



Mining satellite image database of landscapes and application for urban zones: clustering, consensus and categorisation

Ivan Kyrgyzov

► To cite this version:

Ivan Kyrgyzov. Mining satellite image database of landscapes and application for urban zones: clustering, consensus and categorisation. domain_other. Télécom ParisTech, 2008. English. NNT : . pastel-00004084

HAL Id: pastel-00004084

<https://pastel.hal.science/pastel-00004084>

Submitted on 9 Jan 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Thèse

Présenté pour obtenir le grade de docteur
de l'École Nationale Supérieure des Télécommunications

Spécialité : **Signal et Images**

Ivan O. Kyrgyzov

Sujet :

RECHERCHE DANS LES BASES DE DONNÉES SATELLITAIRES DES
PAYSAGES ET APPLICATION AU MILIEU URBAIN: CLUSTERING,
CONSENSUS ET CATÉGORISATION

MINING SATELLITE IMAGE DATABASE OF LANDSCAPES AND
APPLICATION FOR URBAN ZONES: CLUSTERING, CONSENSUS AND
CATEGORISATION

GÉRARD GOVAERT

BERNARD MERIALDO

JEAN-FRANÇOIS MARCOTORCHINO

DANIELLE DUCROT

MARINE CAMPEDEL

HENRI MAÎTRE

Rapporteurs

Examineurs

Directeur de thèse

Contents

Résumé des travaux	1
Notations and Definitions	29
1 Introduction	31
1.1 Content of optical satellite images	32
1.2 Pattern recognition	33
1.3 Contribution of this thesis	34
I Problem statement	37
2 Data mining in satellite images	39
2.1 Data mining	39
2.2 Satellite image models and their application	41
SPOT5 images	41
2.3 Data mining in high resolution satellite images	42
2.4 Conclusions	44
3 Feature extraction	45
3.1 Image intensity features	45
3.2 Texture features	46
Haralick features	46
Gabor features	47
QMF features	48
3.3 Geometrical features	49
Adaptive edge detection	49
Edge detection	49
Adaptive thresholding	50
Edge approximation by line segments	50
3.4 Conclusions	55
II Pattern recognition in satellite images	57
4 Supervised classification	59
4.1 Support Vector Machines (SVM) classification	59
4.2 Curse of dimensionality and feature selection algorithms	61
4.3 SVM classification of satellite images	62

4.4	Conclusions	68
5	Unsupervised classification. Clustering algorithms	71
5.1	State of the art	71
5.2	Combinatorial search	73
5.3	Hierarchical clustering algorithms	73
	Hierarchical agglomerative clustering algorithms	73
	Single-link method	74
	Complete-link method	75
	Average-link method	75
	Centroid-link method	75
	Median-link method	75
	Ward's method	76
	General agglomerative algorithm	76
	Hierarchical divisive clustering algorithms	77
	Bi-section clustering algorithms	77
	K-section clustering algorithms	78
5.4	Partitional clustering algorithms	78
	K-means clustering algorithm	78
	Kernel K-means	79
	Spectral K-means	81
5.5	Bayesian decision theory	81
	Maximum Likelihood Classification	82
	Gaussian Mixture Model	83
	Expectation-Maximisation algorithm	84
5.6	Conclusions	85
6	Model selection	87
6.1	Estimation of the clustering solution	89
6.2	Between-, within- cluster criteria	90
	Validity criteria for hierarchical clustering	90
	Validity criteria for partitional clustering	90
6.3	Information measure	92
	Bayesian information criterion	92
	Akaike information criterion	93
	Minimum description length criterion	93
	Stochastic complexity	93
	2-parts description length	94
6.4	MDL for the Gaussian Mixture Model	94
	MDL for the Complete Log-likelihood of GMM	95
	Graph of MDL to determine the number of clusters	96
	The optimal number of clusters and features	98
	Kernel MDL	98
	Experiments with synthetic data	100
	Experiments with real data: satellite images	102
	Conclusions	106
6.5	An unsupervised hierarchical clustering based on KMDL	106
	An unsupervised hierarchical clustering algorithm, MDL	110

An unsupervised hierarchical clustering algorithm, KMDL	110
Direct error computation	110
Eigen values for error computation	111
6.6 Conclusions	113
7 Combination of clustering results	117
7.1 Introduction	117
7.2 Related works	118
7.3 Nominal data clustering	121
Partitional clustering	121
Combinatorial search	123
Partitional algorithms	123
Binomial distribution	124
Bernoulli mixture model	125
Multinomial mixture model	128
7.4 Combination using a co-association matrix	130
Problem statement	131
Eigen vector decomposition	132
Bounds of square error E	133
Cholesky decomposition	134
Quadratic programming	135
Proposed solution	136
Combination algorithm	136
Approximate solution. Initialisation	138
Gradient descent optimisation and storage reduction	140
A complete iterative algorithm	142
Examples of combining	143
7.5 Proposed Mean Shift combination	147
Proving convergence with mean shift	147
Optimal adaptive radius for mean shift combination	149
Practical aspects of mean shift	150
Results	152
7.6 Measure of clustering stability, stable patterns	155
Examples of stable patterns and clustering stability	156
Self-optimising effect	157
7.7 Conclusions	158
8 Clustering combination and image analysis	159
8.1 Comparing clustering combination methods	160
Clustering error E_c	160
Synthetic data	161
S-link combination: NMI and error E	162
K-means combination: error E and MDL	162
MMM with EM-algorithm combination: error E and MDL	165
Discussion	165
8.2 Combining via reclustering	167
8.3 Combining of satellite image segmentations	168
8.4 Combining of images with artefacts	168

Synthetic segmentations	170
Combination of clustered images with clouds	172
8.5 Determining the optimal number of clusters for image series	174
8.6 Combining for image deblurring	175
8.7 Clustering of nominal data	177
8.8 Unsupervised feature selection algorithm	178
8.9 Conclusions	178
 III Semantic construction	 181
9 Semantic construction for images	183
9.1 Visualisation of clusterings	184
9.2 Extraction of relations among concepts	185
9.3 Semantic construction for multimedia images	185
Combining of classifications	186
Combining descriptions of classifications	191
Discussions	192
9.4 Semantic construction for satellite images	195
Combining of samples of satellite images	195
Unsupervised image clustering of urban content (QuickBird)	197
Satellite image of general content (SPOT 5)	199
Satellite image of urban areas (SPOT 5)	209
9.5 Conclusions	216
 10 Conclusions	 219
10.1 Summary	219
10.2 Perspectives	221
 A Haralick features	 225
 B Features of line segments and edges	 227
 C MDL for the Complete Log-likelihood of GMM	 231
 D Proof of Theorem 7.5.1	 233
 E Dictionary of image classes	 235
 F Human-computer interface for unsupervised image clustering	 237
 Bibliography	 240

Acknowledgements

To my family

I am grateful to my thesis supervisor Professor Henri Maître for the guidance, encouragement, fruitful discussions and constructive comments.

I am thankful to the jury members for their participation in my PhD defence: Jean-François Marcotorchino (president of the jury), Gérard Govaert et Bernard Merialdo (reporters), Marine Campedel et Danielle Ducrot (examiners). I thank them for attentive reviews and comments which helped me to improve my thesis manuscript.

I would like to thank all members of a project "Competence Centre" (TELECOM Paris/ CNES/ DLR) in the frame of which my thesis has been realised. Especially I thank Alain Giros and Mihai Datcu for their fruitful comments.

I want to express many thanks to Marine Campedel for numerous discussions during my thesis. I also thank Hichem Sahbi for his constructive remarks. I am very grateful to all colleagues at TELECOM Paris for their help, advices and an excellent working atmosphere.

I would also thank my "co-bureaux" Luo Bin, Marie Lienou, Mihai Costache, Julien Rabin, Xavier Perrotton and many others PhD students at TELECOM Paris.

I wish to thank my friends Sergiy Redko, Michael Lemarenko, Julie Qian, Julien Chabas and François Faragot for the interesting time spent together.

Of course I am very grateful to my wife Maryna for her unconditional support.

Abstract

Remote sensed satellite images have found a wide application for analysing and managing natural resources and human activities. Satellite images of high resolution, e.g., SPOT5, have large sizes and are very numerous. This gives a large interest to develop new theoretical aspects and practical tools for satellite image mining.

The objective of the thesis is unsupervised satellite image mining and includes three main parts. In the first part of the thesis we demonstrate content of high resolution optical satellite images. We describe image zones by texture and geometrical features.

Unsupervised clustering algorithms are presented in the second part of the thesis. A review of validity criteria and information measures is given in order to estimate the quality of clustering solutions. A new criterion based on Minimum Description Length (MDL) is proposed for estimating the optimal number of clusters. In addition, we propose a new kernel hierarchical clustering algorithm based on kernel MDL criterion.

A new method of "clustering combination" is presented in the thesis in order to benefit from several clusterings issued from different algorithms. We develop a hierarchical algorithm to optimise the objective function based on a co-association matrix. A second method is proposed which converges to a global solution. We prove that the global minimum may be found using the gradient density function estimation by the mean shift procedure. Advantages of this method are a fast convergence and a linear complexity.

In the third part of the thesis a complete protocol of unsupervised satellite images mining is proposed. Different clustering results are represented via semantic relations between concepts.

Keywords : Satellite image, feature, class, cluster, clustering, combination, consensus, algorithm, co-association, number of clusters, square error, minimum description length, mean shift, semantic.

Résumé

Les images satellitaires ont trouvées une large application pour l'analyse des ressources naturelles et des activités humaines. Les images à haute résolution, e.g., SPOT5, sont très nombreuses. Ceci donne un grand intérêt afin de développer de nouveaux aspects théoriques et des outils pour la fouille d'images.

L'objectif de la thèse est la fouille non-supervisée d'images et inclut trois parties principales. Dans la première partie nous démontrons le contenu d'images à haute résolution. Nous décrivons les zones d'images par les caractéristiques texturelles et géométriques.

Les algorithmes de clustering sont présentés dans la deuxième partie. Une étude de critères de validité et de mesures d'information est donnée pour estimer la qualité de clustering. Un nouveau critère basé sur la Longueur de Description Minimale (LDM) est proposé pour estimer le nombre optimal de clusters. Par ailleurs, nous proposons un nouveau algorithme hiérarchique basé sur le critère LDM à noyau.

Une nouvelle méthode de "combinaison de clustering" est présentée dans la thèse pour profiter de différents algorithmes de clustering. Nous développons un algorithme hiérarchique pour optimiser la fonction objective basée sur une matrice de co-association. Une deuxième méthode est proposée qui converge à une solution globale. Nous prouvons que le minimum global peut être trouvé en utilisant l'algorithme de type "mean shift". Les avantages de cette méthode sont une convergence rapide et une complexité linéaire.

Dans la troisième partie de la thèse un protocole complet de la fouille d'images est proposé. Différents clusterings sont représentés via les relations sémantiques entre les concepts.

Mots-clés : Image satellitaire, caractéristique, classe, cluster, clustering, combinaison, consensus, algorithme, co-association, nombre de clusters, erreur quadratique, longueur de description minimale, mean shift, sémantique.

Résumé des travaux

Introduction

L'observation de la Terre est un domaine de la science qui a trouvé un large champ d'application au cours des dernières décennies pour l'analyse, la surveillance, la prévision et la gestion des ressources naturelles et des activités humaines. Les scientifiques et les spécialistes de différents domaines sont intéressés dans les observations de vastes zones de la Terre et même sa surface globale. Les techniques de télédétection sont capables de réaliser de telles observations. La télédétection est l'acquisition de données (des images) qui ont la relation spatiale dans les scènes détectées. Les instruments de télédétection (e.g., des caméras ou des capteurs) mesurent différents éléments d'information tels que les différents domaines du spectre électromagnétique.

Nous nous sommes intéressés au traitement des images optiques telles que les images SPOT5 "Satellites Pour l'Observation de la Terre" ¹. Les images capturées par un satellite reflètent de grandes surfaces dans les détails. Il en résulte de gros volumes de données, e.g., une image SPOT5 de taille 12000×12000 pixels couvre $60 \times 60 \text{ km}^2$ [Gleyzes et al., 2003]. Les images comme SPOT5 sont très nombreuses, près de 10^6 images ont été acquises depuis 1986 ². La haute résolution d'images permet d'identifier de nombreuses structures comme les bâtiments, les routes, les aéroports, les gares, les ponts, les forêts, les domaines agricoles, les nuages, l'eau, la neige et beaucoup d'autres. A l'heure actuelle, elles sont faiblement exploitées en raison de leur grande taille et de temps d'analyse visuelle. Cela donne un grand intérêt et fournit une demande afin de développer de nouveaux aspects théoriques et des outils pour la fouille d'images satellitaires.

Dans cette thèse nous étudions et proposons de nouvelles méthodes d'analyse d'images satellitaires dans le cadre de la fouille d'images. L'objectif de la thèse consiste en l'extraction d'information (des caractéristiques, des modèles et des classes) à partir des images dans une forme compacte et offrant une sémantique des images à l'utilisateur. L'idée défendue ici est de mener différentes approches possibles de fouille d'images et de combiner leurs résultats au lieu d'utiliser une seule approche. L'attention dans cette thèse est concentrée sur les méthodes de clustering non supervisées. La procédure d'extraction de données proposée dans cette thèse n'est pas limitée par une tâche spécifique et peut être appliquée comme une approche générale sur les différents types de données, e.g., les images multimédia. Une caractéristique du schéma proposé consiste en la possibilité de son application à de très grandes bases des données, ce qui est le cas de bases de données d'images satellitaires.

La fouille de données est la direction de science qui combine différents aspects d'apprentissage statistique, de sélection de modèle et d'estimation de paramètres

¹<http://www.spotimage.fr>

²<http://www.cnes.fr/web/258-spot.php>

[Witten & Frank, 1999]. Dans cette thèse la fouille de données est considérée comme une tâche combinant la reconnaissance des formes, la classification et la représentation des relations entre les données [Jain & Dubes, 1988; Fukunaga, 1990; Duda et al., 2000; Theodoridis & Koutroumbas, 2003]. La fouille de données dans des images satellitaires est montrée dans les nombreux travaux [Datcu & Seidel, 2000; Stein et al., 2002].

Un exemple de l'analyse de la surface de la Terre par des images satellitaires est le projet Corine Landcover [Bossard et al., 2000]. L'idée principale de ce travail est de déterminer les caractéristiques et les catégories des surfaces et les classer en utilisant des images satellitaires. Il y a une liste des formes et des classes prédéterminées par les différents experts qui sont utilisés pour la classification supervisée d'images. Les classes sont représentées par un arbre hiérarchique avec plusieurs niveaux de la hiérarchie. Cette représentation est utilisée par différents experts pour analyser la surface. La principale limite de cette approche de fouille de données est la sélection supervisée et la classification supervisée des classes. Souvent, les experts classent les données visuellement, mais il existe également de nombreux algorithmes d'apprentissage pour la classification [Gorte & Stein, 1998]. Ces expériences ont été réalisées pour la plupart avec des images à faible résolution (de plusieurs dizaines à plusieurs centaines de mètres par pixel).

L'apprentissage statistique, la modélisation Bayésienne par les modèles probabilistes sont des approches largement utilisées pour la reconnaissance des formes [Fukunaga, 1990]. Les principes pour estimer et sélectionner des meilleurs modèles probabilistes des données sont expliqués dans [Friedman et al., 2001; Mackay, 2002]. Les modèles probabilistes pour des données continues sont souvent considérés comme un mélange de modèles, e.g., la mélange de distributions Gaussiennes. Une très bonne étude de cette question peut être trouvée dans [Mclachlan & Peel, 2000].

Dans la pratique, des modèles d'apprentissage statistique et des algorithmes parfois atteignent leurs limites en raison des hypothèses fortes sur les distributions probabilistes. Pour surmonter ces limites, une approche basée sur un noyau est récemment devenu très populaire [Vapnik, 1998]. Une étude détaillée d'approches à noyau pour l'apprentissage statistique est présentée dans [Vapnik, 1998]. D'autres idées pratiques et intéressantes pour l'apprentissage par les noyaux sont bien expliquées dans [Shawe-Taylor & Cristianini, 2004].

Les exemples de systèmes de fouille d'image satellitaire et de leurs aspects théoriques sont présentés dans [Datcu & Seidel, 2000; Datcu et al., 2003; Datcu & Seidel, 2005; Barnes, 2007]. L'une des études récentes de la fouille peut être trouvée dans [Heas & Datcu, 2005; Gueguen & Datcu, 2007]. Bien que ce travail porte sur les séries temporelles d'images satellitaires, certains aspects de la modélisation des données peuvent être prises en considération pour différents types d'images satellitaires.

Le but de cette thèse est de contribuer à la fouille non supervisée d'images satellitaires. Pour cette tâche, les sujets suivants ont été abordés :

1. l'extraction d'information provenant d'images satellitaires et sa représentation par des caractéristiques ;
 2. la sélection d'éléments d'information et la réduction de l'espace de données pour les algorithmes de clustering ;
 3. la modélisation des données par le clustering en utilisant différents algorithmes non supervisés avec la sélection de la solution optimale pour chaque algorithme ;
-

4. la combinaison des différents résultats obtenus par les algorithmes non supervisés de clustering ;
5. la représentation sémantique de clusterings pour satisfaire les besoins de l'utilisateur.

La principale problématique de la thèse est de trouver des catégories de zones d'images et de faire le clustering sans connaissance a priori sur le type et le nombre de catégories.

La thèse est organisée de la manière suivante : la problématique de cette thèse est posée dans le Chapitre 2, où une description des images satellitaires à haute résolution est également présentée. Dans ce Chapitre les problèmes de fouille de données et reconnaissance des formes pour les grandes bases de données d'images satellitaires sont également introduits. Le Chapitre 3 présente l'information qui peut être extraite à partir des images satellitaires optiques. L'information est représentée par des caractéristiques qui décrivent les différentes propriétés de la surface de la Terre. Le problème de la reconnaissance des formes est abordé dans le Chapitre 4 où la classification supervisée est présentée. Ensuite, dans le Chapitre 5 nous présentons des algorithmes de clustering. Le problème et les solutions pour la sélection non supervisée de modèles de clustering sont proposés dans le Chapitre 6. La formulation du problème de la combinaison de clustering ainsi que ses solutions sont proposées au Chapitre 7. Le Chapitre 8 présente une variété d'exemples d'application de combinaison. Un protocole complet de fouille non supervisée d'images satellitaires est montré dans le Chapitre 9. Enfin, les conclusions et les perspectives de la thèse sont données au Chapitre 10.

Fouille d'images satellitaires

Dans ce Chapitre, nous donnons une définition de fouille de données et des exemples d'applications dans différents domaines. L'un de ses principaux rôles est la prise de décision. La fouille de données (Data Mining) est un processus de découverte de modèles de données et les relations entre eux. Il couvre également les aspects de classification, la prévision des données et de représentation des résultats découverts. La représentation peut être faite par des indicateurs statistiques ou par l'intermédiaire de la visualisation des images, des arbres ou des graphiques [Larose, 2006]. Une forme (pattern) est un exemple représentatif d'une partie des données et en fonction de l'application peut être une image, un signal ou n'importe quel type de mesures soit à être classées ou reconnues [Marques de Sá, 2001; Larose, 2006].

A l'heure actuelle, la fouille de données est utilisée dans de nombreux domaines : scientifique, industriel et commercial [Marques de Sá, 2001; Theodoridis & Koutroumbas, 2003; Duda et al., 2000; Larose, 2006]. Les exemples d'applications de fouille de données sont :

- ★ **Imagerie.** Il y a de plus en plus d'images satellitaires et une demande de la gestion des données et le traitement intelligent augmente : l'analyse, la détection et la classification des images satellitaires et aériennes. Il y a de nombreuses applications, e.g., pour la gestion des zones urbaines et agricultures : analyse et la gestion des sols ; pour la géologie : la classification des couvertures des terres (eau, sol, des forêts, urbain, etc.), l'estimation et l'analyse des ressources minières, l'analyse sismique ; pour l'astronomie : l'analyse des images télescopiques. De nombreuses organisations commerciales sont intéressées par l'analyse des images multimédia afin de trouver des groupes de mêmes images et d'analyser les besoins des utilisa-

teurs, l'analyse de vidéo, la classification, la description du contenu et l'indexation des images et des video (e.g., Google).

- ★ **Bio-informatique.** En bio-informatique l'une de tâches connue d'extraction de données est l'étude du comportement des gènes au cours d'expériences. Les gènes peuvent être regroupés automatiquement où chaque groupe de gènes représente le même comportement ou les caractéristiques.
- ★ **Industrie.** La fouille de données peut être considérée pour l'industrie lourde et pour l'industrie de haute technologie. Un exemple très courant est la détection automatique et la classification des objets sur une chaîne de montage d'une usine. Cela réduit le temps et d'améliorer la qualité de l'assemblage de produits.
- ★ **Commerce.** La fouille de données pour le commerce peut-être l'analyse du marché et des produits, etc. L'un des problèmes intéressants est l'étude des besoins des consommateurs.

Nous nous intéressons à la fouille d'images satellitaires SPOT5. Une telle image a deux dimensions et est enregistré par un scanner optique multispectral. Pour chaque appareil SPOT5 une ligne a 12000 éléments qui correspondent aux pixels sur une image³. Le traitement de l'image satellitaire comprend les étapes suivantes : tout d'abord, les images sont enregistrées par des appareils d'enregistrement numérique. Puis elles sont corrigées par la correction, la restauration ou la reconstruction d'images. Enfin, les images sont classées en utilisant leurs caractéristiques. Cela se fait par la classification supervisée, semi-supervisée ou non supervisée. La dernière étape est la présentation des résultats soit directement à l'utilisateur, soit par l'enregistrement sur un système d'information géographique (SIG). Dans cette thèse la classification non supervisée d'image satellitaires et la représentation des résultats sont pris en compte.

La fouille des images satellitaires à haute résolution (HR) est considérée dans ce Chapitre. Les images à HR fournissent diverses informations sur la surface de la Terre et sont très intéressants pour les experts dans différents domaines : l'urbain, l'agriculture, l'environnement, la militaire, etc. Les exigences relatives à la classification d'images et sa validation peuvent être trouvées dans [Muchoney & Strahler, 1996; Atkinson & Lewis, 2000]. L'une des principales applications de l'imagerie satellitaire est la construction des cartes pour la détection des routes et des zones urbaines, pour l'analyse des champs agricoles ou des forêts (les classes). Les systèmes actuels d'analyse d'images satellitaire impliquent très souvent l'information fournie par un expert. Ce type de travail nécessite un effort humain considérable. La demande pour les systèmes automatiques est très grande car ils peuvent améliorer la qualité des décisions et réduire le temps d'analyse nécessaire. Un autre intérêt de l'utilisation de systèmes automatiques d'analyse d'images satellitaires consiste à découvrir de nouvelles informations et connaissances (nouvelles relations entre les classes, de nouvelles classes).

Les étapes principales d'un système pour analyser les images satellitaires sont généralement les mêmes, indépendamment de l'application :

1. Extraction des caractéristiques (feature extraction) - la modélisation et l'extraction d'information d'image satellitaire ;

³<http://www.cnes.fr>

2. Reconnaissance des formes (pattern recognition) - la sélection de modèles et l'optimisation de leurs paramètres pour l'analyse, la classification et le clustering ;
3. Représentation des résultats - la visualisation des résultats de la reconnaissance des formes.

Dans ce Chapitre nous avons illustré des exemples de fouille de données. Un bref exemple d'analyse d'images satellitaires et de fouille de données a été illustré. Les sujets suivants sont passés en revue : les demandes d'extraction de données dans différents domaines (scientifique, industriel et commercial). L'importance de l'utilisation des systèmes de fouille de données a été argumentée. Les exemples d'analyse d'images satellitaires ont été présentés. Les étapes de l'extraction de données ont été démontrées dans ce Chapitre. Une interaction entre un utilisateur et un ordinateur pour fouiller des images a été décrite.

Extraction des caractéristiques

Dans ce Chapitre, nous revoyons la définition et les notations de descripteurs (également appelé les caractéristiques) et montrons les modèles de caractéristiques ainsi que leur extraction. Une forme (pattern) est considérée comme une partie d'une image satellitaire.

Une image naturelle, par exemple une image satellitaire, contient des régions qui ont des propriétés communes pour la perception visuelle. Les régions homogènes de la ville, des forêts et les nuages sont faciles à distinguer. Chacune de ces régions est caractérisée ou décrit par des caractéristiques, e.g., les pixels de niveau gris d'intensité ou les textures. Il y a deux groupes de caractéristiques : (i) naturelle et (ii) artificielle. Les caractéristiques naturelles correspondent à la perception visuelle, e.g., niveau d'intensité, de régions texturales, tandis que les caractéristiques artificielles sont obtenues après la manipulation d'images, e.g., un histogramme d'une image, les spectres de fréquence spatiale, etc. [Pratt, 2001].

Les caractéristiques d'une image sont utilisées pour la segmentation d'images, la classification et le regroupement afin de trouver des régions avec des propriétés communes. En traitement d'images de nombreuses caractéristiques ont été proposées pour décrire une image. En règle générale, pour des images statiques (dans notre cas SPOT5) les modèles de caractéristiques sont principalement les suivants : les descripteurs statistiques de l'intensité de l'image, de textures et de la géométrie [Pratt, 2001; Forsyth & Ponce, 2002]. Dans ce Chapitre, nous proposons les caractéristiques géométriques et donnons des exemples des caractéristiques de textures qui ont été réalisés dans [Campedel et al., 2004, 2005].

Pour obtenir les caractéristiques de l'intensité d'image nous calculons les moments statistiques telles que les moments centraux du premier ordre (valeur moyenne) et du second ordre (écart type). Ce type de descripteurs statistiques est en mesure de distinguer sur une image SPOT5 une partie lumineuse d'image avec une haute intensité des niveaux de gris (e.g., les nuages, la neige) d'une partie sombre à faible intensité (e.g., la mer et la terre). Certaines statistiques d'ordre supérieur peuvent être extraites de l'image : l'asymétrie, le kurtosis, l'entropie.

Une texture est définie comme une image d'une surface qui est facile à reconnaître, mais difficile à décrire et il est représenté par de nombreux objets [Forsyth & Ponce, 2002]. Différentes propriétés d'échantillons d'images peuvent être caractérisées par la dépendance spatiale de l'intensité des pixels. Certains modèles de dépendance spatiale

sont représentés soit par l'extraction des statistiques d'une image, soit par un filtrage d'image. Les caractéristiques obtenues par le filtrage sont appelées les caractéristiques de texture. Maintenant, nous donnons quelques modèles basiques de caractéristiques de texture.

Des caractéristiques réputées pour décrire la texture sont les caractéristiques de Haralick [Haralick et al., 1977]. Ces caractéristiques sont calculées sur un histogramme du deuxième ordre de la distribution de probabilité conjointe d'une paire de pixels que l'on appelle une matrice de co-occurrence (MC). Les caractéristiques calculées sur MC sont des descripteurs statistiques qui reflètent différentes propriétés de textures. Les filtres de Gabor représentent des modèles de la perception visuelle d'une texture [Daugman, 1985]. Ils ont été largement étudiés et appliqués pour la classification et la segmentation d'images [Dunn et al., 1994; Dunn & Higgins, 1995; Jain & Farrokhnia, 1991; Weldon et al., 1996]. Les caractéristiques extraites par des filtres de Gabor sont les valeurs moyennes et les écarts-types des images filtrées. Ces caractéristiques permettent d'obtenir des caractéristiques qui sont invariantes par rotation [Manthalkar et al., 2003]. Quadratic Mirror Filters (QMF) appliquent un filtrage qui peut reconstruire une image exactement [Vetterli, 1986].

Nous considérons les caractéristiques géométriques comme celles qui décrivent les propriétés géométriques des objets visibles sur l'image, e.g., les propriétés statistiques des segments linéaires et des bords détectés sur l'image. Nous présentons une approche de détection de bords, l'approximation des bords par les segments linéaires et l'extraction de caractéristiques géométriques. L'image est filtrée par le filtre de Deriche [Deriche, 1987b] suivant le seuillage de Hysteresis et la détection des bords. Les segments linéaires sont les approximations des bords [Papakonstantinou, 1985]. Les caractéristiques sont extraites des bords et des segments linéaires.

Les modèles des caractéristiques décrivant des propriétés différentes de la surface de la Terre dans les images satellitaires ont été présentés dans ce Chapitre. Ces modèles reflètent la texture et les caractéristiques géométriques. Les principaux thèmes abordés dans ce Chapitre sont les suivants : les statistiques calculées sur les modèles Haralick, Gabor et QMF ont été considérés comme les caractéristiques de texture. Les éléments géométriques sont des valeurs statistiques de bords et de segments linéaires. Les bords ont été détectés en utilisant le filtre de Deriche. L'espace des caractéristiques introduit dans ce Chapitre est utilisé ensuite pour la fouille d'images satellitaires par des algorithmes supervisés et non supervisés.

Reconnaissance des formes pour l'imagerie satellitaire

La reconnaissance des formes (pattern recognition) est la partie principale de la fouille de données et a été développée dans les 20 dernières années. L'une des problématiques de reconnaissance des formes est la description statistique des données. Une telle description est basée sur des modèles qui sont utilisés pour la classification : supervisée, semi-supervisée ou non supervisée. La tâche de classification supervisée est d'attribuer des étiquettes ou des classes à des échantillons, en sachant que les classes existent et quel échantillon appartient à quelle classe. La classification semi-supervisée s'effectue par l'intégration de l'interaction humaine.

Dans ce Chapitre, nous considérons la classification supervisée qui assigne une forme à l'une des classes. Cette forme est décrite par un ensemble de caractéristiques. La plus simple classification supervisée est la classification en deux classes où les classes sont

linéairement séparées. Dans ce cas, le modèle du classificateur est un hyperplan qui sépare les modes dans l'espace des caractéristiques.

Les études récentes sur la classification ont indiqué le potentiel considérable de la machine à vecteurs de support (Support Vector Machines, SVM) [Vapnik, 1998; Chapelle et al., 2002; Shawe-Taylor & Cristianini, 2004]. L'une des applications de reconnaissance de formes est la classification supervisée de données de télédétection [Huang et al., 2002]. Des études comparatives ont montré que la classification SVM peut être plus précise que des techniques connues telles que les réseaux de neurones et les arbres de décision ainsi que les classificateurs probabilistes classiques tels que la classification de maximum de vraisemblance (maximum likelihood classification) [Chapelle et al., 2002]. SVM a été conçu pour la classification binaire mais plusieurs méthodes existent pour étendre cette approche vers la classification des multi-classes [Vapnik, 1998; Hsu & Lin, 2002]. La classification par SVM est basée sur l'estimation d'un hyperplan optimal de séparation entre les classes en mettant l'accent sur des échantillons qui sont au bord des distributions des classes: les vecteurs de support. L'approche SVM est largement utilisée dans de nombreuses applications de classification supervisée. Elle est surtout mise en oeuvre dans de nombreux systèmes de traitement d'images satellitaires [Parulekar et al., 2005] : pour la classification supervisée [Bhattacharya et al., 2007; Zammit et al., 2007] ainsi que pour la classification semi-supervisée ou la boucle de retour de pertinence (relevance feedback) [Ferecatu & Boujemaa, 2007; Costache & Datcu, 2007].

Dans ce Chapitre nous considérons le problème de la malédiction de la dimension. Les images sont décrites par un "grand" ensemble des caractéristiques ("grand" signifie de dizaines à des centaines des caractéristiques). Il est très important de prendre en compte la dimension de données. La dimension influence les résultats de classification de manière significative [Bishop, 2006]. L'auteur montre que certains algorithmes de reconnaissance des formes ne peuvent être directement appliquées à des données de grande dimension et devraient être utilisés avec prudence. L'une des solutions à ce problème peut être soit une pondération, soit une sélection des caractéristiques. La pondération est une procédure d'attribution d'un poids à une caractéristique soit par une connaissance préalable, soit par l'intermédiaire d'un algorithme qui estime le poids au cours du processus de classification. La sélection est en mesure de déterminer un ensemble des caractéristiques plus approprié pour représenter l'information utile dans les données. En outre, le temps de traitement des données est en baisse après la sélection. La réduction de la dimension ou des caractéristiques pourrait être utilisée pour diminuer le sur-apprentissage et améliorer la classification. L'une de ces techniques est la sélection récursive (recursive feature elimination, RFE) [Guyon, 2002]. RFE élimine certaines caractéristiques et conserve le sous-ensemble qui fournit la meilleure performance de classification [Campedel et al., 2004].

Pour classer une base de données d'images satellitaires SPOT5, quatre classes de textures sont utilisées : les champs, les villes, les nuages et la mer. Les images satellitaires de différentes villes du monde ont été sélectionnées : Béziers, Paris, Los Angeles et Hong Kong. Nous supposons que chaque image contient différentes textures de la surface de la Terre qui reflètent la nature et l'architecture des villes diverses. Pour aborder l'absence de la description géométrique nous démontrons un exemple de la classification des images avec des caractéristiques géométriques et de texture.

Dans ce Chapitre la classification supervisée et la sélection des caractéristiques ont été introduites et appliquée aux images satellitaires. Le problème d'un grand ensemble de caractéristiques appelé "la malédiction de la dimension" a également été présenté. La

classification des images satellitaires par SVM avec les caractéristiques géométriques et textuelles a été démontrée. Comme il a été illustré, un très grand espace de données peut conduire à une mauvaise classification des résultats et prendre beaucoup de temps. La dimension de données a été réduite via la sélection par SVM-RFE. La classification supervisée par SVM qui utilise les caractéristiques géométriques montre des propriétés intéressantes des images satellitaires, e.g., la rugosité des structures détectées. Cependant, l'utilisation des caractéristiques géométriques est une approche limitée et elle doit être complétée par les caractéristiques de texture pour refléter des différentes propriétés des surfaces.

Classification non supervisée (clustering)

La modélisation des données est chargée de représenter la connaissance et l'extraction d'information. Quand il n'y a pas ou peu d'information a priori sur les données alors les méthodes de classification non supervisée doivent être utilisées. La classification non supervisée est un moyen de la modéliser les données. Pendant la modélisation, les paramètres optimaux du modèle de données sont estimés et la qualité des données qui sont remplacées par le modèle est vérifiée. Il y a plusieurs problèmes d'estimation de modèles pour lequel nous devrions prêter attention :

1. le choix du modèle de données,
2. si les paramètres du modèle sont partiellement connus ou pas connus, ils devront être estimés,
3. l'approche d'estimation devra également être sélectionnée et argumentée.

Les modèles de classification nous serviront pour la description du contenu de données. Pour comprendre le contenu d'un ensemble de données, nous devons d'abord trouver les éléments composant (les clusters et / ou les classes). L'un des moyens de trouver des groupes ou des classes de données peut être l'estimation du modèle de données. Le modèle indique comment les données sont distribuées dans les clusters (classes). Ensuite, les relations entre les clusters peuvent être présentées par des liens entre eux, par exemple sous la forme d'un graphe ou d'un arbre hiérarchique.

Il y a une variété de directions pour découvrir les classes par la classification non supervisée. L'un d'entre eux est fait référence dans la littérature - le clustering. Les références sur les méthodes de regroupement (clustering) et de reconnaissance des formes peuvent être trouvées dans [Diday, 1979; Jain & Dubes, 1988; Fukunaga, 1990] tandis que les approches et les formulations sont proposées dans [Mclachlan & Peel, 2000; Duda et al., 2000; Friedman et al., 2001; Rencher, 2002; Theodoridis & Koutroumbas, 2003; Rowe, 2002; Mackay, 2002; Hardle et al., 2003; Bishop, 2006]. Le regroupement (clustering) est un processus automatique qui découvre des groupes (les clusters, les groupes de données similaires) et assigne un échantillon de données à chacun des groupes.

L'une des études précédentes sur les différentes méthodes et les algorithmes de clustering est présenté dans [Diday, 1979; Jain & Dubes, 1988]. Habituellement, les techniques de regroupement sont soit partitionnelles, soit hiérarchiques. Le regroupement par la partition est une division des échantillons en groupes (clusters), tels que les échantillons dans un groupe sont plus proches les uns des autres que d'échantillons dans différents groupes. Le regroupement par les méthodes hiérarchiques est classé comme la division et l'agglomération et est également appelé de bas en haut et de haut en bas (bottom-up and

top-down) [Diday, 1979; Jain & Dubes, 1988; Rencher, 2002; Friedman et al., 2001]. Le regroupement par la division commence par un groupe qui contient tous les échantillons et les sépare dans les clusters de la manière récursive. L'avantage des algorithmes de division est que le temps, la complexité et la mémoire sont très petits. Au contraire, il souffre d'optimalité locale de trouver des solutions de regroupement. Le regroupement par l'agglomération commence par un point (singleton) et fusionne des groupes de deux ou plusieurs clusters. Les algorithmes de regroupement (clustering) traitent le problème de regroupement comme un processus d'optimisation qui cherche à maximiser ou minimiser un critère particulier de regroupement [Friedman et al., 2001; Rencher, 2002; Webb, 2002].

Le nombre de solutions possibles pour obtenir toutes les partitions de données est trop élevé pour la plupart des cas pratiques [Jain & Dubes, 1988]. La recherche directe du regroupement pourrait être appliquée qu'à un très petit nombre d'échantillons.

Les méthodes hiérarchiques de regroupement (clustering).

Le clustering hiérarchique est une partition de données imbriquées. Il est représenté par un arbre hiérarchique ou un *dendrogramme*. Chaque regroupement correspond à un certain niveau de l'arbre hiérarchique [Diday, 1979; Theodoridis & Koutroumbas, 2003]. Nous considérons le cas de regroupement hiérarchique lorsque les groupes d'une partition à un certain niveau sont complètement inclus dans les groupes de niveau supérieur. Le haut niveau hiérarchique de l'arbre est la racine et contient toutes les données, le niveau inférieur de l'arbre peut contenir les feuilles qui correspondent aux échantillons, chaque groupe de ce niveau a un seul échantillon.

Les méthodes hiérarchiques d'agglomération. Les méthodes hiérarchiques d'agglomération consistent en fusion des groupes à un certain niveau de l'arbre. Nous ne considérons que la fusion de paires de clusters (un problème fréquent pour résoudre des problèmes pratiques [Rencher, 2002]). Il n'existe pas de garantie, en général, que la fusion par paires peut produire la représentation optimale de données ou la solution optimale d'une fonction objective. De l'autre côté, plusieurs clusters peuvent être fusionnés à chaque étape, cependant cette méthode implique le temps de calcul exponentiel et beaucoup de mémoire. Le regroupement par l'agglomération hiérarchique construit un arbre hiérarchique à partir de la première partition en utilisant une matrice de paires de distances entre les clusters (clusters peuvent contenir un seul échantillon). Selon la méthode de choix de deux groupes à fusionner et la méthode pour calculer la distance, plusieurs méthodes du regroupement hiérarchique existent : lien-unique (single-link), lien-complète (complete-link), lien-moyenne (average-link), lien-médiane (median-link) et agglomération de Ward. La généralisation de ces approches également existe [Diday, 1979; Jain & Dubes, 1988; Duda et al., 2000; Rencher, 2002; Theodoridis & Koutroumbas, 2003].

A chaque étape de la fusion par l'algorithme de single-link, deux groupes voisins sont fusionnés. La distance minimale entre les clusters est la distance entre deux échantillons les plus proches de ces groupes. Pour la méthode de complete-link, là encore, comme dans les cas précédents, un arbre hiérarchique est construit par la fusion de deux groupes (les plus proches). La distance qui les sépare est calculée pour les deux échantillons plus éloignés appartenant à ces groupes. Cet algorithme diffère de la single-link par le calcul de la plus grande distance entre les clusters. L'algorithme de complete-link cherche la cohésion des groupes, contrairement à single-link qui cherche des groupes isolés. L'approche average-link cherche de fusionner les deux groupes voisins lorsque la distance entre eux est le moyen par paires à distance entre les points de ces groupes. Cet algorithme diffère de deux présentées ci-dessus dans le sens où il est moins sensi-

ble au bruit. D'autre part, cet algorithme a tendance à trouver les clusters globulaires et pas à trouver des groupes avec des formes complexes. Ses propriétés statistiques sont mentionnées dans [Friedman et al., 2001]. L'approche de centroid-link regroupe deux clusters les plus proches et la distance entre eux est calculée comme étant la distance entre les centroïdes de ces groupes. Pour la méthode de median-link un cluster ayant le plus grand nombre de points a un poids supérieur dans le calcul de distance [Rencher, 2002]. La méthode de Ward (Ward's method) est basée sur la réduction de l'erreur dans chaque groupe et elle est nommée comme la méthode de la variance minimale. Les méthodes hiérarchiques présentées ci-dessus peuvent être considérées comme une seule méthode avec des paramètres différents pour mettre à jour la matrice des distances [Lance & Williams, 1967].

Les méthodes hiérarchiques de division

Les méthodes hiérarchiques de division ne sont pas populaires et sont rarement rencontrées dans la littérature [Jain & Dubes, 1988; Rencher, 2002; Webb, 2002]. Cependant, nous expliquons brièvement cette approche pour une observation complète de méthodes hiérarchiques et pour montrer certains de leurs aspects intéressants. Ces algorithmes divisent itérativement les données en clusters. Au cours de la division, ils construisent un arbre hiérarchique. Dans la littérature [Jain & Dubes, 1988; Rencher, 2002] le regroupement hiérarchique de division sont considérés en deux groupes : monothetic et polythetic. Les algorithmes monothetics utilisent les caractéristiques consécutivement une par une pour diviser les données tandis que les algorithmes polythetics utilisent toutes les caractéristiques pour diviser les données. Pour les algorithmes monothetics un ordre de caractéristiques devrait être fixé ou estimé. Seulement les algorithmes polythetic sont considérés dans cette thèse parce qu'un ensemble complet des caractéristiques est plus informatif que leur sous-ensemble.

Deux exemples des algorithmes hiérarchique de division sont Bi-section et K-section algorithmes de regroupement [Chan et al., 1994]. L'algorithme Bi-section divise les données en deux groupes et ainsi de suite chaque subcluster est divisé en deux. De même, l'algorithme K-section divise les données en K clusters [Jain & Dubes, 1988]. L'avantage des algorithmes de clustering par la division est la construction rapide des arbres hiérarchiques pour un volume de données élevé.

Les algorithmes de regroupement par la partition (partitional clustering)

K-moyens (K-means) algorithme de clustering. K-means algorithme peut être trouvé dans les nombreux travaux sur le clustering des données [Diday, 1979; Jain & Dubes, 1988; Duda et al., 2000; Webb, 2002; Mackay, 2002; Theodoridis & Koutroumbas, 2003; Friedman et al., 2001]. La version classique de K-means regroupe l'ensemble de données en un nombre prédéterminé de clusters. Chaque groupe est paramétré par ses vecteurs moyens. L'algorithme a deux étapes :

1. attribution : chaque échantillon est attribué à son plus proches vecteur moyenne,
2. mise à jour : les vecteurs moyenne sont re-estimés.

Nous devrions mentionner l'algorithme de regroupement K-medoid qui a été proposés dans [Kaufman & Rousseeuw, 1990; Diday, 1979]. Sa principale différence d'algorithme K-means consiste à remplacer les vecteurs moyens par des échantillons qui minimisent l'erreur quadratique.

K-means à noyau. Dans le cas lorsque les données ont une structure complexe (e.g., les données ne sont pas séparables linéairement) l'application directe de K-means est

inappropriée en raison de la tendance de K-means de détecter des groupes en forme des globes. L'une des solutions est de regrouper les données par un noyau sur une nouvelle espace caractéristique où les clusters sont linéairement séparables. Le noyau est défini comme le produit scalaire. Avec le noyau et la fonction objective, les étapes d'algorithme K-means peuvent être appliquées [Shawe-Taylor & Cristianini, 2004]. Lorsque le noyau non-linéaire est appliqué, ce regroupement peut trouver des groupes de clusters qui ont des formes non-linéaires.

K-means spectrale. Dans [Ng et al., 2002], les auteurs ont proposé l'algorithme de clustering spectrale. L'idée générale de cette approche est d'utiliser les vecteurs propres du noyau comme une matrice de données sur lesquelles un algorithme de clustering est appliqué. Le point essentiel est de fixer le nombre de vecteurs propres comme le nombre de clusters. La relation entre le noyau et le spectrale K-means sont mises en évidence par [Schölkopf et al., 1996; Dhillon et al., 2004]. Les références supplémentaires sur le clustering spectrale sont [Lau & Wade, Aug 1991; Kannan et al., 2000; Yu & Shi, 2003].

La théorie de la décision Bayésienne.

La théorie de la décision Bayésienne [R. Hanson & Cheeseman, May, 1991; Duda et al., 2000; Bishop, 2006] est très utilisée pour la reconnaissance des formes [Cheeseman & Stutz, 1996; Mclachlan & Peel, 2000]. Elle est basé sur l'hypothèse que les données peuvent être décrites par des modèles probabilistes. Le problème pratique de la décision Bayésienne se pose lorsque les valeurs de la probabilité ne sont pas connues. En pratique, ces valeurs doivent être estimés sur des données en utilisant l'hypothèse sur un modèle de données. Dans ce Chapitre, nous donnons une brève introduction sur la théorie de la décision Bayésienne et puis nous décrivons un algorithme pour estimer le modèle de données. Nous utilisons le terme "classe" au lieu de "cluster" sans perte de généralité.

La classification par le maximum de vraisemblance. Pour des tâches pratiques de reconnaissance des formes, nous n'avons pas de probabilités (la vraisemblance, les priors et l'évidence), par contre nous avons seulement les échantillons de données. Le problème est de savoir comment évaluer ces probabilités et les utiliser. Dans la littérature sur l'estimation des paramètres du modèle il y a deux approches principales [Duda et al., 2000; Mackay, 2002; Mclachlan & Peel, 2000; Bishop, 2006] :

1. l'estimation du maximum de vraisemblance,
2. l'estimation Bayésienne.

Ces deux approches produisent les résultat similaires à l'estimation des paramètres pour le grand volume de données (c'est le cas de reconnaissance de formes pour les images satellitaires) [Duda et al., 2000].

Dans des cas pratiques, avec de nombreux échantillons (un million et plus) et une dimension élevée (plusieurs dizaines et plus), l'estimation bayésienne demandes beaucoup de calculs. Il est difficile d'appliquer une telle estimation dans le délai raisonnable. Au contraire, les résultats de l'estimation du maximum de vraisemblance sont beaucoup plus faciles à réaliser et plus intuitif à interpréter. C'est pourquoi nous proposons d'examiner l'estimation du maximum de vraisemblance. Cette approche est présentée dans ce Chapitre afin d'estimer un modèle probabiliste par le modèle de mélange gaussien (MMG) et la classification non supervisée.

L'évaluation des probabilités du modèle et ses paramètres sont faites par l'algorithme "Expectation-Maximisation" ou algorithme EM. L'algorithme maximise le logarithme de vraisemblance [Dempster et al., 1977; Mclachlan & Peel, 2000]. Les récents développements

intéressants d'algorithme EM de mélange des modèles sont donnés dans [Govaert & Nadif, 2005, 2003]. Cette optimisation peut se converger vers les solutions optimales qui sont locales, qu'est souvent le cas de problèmes pratiques. Et, en général, il n'existe aucune garantie de la convergence vers le optimum global, sauf dans des cas particuliers [Mclachlan & Peel, 2000; Bishop, 2006]. Certains trucs pratiques sur l'amélioration de la solution optimale sont examinés.

Dans ce Chapitre les algorithmes de regroupement non supervisé (clustering) ont été révisés. Ils sont divisés en deux groupes : clustering partitionnelle et clustering hiérarchique. Les algorithmes de regroupement par la partition sont présentés de le plus simple comme K-means vers les plus complexes comme K-means spectrale et K-means à noyau. Ils sont terminés par un regroupement probabiliste avec le modèle de mélange gaussien des clusters. Les paramètres de MMG sont estimés par l'algorithme EM. Les algorithmes de regroupement sont comparés via leur complexité et l'optimalité.

Le problème principal du regroupement non supervisé est l'estimation de la qualité de regroupement :

1. comparer et sélectionner les meilleurs regroupements d'un algorithme,
2. déterminer le nombre de clusters.

Ces problèmes ont l'importance cruciale et dépendent de l'algorithme de clustering. Ils seront considérés dans le Chapitre suivant ou de nouvelles idées seront proposées.

Sélection du modèle

L'extraction de connaissances à partir des images satellitaires est l'objectif principal de cette thèse. Notre objectif est d'obtenir un contenu des données grâce à la modélisation des données. A la première étape, cette modélisation doit fournir des clusters. Dans notre cas, l'un des problèmes cruciaux du clustering des données est qu'il n'existe pas d'information a priori sur le nombre de groupes (des clusters, des classes) dans une image. Un groupe ou une classe est considérée comme un type de la surface de la Terre. Pour aborder ce problème, nous proposons d'appliquer le regroupement ou des algorithmes de classification non supervisée pour détecter les clusters dans une image. Comme nous l'avons vu dans le Chapitre précédent l'un des paramètres de ces algorithmes est le nombre de clusters. Dans ce Chapitre, nous analysons plusieurs approches et des critères pour estimer le nombre optimal de clusters. En outre, ces critères sont en mesure de choisir le meilleur regroupement d'un ensemble de clusterings. Il est à noter, que de sélection inappropriés du nombre de groupes et / ou du regroupement peut conduire à l'interprétation des données erronées que peut être le cas des problèmes pratiques.

L'estimation de la qualité de regroupement est appelée la validité de clustering. L'étude sur la validité que peut être trouvée dans [Jain & Dubes, 1988; Mclachlan & Peel, 2000; Friedman et al., 2001; Theodoridis & Koutroumbas, 2003; Mackay, 2002] est divisés en trois groupes [Jain & Dubes, 1988; Theodoridis & Koutroumbas, 2003] : externe, interne et relative. Les critères externes vérifient comment les données confirment une structure qui a été imposée a priori. Ces critères peuvent être vérifiés sans l'application des algorithmes de regroupement. Les critères internes peuvent se basé sur la quantité de valeurs calculées sur les données et le regroupement. Les critères relatifs évaluent le regroupement en comparant différents clusterings obtenu soit à partir du même algorithme, mais

avec des paramètres différents, soit issus des différents algorithmes de regroupement sur les mêmes données. Dans ce Chapitre, nous nous intéressons aux critères internes pour les différents algorithmes.

Pour les algorithmes de regroupement hiérarchique, nous montrons une valeur statistique appelé le coefficient cophenetic de corrélation (CCCP ou Cophenetic Correlation Coefficient (CPCC)) [Jain & Dubes, 1988]. L'estimation des algorithmes de regroupement est montrée à travers l'erreur de regroupement de données et les critères théoriques d'information.

Le nombre de groupes (clusters) dépend de la façon comment un modèle optimal rapproche aux données. Dans ce Chapitre, nous concentrons notre attention sur une mesure théorique d'information. En vertu de cette mesure nous considérons Minimum Description Length (MDL). Nous montrons les relations entre MDL et d'autres critères d'information telles que les mesures d'information Akaike (AIC, Akaike information criterion), le critère d'information de Bayes (BIC, Baeyesian information criterion) et la complexité stochastique d'information (SIC, statistic information criterion). Nous démontrons également la simplification de MDL pour la hypothèse de regroupement "dur" (hard clustering), lorsque les données appartiennent à un seul groupe. Par ailleurs, nous proposons un nouveau critère appelé MDL à noyau (KMDL, kernel MDL) afin d'estimer le nombre de clusters pour l'algorithme de clustering par K-means à noyau. Sur la base de MDL et KMDL critères, nous proposons un nouvel critère MDL (GMDL, general MDL). En outre, plusieurs algorithmes de regroupement hiérarchique proviennent de GMDL. L'intérêt de ces algorithmes est qu'ils trouvent des groupes ayant des formes non-linéaires et dans le même temps, ils sont en mesure d'estimer la qualité de regroupement et le nombre optimal de clusters.

Les critères théoriques sont devenus très connus afin de sélectionner le modèle optimal de données [Mclachlan & Peel, 2000; Mackay, 2002]. En particulier, ils donnent de bons résultats dans le cas où un grand nombre de données sont disponibles, e.g., le traitement d'images satellitaires. Ces critères sont clairement formulées et ont de bonnes bases théoriques. Il existe de nombreux travaux qui montrent l'équivalence des mesures théoriques entre AIC, BIC, MDL [Mackay, 2002; Mclachlan & Peel, 2000]. Certains critères entropiques peuvent également être trouvée dans [Biernacki et al., 1999].

Notre objectif est de trouver des groupes dans les images satellitaires sans connaissance préalable de leur type ou le nombre. Vu la quantité de données disponibles, nous préférons utiliser des algorithmes de regroupement simples, rapides et efficace. K-means est un d'entre eux, mais il souffre de plusieurs inconvénients :

1. il ne peut s'adapter à toutes formes de clusters,
2. la connaissance du nombre de groupes est nécessaire,
3. le résultat dépend fortement du processus d'initialisation.

Pour répondre au premier problème, une solution classique consiste à utiliser K-means algorithme à noyau [Shawe-Taylor & Cristianini, 2004]. Au cours de la dernière décennie des algorithmes à noyau ont attiré beaucoup de chercheurs qui les applique à diverses tâches telles que l'apprentissage automatique, la reconnaissance des formes, etc.

Pour répondre aux deuxième et troisième problèmes nous proposons d'utiliser une approche standard telle que la sélection du meilleur regroupement obtenue en utilisant des différents nombres de clusters et initialisations. Cette sélection est basée sur un minimum d'un critère de regroupement. Il permet également de stabiliser les résultats de

regroupement. La sélection de la meilleure solution pour les initialisations aléatoires a été montré pour être efficace [Biernacki et al., 2003].

Notre proposition sur l'utilisation du critère MDL pour déterminer le nombre de groupes se fonde sur plusieurs arguments. Tout d'abord, MDL est en mesure de donner le meilleur modèle de données [Mackay, 2002], e.g., pour le modèle de mélange Gaussien (GMM). Deuxièmement, ce critère fonctionne bien lorsque beaucoup de données sont disponibles [Heas & Datcu, 2005]. C'est notre cas parce que nous avons un stockage énorme des images satellitaires. Enfin, dans la littérature nous n'avons pas trouvé de travaux précédents sur l'application de critère MDL pour Kernel K-means afin de trouver le nombre optimal des clusters. Tous ces arguments nous ont fourni la motivation de formuler les critères MDL pour l'algorithme de K-means à noyau.

Ce Chapitre couvre les sujets suivants : un critère pour comparer les regroupements en sachant les classes, les critères entre-cluster et inter-cluster pour les groupements hiérarchique et partitionnelle, les critères d'information. Nous révisons la définition de MDL pour GMM et nous montrons une simplification de MDL à travers le logarithme de la vraisemblance de GMM complet. Ensuite, nous formulons MDL à noyau simplifié en utilisant MDL pour GMM. Les résultats sur les données synthétiques et des images satellitaires sont présentés. La qualité d'un regroupement peut être mesurée via la classification connue (e.g., via l'indice de Rand, le nombre d'échantillons correctement classifiés, etc. [Jain & Dubes, 1988]).

Les critères de validité de regroupement. Certains critères de validité peuvent être trouvés dans [Jain & Dubes, 1988]. Le regroupement partitionnel est la partition de données en groupes (clusters). Un bon regroupement partitionnel est tel qu'il réduit la distance entre les points dans le même groupe et au même temps, cela augmente les distances entre les différents groupes. Une série des critères a été calculée dans [Coleman & Andrews, 1979].

Mesures d'information. Les critères donnés ci-dessus pour le regroupement partitionnel sont basés sur la théorie de l'information et sont souvent très efficaces. Dans le cadre d'un ensemble de tels critères nous présentons un ensemble de mesures appliquées à un modèle probabiliste. En outre, nous proposons un nouveau critère sur la base d'une simplification du modèle probabiliste de clustering.

Critère d'information bayésien. Les données sont estimées habituellement en deux étapes de la procédure d'inférence. Dans la première étape, en supposant que les données obéissent à un modèle, nous estimons des paramètres du modèle. Cette estimation est effectuée pour chaque modèle. Dans la deuxième étape en utilisant les paramètres, on compare les modèles et sélectionne le meilleur d'entre eux. Cette procédure à deux étapes est soutenue par le fait que plus le modèle est complexe, mieux cela correspond aux données. Nous devons trouver un compromis entre le modèle et sa complexité. La première étape de modélisation a été examinée dans le Chapitre précédent par l'estimation de maximum de vraisemblance du GMM. La sélection du modèle peut être fait via une approche théorique basée sur le théorème de Bayes [Mackay, 2002]. La minimisation du critère d'information bayésien (BIC) montre le modèle optimal. Akaike a proposé son critère AIC qui est similaire de BIC, mais il est dérivé sur une autre base théorique. BIC propose souvent de choisir des modèles plus simples parce qu'il a une plus grande pénalité que AIC [Friedman et al., 2001] ; AIC a tendance à surestimer un modèle (choisir le modèle le plus complexes) [Mclachlan & Peel, 2000].

Critère de description de longueur minimale (minimum description length, MDL) Comme nous travaillons sur un ensemble fini et discret de données modélisées par une fonction

de la densité, on peut considérer leur logarithme négatif. Ce logarithme est un code entier. Puis, pour ces modèles le code de longueur peut être écrit [Rissanen, 1995]. Ce code est aussi appelé la complexité stochastique d'information du modèle (SIC, stochastic information criteria). L'optimisation de ce critère conduit à la sélection du meilleur modèle de données. La sélection du modèle peut être également démontée à l'aide du code du modèle. Le modèle prévoit une probabilité de données appropriée. L'utilisation de l'entropie de Shannon permet calculer la quantité d'information du modèle. La mesure de cette information est exprimée en bits. Plus de bits sont utilisés pour représenter l'information plus complexes. Par conséquent, le plus complexe est le modèle. Il existe un codage universel proposé dans [Rissanen, 1984] qui a deux parties :

- ★ Partie 1. Description de longueur du modèle décrit le modèle et ses paramètres.
- ★ Partie 2. Description des données décrit la longueur des données sachant le modèle et ses paramètres.

MDL pour le modèle de mélange des Gaussiens détermine le nombre optimal de clusters. Le clustering est obtenu par EM-algorithme qui estime les paramètres de GMM. En outre, nous écrivons critère MDL pour le regroupement "dur" lorsque chaque échantillon appartient à un seul groupe. Pour cela, nous développons le logarithme de la vraisemblance complet de GMM en introduisant une variable supplémentaire qui indique le regroupement "dur". Cette considération conduit à quelques simplifications de critère MDL. La simplification de MDL peut être développée et appliquée aux autres algorithmes (e.g.: K-means à noyau). En outre, des nouveaux algorithmes hiérarchiques sont dérivés de MDL simplifié.

Nous proposons le critère MDL à noyau via une formulation simplifiée de MDL. La distance entre les échantillons (l'erreur) peut être calculée en l'espace d'origine, ainsi que dans l'espace transformé en utilisant un noyau. L'un des principaux avantages de cette formulation est que la moyenne explicite d'un groupe n'est pas nécessaire. Ce point est important quand cette moyenne n'a pas de sens physique, comme c'est souvent le cas pour les clusters non-convexes. Les expériences avec des données synthétiques et des données réelles (des images satellitaires) sont démontrées dans ce Chapitre.

Un regroupement hiérarchique non supervisé basée sur KMDL est proposé dans cette Chapitre. En outre, nous développons deux nouveaux algorithmes de regroupement hiérarchique. Le premier utilise le critère MDL et la deuxième KMDL. Nous formulons un algorithme de clustering hiérarchique qui optimise GMDL. Notre proposition est similaire à celui présentée dans [Heas & Datcu, 2005] mais se distingue par le critère. Dans [Heas & Datcu, 2005] les auteurs proposent un algorithme hiérarchique qui optimise MDL par la combinaison de deux groupes à chaque étape (un niveau de la hiérarchie). L'idée de cette approche est de regrouper les données en grand nombre de "petits groupes" et ensuite optimiser hiérarchiquement le critère MDL pour trouver le nombre optimal de clusters de données. Au lieu de calculer MDL pour chaque nombre de clusters ils considèrent une hiérarchie de modèles et analysent MDL. Nous utilisons la même approche, mais pour le critère proposé GMDL.

Le choix d'une représentation optimale des données à chaque niveau de la modèle hiérarchique est décrit par le critère GMDL. Au lieu de calculer directement ce critère à chaque niveau de la hiérarchie et rechercher sa valeur optimale, il est préférable d'examiner son gradient. La valeur minimale du gradient montre le meilleur GMDL, ainsi que la direction où cet optimum peut être trouvé. En outre, le gradient réduit le temps de calcul et le volume de données traitées.

Dans ce Chapitre, le problème de sélection de modèle a été pris en considération. Différents critères ont été indiqués pour les algorithmes de regroupement hiérarchique, partitionnel et probabilistes. Pour le regroupement hiérarchique un critère basé sur la matrice cophenetic a été présenté, alors que pour le regroupement partitionnel les critères intra- et inter- ont été discutés. Pour les modèles probabilistes comme le modèle de mélange des gaussien les critères théoriques AIC, BIC, MDL et SIC ont été révisés. La similitude entre ces critères a été démontrée. Le regroupement des données a été examiné par GMM avec l'estimation de ses paramètres par l'algorithme EM. La simplification de MDL pour le regroupement "dur" et GMM a été proposée. Le critère MDL simplifié peut être appliqué par K-means algorithme ainsi que par sa modification comme K-means à noyau ou K-means spectrale. L'avantage de K-means à noyau, c'est qu'il permet de séparer les groupes qui ne sont pas séparables linéairement.

L'algorithme hiérarchique basé sur MDL simplifié a été proposé dans ce Chapitre. Ce regroupement hiérarchique est non-supervisé et permet de déterminer le nombre optimal de clusters. Cet algorithme a été élargi afin de formuler le regroupement hiérarchique basé sur MDL à noyau. L'avantage de cet algorithme est qu'il est en mesure de trouver le nombre optimal de clusters et de séparer des groupes qui ne sont pas linéairement séparables.

Combinaison de regroupements (clusterings)

Une étude sur les dernières méthodes de combinaison de clustering est présentée dans ce Chapitre. Elle couvre un large éventail d'approches : de bien formulée avec des bases théoriques à des approches empiriques. La combinaison de clusterings est considérée dans cette thèse comme une tâche non supervisée car nous visons à éviter l'interaction avec l'utilisateur, soit parce qu'il peut prendre beaucoup de temps pour un utilisateur d'analyser les clusterings, soit parce qu'il est très difficile de les interpréter. Nous proposons de combiner les clusterings en utilisant les algorithmes de regroupement (clustering). Nous verrons que chaque approche a ses avantages et ses inconvénients. Les inconvénients nous motivent à poser le problème de la combinaison d'une manière nouvelle afin de les éviter. Après la formulation du problème, nous proposons deux méthodes non-supervisées et deux algorithmes pour combiner les différents clusterings. La première méthode, malgré son efficacité, n'a pas de preuve claire sur la convergence à la solution unique et globale. Concernant la deuxième méthode, le même problème est reformulé, qui mène la solution globale. En outre, un algorithme est proposé pour trouver cette solution. Une preuve de la convergence de cet algorithme à la solution unique et globale est dérivée. Les avantages des méthodes proposées sont examinés. Les résultats sur les données synthétiques et réelles sont fournis à la fin de ce Chapitre.

Nous introduisons le problème de la combinaison du regroupement avec l'application à l'analyse d'images satellitaires. Dans les dernières années, de nombreux capteurs différents d'imagerie de télédétection ont fourni un énorme quantité d'images numériques. L'une des approches d'analyse automatique d'images par des concepts est le regroupement. Dans ce cas, les concepts sont les clusters. Il y a une variété d'algorithmes de regroupement. Chacun d'entre eux a ses avantages et ses inconvénients. Certains algorithmes sont robustes et regroupent est correcte même en cas de fort bruit, mais ils peuvent ne pas être sensibles aux données avec une structure complexe. Au contraire, d'autres algorithmes sont en mesure de trouver les vrais groupes dans les données avec des structures complexes, mais l'influence du bruit sur les résultats de groupement est très forte. La

question est comment choisir ou combiner les algorithmes de clustering. C'est une pratique courante lorsque plusieurs regroupements sont effectués en parallèle, soit parce qu'il y a différents algorithmes, soit parce qu'on change différents paramètres du même algorithme. Différents clusterings fournissent des résultats complémentaires desquelles nous voudrions bénéficier [Fred & Jain, 2005; Strehl & Ghosh, 2002; Topchy et al., 2004a; Ayad & Kamel, 2005; Boulis & Ostendorf, 2004; Li et al., 2004; Y. Qian, 2000]. La principale difficulté est de déterminer un critère judicieux pour combiner les clusterings élémentaires afin d'obtenir une solution finale du regroupement. Un autre problème est comment appliquer efficacement la méthode choisie dans le cas de très grandes bases de données. La contribution de ce Chapitre est d'aborder ces deux problèmes.

Différentes méthodes peuvent être utilisées pour fusionner l'information provenant des différents regroupements [Diday, 1979; Michaud & Marcotorchino, 1979; Kuncheva, 2004; Marcotorchino & Michaud, 1982; Fred & Jain, 2005; Strehl & Ghosh, 2002; Topchy et al., 2004a]. Nous considérons deux approches dans ce Chapitre :

1. combinaison probabiliste,
2. combinaison algébrique.

L'approche probabiliste considère les clusterings comme les données nominales et la combinaison est réalisée par le regroupement non supervisé. L'approche algébrique utilise une matrice de la représentation de clusterings. Les méthodes algébriques sont basées sur la propriété de deux échantillons d'appartenir ou non au même groupe, selon le type de regroupement. Une étude de ces méthodes est donnée dans ce Chapitre. Nous utilisons un critère de combinaison pour l'approche algébrique avec les développements mathématiques. Nous décrivons un algorithme de la combinaison et nous proposons de l'améliorer afin de traiter des données réelles de manière efficace. Le critère de combinaison proposé permet de trouver le optimum globale de la combinaison par un algorithme itératif "mean shift". Les résultats de combinaison sur les données synthétiques et les données réelles sont présentés et discutés. Enfin, l'estimation de la stabilité de regroupement est discutée.

Beaucoup de méthodes ont besoin d'information a priori sur les données afin de combiner les clusterings ou régler manuellement les paramètres de la combinaison. Cela nous motive à poser le problème dans une forme qui ne dépend d'aucun paramètre et connaissances préalables.

Tout d'abord, nous proposons de considérer l'ensemble des clusterings comme le clustering des données nominales. Plusieurs algorithmes peuvent être appliquées : de K-means à EM-algorithme avec un mélange des modèles multinomiaux. La combinaison optimale de ces algorithmes peut être choisi par le critère MDL. Deuxièmement, nous formulons la combinaison en utilisant la matrice de co-association. Elle permet de traiter de gros volumes de données ainsi qu'un grand nombre de classes sans utiliser la matrice de co-association explicitement. Nous proposons une fonction objective et deux algorithmes pour combiner différents clusterings. Le premier algorithme utilise une approche hiérarchique non supervisée et montre des performances compétitives par rapport à ceux qui existent déjà. Il combine les clusterings qui ont un grand volume de données. Malheureusement, il n'existe pas de preuve qu'il réalise toujours un optimum global. Le second algorithme de combinaison est rapide et itératif dont nous prouvons la convergence vers le optimum globale. Il surpasse expérimentalement les combinaisons issues des approches proposées dans la littérature.

Le clustering des données nominales. La combinaison peut être considérée comme le clustering nominale (ou catégoriel) des données, ou les étiquettes des clusterings sont les données nominales. Les algorithmes de regroupement peuvent être appliqués afin de trouver une solution de la combinaison de clustering, e.g., par le regroupement "dur" (K-means [Diday, 1979]) ou par la modélisation probabiliste avec l'algorithme EM [Bishop, 2006; McLachlan & Peel, 2000; Hardle et al., 2003]. Dans ce Chapitre les algorithmes comme K-means, spectral K-means et K-means à noyau ont été pris en considération. Les algorithmes regroupent les données continues pour lesquelles la distance Euclidienne ou une autre distance (e.g., à noyau) sont déterminées [Shawe-Taylor & Cristianini, 2004]. Les données nominales doivent être transformées en un ensemble de données binaires afin d'appliquer les distances et les algorithmes de regroupement comme pour les données continues [Diday, 1979; McLachlan & Peel, 2000; Bishop, 2006]. D'autre part, l'approche probabiliste avec l'algorithme EM peut être appliquée directement aux données nominatives (mais pour des raisons de commodité, nous avons transformé les données afin de montrer clairement les calculs des probabilités).

Regroupement par la partition (partitional clustering) Les algorithmes de regroupement par la partition tels que K-means peuvent être choisis afin de regrouper des données nominales, e.g., des étiquettes ou des noms. Dans notre cas, les données sont les groupes d'étiquettes nominatives. Comme les clusterings peuvent être présentés par des matrices binaires, on peut appliquer les algorithmes de regroupement des données binaires. Une telle approche peut être trouvée dans [Govaert & Nadif, 2007], où K-means et l'algorithme EM avec un modèle de mélange multinomial sont comparés. Ce travail n'est pas considéré pour la combinaison de clusterings, mais il a de nombreux points communs avec ce problème. La comparaison des K-means et EM-algorithme est donnée dans le Chapitre prochain. Dans ce Chapitre, nous montrons comment une approche probabiliste peut être appliquée afin de combiner des différents clusterings. Les modèles probabilistes et l'estimation de ses paramètres par l'algorithme EM seront aussi effectués.

Distribution binomiale. La combinaison de clusterings peut être considérée comme le regroupement de données nominatives via la modélisation probabiliste. Dans ce Chapitre, nous proposons une étude du modèle probabiliste de Bernoulli qui donne un passage au modèle multinomial qui est plus générale. Cette modélisation est basée sur un mélange de distributions de Bernoulli avec l'estimation des paramètres du mélange. Chaque mélange correspond à un ensemble de données nominales. Nous donnons le modèle de mélange de distributions de Bernoulli et l'algorithme EM qui est utilisé pour estimer des paramètres du mélange. L'approche probabiliste comprend :

1. représenter des clusterings par une matrice binaire,
2. modéliser des données binaires via le modèle de mélange de Bernoulli,
3. appliquer l'algorithme EM afin de trouver des groupes des clusterings,
4. sélectionner le meilleur modèle en utilisant des mesures d'information.

Il est bien connu que l'algorithme EM produit le résultat optimal et local. En outre, la classification de cet algorithme dépend de l'initialisation. Pour éviter ces problèmes, la solution suivante peut être proposée : sélectionner la meilleure classification via le modèle de mélange et le critère MDL pour différentes initialisations et le nombre des composants du mélange (le nombre de clusters).

Modèle de mélange multinomial. La distribution de Bernoulli pour les données binaires suppose que les variables binaires (clusterings) sont indépendants, mais pas mutuellement exclusives. Pour ce dernier cas, les données binaires ainsi que les matrices concaténées doivent être modélisées par la distribution multinomiale. Les groupes (les clusters) de clusterings peuvent être trouvés par l'algorithme EM via la modélisation probabiliste. Nous considérons un modèle de mélange des distributions multinomiales. Sans perte de généralité, nous considérons que la classification obtenue (non-supervisée) est une combinaison de clusterings et les classes trouvées représentent des groupes de clusterings. Nous devrions faire une différence entre les distributions binomiales et multinomiales. Les modèles multinomiaux généralisent la distribution binomiale. Toutefois, les deux modèles peuvent être appliqués aux données nominales et très souvent donnent les mêmes résultats. Là encore, les mêmes problèmes se posent lorsque EM-algorithme est utilisé : les résultats optimaux locaux, la dépendance de l'initialisation, la sélection du meilleur modèle et l'estimation du nombre de composants du mélange. Ces problèmes peuvent être résolus via l'estimation du modèle par le critère MDL.

Combinaison par la matrice de co-association. Dans cette section, nous proposons d'étudier les solutions de combinaison de clustering en utilisant la matrice de co-association. Nous présentons également de nouvelles méthodes pour la combinaison afin d'éviter les inconvénients des approches existantes. L'idée de la combinaison proposée est de regrouper les échantillons qui sont dans le même cluster dans la plupart des clusterings. Tout d'abord, nous montrons une fonction objective pour combiner différents clusterings. Ensuite, nous développons un algorithme hiérarchique pour optimiser la fonction objective. Un tel algorithme est compétitif par rapport aux autres algorithmes de combinaison, mais en dépit de ses très bons résultats, il ne garantit pas la convergence vers la solution globale. Après une analyse de la fonction objective, nous proposons une méthode améliorée qui donne la solution globale. De plus, nous décrivons les conditions d'une telle convergence.

Il est intéressant de noter que tel regroupement peut être exprimé en termes de réduction de l'erreur quadratique entre les échantillons présentés par des étiquettes de clusterings. Nous prouvons dans ce Chapitre que la solution globale de l'erreur quadratique minimale peut être trouvée en utilisant le gradient de l'estimation d'une fonction de densité. Tous les modes locaux de la densité forment des groupes d'échantillons et, par conséquent, constituent la solution globale de l'ensemble. L'un des avantages d'une telle méthode est que l'algorithme a une convergence rapide et une complexité linéaire. C'est un avantage important quand une grande quantité de données doivent être traitées comme dans le cas du traitement d'image satellitaire. La combinaison de clusterings est effectuée sur les données synthétiques et réelles. L'efficacité de la méthode proposée et sa supériorité par rapport aux autres approches sont démontrées. Les limites de l'erreur quadratique E de combinaison sont montrés. Ils ont la relation avec la décomposition des valeurs et des vecteurs propres de la matrice de co-association. Mais, pour le problème à part, comme pour le cas précédent, la solution par les vecteurs propres ne permet pas d'assurer l'association a des valeurs positives. Une autre solution peut être envisagée via la décomposition de Cholesky de la matrice de co-association. Le résultat de telle transformation est très dépendant des permutations des lignes et des colonnes de la matrice. Une troisième approche peut être vue par la programmation quadratique. Malheureusement, pour des applications réelles, e.g., la classification d'images, la complexité devient cruciale : il est très difficile de travailler avec une matrice carrée. Ce problème a une formulation non-convexe et quadratique qui est très difficile à résoudre. Toutefois,

il existe des méthodes d'optimisation quadratique visant à trouver un minimum local [Floudas & Visweswaran, 1994].

La solution proposée.

Afin de combiner les clusterings et de trouver une solution consensuelle qui minimise l'erreur quadratique nous proposons d'utiliser l'algorithme single-link [Jain & Dubes, 1988]. Cet algorithme a été expérimentalement démontré de produire très bons résultats. Le nombre optimal de groupes est observé lorsque l'erreur quadratique est minimale. L'un des moyens les plus simples pour aller vers un minimum est une méthode de gradient, qui part d'une bonne initialisation et de façon itérative minimise l'erreur du consensus. Le gradient de l'erreur réduit le temps de calcul ainsi que le volume stocké des données traitées. Nous avons présenté la fonction objective et l'algorithme hiérarchique qui trouve le meilleur consensus des clusterings. Malheureusement, il n'existe pas de preuve que ce algorithme hiérarchique peut atteindre un optimum de la fonction objective. Pour surmonter cette limitation, nous reformulons le processus d'optimisation ainsi que les conditions d'optimalité et nous proposons un algorithme exact pour trouver le minimum de l'erreur. Les exemples de la combinaison sont démontrés.

Combinaison par un algorithme mean shift. Nous proposons de trouver un consensus de clusterings qui, comme précédemment, minimise l'erreur quadratique. Nous prouvons dans ce Chapitre que cette minimisation est équivalente à la réduction de l'erreur quadratique entre les échantillons de clusterings. Une approche non paramétrique pour trouver une solution est l'objectif presque de toutes les tâches de traitement de l'information. La base d'une telle approche en ce qui concerne la reconnaissance des formes est l'estimation non paramétrique de la densité par son gradient [Fukunaga, 1990; Comaniciu & Meer, 2002], ce que l'on appelle l'estimation de la densité par l'algorithme de mean shift. Nous montrons également les propriétés de cet algorithme qui garantissent leur convergence rapide en nombre fini d'itérations.

Théorème. *Le noyau d'Epanechnikov est le meilleur noyau pour trouver le minimum global d'erreur E par l'algorithme mean shift. Un radius adaptatif et optimal est calculé pour la combinaison par mean shift. Les aspects pratiques de mean shift sont aussi discutés :*

- ★ accélération de mean shift par les initialisations appropriées,
- ★ attribution des échantillons aux clusters,
- ★ calcul d'erreur E ,
- ★ fusion des vecteurs mean shift.

Dans ce Chapitre nous présentons une comparaison de différentes méthodes qui est réalisée sur les données synthétiques et réelles. Nous effectuons des expériences de combinaison sur les données réelles de "UCI machine learning repository". En outre, nous comparons les résultats avec les travaux de [Fred & Jain, 2005], où le critère NMI normalisé est étudié. Le but de ces expériences est de montrer que la combinaison obtenue par les algorithmes proposés est compétitive et même mieux que par l'approche proposée dans [Jain & Dubes, 1988]. La combinaison peut être utilisée pour de nombreuses applications de fouille de données : le regroupement des données nominales (e.g., des documents textuelles), la combinaison des différents clusterings ou des segmentations de la même scène, (e.g., en regroupant des différents clusterings de séries temporelle d'images), le regroupement de la vidéo, la détection de mouvement, etc. On peut également stabiliser les résultats de clustering par une mesure de la stabilité.

Dans ce Chapitre, le problème de combinaison de clustering a été pris en considération. L'étude sur les travaux précédents a été faite. Plusieurs algorithmes récents de combinaison ont besoin d'un réglage des paramètres. Combinaison de clustering a été présenté par le regroupement des clusterings. La simple combinaison a été obtenue par K-means algorithme appliqué à la représentation binaire des clusterings. L'équivalence des différentes mesures a été illustrée.

La combinaison plus complexe basée sur l'approche probabiliste a été également prise en considération. Dans ce cas, les clusterings sont considérés comme des données nominales et sont modélisés soit par les mélanges de Bernoullis, soit par les modèles multinomiaux. L'estimation des paramètres du modèle a été faite par l'algorithme EM. Le meilleur modèle probabiliste peut être choisi par le critère MDL. Il a été noté que les mélanges probabiliste souffrent d'initialisations aléatoires de paramètres ce qui donne les résultats de combinaison différents.

Les inconvénients des méthodes analysées nous ont motivés de poser le problème de combinaison comme une tâche non supervisée. La solution pour la combinaison est basée sur la matrice de co-association. La distance quadratique entre le consensus et les clusterings a été utilisée. Deux algorithmes pour optimiser ce critère ont été proposés. Le premier est un algorithme hiérarchique et le second est un itératif. Malgré la bonne performance d'algorithme hiérarchique il n'existe pas de preuve qu'il peut atteindre la solution globale. Au contraire, il a été prouvé (Théorème 1) que l'algorithme itératif de mean shift trouve la solution optimale de la combinaison. Les aspects pratiques d'application de combinaison ont été examinés.

Enfin, quelques mesures pour estimer la stabilité de clustering ont été proposées. Ils indiquent la stabilité des échantillons, des clusters et des clusterings.

Combinaison des clusterings et l'analyse d'images

Dans ce Chapitre, nous démontrons quelques exemples d'application de combinaison. Au début, nous donnons une courte liste des applications avec des brèves explications. Puis nous comparons les performances des algorithmes de combinaison afin de démontrer l'efficacité de méthodes proposées. Différents critères d'évaluation de combinaison sont donnés : supervisé et non-supervisés. Le critère supervisé n'est utilisé que pour comparer les résultats de la combinaison à la classification connue. Les critères non-supervisé sont les critères utilisés pour évaluer la combinaison optimale sans la connaissance préalable de la classification. Les résultats de la comparaison sont discutés. Les applications possibles sont démontrées sur des images.

Nous donnons maintenant une liste des applications de combinaison proposées dans ce Chapitre :

1. Comparaison des méthodes de combinaison. Les performances des différents algorithmes de combinaison et leurs fonctions objectives sont comparées.
 2. Combinaison par le regroupement. Un schéma de la combinaison par le regroupement est donné.
 3. Combinaison de segmentations d'images satellitaires . Un exemple de combinaison de segmentations d'images est présenté.
 4. Combinaison d'images avec des artefacts. Tout d'abord, nous montrons un exemple synthétique pour supprimer des artefacts, puis des images satellitaires segmentées
-

avec des nuages sont utilisées.

5. Détermination du nombre optimal de clusters dans les séries d'images.
6. Combinaison pour la reconstruction d'images (image deblurring). Une brève discussion sur la reconstruction d'images est donnée.
7. Regroupement des données nominales. La combinaison est considérée comme le regroupement de données nominales.
8. Combinaison pour la sélection des caractéristiques. Une méthode non supervisée de sélection des caractéristiques est présentée.

Nous donnons un bref résumé des expériences sur la combinaison des données. Différents algorithmes de combinaison ont été comparés dans ce Chapitre. Nous avons vu que l'erreur quadratique E et le critère MDL montrent le nombre optimal de groupes contrairement au critère NMI [Fred & Jain, 2005] qui ne parvient pas dans certains cas. En outre, il faut noter que l'application directe d'algorithme single-link peut prendre beaucoup de temps et de mémoire pour les grands nombres de données en raison de la complexité quadratique. Au contraire, K-means ou algorithme EM ont le temps, la mémoire et la complexité linéaires et peuvent être appliqués pour tester rapidement les combinaisons. Mais ils souffrent des initialisations dans le cas d'un grand nombre de groupes. Au contraire, la combinaison effectuée par le mean shift pour tous les cas synthétiques fournit la combinaison exacte.

Dans ce Chapitre différents exemples de combinaison de clustering ont été pris en considération. La comparaison des différents algorithmes de combinaison est effectuée. Nous concluons que la fonction objective et l'algorithme MSC prouvent pratiquement leur supériorité par rapport aux autres fonctions objectives et des algorithmes de combinaison. L'efficacité de la combinaison est montrée via :

1. les erreurs de classification et de clustering,
2. la stabilité des solutions,
3. le temps de calculs.

Les applications suivantes pour l'analyse des images ont été prises en considération : la combinaison de différentes segmentations, l'estimation des paramètres et la détection d'objets. Pour l'analyse des données en général, la combinaison permet de combiner des données nominales, estimer et de trouver des modèles stables, analyser et caractériser la stabilité des clusters et clusterings. Une application importante de combinaison consiste à la sélection non supervisée des caractéristiques et montre des bons résultats. D'autres expériences sont données dans le Chapitre suivant où la fouille de données est appliquée aux images multimédia et aux images satellitaires.

La sémantique d'images

Dans ce Chapitre, nous abordons un problème de la construction de la sémantique pour des images. La sémantique peut être considérée comme un ensemble de concepts et de relations entre eux [Suykens & Horvath, 2002]. Cette représentation permet de montrer

une variété de connaissances sur les images (concepts-relations) dans une forme compacte. En outre, la sémantique peut être utilisée pour la gestion des images (classifications, clustering, requête, etc.)

Deux types d'images sont pris en considération pour les expériences dans ce Chapitre : (i) les images multimédia et (ii) les images satellitaires. Nous commençons à construire la sémantique pour les images multimédia. Cette expérience a été réalisée partiellement supervisée (pour obtenir différentes classifications) et partiellement non supervisée (en combinant les classifications). L'objectif de cette expérience est de vérifier l'approche proposée pour la construction de la sémantique. Les images multimédia ont été utilisées en raison de leur interprétation facile par les utilisateurs.

L'une des premières études sur la construction de la sémantique est détaillée dans [Gotlieb & Kumar, 1968]. Les auteurs proposent d'analyser le vocabulaire indexé, où chaque indice exprime une collection de mots ou de phrases. Récemment, la construction de la sémantique d'images est devenue très connue [Kuhn et al., 2007; Carneiro et al., March 2007]. Dans [Kuhn et al., 2007], un regroupement sémantique est proposé, basé sur l'indexation sémantique latente avec le regroupement des éléments textuels qui partagent le même vocabulaire. Les clusters représentent des sujets sémantiques avec des liens entre eux et il sont visualisés sur un graphe 2^D . Une étude de haut niveau du contenu sémantique basée sur la recherche d'images est donnée dans [Liu et al., 2007].

La sémantique de données textuelles est similaire à la construction de la sémantique pour les images, mais pour les images :

1. il n'y a pas de vocabulaire (index) d'images,
2. il n'y a pas de connaissances a priori comment les indices sont liés et comment ils se regroupent pour former des concepts,
3. il n'existe aucun rapport sémantique entre les concepts.

En dépit du manque d'information, nous avons deux hypothèses : (i) les images peuvent être interprétées et (ii) les images représentent l'information utile. En effet, nous sommes capables de détecter dans les images multimédia des objets, des types de textures, des couleurs et donc de classer les images en différents groupes. En outre, il est possible de décrire les images par les mots. Toutes ces hypothèses permettent d'appliquer les méthodes non supervisées de traitement d'images pour extraire des termes d'images (index), des concepts d'images et des relations entre les concepts. La représentation des résultats de clustering est discutée dans ce Chapitre.

Visualisation de clusterings. L'un des objectifs de l'analyse des données non-supervisées est la détection des formes. Lorsque nous avons un grand volume de données, on peut avoir, probablement, de nombreux groupes (des dizaines ou centaines). La navigation dans les résultats devient une tâche plutôt difficile. Pour cela, nous devons extraire d'information provenant du regroupement obtenu, e.g., estimer des paramètres de groupes, des relations entre eux, des degrés de connexions, etc. Différentes distances peuvent être considérées comme les relations entre les groupes (clusters), e.g., la distance Euclidienne. Pour la visualisation, les clusters peuvent être considérés comme des concepts et, pour la simplicité, représentés comme des noeuds, tandis que les relations entre eux peuvent être considérées comme des arêtes qui relient les noeuds. Deux représentations sont possibles : les arbres et les graphes. Un arbre est un graphe sans boucles, non orienté avec un seul noeud en haut et les feuilles à la base. Un arbre peut généraliser les clusters dans un concept. Cette représentation est très utile afin d'analyser comment chaque groupe (noeud)

est liée à d'autres groupes (noeuds) et de mesurer le degré de cette relation. Les arbres et les graphes peuvent être extraites à partir de la matrice des distances ou des similitudes. Un exemple de représentations de regroupement ainsi que l'analyse est donné dans ce Chapitre.

Extraction de relations entre les concepts. Dans ce Chapitre nous introduisons les relations entre les concepts représentés par des arbres et des graphes. Nous concentrons notre attention sur les cas où les données sont regroupées par les algorithmes de classification non supervisée. Les clusterings peuvent être combinées pour obtenir un clustering de consensus. Les relations entre les clusters obtenus de la matrice de co-association sont exploitées dans ce Chapitre. L'algorithme de mean shift estime la combinaison optimale des clusterings. Dans la littérature la relation est exploitée par l'algorithme de single-link [Fred & Jain, 2005] qui sélectionne deux groupes voisins (les plus proches).

Pour construire le graphe de relations, un produit de vecteurs entre les moyens de clusters de consensus est calculé sur la matrice de co-association. L'importance des relations entre les clusters est affichée par l'épaisseur de l'arête : la relation la plus importante a l'arête la plus épaisse.

La sémantique d'images multimédia. Dans ce Chapitre la sémantique d'images multimédia est prise en compte. Les exemples d'analyse sont également donnés. L'idée derrière cette expérience est de demander à plusieurs observateurs de classer un ensemble fini d'images, ensuite, d'exploiter l'ensemble des classifications sémantiques et de tirer des concepts d'images. Cette expérience, en cas de succès, soutiendra l'idée que la sémantique peut être émergée d'un consensus de regroupement. Pour le test 45 images multimédia contenant une variété de sujets ont été sélectionnés. Chaque utilisateur est invité à classer 45 images en fonction de son propre "meilleur critère". Chaque utilisateur peut choisir le plus grand nombre de clusters (ou le plus petit !) et classe les images comme il (elle) veut. En outre, l'utilisateur est invité à donner un nom à chaque classe, et à la fin d'annoter avec un vocabulaire libre chaque classe. L'intérêt de cette expérience est d'avoir des classifications indépendantes de différents utilisateurs ou chaque classification est pertinente et "bonne" comme les autres.

L'objectif de cette expérience est de trouver une classification consensuelle parmi les 50 différentes, fournies par les utilisateurs. Les classifications reflètent les points de vue indépendants et dans le même temps, ils ont une information commune. La combinaison reflète les groupes des classes données par les utilisateurs.

Dans la deuxième partie, nous explorons les classifications des images multimédia, mais au lieu de combiner les étiquettes des classifications nous analysons les descriptions textuelles associées aux classifications. Rappelons que chaque utilisateur, après avoir classé les images avec son propre nombre de classes et ses propres classes, il a été demandé de décrire la description de chaque classe avec un ensemble de mots.

L'objectif de cette expérience est d'analyser et de combiner des descriptions obtenues de différents utilisateurs. Comme auparavant, le résultat de la combinaison fournit la sémantique d'images. En outre, nous comparons la combinaison de classifications et la combinaison des descriptions.

Une expérience sur une combinaison de classifications visuelles et les mots des descriptions d'images a été présentée dans ce Chapitre. Elle ouvre de nouvelles directions dans l'analyse des données. Elle montre également que la tâche de la fouille de données peut être résolue par des approches différentes et illustre la concordance des résultats d'extraction de données. La partie intéressante de l'expérience est que l'analyse des descriptions a été faite entièrement non supervisée. Nous n'avons pas l'information a

priori sur le nombre de clusters et le nombre d'images qui sont distribués dans les clusters. L'information visuelle et textuelle reflète la même représentation sémantique ou en d'autres termes le même sens. La sémantique confirme également la pertinence de l'analyse non-supervisé.

Les clusters ont été considérés comme des concepts. La représentation sémantique des données est en mesure d'indiquer les connexions entre les concepts. En outre, elle peut être exploitée plus précisément pour la recherche ou la fouille de bases de données. Enfin, l'utilisateur peut construire sa propre sémantique des classes par l'analyse des connexions entre les clusters, par les graphes ou par les arbres. Nous devons noter que ce type d'expérience n'est pas limité aux images et peut être appliqué aux différents types de données.

La sémantique d'images satellitaires

Dans cette partie, nous proposons d'aller plus loin dans la fouille de données non supervisée et l'appliquer sur le grand ensemble d'images satellitaires, afin de sortir la sémantique.

Pour cette expérience, la participation d'utilisateurs n'est pas impliquée et toutes les opérations sont entièrement non supervisées. Nous donnons maintenant les étapes essentielles de l'expérience réalisée :

1. Extraction des caractéristiques d'images satellitaires .
2. Sélection non supervisée des caractéristiques.
3. Clustering de données par différents algorithmes non-supervisés.
4. Estimation non supervisée du nombre de clusters pour chaque algorithme.
5. Combinaison non supervisée de différents clusterings.
6. Construction non supervisée de la sémantique d'images satellitaires via la combinaison de regroupement (clustering).

Nous proposons d'analyser les images satellitaires SPOT5 de différentes villes. Les images ont une variété de surfaces qui ont été séparées par le regroupement non-supervisé : ville, terrain, mer, etc. Les images ayant le contenu très complexe sont également prises en compte. Ce contenu est représenté par les zones urbaines qui sont difficiles à distinguer par la sémantique.

Dans ce Chapitre, la notion de la sémantique d'images, ses principes, sa construction et l'analyse ont été présentées. Les exemples de la sémantique sont montrés sur les images multimédia et les images satellitaires. Nous avons démontré comment tirer la sémantique d'images dans le mode non-supervisé et nous l'avons justifiée par une connaissance préalable. Pour les images multimédia à la fois la perception visuelle et la description textuelle ont montré la sémantique pertinente. La sémantique d'images satellitaires a été approuvée par l'interprétation visuelle. La visualisation de combinaison a été faite via les arbres et les graphes des structures.

Dans la première expérience, nous avons construit la sémantique des images multimédia. Dans la deuxième expérience, nous avons présenté les résultats de la combinaison de clusterings des villes. La troisième expérience a été réalisée entièrement non supervisée sur des images satellitaires. La combinaison de différents clusterings a été utilisée afin d'inférer la sémantique d'images satellitaires qui a été représentée par les arbres et les graphes. La perception visuelle du regroupement correspond à la structure

sémantique et justifier la pertinence de l'approche proposée.

Conclusion

Dans cette thèse une approche non-supervisé de fouille de données appliquée aux images satellitaires optiques à haute résolution a été proposée. L'idée générale de la fouille de données comprend l'extraction d'information à partir d'images, la modélisation par les algorithmes de regroupement (clustering), la combinaison de différents clusterings et la représentation des clusterings par une structure sémantique. Un prototype du logiciel avec l'interface graphique pour la fouille d'images satellitaires a été développé.

Résumé La fouille de données non supervisée développée dans cette thèse a été évaluée sur des images satellitaires. Toutefois, l'idée générale de cette approche peut être facilement appliquée aux autres types de données. L'accent de la thèse a été mis sur les méthodes non supervisées en raison de la taille des bases de données qui nécessitent d'être fouillées sans l'interaction humaine afin d'obtenir la modélisation objective.

Nous proposons d'appliquer différents algorithmes pour la modélisation des données, puis de combiner leurs résultats, au contraire de nombreux ouvrages similaires qui appliquent un algorithme unique pour fouiller des données. Certains algorithmes de regroupement sont algébriques, d'autres sont probabilistes.

Nous résumons maintenant les nouvelles idées présentées dans cette thèse :

- ★ Extraction de caractéristiques géométriques d'images satellitaires.
Les caractéristiques sont basées sur les statistiques des bords détectés dans les images. En outre, un ensemble des caractéristiques de texture sont extraites à partir d'images : les descripteurs d'Haralick, les coefficients de Gabor et les caractéristiques QMF.
 - ★ Le problème de la malédiction de la dimension oblige à sélectionner les caractéristiques pertinents. Une nouvelle méthode non-supervisée de la sélection des caractéristiques qui est basée sur leur regroupement a été proposée. Cette approche découle de la combinaison des différents clusterings de l'espace des caractéristiques.
 - ★ Le critère de la Longueur de Description Minimale (minimum description length, MDL) estime les meilleurs regroupements et le nombre optimal de clusters.
 - ★ Les nouveaux algorithmes hiérarchiques ont été dérivés à partir du critère simplifié MDL, adapté pour le K-means à noyau. Les algorithmes sont basés sur l'optimisation du gradient du critère MDL.
 - ★ Une nouvelle méthode non-supervisée de combinaison de clusterings est prouvée d'atteindre la solution globale. Elle est basée sur l'estimation de la densité par l'algorithme "mean shift".
 - ★ Tous les clusterings sont présentés par un arbre ou un graphe. Cette représentation permet à l'utilisateur de visualiser des résultats de regroupement et d'apprendre les structures de données.
-

Les expériences ont été menées sur différents types de données comme les images multimédia ou les images satellitaires. Expérimentalement, nous avons montré que les arbres et les graphes reflètent la sémantique des données.

Les perspectives Plusieurs sujets de recherche peuvent être issus de la thèse. L'un des principaux inconvénients et, par conséquent, le sujet de recherche concerne l'extraction de caractéristiques. Comme cette question n'est pas la question principale de cette thèse elle n'a pas été complètement étudiée. Les paramètres d'algorithmes pour l'extraction de caractéristiques ont été fixés a priori avec la connaissance des propriétés d'images satellitaires. Ils ne reflètent pas exactement la richesse d'information d'images. Une proposition est l'estimation des paramètres optimaux pour chaque algorithme d'extraction. Cette approche peut être réalisée via une modélisation des images et une optimisation des paramètres en fonction de la qualité de modèle.

La taille croissante des bases de données (des images satellitaires, des images multimédia, etc.) pose un problème de la complexité des algorithmes de regroupement. Beaucoup d'algorithmes développés ont une complexité quadratique. Par conséquent, ils ne peuvent être appliqués à de grandes quantités de données dans un délai raisonnable. Un nouvel axe de recherche consiste à développer les algorithmes ayant la complexité de calcul linéaire.

La troisième direction est un problème de la sélection des caractéristiques. Cette procédure devrait également être intégrée dans l'algorithme de clustering.

Le clustering est l'étape de l'extraction de données qui représente les dernières par les clusters et les relations entre eux Kuhn et al. [2007]; Parulekar et al. [2005]. L'analyse des clusters et des relations est une étape vers l'interprétation des données à haut niveau. Une étape intermédiaire entre le clustering et l'interprétation des résultats par l'utilisateur peut être prise en considération. Cette étape est appelée la construction automatique de la sémantique des images. Les clusters à ce niveau sont considérés comme des concepts. L'information spatiale peut-être impliquée pour trouver des groupes de clusters qui ont la même organisation spatiale.

La sémantique des images est un pas vers l'ontologie (une représentation formelle) des concepts et les relations pour décrire la surface de la Terre. L'intérêt de la fouille d'images via l'ontologie est de lier la sémantique des images et les modèles de langues naturelles afin d'améliorer la compréhension des scènes. De nos jours, cette direction de recherche pour l'extraction des données et le raisonnement des connaissances est très prometteuse. L'un des projets sur ce sujet est l'éditeur différentiel et formel de l'ontologie (Differential and Formal Ontology Editor, DAFOE)⁴. En dépit du fait que l'ontologie des images satellitaires n'ait pas encore été construite, de nombreux travaux sur ce sujet ont été effectués : la cartographie de la surface de la Terre, la représentation formelle des concepts, les relations entre eux, etc. Les travaux sur l'analyse de l'environnement en utilisant un formalisme géographique et des images satellitaires sont développés dans les projets de Corine Land Cover Bossard et al. [2000]. La formalisation des images et des concepts par l'ontologie est considérée comme une perspective de recherche.

⁴<http://dafoe4app.fr/>

Notations and Definitions

We give notations and definitions of useful terms which appear throughout the thesis.

Data - all used information about studied subject represented via numerical values (real or nominal values), *e.g.*, data is a satellite image with pixel intensities.

Sample - an item of data represented by a vector of values, *e.g.*, a sample is a subimage, a patch of the image or a pixel.

Feature - named value associated with a *sample*. Each value of a *sample* corresponds to a feature (a variable, an attribute), *e.g.*, a feature of a sample (sub image) is its mean value.

Model - is representation of *data*, *samples* or *features* via a set of parameters (values) and an algorithm which calculates these parameters. We differentiate the model of *data*, the model of *features* and the model of *samples*:

- ★ model of *data* is used to represent data (e.g. raster images represented by with pixels, intensity of each pixel, etc.) and to modify data (image filtering, enhancing, etc.);
- ★ model of *features* is used to extract (calculate) *features* from *samples*;
- ★ model of *samples* is a model of classifier or clusterer and is used for supervised, semi-supervised or unsupervised (clustering) classification of *samples*.

Class - set of *samples* of a predetermined group (each *sample* of this group has the same class name).

Classifier - algorithm which determines to what *class* a *sample* belongs.

Classification - process and a result of a *classifier* on a given set of *samples*.

Cluster - group of samples with similar attributes (each *sample* is associated to the same cluster index).

Clusterer - algorithm which discovers *clusters* using some distance or criteria of similarity or dissimilarity among *samples*.

Clustering - process and a result of a *clusterer* on a given set of *samples*.

i index of samples, $i = 1, \dots, I$.

j an index of an attribute, $j = 1, \dots, J$.

$X = \{X_1, \dots, X_I\}$ - set of samples X_i .

$X_i = \{X_{ij}, \dots, X_{ij}\}$ sample X_i or the vector of attribute values X_{ij} .

m an index of nominal attribute value, $m = 1, \dots, M$.

k an index of a cluster or a class, $k = 1, \dots, K$.

p index of clusterings, $p = 1, \dots, P$.

P the number of clusterings.

$C^p = \{C_1^p, \dots, C_K^p\}$ - set of clusters or classes C_k .

$C = C^p$ if $p = 1$ and $C = \{C_1, \dots, C_k, \dots, C_K\}$.

$B^p = \{b_{ik}^p : b_{ik} = 1, \text{ if } i \in C_k \text{ cluster}, 0 \text{ otherwise}; \forall i\}$.

$\mathbb{B} = [B^1, \dots, B^p, \dots, B^P]$ - concatenation of matrices B^p .

Θ_k set of parameter values associated to a probability density function (p.d.f.).

\mathcal{K} kernel function.

Chapter 1

Introduction

Earth Observation (EO) is a domain of science which has found wide application during last decades for analysing, monitoring, forecasting and managing natural resources and human activities. From particular observations of the Earth surface, scientists and specialists of different domains become interested in observations of large areas of the Earth and even its global surface.

Remote Sensing (RS) techniques for EO are able to realise such observations. RS is the acquisition of geospatially linked data (images) of sensed scenes by instruments of measurements at remote distance. Instruments or devices of remote sensing (*e.g.*, cameras or sensors) may measure different pieces of information such as various domains of the electromagnetic spectrum. Active and passive remote sensing systems exist. Active sensing is made by radars emitting electromagnetic radiation towards the scene and measuring the scattered wave. For Earth Observation, these systems use Synthetic Aperture Radars (SAR) ¹ and produce SAR images. Passive RS systems acquire electromagnetic radiations emitted or reflected by the Earth. Usually, a source of passive radiation is the solar radiation or the emission of infrared radiations by thermal objects. Passive RS systems form optical images like "Satellites Pour l'Observation de la Terre" (SPOT) ². Observed satellite images capture in details wide surfaces resulting in large volumes of data, *e.g.*, one SPOT5 image size of 12000×12000 pixels covers $60 \times 60 \text{ km}^2$. For more details on SPOT5 system and images see [Gleyzes et al., 2003]. Images as SPOT are very numerous, near 10^6 images were acquired since 1986 ³. At present, they are weakly exploited due to their large sizes and time consuming visual analysis. In the near future satellites with new sensors like Pleiades which will take many more new images will be launched. This gives a large interest and provides a demand for new theoretical methods to analyse satellite images and to develop new information systems for exploiting these images.

This thesis studies and proposes new methods to analyse satellite images in the context of satellite image mining. The objective of the thesis consists in extracting information (features, patterns, classes) from the images, representing it in a compact form and providing a semantic of images to the user. The main idea here defended is to carry out different possible approaches of image mining and to combine their inferences instead of using one single approach. Attention in this thesis is paid on unsupervised clustering methods to mine data.

The procedure of data mining proposed in this thesis is not limited by a specific task

¹<http://www.dlr.de>

²<http://www.spotimage.fr>

³<http://www.cnes.fr/web/258-spot.php>

and may be applied as a general approach on different data types, e.g., multimedia images. A characteristic feature of the proposed mining schema consists in the possibility of its application to very large databases that is the case of satellite image databases.

1.1 Content of optical satellite images

An example of optical satellite images is SPOT5 image of size 12000×12000 pixels with a ground resolution of 5 meters per pixel and covering a surface $60 \times 60 \text{ km}^2$ is presented in Figure 1.1. The image covers such a large city like Paris with its suburb. The high resolution of the image allows identifying many different structures like buildings, roads, airports, railway stations, bridges, forest, agricultural fields, clouds, water, snow and many others. As we may see from zoom parts of this image there are a lot of interesting

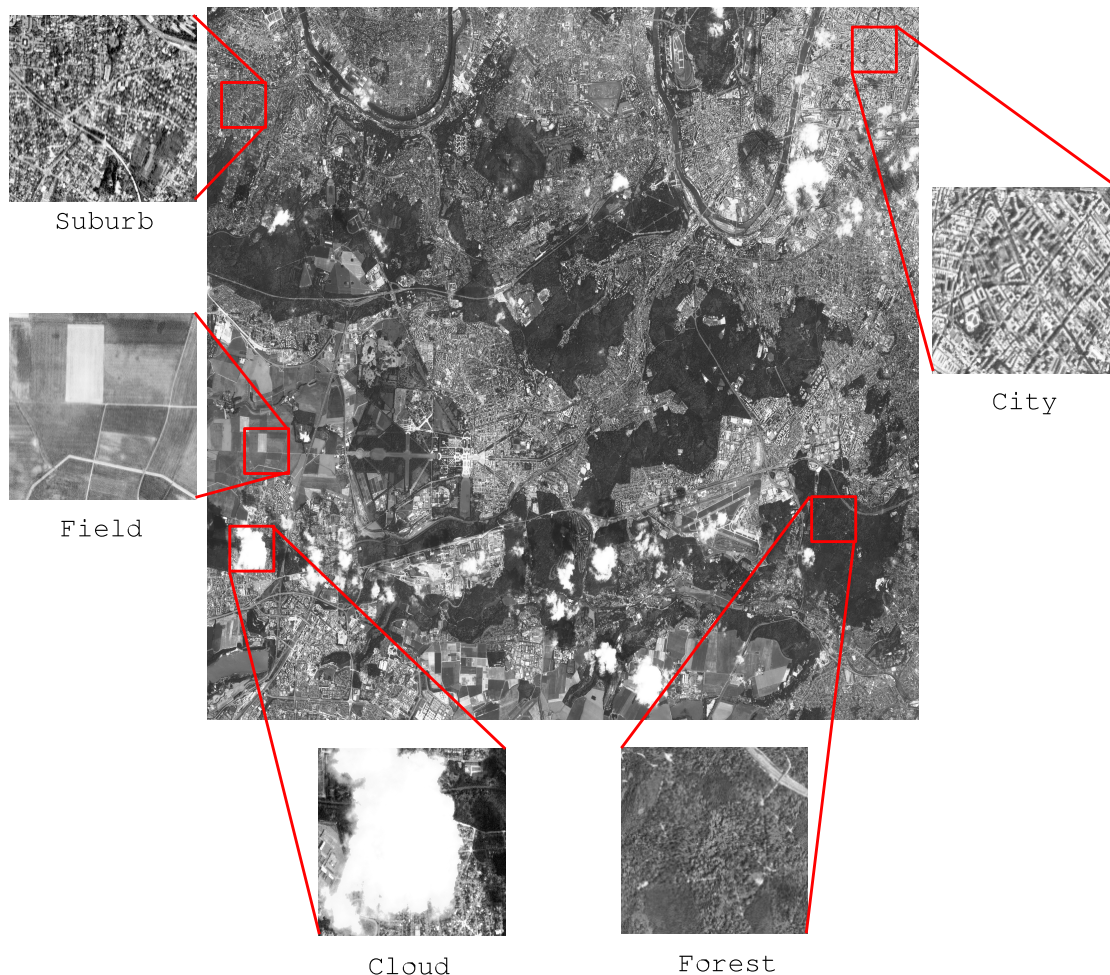


Figure 1.1: An example of a SPOT5 image of Paris with some instances of full resolution areas.

and useful information as different structures and forms.

Let us shortly analyse the content of image in Figure 1.1. A zone of suburb contains several clear lines which are main streets; several pixels grouped in square forms represent houses of size 10×10 meters; trees are viewed as gray dark spots. An image of fields has near uniform areas of size 20×20 pixels (i.e., relatively large areas of 100×100 meters) as well several clear lines of roads. A zone of clouds presents a high level of intensity on a large surface of approximately 500×500 meters. A forest offers quasi uniform gray levels with small deviations. Downtown Paris has high density of buildings of size 10×5 pixels (i.e., 50 meters) and many streets represented by long straight lines.

Very often in satellite image analysis, information is considered at the pixel level [Stein et al., 2002]. It is the case for the images with low resolution, *e.g.* from tens of meters to several kilometres per pixel. Thus dominant information is issued from large homogeneous surfaces like forest, water, snow, clouds, and cities. But with the progress of scanning devices, the resolution of satellite imaging is constantly growing and information at one pixel is no more significant. On the contrary, neighbouring areas should be taken into account and information is carried by groups of pixels. From Figure 1.1 we see that such windows may capture various objects. We may conclude that the higher is the resolution of the image the more complex the content of the image is and the more interesting information which may be analysed.

1.2 Pattern recognition

Data mining is the direction of science which combines different aspects of statistical learning, model selection, parameter estimation, *etc.*, [Witten & Frank, 1999]. To mine data means to find representative samples of data called *patterns* and relations between them as well as to classify and/or predict data. In this thesis data mining is considered as a task combining pattern recognition, classification and representation of relations between data. Some fundamental, important and interesting surveys of pattern recognition and classification may be found in [Jain & Dubes, 1988; Fukunaga, 1990; Duda et al., 2000; Theodoridis & Koutroumbas, 2003]. It is a common approach to mine data when we have not much or no a priory information about data to be analysed. Concerning data mining in satellite images, there are many works which have been done [Datcu & Seidel, 2000; Stein et al., 2002].

One of the reference examples of analysing the Earth surface by satellite images is the Corine Landcover project [Bossard et al., 2000]. The main idea of that work is to determine patterns and classes of surface and classify them using satellite images. There is a list of patterns and classes predetermined by different experts which are used for supervised surface classification. Classes are represented by a hierarchical tree with several levels of hierarchy. Then this compact information is used by different experts to analyse the surface. The main limit of this data mining approach is the supervised selection and supervised classification of patterns and classes. Often, experts classify data visually but there exist also many works as in [Gorte & Stein, 1998] proposing learning algorithms for classification. These experiments were carried out mostly for images with low resolution (from tens to hundreds of meters per pixel).

Statistical learning, Bayesian modelling and probabilistic model inference are widely used approaches for pattern recognition [Fukunaga, 1990]. Principles of estimation and selection of the best probabilistic model for data are explained in [Friedman et al., 2001; Mackay, 2002]. Probabilistic models for continuous data are often supposed to be mixture

models, e.g., mixtures of Gaussian distributions. A very good survey of this topic can be found in [McLachlan & Peel, 2000]. In practice, well formulated statistical learning models and algorithms sometimes reach limits because of strong assumptions on probabilistic distributions. To overcome such limits, *kernel approaches* become recently very popular. A detailed survey to kernel approaches for statistical learning is introduced in [Vapnik, 1998]. Further interesting practical ideas for learning by kernels are well explained in [Shawe-Taylor & Cristianini, 2004].

Examples of a satellite image mining systems and their theoretical aspects are presented in [Datcu & Seidel, 2000; Datcu et al., 2003; Datcu & Seidel, 2005; Barnes, 2007]. One of the recent surveys of satellite image mining may be found in [Heas & Datcu, 2005; Gueguen & Datcu, 2007]. Although this work deals with temporary satellite images, some aspects of general data modelling may be considered for different types of satellite images.

1.3 Contribution of this thesis

The purpose of this thesis is to contribute to unsupervised satellite image mining. Within this task, the following steps have been addressed:

- ★ extraction of information from satellite images and its representation by features;
- ★ selecting of most informative features and reducing the feature space for clustering algorithms;
- ★ data modelling via clustering using different unsupervised algorithms with selection of the best optimal solution for each algorithm;
- ★ combination of the different results obtained from unsupervised clustering algorithms
- ★ semantic representation of unsupervised clusterings to satisfy user's requirements.

The main problem of the thesis is to find categories of zones of images and to cluster them without prior knowledge on the type and number of categories. In this thesis several new approaches to mine and analyse satellite images are proposed:

- ★ Geometrical features describing high resolution satellite images. High resolution satellite images have two different structures: textures (uniform surfaces as sea, forest, clouds, etc.) and geometrical forms (structures with recognised forms: buildings, airports, warehouses, etc.). Geometrical features are good candidates to complement textural descriptors. The first contribution of this thesis is the selection of a set of geometrical features for satellite image.
 - ★ Unsupervised selection of pertinent features. Different descriptions of the same image lead to high dimensional image representations. The problem of minimizing a high dimensional space is considered in this thesis. Feature selection is one of the solutions to overcome the problem. As the second contribution we propose an unsupervised feature selection algorithm which improves the quality of pattern recognition results. This work is published in [Campedel et al., 2007].
-

- ★ Unsupervised clustering with estimating the number of clusters. The third contribution concerns the problem of estimating the number of clusters for unsupervised clustering algorithms. The feature space is supposed to have Gaussian distributions. An Expectation-Maximisation algorithm with a mixture of Gaussians finds the optimal data clustering. A Minimum Description Length (MDL) criterion selects the best model for EM-algorithm as well as the number of clusters. Analysis of MDL for GMM leads to a new simplified MDL criterion [Kyrgyzov et al., 2007b]. Simplified MDL criterion may be applied to different algorithms which minimises square errors, *e.g.*, K-means. It also can be generalised to estimate non square errors via kernel methods. This new criterion has been called Kernel MDL (KMDL) [Kyrgyzov et al., 2007b]. Proposed simplified MDL criterion to determine the optimal number of clusters has been used in different applications [Costache & Datcu, 2007; Bordes & Maître, 2007; Marine Campedel, 2008].
- ★ KMDL criterion leads to a new class of algorithms such as a kernel hierarchical clustering. This algorithm is the fourth contribution and it shows superior performances compared to classical algorithms which operate with square error minimisation. The algorithm also estimates the optimal number of clusters.
- ★ The fifth contribution consists in combining different clustering results obtained from different algorithms. Two unsupervised approaches to estimate the optimal combination are proposed. The first is based on a hierarchical agglomeration of clusterings and a search for an optimal consensus among them by minimising a new proposed criterion [Kyrgyzov et al., 2007a]. This idea has been applied in [Marine Campedel, 2008; Campedel et al., 2007]. The second approach is shown to achieve a global optimum of the clustering combination [Kyrgyzov et al., 2008].
- ★ The sixth contribution consists in representing different clustering results to a user via semantic relations between concepts. An optimal consensus is found for clusterings obtained from different algorithms. Clusters in consensus solution are considered as concepts. Relations among these concepts are shown to a user in the form of a tree or a semantic graph. This idea is presented in the work [Marine Campedel, 2008].

The thesis is constructed in the following way: problem statement for this thesis is expressed in Chapter 2, where a description of satellite images of high resolution is also presented. Problems of data mining and pattern recognition for large data bases of satellite images are introduced. Chapter 3 presents information which can be extracted from optical satellite images. Information is represented by features which describe different properties of the Earth surface. The problem of pattern recognition is introduced in Chapter 4 where supervised classification is presented. Then Chapter 5 presents a review of unsupervised classification via clustering algorithms. The problem and solutions of model selection for unsupervised clustering algorithms are stated in Chapter 6. The formulation of the combination problem of clusterings as well as its solutions is proposed in Chapter 7. Chapter 8 presents a variety of application examples of combination. A complete protocol of unsupervised data mining of satellite images is shown in Chapter 9. Finally, conclusions and perspectives of the thesis are given in Chapter 10.

Part I

Problem statement

Chapter 2

Data mining in satellite images

2.1 Data mining

In this chapter we give a definition of data mining and examples of applications in different domains. One of its main roles is decision making. Data mining is a process of discovering patterns in data and relations among them. It covers also aspects of pattern classification, data prediction and representation of discovered results. The representation may be done by statistical indicators or through visualisation by images, trees or graphs [Larose, 2006]. A pattern is a representative example of some part of data and depending on application can be an image, a signal or any type of measurements to be either classified or recognised [Marques de Sá, 2001; Larose, 2006]. Pattern recognition as well as machine learning includes both theoretical and technical instruments of data mining. Theoretical aspects may be considered as problem statements, theorems, methods and algorithms while technical aspects include systems and methods of data acquisition, programmes of data processing.

At present, data mining is used in many domains: scientific, industrial and commercial [Marques de Sá, 2001; Theodoridis & Koutroumbas, 2003; Duda et al., 2000; Larose, 2006]. These domains are very interdependent because science provides solutions for industrial problems, that influences commercial activities and vice versa commerce demands industrial solutions that poses tasks to scientists. Here we list several domains and applications where data mining is used.

- ★ **Image processing** As we mentioned in previous chapter there is more and more satellite images and a demand of data management and intelligent processing is growing. It may be an analysis, detection and classification of satellite or aerial images, radar and sonar signals, automated target recognition. There are many applications, *e.g.*, for urban management and agriculture needs: soil analysis and management; for geology: land cover classification (water, land, forest, rocks, urban, *etc.*), estimation and analysis of mining resources, seismic analysis; for astronomy: analysis of telescopic images.

With innovations various cameras are accessible to numerous users which take a lot of digital pictures. A user may have thousands of images in several years and may want either to require a specific image or organise images in different groups. For example, indoor-outdoor, travels-events, portraits according to different criteria: data, parenthood, *etc.* Data mining in these images will include feature extraction from photos and automatic grouping of images based on features. A user

may query images with special characteristics. Many commercial organisations are interested in analysis of multimedia images to find groups of similar images and to analyse user's needs, parameters for camera characteristic, image (video) compression and coding, video analysis, shot detection and classification, image and video retrieval, content description and indexing (*e.g.*, Google). Video analysis may be used for vehicle detection, traffic analysis and control, monitoring, navigation systems, robot vision, *etc.*

Another example of image mining is a biometry which includes fingerprint analysis, face and speech detection and recognition, people detection and tracking, human motion recognition and body analysis, human-computer interfaces, surveillance and alarm systems, observation of human activity. Biometry may be used in any public place for control and surveillance (airports, hospitals, schools, organisations, *etc.*). Medical application makes a widespread use of data mining and covers analysis of medical images and data, classification of human body, organs, detection and classification of artefacts and diseases, support of medical decisions, *etc.*

- ★ **Bioinformatics** In bioinformatics one of the popular data mining tasks is studying of behaviour of genes during experiments. Genes may be grouped automatically where each group of genes represents the same behaviour or characteristics. Automated cytology, properties of chromosomes, genetic studies, *etc.*
 - ★ **Industrial domain** Industrial application is very close to commercial application. For example, in automobile industry a producer of cars wants to analyse a data bases of world cars and either to estimate how many cars with the same characteristic are or to classify cars. The estimation of tendency in construction is also very important that directly influences on the commercial proposition. Mining may be considered not only to cars but any products, from light industry to heavy and high-tech industry. For example, a very popular example is the automatic detection and classification of objects on an assembly line of a factory. That reduces the time and increase the quality of product assembly.
 - ★ **Commercial application** Data mining for commercial needs involves tasks quoted above and may include fault detection in products, character recognition, watermarking, market analysis, *etc.* One of the interesting tasks is the study of consumer needs. For example, it is very important to find groups of consumers with the same needs, which help to better manage their demands and to elaborate good propositions. Selling in markets includes a lot of different groups of consumers like students, working people, young, seniors *etc.*. Indeed, consumers differencing by characteristics such as age, social position, earning level, *etc.*, are expected to buy different families of products. It is important to determine such families in an automatic way that allows retaining the customer by providing more adapted choice of new, cheaper and convenient groups of products, being sure that such groups of consumers will always buy certain groups of products. It concerns not only market but a selling process in general, *e.g.*, high-tech technologies (mobile phones, computers, services, *etc.*), bank services (credits, insurances, market of securities, *etc.*), and all possible commerce.
-

2.2 Satellite image models and their application

In this section a short introduction to models of satellite images as well as their applications is given. As mentioned in the previous chapter there are mostly two types of remote sensing domains depending on the wavelength: (i) visible and infrared remote sensing and (ii) microwave remote sensing. In this thesis the accent is on visible remote sensing for optical images of a high resolution.

A two dimensional image is recorded by an optical multispectral scanner along the flight of the satellite platform. For example, the SPOT5 scanner is a linear array of solid semiconductive elements¹. These elements detect average intensity and correspond to pixels of a digital image with integer values. Such a kind of sensors is called Charge Coupled Devices (CCD). For SPOT5 camera each line has 12000 elements which correspond to pixels on an image. Three consecutive lines of camera record the same line of data but with different characteristics of a filter (near infrared, red, green). A digital image has coordinates of pixel number, counted from left to right and from top to bottom. The size of pixels corresponds to sampling frequency, the larger the pixel size the lower the sampling frequency and the worse the image. On the contrary, the smaller the pixel size the more details may be seen on the image and the bigger image size.

Satellite image processing includes main steps: input images are registered by digital recording devices. Then images are corrected or reconstructed via image intensity or geometry correction, image restoration or reconstruction. Finally, images are classified by grouping image descriptors or features in classes. This is done by supervised, semi supervised or unsupervised classification, segmentation or image matching. The last step is presentation of results either directly to a user or by saving them to a Geographical Information System (GIS). In this thesis unsupervised image classification and representation of results are considered.

SPOT5 images

High Resolution (HR) satellite images are considered to have a metric resolution, *e.g.*, SPOT5 images have the resolution of 5 meter per pixel, and images with the sub-metric resolution have a Very High Resolution (VHR), *e.g.*, QuickBird (QB)² images have 60 cm. per pixel and Pléiades images will have 70 cm. per pixel³. Examples of images of the same place provided by satellites SPOT5 and QuickBird are presented in Figure 2.1. On HR images as SPOT5 Figure 1.1 each pixel has meaning for objects no less than houses, big trees, trucks, *etc.*; neighbour pixels may represent bigger objects such as roads, buildings, bridges, crossroads, airplanes, *etc.* A square of size 64×64 pixels takes a ground surface 320×320 meters and capture such textures as homogeneous area of forest, sea, desert, snow, cloud, agriculture field, suburb, downtown, industrial zones, *etc.* Examples of several such surfaces are presented in Figure 2.2, they have been extracted from SPOT5 image of Paris Figure 1.1.

¹<http://www.cnes.fr>

²<http://www.digitalglobe.com/>

³<http://www.cnes.fr/web/3227-pleiades.php>

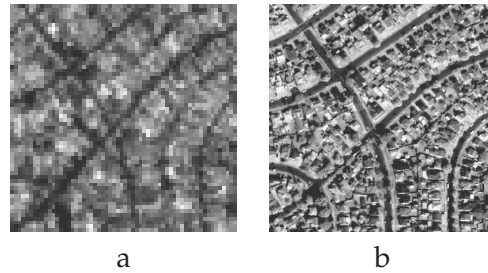


Figure 2.1: Samples of satellite images of the same place in Los Angeles: a - SPOT5 (5 m/pixel), b - QuickBird (0.6 m/pixel). ©Copyright CNES

2.3 Data mining in high resolution satellite images

Images of HR provide diverse information about the Earth surface and are very interesting for experts in different domains: urban, agriculture, environment, military, *etc.*. Satellite images reflect useful information about resources of different countries and allow quickly analysing the Earth surface and make decisions; it is only limited by the quantity and capacity of satellites to capture images. Requirements for land cover classification and its validation via satellite images can be found in [Muchoney & Strahler, 1996]. Another review for classification of remote sensed images is given in [Atkinson & Lewis, 2000].

One of the main applications of satellite imagery is map constructing. Knowing ground coordinates and setting them on a satellite image a geographical expert may compare actual ground state to existing maps. Also images are utilised to help up-date maps, *e.g.*, for monitoring density of buildings in urban zones which may increase or decrease, for detecting roads which may appear or disappear, for analysing developing or degrading agriculture fields or forest, *etc.* Another possible application of satellite images may be in using them as a tourist guide, virtual tourism or visit. Before departure in different places tourists may be interested in viewing these new places. Images with different scales may give an idea to visit some neighbour regions.

Actual systems of satellite image analysis very often implicate information provided by an expert. *e.g.*, recognition of urban and natural zones, detection of road nets, *etc.* Such a kind of work needs a lot of experts and efforts.

Currently, analysis of satellite images is done by users visually. Of course the visual observation is a restricted approach. As we know satellite images have large sizes, they are numerous and in the future more and more images will come. To do an analysis of a large surface several experts should be involved, but it takes several years to educate an expert that may be expensive. In addition, the more human is introduced in taking decision in a complex process the more probable to have mistake and subjective results. A term "subjective" means that results of the image analysis will be different from one user to another. An example of analysis is the road detection on satellite images. Indeed, in the world there are a lot of regions when there are no maps or they are composed only on the paper and not in the electronic version. HR or VHR satellite images are used to detect a road on the image. But these images are very large for relatively small surface (12000×12000 pixels for 60×60 km²). To find roads one person may analyse such an image approximately one week. Roughly speaking, an expert who uses only satellite

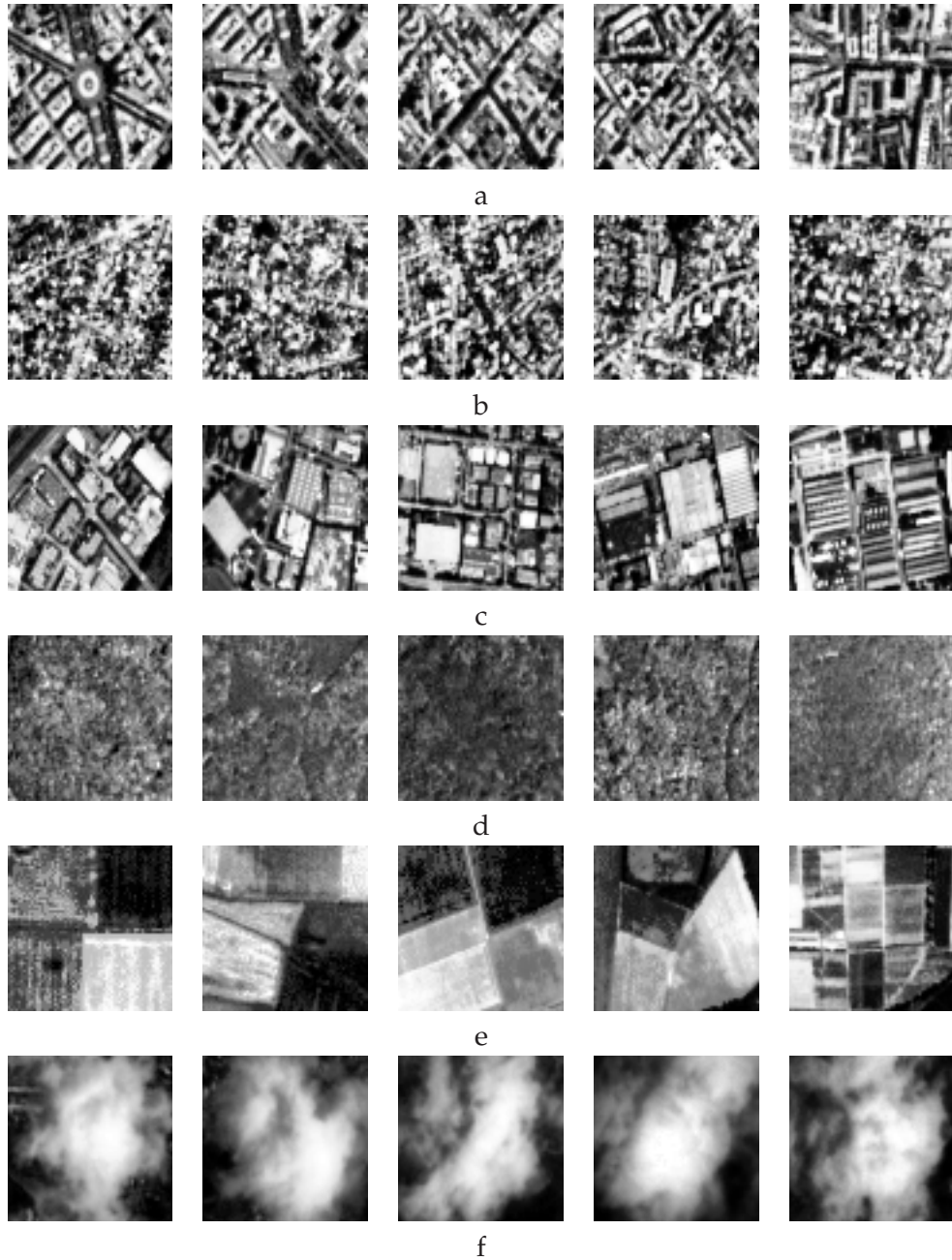


Figure 2.2: Instances of typical image content issued from SPOT5 images of Paris (64×64 pixels per sample) : a - Downtown, b - Suburb, c - Industrial zone, d - Forest, e - Field f- Cloud. ©Copyright CNES

images to analyse the surface of Europe which is 10^7km^2 would take near 53 years. Either 53 experts can find only roads for one year.

We see, that the demand in automatic system is very big, because they may significantly improve the quality of decisions and reduce the time needed for. Human analysis has advantage in its a priori knowledge but subjective decisions are very time consuming.

To overpass lacks of human decisions automated analysis should be applied everywhere where it is possible. Now, the demand in automatic systems is very high and less or even no user interaction is preferred. Under such a kind of interaction we understand setting some a priori parameters or tuning them during analysis. An automated system should analyse and process data and provide a user with all possible results in a compact and clear form.

An addition interest of using automated systems of satellite image analysis consists in possibility to discover new information and knowledge. For example, such new information may be new classes, that user does not realise before, new relations among classes, which will help to understand interdependence among them. Another possible analysis may be in discovering whether the same arrangement or interdependence of classes may be observed in other parts of the Earth surface, *etc.*

Main steps of a system to analyse satellite images are usually the same, independently from application, and consists of:

- ★ Feature extraction - modelling and extraction of information from satellite image;
- ★ Pattern recognition - selecting models and optimising their parameters for analysis, classification, clustering and learning process;
- ★ Representation of results - visualising results of pattern recognition.

Some intermediate steps as well as loops among them may also be included. The following chapter is devoted to feature extraction from satellite images.

2.4 Conclusions

In this chapter examples of data mining have been illustrated. A brief review of different problems of satellite image analysis and mining has been considered. Models of optical satellite images have also been discussed. Following subjects have been revised:

- ★ Applications of data mining in different domains: scientific, industrial and commercial. The importance of mining systems has been argued.
 - ★ Existing numerous volumes of satellite images have been noted. Examples of satellite image analysis have been presented. Available diversity of the Earth surfaces has been shown by satellite image samples. Image content shows its richness and complexity. A comparison of human and automatic analysis of satellite images has been provided and the growing needs in automatic systems have been justified.
 - ★ Main steps of data mining have been presented in this chapter. An interaction between a user and a computer during mining process has been described.
-

Chapter 3

Feature extraction

In this chapter we give definitions and notations for pattern descriptors (also called later features) and show models of features as well as feature extraction. Here a pattern is considered as a part of a satellite image.

A natural image, e.g., a satellite image, contains regions which have common properties for visual perception. For example, in Figure 1.1, we see homogeneous regions of city, forest and clouds which are easily distinguishable. Each of these regions is characterised or described by features, e.g., by similar pixel gray level intensity or texture. There are two groups of features: (i) natural and (ii) artificial. Natural features correspond to visual image perception, e.g., intensity level, textural regions, while artificial features are obtained after image manipulation, e.g., image histogram, spatial frequency spectra [Pratt, 2001], etc. Image features are used for image segmentation, classification and clustering to find regions with common properties.

In image processing domain many features have been proposed to describe an image. Typically, for static images (in our case SPOT5) feature models mainly are: statistical descriptors of image intensity, textures and geometry [Pratt, 2001; Forsyth & Ponce, 2002]. In this chapter we propose developed geometrical features and give examples of texture features which have been realised in [Campedel et al., 2004, 2005].

Let us have I patterns presented as grayscale images \mathcal{I}_i of size $N_r \times N_c$ pixels, where $i = 1, \dots, I$ (later \mathcal{I} , for simplicity). The image \mathcal{I} is a square matrix which has N_r pixels of rows from top to bottom and N_c of columns from left to right: $\mathcal{I} = \{\mathcal{I}_{rc} : r = 1, \dots, N_r, c = 1, \dots, N_c\}$, $\mathcal{I}_{rc} \in \{0, \dots, L - 1\}$, where L is the number of gray levels. Let feature set $\{X_{ij}\}$ have I elements, where X_j has a value describing feature j of pattern \mathcal{I} . Later we use X_j to indicate the feature j of sample i for the simplicity.

3.1 Image intensity features

For image intensity features we compute statistical moments such as central moments of the first (mean value) and second order (standard deviation). A mean value of gray level intensity of image \mathcal{I} is:

$$X_1 = \frac{1}{N_r N_c} \sum_{r=1}^{N_r} \sum_{c=1}^{N_c} \mathcal{I}_{rc}, \quad (3.1)$$

and a standard deviation of intensity level is:

$$X_2 = \sqrt{\frac{1}{N_r N_c} \left(\sum_{r=1}^{N_r} \sum_{c=1}^{N_c} \mathcal{I}_{rc} - X_1 \right)^2} \quad (3.2)$$

This kind of statistical descriptors for SPOT5 is able to distinguish, *e.g.*, a bright part of the image with a high intensity gray level (*e.g.*, clouds, snow) from a dark part with low intensity (*e.g.*, sea, land), see Figure 1.1. Some higher order statistical features may be extracted from the image:

skewness

$$X_3 = \frac{1}{N_r N_c X_2^3} \left(\sum_{r=1}^{N_r} \sum_{c=1}^{N_c} \mathcal{I}_{rc} - X_1 \right)^3, \quad (3.3)$$

kurtosis

$$X_4 = \frac{1}{N_r N_c X_2^4} \left(\sum_{r=1}^{N_r} \sum_{c=1}^{N_c} \mathcal{I}_{rc} - X_1 \right)^4, \quad (3.4)$$

information theoretic measures, like entropy

$$X_5 = - \sum_{l=0}^{L-1} p_l \log p_l, \quad (3.5)$$

where $p_l = \frac{N_l}{N_r N_c}$ and N_l is the number of pixels of gray level $l : 0 \leq l \leq L - 1$.

3.2 Texture features

A texture is defined as an image of a surface which is easy to recognise but difficult to describe and it is represented by many objects [Forsyth & Ponce, 2002].

Image samples in Figures 1.1 and 2.2 represent examples of different textures. Samples of a forest have small gray level deviations while samples of clouds have high gray level deviations with dominant bright part. Samples of suburb, city and industrial zones have sharp gray level surface. Image samples of fields can have geometrical forms (*e.g.*, squares and triangles) with dark and bright gray levels. These different properties of image samples can be characterised by spatial dependency of pixel intensity. Some models of spatial dependency are represented either via extracting of image statistics or via image filtering. Characteristics obtained from filtering are called texture features. Now we give some basic models of texture features.

Haralick features

One of the famous features to describe texture presented by an image are Haralick features [Haralick et al., 1977]. These features are calculated on the second-order histogram of the joint probability distribution of a pair of pixels which are called a co-occurrence matrix. This matrix is computed for a pair of pixels with coordinates (m, n) and $(m \pm \rho, n \pm \rho)$, which are separated by ρ pixels and have an angle θ with respect to the horizontal axis:

$$P(l_a, l_b, \rho, \theta) = \#(\mathcal{I}_{mn} = l_a, \mathcal{I}_{m \pm \rho, n \pm \rho} = l_b, \theta). \quad (3.6)$$

Here $P(l_a, l_b, \rho, \theta)$ is the number of occurrences $\mathcal{I}_{mn} = l_a$ and $\mathcal{I}_{m \pm \rho n \pm \rho} = l_b$, for $0 \leq l_a, l_b \leq L - 1$ [Theodoridis & Koutroumbas, 2003]. Usually, ρ can take some units and θ four angles $0^\circ, 45^\circ, 90^\circ$ and 135° . Size of co-occurrence matrix (CM) corresponds to the number of gray levels of the image. Let p_{ij} be an element of normalised CM P_{ij} Eq. (3.6) for some ρ and θ such as $p_{ij} = P_{ij} / \sum_{ij} P_{ij}$. Image features calculated on CM are statistical descriptors which reflect properties of textures (smoothness, coarseness, etc.). They are listed in Appendix A.

The number of Haralick features is 78: 13 Haralick features for four directions $0^\circ, 45^\circ, 90^\circ$ and 135° and one fixed $\rho = 3$ with their mean and standard deviation values.

Gabor features

Gabor filters represent models of visual perception of a texture [Daugman, 1985] and was widely studied and applied for texture classification and segmentation [Dunn et al., 1994; Dunn & Higgins, 1995; Jain & Farrokhnia, 1991; Weldon et al., 1996]. These filters in the spatial domain are presented as [Manthalkar et al., 2003]:

$$h(x, y; u, \theta) = \exp \left(-\frac{1}{2} \left[\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2} \right] \right) \cos(2\pi u x'), \quad (3.7)$$

where $x' = x \cos \theta + y \sin \theta$, $y' = -x \sin \theta + y \cos \theta$ and u is the frequency along the direction θ from the axis x . Variances σ_x and σ_y correspond to a width of Gaussian for x and y axes respectively and they determine the bandwidth of the Gabor filter. The Fourier transformation of Eq. (3.7) is:

$$H(U, V) = 2\pi\sigma_x\sigma_y \left(\exp \left\{ -\frac{1}{2} \left[\frac{(U - u)^2}{\sigma_u^2} + \frac{V^2}{\sigma_v^2} \right] \right\} + \exp \left\{ -\frac{1}{2} \left[\frac{(U + u)^2}{\sigma_u^2} + \frac{V^2}{\sigma_v^2} \right] \right\} \right), \quad (3.8)$$

where $\sigma_u = 1/2\pi\sigma_x$, $\sigma_v = 1/2\pi\sigma_y$. An example of Gabor filter for 128 points in the spatial domain and its corresponding Fourier transform in frequency domain are in Figure (3.1 a) and (3.1 b). The Gabor filter in the frequency domain for 6 orientations $0^\circ, 30^\circ, 60^\circ, 90^\circ, 120^\circ, 150^\circ$ and 4 scales from 0.05 to 0.4 are presented in Figure (3.1 c).

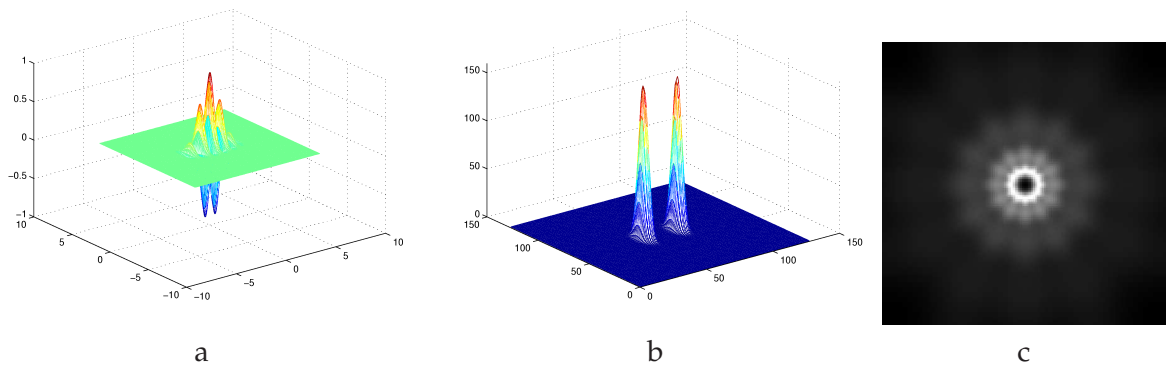


Figure 3.1: Examples of the Gabor filter: a - in the spatial domain; b - corresponding frequency domain of (a); c - in the frequency domain for 6 orientations $0^\circ, 30^\circ, 60^\circ, 90^\circ, 120^\circ, 150^\circ$ and 4 scales from 0.05 to 0.4.

For m scales and n orientations we write:

$$g_{mn}(x, y) = a^{-m}h(x', y'), \quad (3.9)$$

where $x' = a^{-m}(x \cos(\theta) + x \sin(\theta))$, $y' = a^{-m}(-x \sin(\theta) + x \cos(\theta))$, $\theta = n\pi/K$ and K is the total number of orientations. Let U_h and U_l be the higher and lower frequencies and K and S be the number of orientations and scales. We may write filter parameters in frequency domain.

$$\begin{aligned} a &= (U_h/U_l)^{1/(S-1)}, \sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2\ln 2}}, \\ \sigma_v &= \tan\left(\frac{\pi}{2K}\right) \left(U_h - 2 \ln 2 \left(\frac{\sigma_u^2}{U_h} \right) \right) \times \left[2 \ln 2 - \frac{(2 \ln 2)^2 \sigma_u^2}{U_h^2} \right]^{-1/2} \end{aligned} \quad (3.10)$$

where $u = U_h$, $\theta = \pi/K$, $m = 0, 1, \dots, S-1$. a^m ensures that the energy of signal Eq. (3.9) does not depend on scale m . Features extracted by Gabor filters are mean values and variances of filtered images. These features are used to obtain characteristics which are invariant to rotation [Manthalkar et al., 2003].

QMF features

Quadratic mirror filters (QMF) applies a filtering scheme which can reconstruct an image exactly [Vetterli, 1986]. It was proposed to consider a general problem of signal reconstruction after applying filters. Generally, for image processing two filters are used low-pass H_1 and high-pass H_0 and their combination in vertical and horizontal directions gives four output.

The simplified scheme of such reconstruction is presented in Figure (3.2). In the two

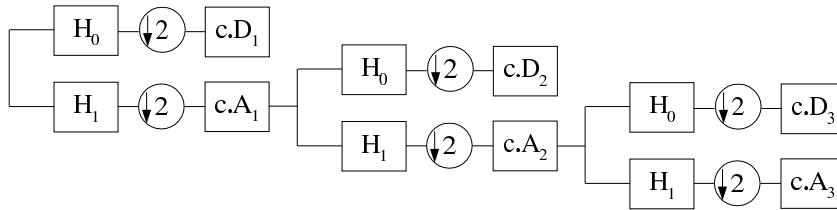


Figure 3.2: Simplified scheme of QMF filtering

dimensional case (as for images), filtering is made in horizontal and vertical directions by separable filters. Therefore we obtain four outputs: LL - low-pass filtering in horizontal and vertical directions, LH - low-pass filtering in horizontal direction and high-pass filtering in vertical direction, HL - high-pass filtering in horizontal direction and low-pass filtering in vertical direction, HH - high-pass filtering in horizontal and vertical directions. Then the same scheme is applied to LL output after resampling. Here again, image features are statistical values as mean and standard deviation for each output of filter. So, we have to calculate 2 features (mean and variance) for LH , HL and HH outputs. For the number of scales (decompositions) s we have $s \times 3 \times 2$ features.

3.3 Geometrical features

In this section we propose extracting geometrical features from optical satellite images. We consider geometrical features as that which describe geometrical properties of objects seen on the image, *e.g.*, line segments and statistical properties of edges detected on the image. The resolution of satellite images (*e.g.*, SPOT5) allows distinguishing private buildings, big buildings, warehouses, borders of roads, rivers, fields. Such objects have line segments of different length, angle and density. We want to benefit from such properties to have more rich set of features and to cover more image classes.

We present detection of edges, edge approximation by line segments and extraction of geometrical features. An image is filtered by Deriche filter [?] following hysteresis thresholding and edge detection. Linear segments approximate edges using Papakonstantinou [Papakonstantinou, 1985] algorithm. Features are extracted from both edges and line segments. For such big images as satellite, edge detection should be adaptive because the image covers different landscapes.

Adaptive edge detection

Edge detection

A Deriche's approach of edge detection [?], [Deriche, 1987a] is based on Canny's filter design [Canny, 1983], [Canny, 1986]. Canny had defined a one - dimensional Finite Impulse Response filter for edges by optimising specific criteria for the quality of edge detectors. Deriche used the same method to construct an Infinite Impulse Response edge filter. Let $I(x, y)$ denote the image with size of $N \times M$ pixels. The signal I is filtered by a low pas filter ψ , then derived. In a similar way, it may be filtered by the derivative ϕ of ψ :

$$\psi(x) = -cx \exp(-\alpha |x|) \quad (3.11)$$

$$c = \frac{[1 - \exp(-\alpha)]^2}{\exp(-\alpha)}$$

$$\phi(x) = c(\alpha |x| + 1) \exp(-\alpha |x|) \quad (3.12)$$

where α is a scale parameter which controls the minimal distance between two adjacent edges.

General steps of the edge detection are:

1. Convolve the image with a separable Deriche filter in x and y direction of the image.
2. Take the first derivatives in horizontal and vertical directions.
3. Compute the magnitude of the gradient $M_g(x, y)$.
4. Perform non-maximal suppression of the gradient in horizontal and vertical directions.
5. Perform hysteresis thresholding.

Because the contrast in the images is varying and image dynamics are highly changing, we make use of adaptive thresholding in order to keep the main edges of any area

whatever the local contrast.

We choose a threshold under the form:

$$t = m + \beta s \quad (3.13)$$

where

$$m = \frac{1}{NM} \sum_{(x,y)} M_g(x, y)$$

$$s = \sqrt{\frac{1}{NM} \sum_{(x,y)} (M_g(x, y) - m(x, y))^2}$$

m is the mean and s is the standard deviation of the image magnitude derivation respectively and $\beta=0.8$.

For hysteresis thresholding the high threshold is t and depends on standard deviation of magnitude of the smoothed image. The low threshold is set 50% of the high threshold.

Adaptive thresholding

For adaptive edge detection of a large size image (typical size = 3000×3000 pixels) we need to compute local thresholds. In later stage of the process, we are interested in measuring density of edges on small windows of typical size 64×64 pixels. We have chosen to compute the threshold on windows of size 300×300 . Then thresholds are interpolated for windows with size 64×64 .

The most commonly-used method of interpolation is bilinear (also called twisted-plane, area-weighting, or four-point). Let $D(x, y)$ be the map of thresholds and $I'(x, y)$ be interpolated values of $D(x, y)$. Suppose that we need the value at location $(i+p, j+q)$, where i and j are integers, p and q are in $[0, 1.0)$. We can approximate $I(i+p, j+q)$ using the values at the four nearest integer locations using the formula

$$I(x, y) = (1 - p)(1 - q)D(i, j) + p(1 - q)D(i + 1, j) +$$

$$+ q(1 - p)D(i, j + 1) + pqD(i + 1, j + 1) \quad (3.14)$$

A scheme of adaptive edge detection is presented in Figure3.3.

Edge approximation by line segments

Edge detection provides an image of edge which can be used to detect road network, to identify fields, etc. Some objects in high resolution satellite images have elongated edges which can be modelled by line segments. For this reason, we propose extracting linear segments using image edges.

Piecewise linear approximation (PLA) is widely used in signal and image processing, and pattern recognition. PLA is able to approximate digitised curves (edges) using consecutive line segments.

An edge can be presented as an open N -vertex polygonal curve P in 2-dimensional space. P is the ordered set of vertices $P = \{p_1, \dots, p_N\} = \{(x_1, y_1), \dots, (x_N, y_N)\}$. An polygonal Q is approximation of P and consists of $(M+1)$ vertices: $Q = \{q_1, \dots, q_{M+1}\}$ where the set of vertices q_m is a subset of P and $M < N$. The end points of Q and P

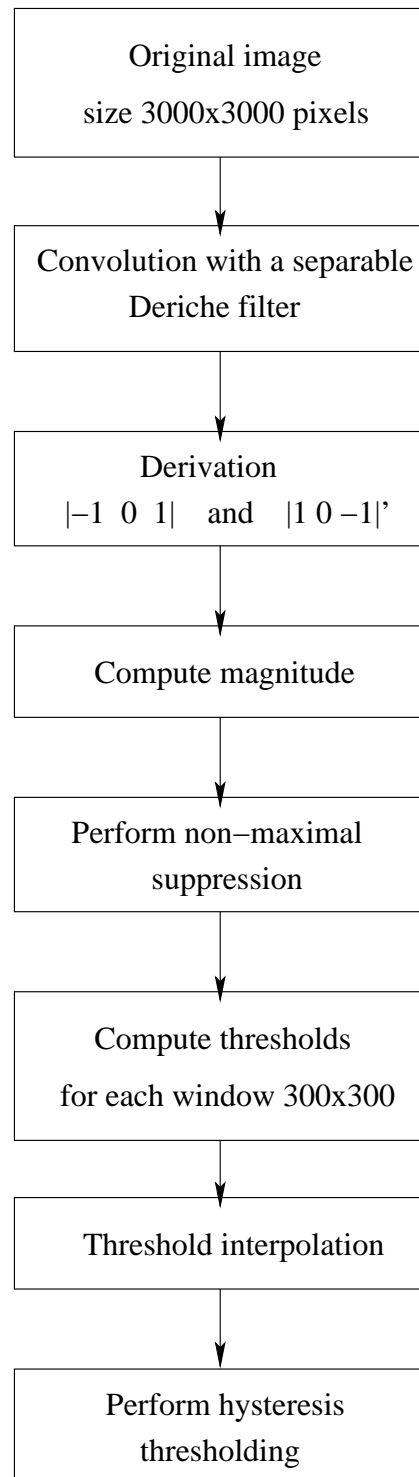
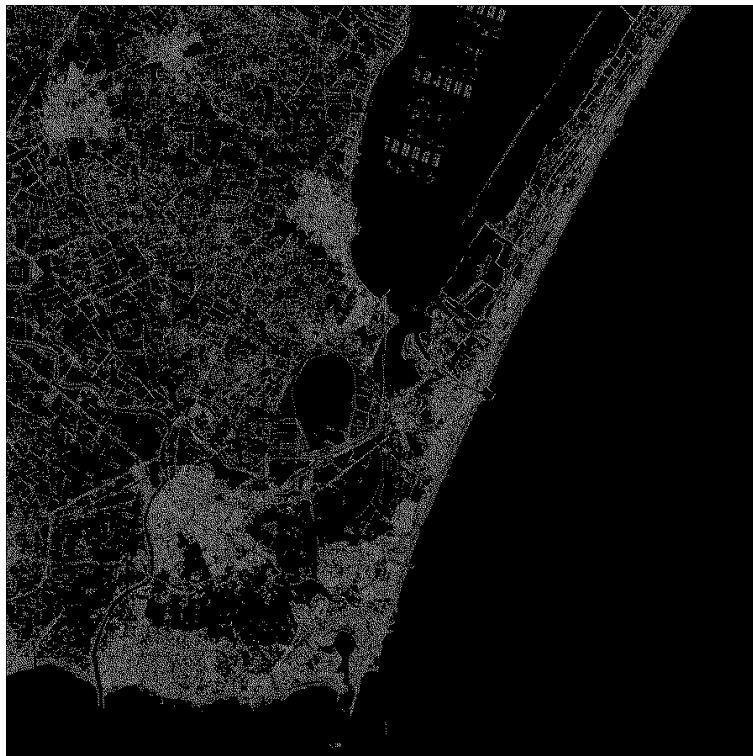


Figure 3.3: Diagram of adaptive edge detection



a



b

Figure 3.4: Satellite image SPOT5 of Béziers ©CNES: a - original image 3000×3000 pixels, b - image of edges.

are the same: $q_1 = p_1, q_{M+1} = p_N$. The approximation of the linear segment (q_m, q_{m+1}) of Q for curve segment p_i, \dots, p_j of P is defined by the end points p_i and p_j : $q_m = p_i, q_{m+1} = p_j$ and $(q_m, q_{m+1}) = (p_i, p_j)$.

There are two optimisation problems [Imai & Iri, 1988; Kurozumi & Davis, 1982] connected to polygonal approximation:

1. *min - e problem*: a polygonal curve P is approximated by another polygonal curve Q with a given number of line segments M so that approximation error $E(P)$ is minimised.
2. *min - # problem*: a polygonal curve P is approximated by another polygonal curve Q with the minimum number of segments M so that approximation error $E(P)$ less than a given maximum error E_{max} .

An approximation of the curve must satisfy some error criterion. The most of practical error measures in use are based on distance between vertices of the curve and linear segments.

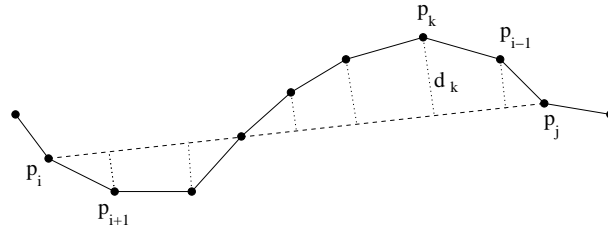


Figure 3.5: Line approximation

Let $d_k(i, j)$ be the distance from curve vertex $p_k = (x_k, y_k)$ to the corresponding approximation linear segments (p_i, p_j) :

$$d_k(i, j) = \frac{|y_k - a_{i,j}x_k - b_{i,j}|}{\sqrt{1 + a_{i,j}^2}} \quad (3.15)$$

where the coefficients $a_{i,j}$ and $b_{i,j}$ are defined from the parameters of linear segment (p_i, p_j) :

$$a(i, j) = \frac{y_j - y_i}{x_j - x_i} \quad (3.16)$$

$$b(i, j) = y_i - a_{i,j}x_i \quad (3.17)$$

The approximation error is defined as the maximum deviation from curve to approximation linear segment:

$$E_{max}(i, j) = \max_{i < k < j} d_k(i, j) \quad (3.18)$$

We consider edge approximation as a *min - #problem* because we are interested only in minimal error approximation. One of the optimal algorithms for *min - #problem* has been proposed by Papakonstantinou [Papakonstantinou, 1985] for error criterion E_{max} .

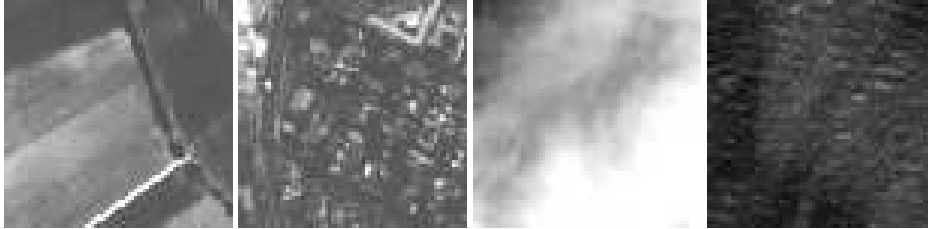


Figure 3.6: Textures: field, city, cloud, sea.

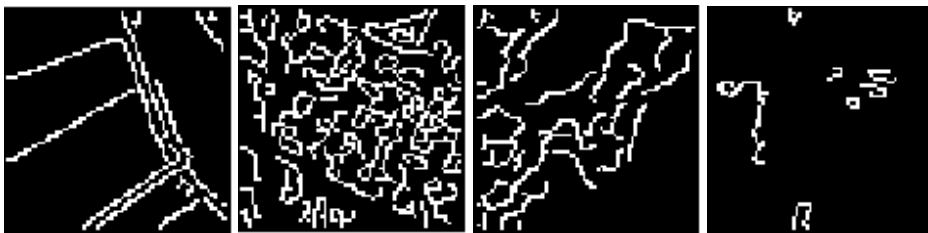


Figure 3.7: Adaptive edge detection: field, city, cloud, sea.

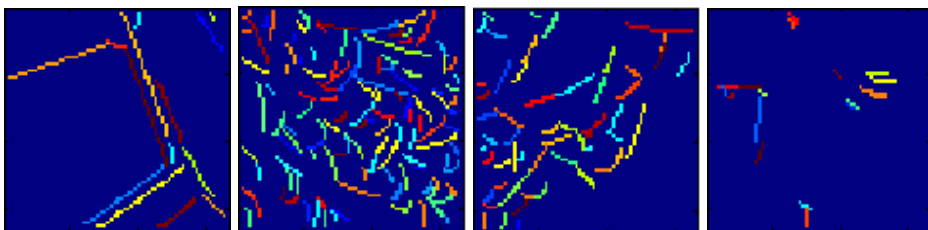


Figure 3.8: Linear approximation: field, city, cloud, sea.

An example of textures is presented in Figure 3.6 and respective images of edges is in Figure 3.7. Figure 3.8 shows a result of linear approximation.

Geometrical features extracted from edges and their approximations by line segments are presented in Appendix B.

An example of geometrical features (histograms of line segment rotations) is presented in Figure 3.9. The first histogram of directions (in Figure 3.9) has one value that

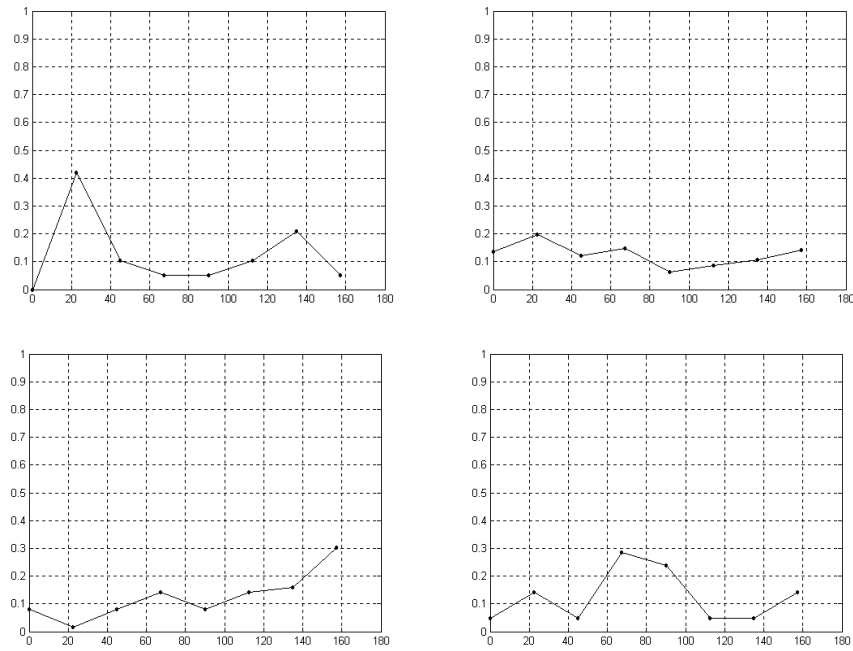


Figure 3.9: The directions histogram of linear segments: a field, a city, a cloud, a sea.

exceeds the level 0.4. Thus, corresponding first image of field in Figure 3.8 has one direction. Other histograms in Figure 3.9 show the images of a city, a cloud and a sea have not directions.

3.4 Conclusions

Feature models describing different properties of the Earth surface in satellite images have been presented in this chapter. These models reflect texture and geometrical features. With the growing quality, size and diversity of satellite images as many informative features as possible should be extracted. Main topics revealed in this chapter are following:

- ★ Statistics calculated on Haralick, Gabor and QMF image models have been considered as texture features. The interest of those features is that they can model pixel intensity and spatial relations among them. A drawback of these models is a priori parameter setting. In the case of satellite images we have predetermined and known characteristics, e.g., image resolution, relative size of objects, etc. It allows fixing parameters of models to get more informative features. But a priori parameters have a limit: models are not sensitive to images and process them equally. It

sometimes may produce insufficient quality of features that influences directly the accuracy of image classification or clustering. This problem should be solved by adaptive parameter estimation. For example, the model of Haralick features has the distance for which pixels are analysed. This parameter should be estimated. Another drawback of Haralick features is that the number of pixels intensities is different from one image to other. This number affects the size of the co-occurrence matrix and computation of statistic. Gabor and QMF features are statistics of filtered images. Here again filter parameters (rotation, size, depth of decomposition) have been fixed, while their estimation is an important and interesting challenge.

- ★ Geometrical features are statistical values of edges and line segments extracted from images. The edge detector using the Deriche filter has been built to get main edges. The estimated adaptive threshold for the detector depends on the mean and the standard deviation of image gradient magnitude. The algorithm of Papakonstantinou has been used to approximate edges by linear segments. But the parameter of the Deriche filter, i.e. scale, has been fixed during filtering. Parameters of geometrical features have been also fixed: the error of the edge approximation by line segments, parameters for binary edge co-occurrences. These features can be improved via adaptive parameter estimating.

Introduced feature spaces describing image textures is used farther for mining satellite images via supervised and unsupervised algorithms.

Part II

Pattern recognition in satellite images

Chapter 4

Supervised classification

Pattern recognition is the main part of data mining and was deeply developed in the last 20 years. Many successful applications have been found in decision-making problems. One of the possible bases of pattern recognition problems is statistical description of data. Such a description is based on models which are used for classification: supervised, semi-supervised or unsupervised. The task of supervised classification is to attribute labels or classes to samples, knowing which classes exist and instances of samples for each class. Semi-supervised classification is made by incorporating in classification human interaction.

In this section we consider supervised classification which assigns a pattern to one of the given classes through a model of classifier. A pattern is described by a set of features. The simplest supervised classification case is the classification into two classes where classes are linearly separated. In this case the classifier model is a hyperplane which separates patterns in the feature space.

4.1 Support Vector Machines (SVM) classification

Recent research has indicated the considerable potential of SVM-based approaches for classification tasks [Vapnik, 1998; ?; Shawe-Taylor & Cristianini, 2004]. One of the recent applications of pattern recognition is supervised classification of remotely sensed data [Huang et al., 2002]. Comparative studies have shown that SVM classification can be more accurate than popular techniques such as neural networks and decision trees as well as conventional probabilistic classifiers such as the maximum likelihood classification [?]. SVMs were designed for binary classification but various methods exist to extend the binary approach to multiclass classification [Vapnik, 1998; Hsu & Lin, 2002]. SVM classification is based on fitting an optimal separating hyperplane between classes by focusing on the training samples that lie at the edge of the class distributions, the support vectors. In other words, it maximises the margin between positive and negative examples. The basis of SVM classification for two classes is illustrated in Fig.4.1.

A training set of patterns \mathbf{x} with known class labels \mathbf{y} is $\{X_i, y_i\}, y_i \in \{1, -1\}, i = 1, \dots, I$, are used to build an optimal hyperplane which should be located between the two classes such that the distance to the closest training data samples in both of the classes is as large as possible. This hyperplane is a decision function defined by the equation of $wX + b = 0$, where X are points lying on the hyperplane, w is the normal to the hyperplane and b is the bias. A separating hyperplane can be defined for the two classes as: $wX_i + b \geq 1$

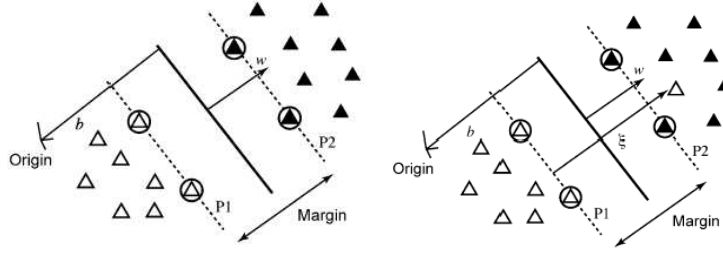


Figure 4.1: Basis of SVM classification. (a) Linearly separable classes and (b) non-linear case.

(for the class $y_i = +1$) and $wX_i + b \leq 1$ (for the class $y_i = -1$). These two equations are combined:

$$y_i(wX_i + b) - 1 \geq 0 \quad (4.1)$$

The support vectors of the two classes lie on two hyperplanes, which themselves are parallel to the optimal hyperplane and are defined by $wX_i + b = \pm 1$. The margin between these planes is $2/\|w\|$ and the analysis aims to maximise this margin through:

$$\min \left\{ \frac{1}{2} \|w^2\| \right\} \quad (4.2)$$

This optimisation problem is solved using Lagrange multipliers:

$$\max J(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle X_i X_j \rangle \quad (4.3)$$

In a case when the classes are not linearly separable (Fig.4.1 (b)), slack variables, $\{\xi_i\}_{i=1}^r$, introduced and Eq.(4.1) may be rewritten as:

$$y_i(wX_i + b) > 1 - \xi_i \quad (4.4)$$

If outliers exist in the data set, Eq.(4.4) can always be satisfied by making ξ_i very large and, so, a penalty term, $C \sum_{i=1}^r \xi_i$ is added to penalise solutions for which ξ_i are very large. Thus, the optimisation problem becomes:

$$\min \left[\frac{\|w^2\|}{2} + C \sum_{i=1}^r \xi_i \right] \quad (4.5)$$

The first part of Eq.(4.5) seeks to maximise the margin between the classes while the second part aims to penalise samples located on the incorrect side of the hyperplane with C .

The basic approach to SVM classification may be extended for nonlinear decision surfaces. In this case, the input data (\mathbf{x}, \mathbf{y}) are mapped into a high dimensional space (\mathbf{X}, \mathbf{y}) through some nonlinear mapping $\phi: \mathbf{z} \rightarrow \mathbf{y}$. Thus, the optimisation of Eq.(4.5) is:

$$\max J(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \phi(X_i), \phi(X_j) \rangle \quad (4.6)$$

subject to $0 \leq \alpha_i \leq C$ and $\sum_i \alpha_i y_i = 0$.
Introducing kernel function (see definition in [Shawe-Taylor & Cristianini, 2004])

$$\mathcal{K}(X, X') = \langle \phi(X), \phi(X') \rangle \quad (4.7)$$

we may represent resulting decision function of an input vector X :

$$f(X) = \text{sgn} \left(\sum_{i=1}^r \alpha_i y_i \mathcal{K}(X, X_i) + b \right) \quad (4.8)$$

where α_i are Lagrange multipliers and $\mathcal{K}(X, X_i)$ is a kernel function. The magnitude of α_i is determined by the parameter C and lies on a scale of 0 - C . A kernel widely used is the radial basis function:

$$\mathcal{K}(X, X_i) = e^{-\gamma \|X - X_i\|^2} \quad (4.9)$$

where γ is the parameter controlling the width of the Gaussian kernel. There are several arguments for radial basis function: (i) only one parameter is needed for this kernel and (ii) practically proved to be efficient in numerous applications [Vapnik, 1999]. The classification accuracy of SVM depends on the magnitudes of the parameters C and γ . With a large value of γ and/or C , there is a tendency for the SVM to overfit the training data.

4.2 Curse of dimensionality and feature selection algorithms

We have seen from previous Chapter that images are described as "large" set of features ("large" means from tens features to hundreds). It is very important to take into account the dataset dimension (the number D of features). This number may significantly influence classification results as shown in [Bishop, 2006]. For example, we consider a sphere with radius $r = 1$ in D dimensions and calculate ratio between the sphere volume $V(r, D)$ and the sphere volume with radius $r = 1 - \epsilon$. Without loss of generality we consider the volume of sphere with radius r as $V(r, D) = c(D)r^D$, where $c(D)$ is a constant. Then the ratio for radius $r = 1$ and $r = 1 - \epsilon$, $0 < \epsilon$ is:

$$\frac{V(1, D) - V(1 - \epsilon, D)}{V(1, D)} = \frac{c(D) - c(D)(1 - \epsilon)^D}{c(D)} = 1 - (1 - \epsilon)^D. \quad (4.10)$$

From this ratio we see that for "large" values of D and even for small value of ϵ the ratio tends to 1. It means that the volume also tends to 1 and points which fill the sphere are located near the border of the sphere. Consequently, the distance among uniformly distributed points in the D dimensional sphere tends to 1 as can be seen from Figure (4.2a). Assume that samples belong to several classes in the sphere and we want to attribute a new sample to one class. From Eq. (4.10) we see that the Euclidean distance from a new sample to samples of different classes also tends to 1. In addition, if we consider Gaussian probability distribution of points in a space of dimension D , then from Figure (4.2b) we see that probability mass of a Gaussian distribution is mainly located within a thin layer at some radius [Bishop, 2006].

This aspect is very important in pattern recognition and data mining tasks. It shows that some basic algorithms of pattern recognition can not be directly applied to data of high dimension and should be used carefully. One of the solutions to this problem may be in feature weighting or feature selection. Feature weighting is a procedure assigning to

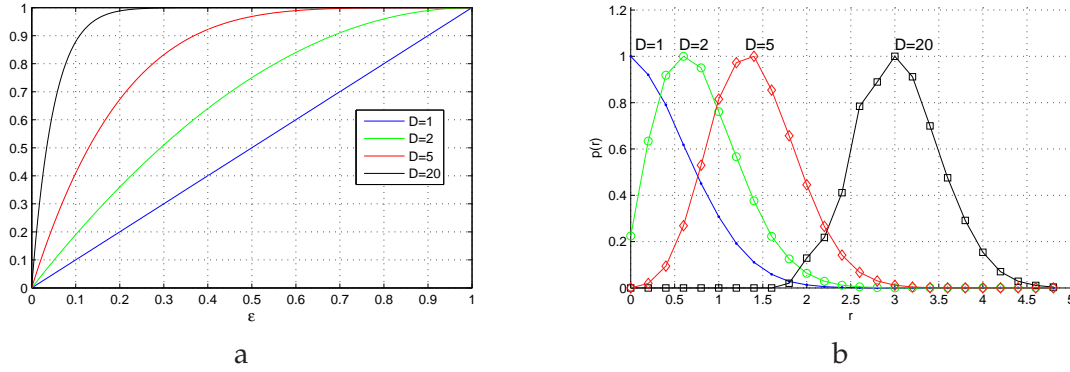


Figure 4.2: Curse of dimensionality for various values of the dimensionality D . a - Relation of volumes of radius $r = 1 - \epsilon$ to r . b - Normalised probability density *w.r.t.* radius r of a Gaussian distribution for different values of D .

a feature a weight either through prior knowledge (often difficult to have) or through an algorithm which estimates feature weights during classification process. Feature selection is able to determine a more appropriate feature subset to reflect useful information in data. In addition, data processing time is decreasing after feature selection.

Reduction of feature space dimension could be used to decrease overfitting and improve classification. One of these pruning techniques is Recursive feature selection (RFE) Guyon [2002]. RFE eliminates some of the original input features and retains a feature subset that provides best classification performance. Detailed survey of feature selection approaches, especially for satellite image processing, may be found in [Campedel et al., 2004]. For an illustrative example we use RFE feature selection based on the following iterative procedure:

1. Train the SVM classifier;
2. Compute the ranking criterion for all features w_i^2 ;
3. Remove the feature with smallest ranking criterion.

The iterative procedure stops when the desired number of features is obtained. This selection approach produces a very good result for the classification but a priori information about classes should be known. We are rather interested in unsupervised feature selection because we have a lot of unstructured information. In Chapter 8 a new approach for unsupervised feature selection is presented. It selects features and gives very good classification results.

4.3 SVM classification of satellite images

SVM approach is widely used in many applications of supervised classification. Especially it is implemented in many systems of satellite image processing [Parulekar et al., 2005]. It also has been shown its usefulness in many works to classify satellite images: for supervised classification [Bhattacharya et al., 2007; Zammit et al., 2007] as well

as for semi supervised classification or relevance feedback [Ferecatu & Boujemaa, 2007; Costache & Datcu, 2007].

In this Section we propose three experiments:

1. classification of satellite image samples;
2. classification of samples with feature selection;
3. classification of complete scenes of satellite images.

To classify data base of SPOT5 satellite images four classes of textures are used: fields, cities, clouds and sea Fig.3.6. Each class contains 100 examples issued from different satellite images. Each sample has size of 64×64 pixels. 15 geometrical features presented in Section 3.3 have been extracted from each sample. These features reflect geometrical structure of image samples (distribution of edges, line segments, etc.).

In the first experiment we classify samples using the complete feature set. We apply a cross validation procedure with a training set (75% of images) and a test set (25% of images). Results of image classification are presented in Table 4.1.

Table 4.1: Classification table of database of textures for the whole set of 15 geometrical features. Classification accuracy is 89 percent

	Assigned class				Total
	Cloud	Sea	City	Field	
Cloud	93	3	2	2	100
Sea	10	89	0	1	100
City	3	0	86	11	100
Field	3	1	7	89	100
Total	109	93	95	103	400

The general accuracy of classification is obtained as the sum of diagonal elements in the table, divided by the total number of samples. The classification accuracy is 89 percent that is a good enough result. In Table 4.1 most of the samples from all four classes have been correctly classified and a maximal confusion between classes is less than 11 percent.

In the second experiment we perform classification with feature selection. To determine the most significant features and suppress insignificant ones a recursive feature elimination method based on SVM (SVM-RFE) has been applied [Campedel et al., 2004]. The number of selected features should be set for the SVM-RFE approach. Figure 4.3 shows the classification error as a function of the number of features. A minimal error of classification is obtained for 10 features: $X_1, X_2, X_3, X_5, X_9, X_{10}, X_{12}, X_{13}, X_{14}, X_{15}$, see Appendix B.

It can be seen from Figure 4.3 that selected features are: the number of line segments (X_1 and X_2), their length (X_3 and X_5), direction of line segments (X_{12}), statistical features of frequencies (X_{13}, X_{14} and X_{15}) and statistics of pixel distribution (X_9 and X_{10}). They are more important and decrease classification error.

Now a cross validation procedure is applied to classify samples, using selected 10 features (Table 4.2). The accuracy of classification is 92 percent which improves the previous classification result. From Table 4.2 we see that the maximal confusion between any two classes equals 7 samples that is less than 11 for classification with the complete feature set.

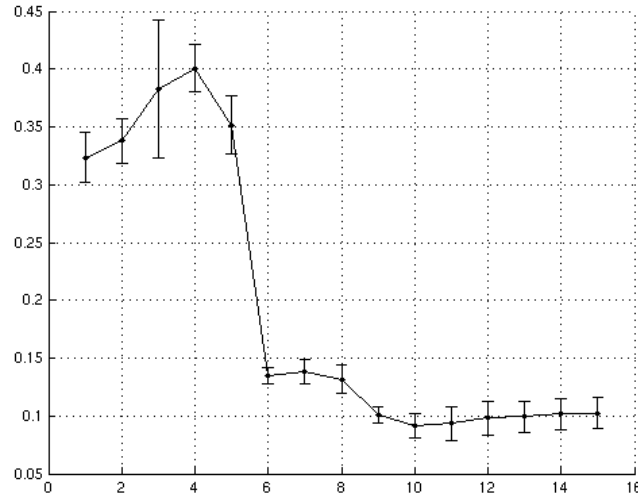


Figure 4.3: Classification error versus the number of features.

Table 4.2: Classification table of database of textures for 10 selected geometrical features ($X_1, X_2, X_3, X_5, X_9, X_{10}, X_{12}, X_{13}, X_{14}, X_{15}$). Classification accuracy is 92 percent.

	Assigned class				Total
	Cloud	Sea	City	Field	
Cloud	96	2	1	1	100
Sea	4	95	0	1	100
City	7	0	86	7	100
Field	3	0	6	91	100
Total	110	97	93	100	400

In the third experiment we demonstrate results of classification on satellite images. Satellite images of different world cities have been selected: Béziers, Paris, Los Angeles and Hong Kong. We suppose that each image contains different textures of the Earth surface which reflect different natural and architectural configurations.

The protocol of feature extraction is the same as in previous experiments: original images are cut into samples of size 64×64 pixels and then features are calculated for each sample. Training of SVM classifier is performed on the same data base of 4 classes, 400 samples and 10 best features. Testing (classification) is given for samples of each image.

First we propose to classify images of Béziers and Paris which have a size of 3000×3000 pixels. The original images of Bezier and Paris are presented in Figures 4.4a and 4.5a, and their images of edges are shown in Figures 4.4b and 4.5b, respectively. Images of edges in Figures 4.4b and 4.5b shows that "smooth" surfaces as "Sea", "Forest", and "Cloud" have no edges. It shows that geometrical features may distinguish textures with edges and without edges. The best 10 geometrical features selected in the previous experiment have been calculated for image samples. The classified satellite images of Bezier and Paris are presented in Figures 4.4c and 4.5c, respectively.

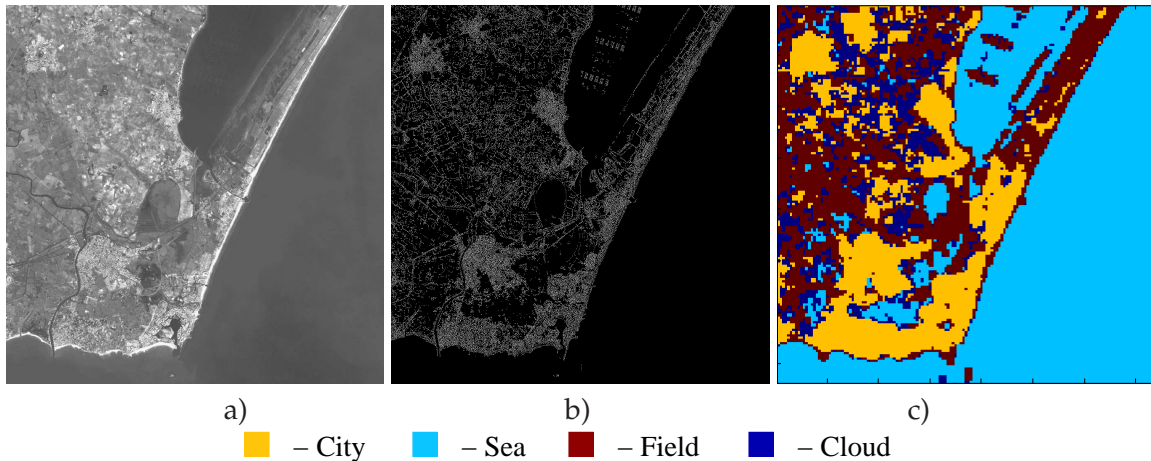


Figure 4.4: Classification of SPOT5 image of Béziers, ©CNES. a - an original satellite image SPOT5, b - image of edges, c - classification result.

With the selected set of features, smoothed regions such as a sea, clouds, forests and fields which have no edges are classified as the class "Sea". We observe in Figure 4.4c that a coast line is classified as class "Field". It explained by the fact that the coast has long strong lines like fields. The image of Paris in Figure 4.5c mainly contains class "City" and shows a good enough classification.

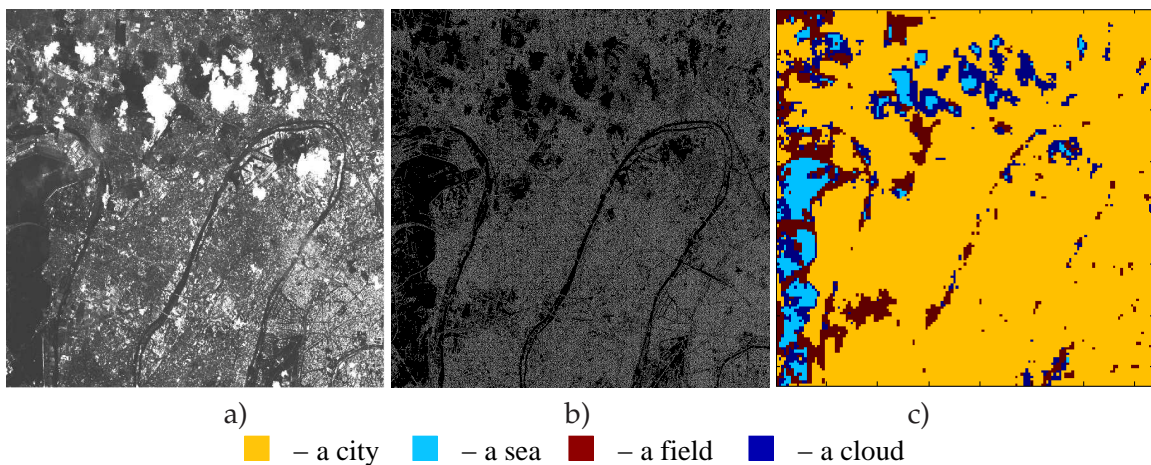


Figure 4.5: Classification of SPOT5 image of Paris, ©CNES. a - an original satellite image SPOT5, b - image of edges, c - classification result.

Another experiment consists in classifying images of Los Angeles, Hong Kong and Peking presented in Figures 4.6a, 4.7a and 4.8a, respectively. These cities have been selected to demonstrate diversity of landscapes.

Each image has a size of 512×512 pixels. Corresponding images of edges are shown in Figures 4.6b, 4.7b and 4.8b. Here again, we cut images into samples of size 64×64 pixels and compute 10 geometrical features for each sample. After, trained SVM classify samples in 4 classes.

Results of classification for Los Angeles, Hong Kong and Peking are given in Figures 4.6c, 4.7c and 4.8c, respectively. Visually two classes ("City" and "Field") are dominant

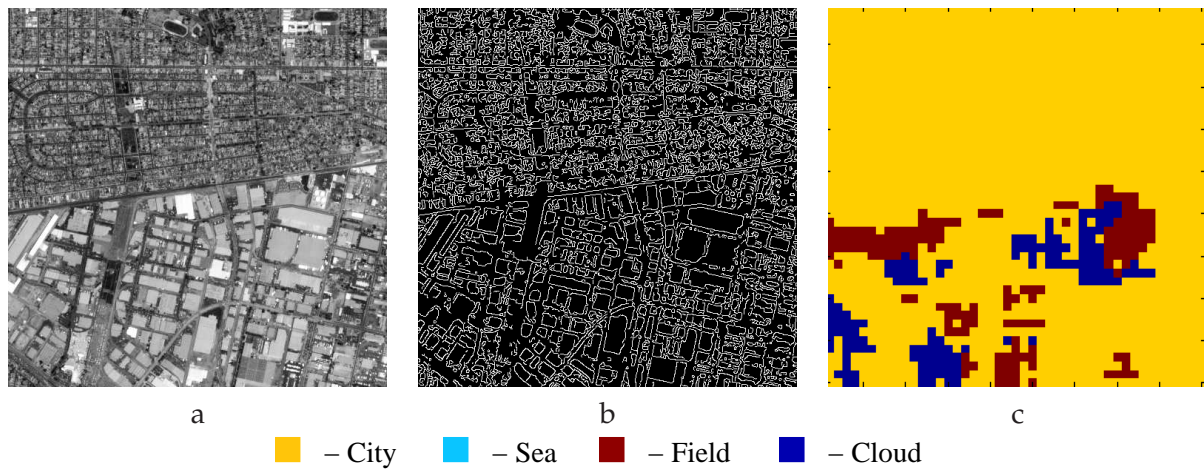


Figure 4.6: Classification of SPOT 5 image of Los Angeles, ©CNES.

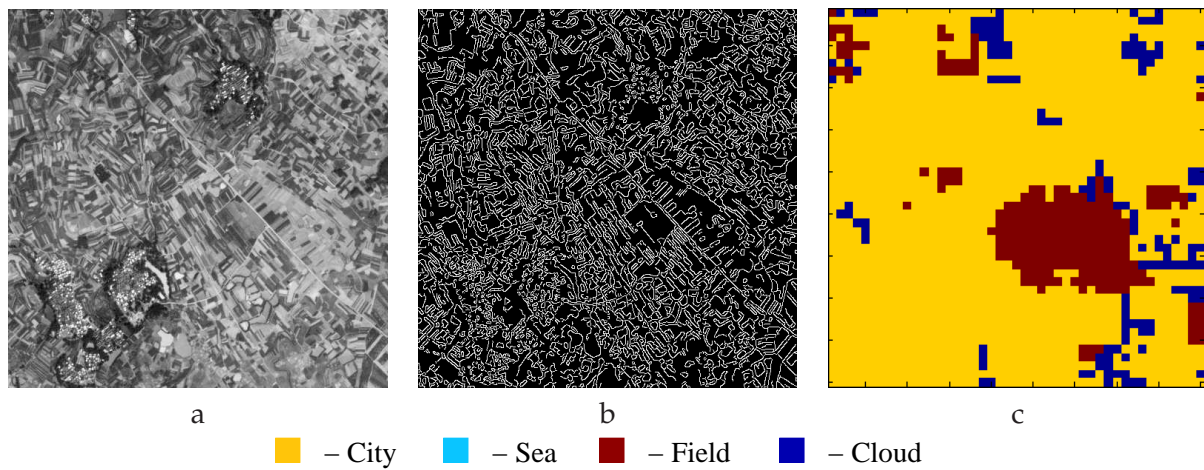


Figure 4.7: Classification of SPOT 5 image of Hong Kong, ©CNES.

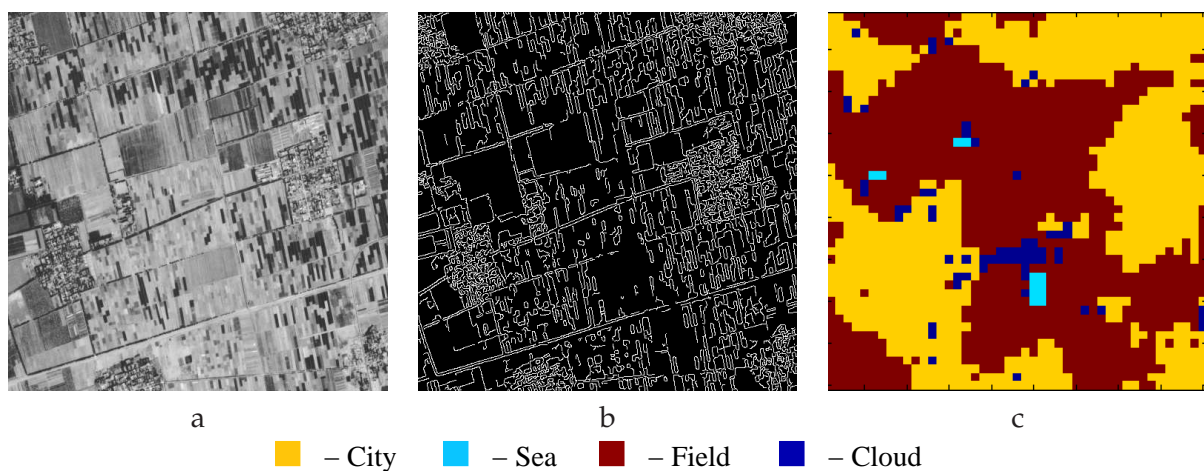


Figure 4.8: Classification of SPOT 5 image of Peking, ©CNES.

in the images, Figures 4.6a, 4.7a and 4.8a. The image of Los Angeles in Figure 4.6c is well classified except the small part of class "Field". The presence of class "Field" in the urban zone is explained by the fact that warehouses in industrial zones have the same geometrical features as fields (long straight lines, etc.). Another confusion between classes "City" and "Field" is observed from the image classification of Hong Kong, Figure 4.7c. The classified image in Figure 4.7c contains mostly class "City" and small part of class "Field", while the original image in Figure 4.7a has only several small urban zones and mostly has fields. This confusion comes from the tradition of organising villages and agriculture field: fields are of very small sizes and have the near same geometrical features as urban zones. The same type of confusion is observed from image classification of Peking, Figure 4.8c.

Image classifications in Figures 4.4c - 4.8c show that in spite of capturing information about structures by geometrical features, sometimes they are not sufficient to discriminate semantically different classes (e.g., "Field" and "City").

To overcome the lack of geometrical description we give an example of image classification with geometrical and texture features. An example of satellite image of Honk Kong (3000 \times 3000 pixels) is shown in Figure 4.10a. Samples of size 64 \times 64 pixels have been issued from the image. Three classes "Mountain", "Village" and "Field" each of size 25 samples have been defined for SVM classification, Figure 4.9.

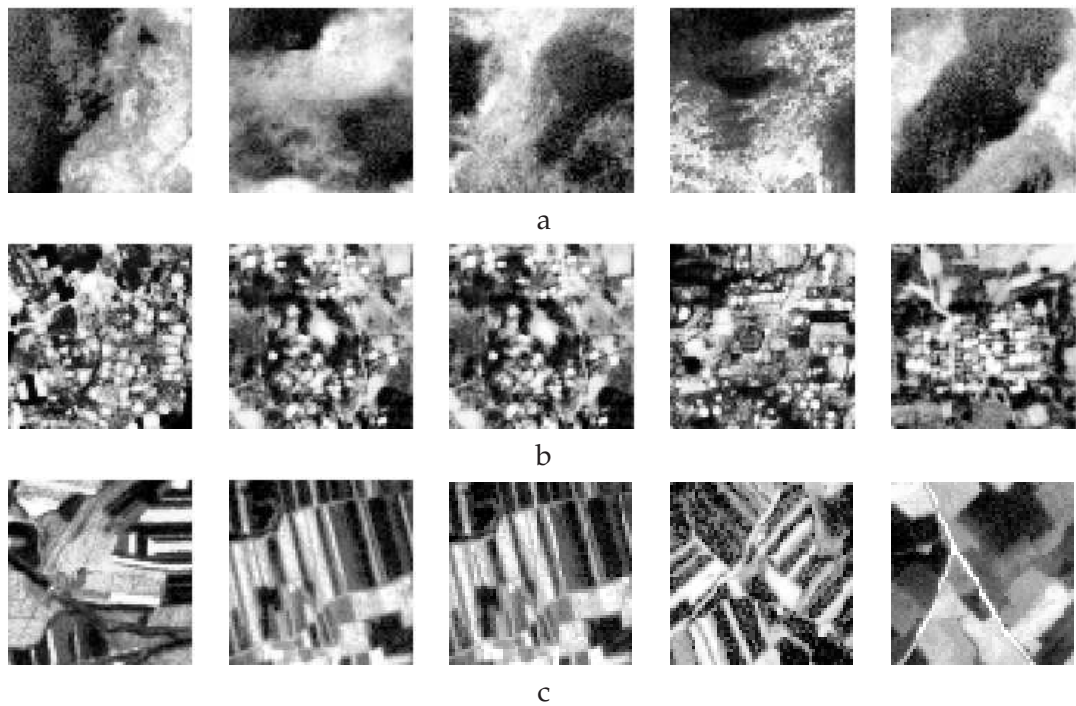


Figure 4.9: Samples of classes issued from SPOT5 image in Figure 4.10a: a - "Mountain", b - "Village", c - "Field". ©CNES

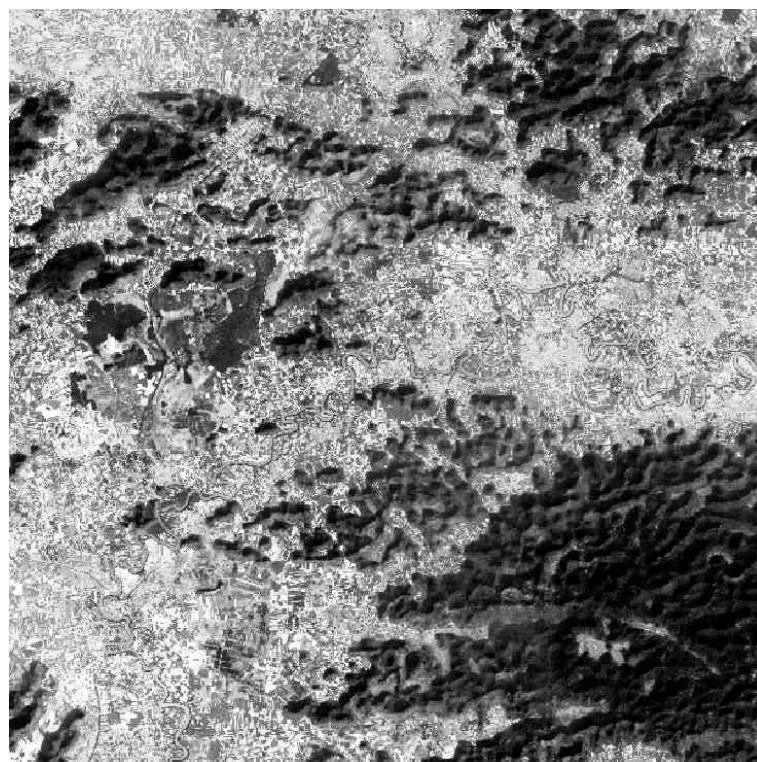
Textures and geometrical features have been extracted from each sample. Then SVM-RFE approach has selected 30 best features by classifying 75 samples (25 per class). Finally, trained SVM result the supervised image classification which is shown in Figure 4.10b. We observe from Figures 4.10a and 4.10b that the original image is well classified by three classes ("Mountain", "Village" and "Field"). This experiment demonstrates that

geometrical features complement textural features.

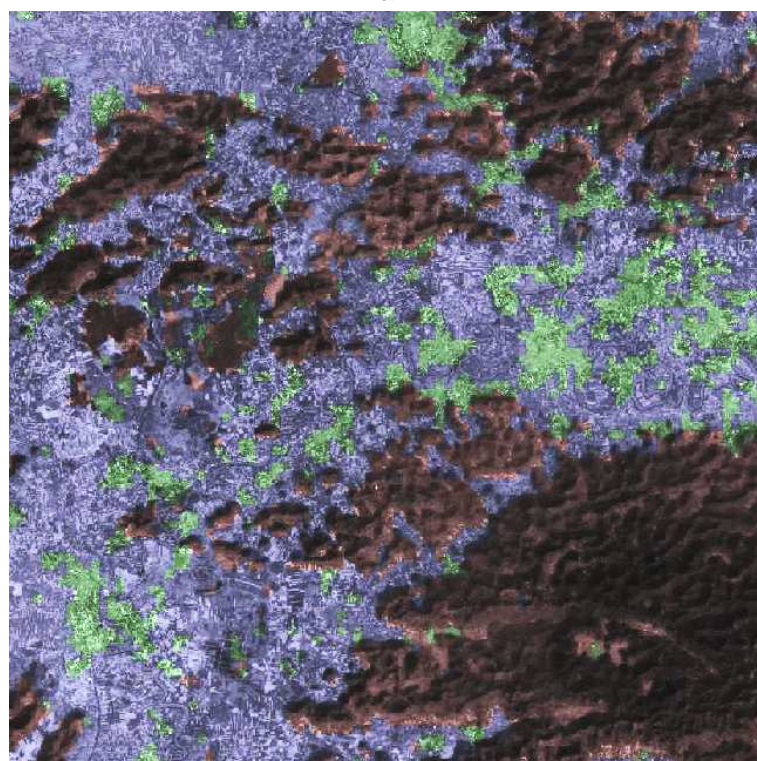
4.4 Conclusions

In this Chapter supervised classification and feature selection have been introduced and applied to satellite images. The problem of a large feature set called curse of dimensionality has been also presented. SVM classification of satellite images with structural features has been demonstrated. As it has been shown, very large space of data may lead to wrong classification results. In addition, mining high dimensional data is time consuming. Data dimension has been reduced via feature selection by SVM-RFE.

Supervised SVM classification using geometrical features shows interesting properties of satellite images, e.g., sharpness of detected structures. However using only geometrical features is limited approach and they should be completed by texture features to reflect different properties of surfaces.



a



b

Figure 4.10: Satellite image SPOT5 of Honk Kong ©CNES: a - original image 3000×3000 pixels, b - SVM classification for texture and geometrical features (red - "Mountain", green - "Village" and blue - "Field").

Chapter 5

Unsupervised classification. Clustering algorithms

Data modelling is in charge of representing knowledge. It may also help in extracting information. When there is no or a few prior information about data then unsupervised methods should be used. Unsupervised classification is one way of modelling data. It estimates optimal data model parameters and verifies how good data are replaced by the model. There are several problems of model estimation for which we should pay attention:

1. the choice of the data model,
2. if the model parameters are partially known or not known they should be estimated,
3. the estimation approach should also be selected and argued.

To understand a content of data set we should first find their composing elements (clusters and/or classes). One of the ways to find clusters or classes in data may be considered via estimation of the data model. The model indicates how data distributed in clusters (classes). Then, relationships between clusters may be presented as links between them, as for instance in the form of a graph or a hierarchical tree.

There is a variety of directions to discover classes via unsupervised classification. One of them is referred in the literature as clustering. Earlier references about clustering and pattern recognition methods may be found in [Jain & Dubes, 1988; Fukunaga, 1990] while recent approaches and formulations are proposed in [Mclachlan & Peel, 2000; Duda et al., 2000; Friedman et al., 2001; Rencher, 2002; Rowe, 2002; Theodoridis & Koutroumbas, 2003; Mackay, 2002; Hardle et al., 2003; Bishop, 2006].

Clustering is an automatic process which discovers clusters (groups of similar data) and assigns a data sample to each of cluster. In the next Section we represent a hierarchy of clustering algorithms.

5.1 State of the art

One of the earlier surveys of different clustering methods and algorithms has been presented in [Jain & Dubes, 1988]. Usually clustering techniques are divided in partitional and hierarchical. Partitional clustering is a division of the samples into K groups, or

clusters, such that the samples in a cluster are more similar to each other than to samples in different clusters. Hierarchical clustering methods are categorised into divisive and agglomerative also called bottom-up and top-down [Jain & Dubes, 1988; Rencher, 2002; Friedman et al., 2001]. A divisive clustering starts with one cluster which contains all samples and splits it into the most appropriate clusters in a recursive way. We present a schema of the various clustering algorithms for pattern recognition in Figure 5.1. The

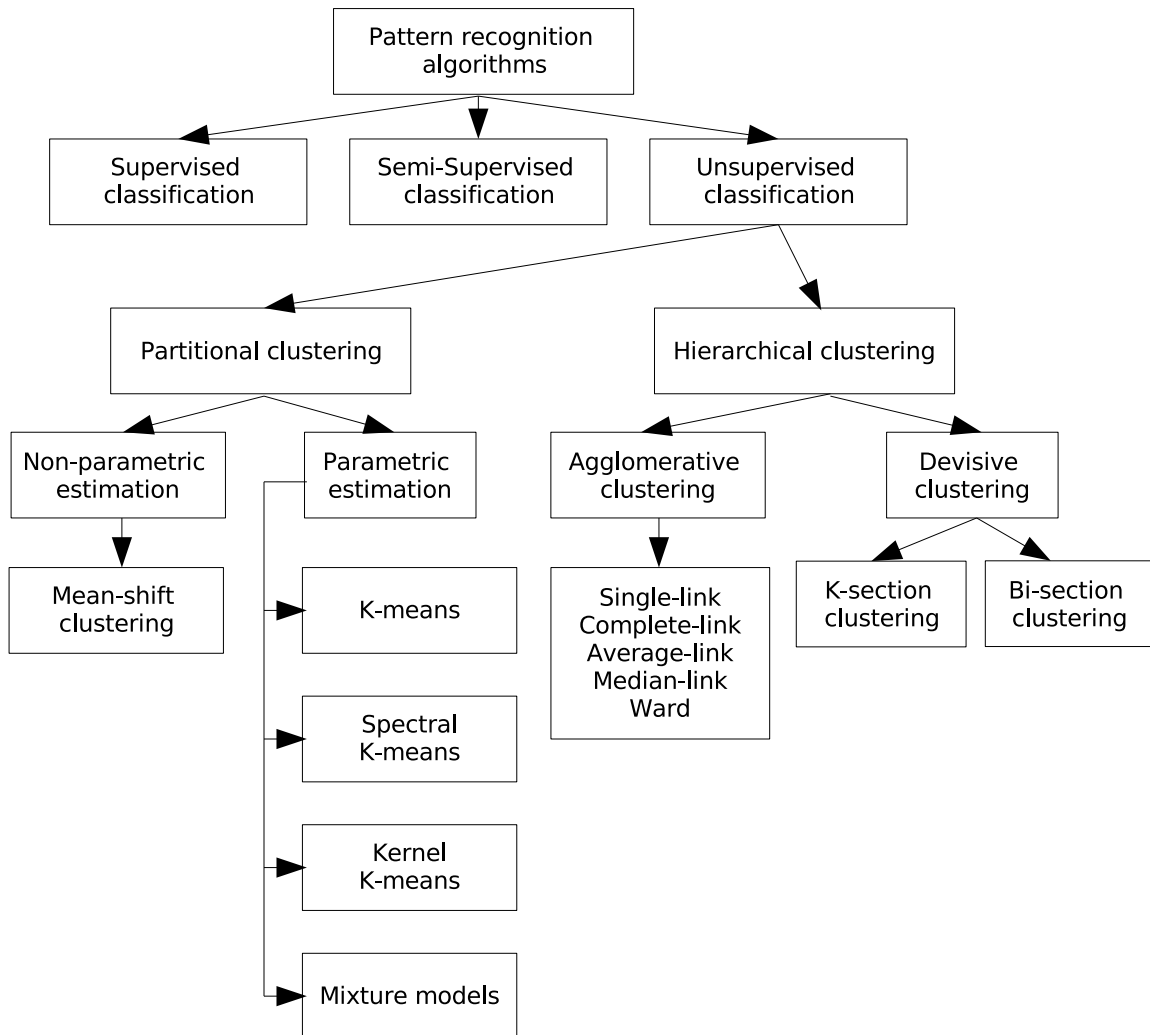


Figure 5.1: An overview of clustering algorithms for pattern recognition.

advantage of divisive algorithms is that the time and memory complexity is very low. On the contrary, it suffers from local optimality of found clustering solutions. An agglomerative clustering starts with one-point (singleton) clusters and merges two or more most appropriate clusters. Clustering algorithms treat the clustering problem as an optimisation process which tries to maximise or minimise a particular clustering criterion function [Friedman et al., 2001; Rencher, 2002; Webb, 2002].

5.2 Combinatorial search

Let us have a look to the combinatorial issue of the search for optimal clustering.

The number of possible solutions to obtain all possible data partitions of size I is given by Stirling number [Jain & Dubes, 1988]:

$$\frac{1}{K!} \sum_{k=1}^K (-1)^{(K-k)} \binom{K}{k} (k)^I, \quad (5.1)$$

where K is the number of clusters of a desired partition. Clearly this number is too high for most of practical cases. It grows up to 10^{155} with 100 samples. The direct search of clustering could be applied only to a very small set of samples.

One of the ways to work with the direct search may be in computing some statistical quantities on data to restrict the search range. For example, to compute the lower and higher bounds of possible partition numbers. Other links on this topic for a few samples could be found in [Jain & Dubes, 1988].

5.3 Hierarchical clustering algorithms

Hierarchical clustering is a nested data partition. It is represented by a hierarchical tree or a *dendrogram*. Each clustering corresponds to a certain level of the hierarchical tree [Theodoridis & Koutroumbas, 2003]. We consider the case of hierarchical clustering when clusters of a partition at a certain level are completely included in clusters of higher level. The top level of hierarchical tree is the root and contains all data set, the lower level of the tree may contain leaves which correspond to data samples, so that each cluster of this level has one sample only.

Hierarchical agglomerative clustering algorithms

Methods of agglomerative clustering consist in merging clusters at a certain level of tree. Here we consider only pairwise merging of clusters (a frequent issue to solve practical problems [Rencher, 2002]). There is no guarantee, in general, that pairwise merging may produce a global optimal representation of data or a global optimal solution of an objective function. From the other hand, several clusters can be merged at each step, however this method may entail an exponential time and memory complexity. An agglomerative hierarchical clustering builds a hierarchical tree from initial partition using a pairwise matrix of distances between clusters (clusters may contain only one sample). After merging two clusters, a distance matrix should be updated: distances from the merged cluster to other clusters should be reestimated. Depending on the method of choice of two clusters to be merged and the method to calculate distance matrix updating, several hierarchical clustering methods exist.

A pseudo code of hierarchical clustering algorithm is given here (**Algorithm 5.3**), with the same definitions as in Chapter 4. Let C_u and C_v be any two different clusters in cluster set C , $C_u, C_v \in C$, where $u, v = 1, \dots, K$, $u \neq v$ and K is the cluster number in C . Let clustering C^1 be included in C^2 : $C^1 \in C^2$. The agglomerative algorithm produces a hierarchical tree of included clusterings $C^1 \in C^2 \in \dots C^{k-1} \in C^K$. Let $d(C_u, C_v)$ be a dissimilarity function between any two clusters C_u and C_v , e.g., a distance, and t be the current level of hierarchy.

Algorithm 5.3 Pseudo code of agglomerative algorithm

-
- 1: Initialise $t = 0$ and C^0 as the initial data clustering, *e.g.*, $C_i^0 = X_i, i = 1, \dots, I$
 - 2: $t = t + 1$
 - 3: For all $(C_u, C_v) \in C^{t-1}, \forall u, v$ find (r, s) as
 - 3.1: $d(C_r, C_s) = \min_{u,v} d(C_u, C_v)$
 - 3.2: Set $C_q = C_r \cup C_s$
 - 3.3: New clustering $C^t = (C^{t-1} - \{C_r, C_s\}) \cup C_q$
 - 4: Go to **Step 2**
-

Direct search for all possible pairs (u, v) in the matrix d is able to construct a hierarchical tree with K levels, when $K = I$. The algorithm complexity is $\mathcal{O}(I^3)$ that makes difficult to apply it for large datasets [Theodoridis & Koutroumbas, 2003], and it may be inapplicable for short time calculations when a large volume of data is processed. In addition, the computation complexity of the similarity matrix d should be taken into account. For practical realisation of the hierarchical algorithm and in order to reduce computation complexity we suggest to choose C^0 with a small number of clusters. The initial clustering C^0 may be obtained by simple algorithms, *e.g.*, K-means partitional clustering which has a linear complexity. This algorithm will be presented later in Section 5.4.

Different agglomerative clustering algorithms depend on the choice of matrix d and the method which is used to merge two clusters. Below we consider the main ones: single-link, complete-link, average-link, median-link and Ward agglomerative clustering algorithms. The generalisation of these approaches will also be done. These algorithms are general and may be found in [Jain & Dubes, 1988; Duda et al., 2000; Rencher, 2002; Theodoridis & Koutroumbas, 2003]. For the simplicity of notations, we will use u and v instead of C_u and C_v .

Single-link method

At each merging stage of single-link algorithm two neighbouring clusters u and v are merged to one cluster. The minimal distance between clusters is the distance between two closest samples of these clusters:

$$d_{uv} = \min_{i \in u, l \in v} \text{dist}(X_i X_l), \quad (5.2)$$

where $i = 1, \dots, n_u, l = 1, \dots, n_v$ are the number of samples in clusters u and v , respectively. As we can see this algorithm has a quadratic computation complexity $\mathcal{O}(I^2)$ to find nearest neighbour elements. Different techniques or a priori knowledge may reduce this complexity, *e.g.*, in image processing for searching we may consider only neighbouring pixels. This approach may find clusters with complex shapes, *e.g.*, elongated, spirals etc. But it is influenced by noise: small noise may lead for merging two different clusters. Some extensions of this approach may be developed by considering several neighbours when calculating the distance between clusters. But one should set a priori or estimate the optimal neighbours' number.

Complete-link method

Here again, as in previous case a hierarchical tree is built by merging two nearest clusters u and v ; but distance between them is calculated as two farthest-neighbour samples belonging to these clusters:

$$d_{uv} = \max_{i \in u, l \in v} \text{dist}(X_i X_l). \quad (5.3)$$

This algorithm differs from single-link by computation of the farthest distance between clusters. Complete-link algorithm seeks cohesion of the clusters, contrary to single-link which looks for isolated clusters.

Average-link method

The average-link approach search to merge two neighbour clusters u and v when the distance between them is the average pairwise distance between points of these clusters:

$$d_{uv} = \frac{1}{n_u n_v} \sum_{i=1}^{n_u} \sum_{l=1}^{n_v} \text{dist}(X_i X_l), \quad (5.4)$$

where n_u and n_v are the numbers of samples in clusters n_{j_1} and n_{j_2} , respectively. This algorithm differs from previous two in the way that it is less sensitive to noise. From the other hand, this algorithm tends to find globular clusters and will not find clusters with complex shapes. Its statistical properties are mentioned in [Friedman et al., 2001].

Centroid-link method

For the centroid link method two nearest clusters are merged and the distance between them is computed as the distance between centroids of these clusters:

$$d_{uv} = \text{dist}(\bar{X}_u, \bar{X}_v), \quad (5.5)$$

where \bar{X}_u and \bar{X}_v are mean vectors of clusters u and v , respectively. Mean vectors are $\bar{X}_u = \sum_{i=1, X_i \in u}^{n_u} X_i$ and $\bar{X}_v = \sum_{i=1, X_i \in v}^{n_v} X_i$. After merging two clusters u and v the new centroid is found as:

$$\bar{X}_{uv} = \frac{n_u \bar{X}_u + n_v \bar{X}_v}{n_u + n_v}. \quad (5.6)$$

Median-link method

For the centroid-link method a cluster with higher number of points has higher weight in calculating a distance. To avoid this problem a median approach may be applied:

$$\mathbf{m}_{uv} = \frac{n_u \bar{X}_u + n_v \bar{X}_v}{2}, \quad (5.7)$$

where \mathbf{m}_{uv} is a median distance between two clusters which corresponds to the midpoint of a line connecting two clusters u and v . This median has no relation to a statistical median, it is related to the geometrical median of a triangle which connects a vertex with a midpoint of the opposite side [Rencher, 2002].

Ward's method

This approach is based on the minimisation of the square error in each cluster and is named as minimum variance method. It has been shown that Ward's method [Ward, 1963] outperforms hierarchical methods described above. Let the sum of within cluster distances for cluster u , v and their combination C_{uv} be:

$$SSE_u = \sum_{i=1}^{n_u} (X_i - \bar{X}_u)'(X_i - \bar{X}_u), \quad (5.8)$$

$$SSE_v = \sum_{i=1}^{n_v} (X_i - \bar{X}_v)'(X_i - \bar{X}_v), \quad (5.9)$$

$$SSE_{uv} = \sum_{i=1}^{n_{uv}} (X_i - \bar{X}_{uv})'(X_i - \bar{X}_{uv}). \quad (5.10)$$

where \bar{X}_{uv} is as in Eq. (5.6) and $n_{uv} = n_u + n_v$ is the number of points in the merged cluster $u \cup v$. Ward's method aggregates two clusters u and v which minimise the increasing within-cluster distance of their merging:

$$\Delta = SSE_{uv} - (SSE_u + SSE_v) = \frac{n_{uv}}{n_u + n_v} (\bar{X}_u - \bar{X}_v)'(\bar{X}_u - \bar{X}_v) \quad (5.11)$$

Ward's method tends to merge clusters with a small or equal number of samples as shown in [Jain & Dubes, 1988; Rencher, 2002]. Updated distance between any of the remaining clusters k , where $k = 1, \dots, K$ and merged clusters uv is:

$$d(k, uv) = \frac{n_k + n_u}{n_k + n_{uv}} dist(k, u) + \frac{n_k + n_v}{n_k + n_{uv}} dist(k, v) + \frac{n_k}{n_k + n_{uv}} dist(u, v) \quad (5.12)$$

General agglomerative algorithm

Hierarchical clustering methods above presented may be viewed as methods with special parameters for updating the matrix of distances [Lance & Williams, 1967]. After merging of clusters C_u and C_v the pairwise distance from this new cluster to any other r is updated as:

$$d(r, uv) = \alpha_u dist(r, u) + \alpha_v dist(r, v) + \beta dist(u, v) + \gamma |dist(r, u) - dist(r, v)| \quad (5.13)$$

The authors in [Lance & Williams, 1967] propose to simplify Eq. (5.13) by introducing constraints:

$$\begin{cases} \alpha_u + \alpha_v + \beta = 1, \\ \alpha_u = \alpha_v, \\ \gamma = 0, \\ \beta < 1. \end{cases} \quad (5.14)$$

From the equation (5.14) we see that $2\alpha_u = 1 - \beta$ and $\alpha_u = \alpha_v = (1 - \beta)/2$. It means that for the general hierarchical clustering algorithm with updating distances Eq. (5.14) we should define only one parameter β . This hierarchical clustering algorithm is called the flexible beta method. Parameters are summarised in Table 5.1.

General equation of the hierarchical clustering Eq. (5.13) with parameters from Table 5.1 is added to Algorithm 5.3 and produces a general approach to cluster data by the hierarchical agglomerative algorithm. Complexity of this algorithm may be reduced using relatively small number of clusters to construct a hierarchical tree.

Table 5.1: Parameters of the flexible beta method

Clustering Method	α_u	α_v	β	γ
Single-link	1/2	1/2	0	-1/2
Complete-link	1/2	1/2	0	1/2
Average-link	$\frac{n_u}{n_u+n_v}$	$\frac{n_v}{n_v+n_u}$	0	0
Centroid	$\frac{n_u}{n_u+n_v}$	$\frac{n_v}{n_v+n_u}$	$\frac{-n_v n_u}{(n_v+n_u)^2}$	0
Median	1/2	1/2	-1/4	0
Ward's method	$\frac{n_v+n_{vu}}{n_v+n_u+n_{vu}}$	$\frac{n_u+n_{vu}}{n_v+n_u+n_{vu}}$	$\frac{-n_{vu}}{n_v+n_u+n_{vu}}$	
Flexible beta	$(1-\beta)/2$	$(1-\beta)/2$	$\beta(<1)$	0

Hierarchical divisive clustering algorithms

Dividing methods are not popular and are rarely met in the literature [Jain & Dubes, 1988; Rencher, 2002; Webb, 2002]. However we should list this approaches for a complete observation of hierarchical methods and to show some of their interesting aspects. Divisive hierarchical algorithms divide iteratively data into clusters. During division they construct a hierarchical tree. In the literature [Jain & Dubes, 1988; Rencher, 2002] divisive hierarchical clustering are considered into two groups: monothetic and polythetic. Monothetic algorithms use consequently one by one feature to divide data while polythetic algorithms use all features to divide data. For monothetic algorithms an order of features should be set or estimated. Only polythetic algorithms are considered in the thesis because a complete set of features is more informative than each separate feature.

Another classification of divisive hierarchical algorithms may be presented as Bi-section and K-section or two- or multi-way clustering [Chan et al., 1994]. For bi-section we use an algorithm to divide data into two clusters and so on we divide each subcluster in two. Similarly, for K-section algorithm on each step we divide data into K clusters [Jain & Dubes, 1988]. Bi-section algorithms may be successfully applied to data which have linearly separated clusters in the feature space, if it is not the case, it may fail. K-section algorithm is used when clusters have more complex shapes. In general, divisive hierarchical clustering algorithms do not provide an optimal solution and may result a local optimum of clustering. The advantage of divisive clustering algorithms is that they may build a hierarchical tree for a high volume of data very quickly and with significantly less memory and time complexity than agglomerative hierarchical clustering. Next, we give examples and descriptions of polythetic Bi- and K-section algorithms.

Bi-section clustering algorithms

Bi-section algorithm may be considered to divide data by some criterion. Here we propose to draw our attention to maximisation of square Euclidean distances between divided clusters. As it is easy to show, this distance also minimises square errors of each cluster or within-clusters distances. Therefore we may apply such an algorithm as K-means [Jain & Dubes, 1988] to divide successively data into two clusters. Introduction to this and other partitional algorithm is given in the next section. In [Shawe-Taylor & Cristianini, 2004], the authors show that optimisation of partitional clustering into two clusters may

be solved in the closed form. The solution of this problem is presented by the second eigen vector of the kernel matrix or Laplacian [Shawe-Taylor & Cristianini, 2004]. Negative elements of this vector correspond to the first cluster while positive elements with positive values belong to the second cluster. Applying this division we obtain hierarchical clustering with a binary tree.

K-section clustering algorithms

For K-section algorithms we may also apply partitional K-means. Here we have a choice to divide data into several subclusters at each level of hierarchy. This number should be set a priori (if we have such information) or estimated. Unfortunately, K-section clustering cannot be obtained directly from eigen vectors of the similarity matrix. Note, that K-means may produce a local optimum of clustering and should be applied carefully because it may give different results for different starting points.

Here again as in the case of the agglomerative clustering we have included clusters, but from top to bottom. Let C_u and C_v be any two different clusters in cluster set C , $C_u, C_v \in C$, where $u, v = 1, \dots, K$, $u \neq v$ and K is the number of clusters in C . Let clustering C^2 be included in C^1 : $C^2 \in C^1$. The divisive algorithm produces a hierarchical tree of included clusterings $C^K \in C^{K-1} \in \dots C^2 \in C^1$. Let t be the current level of hierarchy. Now a pseudo code for a general K-section divisive algorithm may be presented, **Algorithm 5.3**. We should say that for data coding there is an optimal hierarchical clustering presented

Algorithm 5.3 Pseudo code of K-section divisive algorithm

- 1: Initialise $t = I$ and C^t is the initial data clustering, $X_i \in C_1^t = X_i, i = 1, \dots, I$
 - 2: Divide each cluster C_r of C^t into $\{C_q\}$ clusters, where $q = 1, \dots, K$.
 - 3: $C^t = C^t - C_r$
 - 4: For $q = 1, \dots, K$ do
 - 4.1: New clustering $C^{t-1} = C^t \cup C_q$
 - 5: $t = t - 1$
 - 6: If $t > 0$ then Go to Step 2
-

in [Feder & Merhav, 1996]. In the following Section we give algorithms of partitional clustering.

5.4 Partitional clustering algorithms

K-means clustering algorithm

K-means algorithm was proposed in 1960s and may be found in numerous literature references about data clustering [Jain & Dubes, 1988; Duda et al., 2000; Webb, 2002; Mackay, 2002; Theodoridis & Koutroumbas, 2003; Friedman et al., 2001] The classical version of K-means algorithm clusters the data set X into a predetermined number of K clusters. Each cluster $k, k = 1, \dots, K$ is parameterised by its means vector μ_k :

$$\mu_k = \frac{1}{n_k} \sum_{l=1}^{n_k} X_l, \quad (5.15)$$

where n_k is the number of points in cluster k .

K-means clustering is minimising the sum-of-squares criterion:

$$\min \sum_{k=1}^K \sum_{i \in C_k} \|X_i - \mu_k\|^2, \quad (5.16)$$

We present a pseudo code of the K-means in **Algorithm 5.4**. This algorithm consists in two steps:

1. assignment step, when each sample X_i is assigned to its nearest mean vector,
2. updating step, when mean vectors μ_k are reestimated for the assigned samples.

This procedure is shown to minimise the square error Eq. (5.16).

Algorithm 5.4 Pseudo code of K-means algorithm

- 1: Initialise K and mean vectors μ_k
 - 2: Assign all points X_i to its nearest cluster $C_k = \operatorname{argmin}_k \{d(X_i, \mu_u)\}, u = 1, \dots, K$
 - 3: Update μ_k as $\mu_k = \frac{1}{n_k} \sum_{l=1}^{n_k} X_l$, where $X_l \in C_k$
 - 4: Go to **Step 2** until the assignments do not change
-

Discussions

At the beginning the algorithm randomly initialises mean vectors and at the second step assign every sample to the closest cluster (in the sense of a given distance). At the third step it updates the mean of each cluster. Repeating the second and the third steps the algorithm converges to some local optimum with a stopping criterion, so that the mean of each cluster does not change or there is no change in assignment.

At the first step we should initialise mean vectors μ_k . We can set its values if we know a priori information where clusters are located. If we do not have this information we can initialise μ_k either by random values in the range of data or by random selection of samples from data and assigning them to mean vectors. We note, that random initialisation may produce different results from one run to the other, especially in the case when we have relatively high dimension of data (e.g., higher than 10), many clusters in data (e.g., higher than 10) and not enough data samples. One of the practical suggestions to avoid problems with different clustering may be in application of Ward's hierarchical clustering of data and then using hierarchical clusters to calculate initial mean values for K-means.

We should mention K-medoid clustering algorithm which have been proposed in [Kaufman & Rousseeuw, 1990]. Its main difference from K-means algorithm consists in replacing mean vectors by samples which minimise the square error Eq. (5.16). This algorithm is effective and robust to outliers but the main drawback is its complexity $O(I^3)$. Several solutions to reduce this complexity have been revised in [Friedman et al., 2001], but they do not guarantee the optimal clustering.

Kernel K-means

In the case when data have a complex structure (e.g. data are nonlinearly separable) a direct application of K-means is inappropriate because of the tendency of K-means to

group data into globe-shaped clusters. One of the solutions is to map data by a kernel into a new feature space where samples are linearly separable. The kernel $\mathcal{K}(\cdot)$ is defined as the inner product :

$$\mathcal{K}(X_i, X_l) = \langle \phi(X_i) \phi(X_l) \rangle \quad (5.17)$$

where $\phi(\cdot)$ is a mapping of X to an inner product feature space and i, l take values $[1, \dots, I]$. The simplest kernel is called "linear":

$$\mathcal{K}(X_i, X_l) = X_i X_l, \quad (5.18)$$

and one of the frequently used kernels is the Gaussian kernel

$$\mathcal{K}(X_i, X_l) = e^{-\frac{\|X_i - X_l\|^2}{2\sigma^2}}, \quad (5.19)$$

where σ is a kernel parameter. As in the previous case Kernel K-means minimises the same optimisation function but on transformed data :

$$\min \sum_{j=1}^J \sum_{X_i \in C_k} \|\phi(X_i) - \bar{\phi}(X_i)\|^2, \quad (5.20)$$

where $\bar{\phi}(X_i) = \frac{1}{n_k} \sum_{X_i \in C_k} \phi(X_i)$ corresponds to the mean of cluster C_k with n_k number of samples. To solve 5.20 we do not operate with the explicit representation of function $\phi(\cdot)$ but we calculate the distance $\|\phi(X_i) - \bar{\phi}(X_i)\|^2$ with the inner product $\langle \phi(\cdot) \phi(\cdot) \rangle$.

The kernel distance between sample X_i and cluster C_k as in Eq. (5.21) is:

$$\|\phi(X_i) - \bar{\phi}(X_i)\|^2 = \langle \phi(X_i) \phi(X_i) \rangle - \frac{\sum_{X_l \in C_k} \langle \phi(X_i) \phi(X_l) \rangle}{n_k} + \frac{\sum_{X_j \in C_k} \sum_{X_l \in C_k} \langle \phi(X_j) \phi(X_l) \rangle}{n_k^2}, \quad (5.21)$$

which can be rewritten as :

$$\|\phi(X_i) - \bar{\phi}(X_i)\|^2 = \mathcal{K}(X_i, X_i) - \frac{\sum_{X_l \in C_k} \mathcal{K}(X_i, X_l)}{n_k} + \frac{\sum_{X_j \in C_k} \sum_{X_l \in C_k} \mathcal{K}(X_j, X_l)}{n_k^2}. \quad (5.22)$$

With the objective function Eq. (5.20) and Eq. (5.22), the standard steps of K-means algorithm are applied [Shawe-Taylor & Cristianini, 2004]. Kernel K-means algorithm is an interesting way to exploit data. When nonlinear kernel is applied this clustering may find groups of clusters which have non-linear shapes. In addition, the authors in [Shawe-Taylor & Cristianini, 2004] prove that optimal clustering may be found from eigen decomposition of the kernel matrix. They show that K clusters can be detected by K eigen vectors and values as the minimisation of the least squares distances between samples and mean vectors of corresponding clusters in the kernel space. It means that simple K-means like algorithms may be applied directly on eigen space. In addition, kernel K-mean is able to separate clusters which are not linearly separated in original space.

An algorithm of kernel K-means is presented in **Algorithm 5.4**. As can be seen Kernel K-means algorithm is equal to K-means when the linear kernel (5.18) is used.

Algorithm 5.4 Pseudo code of kernel K-means algorithm

-
- 1: Initialise data clustering X_i into K clusters
 - 2: Calculate the distance $d(X_i, \mu_k)$ Eq. (5.22) from all points X_i to all mean vectors μ_k presented by points $X_i \in C_k, k = 1, \dots, K$
 - 3: Assign all points X_i to its nearest cluster $C_k = \operatorname{argmin}_k \{d(X_i, \mu_k)\}, k = 1, \dots, K$
 - 4: Go to **Step 2** until the assignment do not change
-

Spectral K-means

In [Ng et al., 2002], the authors proposed the spectral clustering algorithm. The general idea of this approach is to use eigen vectors of the kernel matrix as a dataset on which a clustering algorithm is applied. The key point is to fix the number of eigen vectors as the number of desired clusters. The algorithm consists in the next main steps :

Algorithm 4 Pseudo code of spectral K-means algorithm

-
- 1: Compute the matrix A based on Gaussian kernel
 - 2: Construct a matrix $L = D^{-1/2}AD^{1/2}$, where D is a diagonal matrix with diagonal elements which correspond to the sum of rows of matrix A .
 - 3: Calculate matrix X as the eigen decomposition of K vectors of matrix L .
 - 4: Normalise matrix X so that each row has a unit length : $Y_{ij} = \frac{X_{ij}}{\sum_i X_{ij}}$.
 - 5: Obtain a clustering solution by applying K-means algorithm to the matrix Y for K clusters.
-

As in the case of kernel K-means the parameter σ may be taken equal to 1, if each attribute has been normalised by subtracting its mean and dividing by its standard deviation. A kernel normalisation may also be done as in [Shawe-Taylor & Cristianini, 2004]:

$$\mathcal{K}(X_{i_1}, X_{i_2}) = \frac{\mathcal{K}(X_{i_1}, X_{i_2})}{\sqrt{\mathcal{K}(X_{i_1}, X_{i_1})\mathcal{K}(X_{i_2}, X_{i_2})}} \quad (5.23)$$

Relation between spectral and kernel K-means are demonstrated in [Schölkopf et al., 1996; Dhillon et al., 2004]. Addition references for spectral clustering are in [Lau & Wade, Aug 1991; Kannan et al., 2000; Yu & Shi, 2003].

5.5 Bayesian decision theory

Bayesian decision theory has a well formulated theoretical background, clearly explained in the literature [R. Hanson & Cheeseman, May, 1991; Duda et al., 2000; Bishop, 2006] and widely used for pattern recognition tasks [Cheeseman & Stutz, 1996; Mclachlan & Peel, 2000]. It is based on the assumption that data may be described by probabilistic models. The practical problem of Bayesian decision arises when probability values are done partially or are unknown. Practically, these values should be estimated on data using

hypothesis about a model of data. Firstly, we give a small introduction about Bayesian decision theory and then we describe an algorithm to estimate the data model.

In this section we use the term "class" instead of "cluster" without loss of generality. Bayesian decision theory is a statistical approach to model and extract information from data. Let $\{C_k\} \in C$ be a finite set of K classes with a priori probabilities $P(C_k)$, where $k = 1, \dots, K$ and $P(X_i | C_k)$ is the class-conditional probability density function of random variable X whose distribution depends on the class C_k . When $P(C_k)$ and $P(X_i | C_k)$ are provided then the probability density function of sample X_i which belongs to the class C_k is

$$P(X_i, C_k) = P(X_i | C_k)P(C_k) = P(C_k | X_i)P(X_i).$$

The a posteriori probability of class C_k given sample X_i is presented via the Bayesian decision formula:

$$P(C_k | X_i) = \frac{P(X_i | C_k)P(C_k)}{P(X_i)}, \quad (5.24)$$

where

$$P(X_i) = \sum_{k=1}^K P(X_i | C_k)P(C_k), \quad (5.25)$$

or in the other word

$$\text{Posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}. \quad (5.26)$$

$P(X_i | C_k)$ is the likelihood of class C_k with respect to sample X_i and $P(X_i)$ is the evidence factor to normalise the probability to 1. The decision that sample X_i belongs to cluster C_k is done for maximal value of $P(C_k | X_i)$.

Maximum Likelihood Classification

The equation (5.24) shows how to construct an optimal classifier when all probabilities are known. But in many practical tasks of pattern recognition we have no these probabilities (likelihood, priors and evidence), instead of it we are disposing only data samples X_i . The problem is how to estimate these probabilities and use them instead of true ones. We refer to an approach based on probabilistic models for which we should estimate parameters. In the literature for the parameter model estimation we may found two main approaches [Duda et al., 2000; Mackay, 2002; Mclachlan & Peel, 2000; Bishop, 2006]:

- ★ maximum likelihood estimation
- ★ Bayesian estimation.

These two approaches result similar estimation of parameters for the large volume of data (that is the case of pattern recognition in satellite images). But the nature of estimation between them is quite different. For maximum likelihood the estimation of fixed unknown parameters maximises the probability of observed samples. While the Bayesian approach considers parameters as random variables with known a priori distribution. This distribution together with observed samples estimates a posterior density. A reader may find a brief comparison of these two approaches in [Duda et al., 2000].

We should note, that in practical cases, *e.g.*, image processing tasks, with many samples (a million and more) and high dimensions (several tens and more), Bayesian estimation is computationally consuming, that makes it difficult for application in the reasonable time. Whereas results of maximum likelihood estimation are much easier to

realise and more intuitive to interpret. In addition, as we mentioned for large data samples the estimation of parameters and classification of these two methods converge to the same results. That is why we propose to consider maximum likelihood estimation. This approach is presented in the following section and estimates a probabilistic model via Gaussian Mixture Model (GMM) and unsupervised classification.

Gaussian Mixture Model

The finite mixture model is widely used to represent data in statistical pattern recognition. Let $X = \{X_1, \dots, X_I\}$ denote the data set of samples X_i , where each X_i is a vector $X_i = (X_{i1}, \dots, X_{iJ})$ of feature values X_{ij} . The set X is modeled by a finite mixture model consisting of two parts [Cheeseman & Stutz, 1996]:

1. the prior probability $P(X_i \in k \mid \Theta_k) = \alpha_k$ that every sample X_i is a member of only one mixture component k , ($k = 1, \dots, K$), where $\alpha_k = n_k/I$, (n_k denoting the number of samples belonging to the mixture component k);
2. the conditional probability modelling each component k by the parameterised probability density function (pdf) $P_k(X_i \mid \Theta_k)$, where Θ_k denotes the parameter set.

Let $P_k(X_i \mid \Theta_k)$ denote the class-probability of observing sample X_i conditionally to X_i belonging to the component k . The finite mixture model expresses the probability of observing X_i as a sum of pdf:

$$P(X_i \mid \Theta) = \sum_{k=1}^K \alpha_k P_k(X_i \mid \Theta_k). \quad (5.27)$$

With the assumption that data instances $X_i, i = 1, \dots, I$ are independently distributed the joint data probability (probability of observing data set X or likelihood function) is the product of the individual instance probabilities:

$$P(X \mid \Theta) = \prod_{i=1}^I \sum_{k=1}^K \alpha_k P_k(X_i \mid \Theta_k) \quad (5.28)$$

An important sub-class of mixture models is the multivariate Gaussian distribution, based on a Gaussian class-distribution:

$$P_k(X_i \mid \Theta_k) = \mathcal{N}(X_i \mid \mu_k, \Sigma_k) = \frac{e^{-\frac{1}{2}((X_i - \mu_k)\Sigma_k^{-1}(X_i - \mu_k)^T)}}{(2\pi)^{D/2} |\Sigma_k|^{1/2}}, \quad (5.29)$$

where μ_k and Σ_k are the mean and the covariance matrix of the k^{th} component, respectively. Estimates of the k^{th} mean and covariance matrix are obtained in the sense of maximum likelihood estimation as [Dempster et al., 1977; McLachlan & Peel, 2000; Duda et al., 2000]:

$$\mu_k = \frac{1}{n_k} \sum_{l=1}^{n_k} X_l, \quad (5.30)$$

$$\Sigma_k = \frac{1}{n_k} \sum_{l=1}^{n_k} (X_l - \mu_k)^T (X_l - \mu_k), \quad (5.31)$$

where $X_l \subseteq k$.

Expectation-Maximisation algorithm

Evaluation of conditional probability $P_k(X_i | \Theta_k)$ and their parameters are made by Expectation-Maximisation algorithm or EM-algorithm. The algorithm maximises the log-likelihood [Dempster et al., 1977; McLachlan & Peel, 2000]. We should note that this optimisation may converge, and it is often a case for practical problems, to different local optimal solutions. And, in general, there is no guarantee of convergence to a global optimum, except in several particular cases. But some practical tricks how to improve optimal solution will be discussed later.

The equation (5.29) is used to get values $\hat{\Theta}_k$ and α_k which allow to estimate the weight w_{ik} or the conditional probability that instance X_i belongs to class k :

$$w_{ik} = \frac{\alpha_k P_k(X_i | \hat{\Theta}_k)}{\sum_{l=1}^K \alpha_l P_l(X_i | \hat{\Theta}_l)} \quad (5.32)$$

EM-algorithm [Bishop, 2006] is presented in **Algorithm 5.5**. It estimates parameters of Gaussian mixture model via unsupervised classification and maximisation of the log-likelihood function.

Algorithm 5.5 Pseudo code of EM-algorithm

- 1: Initialise K means μ_k Eq. (5.30), covariance matrices Σ_k Eq. (5.31) and α_k .
 - 2: **E-step** Calculate w_{ik}

$$w_{ik} = \frac{\alpha_k \mathcal{N}(X_i | \mu_k, \Sigma_k)}{\sum_{l=1}^K \alpha_l \mathcal{N}(X_i | \mu_l, \Sigma_l)}$$
 - 3: **M-step** Re-estimate the parameters
 - 3.1: $\mu_k = \frac{1}{n_k} \sum_{i=1}^I w_{ik} X_i$,
 - 3.2: $\Sigma_k = \frac{1}{n_k} \sum_{i=1}^I w_{ik} (X_i - \mu_k)^T (X_i - \mu_k)$,
 - 3.3: $\alpha_k = \frac{n_k}{I}$, where $n_k = \sum_{i=1}^I w_{ik}$.
 - 4: Evaluate the log-likelihood function:

$$\ln(P(X | \mu, \Sigma, \alpha)) = \sum_{i=1}^I \ln \left\{ \sum_{k=1}^K \alpha_k \mathcal{N}(X_i | \mu_k, \Sigma_k) \right\}.$$
 - 5: **If** log-likelihood converged,
 then stop,
 else go to **Step 2**.
-

Other variants of EM-algorithm as well as a survey of other probabilistic models may be found in [McLachlan & Peel, 2000; Bishop, 2006]. We mention that the Gaussian mixture model sometimes in practice are replaced by more robust t -mixtures [McLachlan & Peel, 2000]. In case of multinomial data, a mixture of multinomial distributions may be considered [Bishop, 2006].

Recent interesting developments of EM-algorithm for mixture models are given in [Govaert & Nadif, 2005, 2003]. Authors propose to cluster data simultaneously with feature clustering. This representation indicates that each cluster of data have a corresponding cluster of features.

5.6 Conclusions

In this Chapter unsupervised clustering algorithms have been revised. They are divided into two groups: hierarchical and partitional clustering. Partitional clustering algorithms are presented from simple as K-means clustering to more complex as kernel and spectral K-means. They are ended by a probabilistic clustering with the Gaussian mixture model of clusters. Parameters of GMM are estimated by Expectation-Maximisation algorithm. The clustering algorithms are compared through their complexity and optimality.

The main problem of unsupervised clustering is the estimation of clustering quality. Under this notion we understand:

1. comparing and selecting the best clustering result of an algorithm,
2. determining the number of clusters.

These problems are crucial and depend on the clustering algorithm and its measure. They will be considered in the following Chapter 6 and some new ideas will be proposed.

We have seen that the basis of these algorithms is different. In the case of large volumes of data we consider the data to have "complex" distributions and shapes of clusters. That is why it is also assumed that application of different clustering algorithms will provide us with different clustering results. The interpretation of those results is considered in Chapter 7.

Chapter 6

Model selection

Knowledge extraction from satellite images is the main purpose of this thesis. At the beginning of this extraction we aim to obtain content of data through data modelling. At the first step this modelling should provide clusters. In our case, one of the crucial problems of data clustering is that there is no prior information about how many clusters or classes an image has. A cluster or a class is considered as a type of the Earth surface. Moreover, it is quite difficult to analyse visually a large satellite image and to delimit different types of surfaces. To overcome this problem we propose to apply clustering or unsupervised classification algorithms to detect image clusters. As we have seen in previous Chapter 5 one of the parameters of these algorithms is the number of clusters. In this Chapter we give a review of several approaches and criteria to estimate the optimal number of clusters. Furthermore these criteria are able to select the best clustering from a set of clusterings. We should note, that inappropriate selection of the number of clusters and/or clustering results may lead to different and wrong data interpretations that may be the case in real practical problems.

The estimation of the clustering quality is called cluster validity. A survey of the validity may be found in [Jain & Dubes, 1988; McLachlan & Peel, 2000; Friedman et al., 2001; Theodoridis & Koutroumbas, 2003; Mackay, 2002] and divided into three groups [Jain & Dubes, 1988; Theodoridis & Koutroumbas, 2003]: external, internal and relative criteria. External criteria verify whether or how data confirm a structure which were a priori imposed. These criteria may be verified without application of clustering algorithms. Internal criteria may be based on the quantity values calculated on data and clustering results. Relative criteria evaluate clustering by comparing different clusterings obtained either from the same algorithm, but with different parameters or issued from different clustering algorithms on the same data.

One of the first clustering quality analyses is to check whether data represent some clusters or they have a random distribution and are not structured in the original space. This analysis is revised in [Jain & Dubes, 1988] and based on the Hubert's statistics. This approach is not considered in the thesis because we know a priori that our data presented by satellite images have clusters. When the nature about data is not known or proportion of noise or random samples in data may be considered as significant then this analysis should be carried out to verify whether data may be clustered or not.

In this chapter we survey internal criteria for different algorithms. Relative criterion represents an essential part of this thesis and will be revised in Chapter 7.

For hierarchical clustering algorithms we show a statistical value called Cophenetic Correlation Coefficient (CPCC). The estimation of partitional clustering algorithms will

be shown through the clustering error of data and information theoretic criteria.

The number of clusters depends on how optimally a model approximates data. In this Chapter we concentrate our attention on an information theoretic measure to estimate how well a model fits data. Under such a measure we consider Minimum description length (MDL). We show relations between MDL and other information measures such as Akaike information criterion (AIC), Bayesian information criterion (BIC) and Stochastic Information Complexity (SIC). We demonstrate also simplification of MDL criteria under the hypothesis of "hard" clustering, when data belongs only to one cluster. Further we derive a new criterion called kernel MDL (KMDL) to estimate the number of clusters for kernel clustering algorithm. Based on MDL and KMDL criteria we propose a general MDL (GMDL) criterion. In addition, several hierarchical clustering algorithms are derived from GMDL. The interest of such algorithms is that they find clusters with nonlinear shapes and in the same time they are able to estimate the quality of the clustering solution and the optimal number of clusters.

Information theoretic criteria have become very popular to select the optimal model for data fitting [Mclachlan & Peel, 2000; Mackay, 2002]. Especially it produces good results in the case when a lot of data are available, e.g., satellite image processing. These criteria are clearly formulated and have good theoretical bases. There are many works discussing equivalence of information theoretic measures AIC, BIC, MDL, which show that sometimes these measures are equivalent [Mackay, 2002; Mclachlan & Peel, 2000]. Some entropic criterion can also be found in [Biernacki et al., 1999].

We aim to find image clusters and to cluster them without prior knowledge on their type or number. Considering the amount of available data we prefer using simple, fast and efficient clustering algorithms. K-means is one of them but suffers from several drawbacks:

1. it cannot adapt to any cluster shape,
2. the knowledge of the number of clusters is necessary,
3. the result strongly depends on the initialisation process.

To answer the first problem, a classical solution is to use kernel K-means algorithm [Shawe-Taylor & Cristianini, 2004] given in Chapter 5. During the last decade kernel-based algorithms attracted lots of researchers who applied them to various tasks such as machine learning, pattern recognition, computer vision, etc. The success of these approaches is related to the fact that using a kernel (see definition and properties of kernel in [Scholkopf & Smola, 2001] [Shawe-Taylor & Cristianini, 2004]) is equivalent to define a feature space transform. This feature space depends on kernel parameters; several approaches are proposed in the literature to determine the optimal parameters [?]. The resulting feature space leads to linear separation of clusters. Therefore, classical algorithms (like K-means) can be applied.

To answer the second and third problems we propose to use a standard approach such as selection of a clustering solution obtained using different numbers of clusters and initialisations. This selection is based on the minimum of a clustering criterion. It allows also stabilising clustering results. The selection of the best solution for random initialisations have been shown to be effective [Biernacki et al., 2003].

Our proposition about using MDL criterion to determine the number of clusters is based on several arguments. Firstly, MDL is able to give the optimal code or the optimal data model [Mackay, 2002], e.g., for the Gaussian mixture model. Secondly, this criterion

works well when lots of data are available [Heas & Datcu, 2005]. This is our case because we have a huge storage of satellite images. Finally, in the literature we have not found previous works about applying MDL criteria to Kernel K-means to find the optimally associated number of clusters. This provided us with the motivation to formulate MDL criteria for Kernel K-means clustering.

In this Chapter we propose a new criterion, based on Minimum Description Length, to estimate the optimal number of clusters. The criterion, called Kernel MDL (KMDL), is particularly adapted to the use of kernel K-means clustering algorithm. Its formulation is based on the definition of MDL derived for Gaussian Mixture Model (GMM). We demonstrate the efficiency of our approach on both synthetic and real data.

This Chapter covers the following topics: a criterion to compare clustering knowing classes is presented in Section 6.1. Between- and within-cluster criteria for hierarchical and partitional clustering are given in Section 6.2. A survey of information criteria is presented in Section 6.3 We revise the main definition of MDL for GMM and we show a simplification of MDL through the complete log-likelihood of GMM in Section 6.4. Then we formulate Kernel MDL in Section 6.4 using the simplified MDL for GMM. Results on synthetic data and real satellite images are presented in Sect. 6.4 and Sect. 6.4, respectively.

6.1 Estimation of the clustering solution

The quality of a clustering solution can be measured in regards to the true classification (e.g., Rand index, the number of missclassified samples, etc. [Jain & Dubes, 1988]). Another measures may look at the class labels of the samples assigned to each cluster: entropy and purity of clustering [Zhao & Karypis, 2004]. The first measure is the widely used *entropy* measure that looks at how the various classes of samples are distributed within each cluster, and the second measure is the *purity* that measures how good a cluster fills a given class. Given a cluster C_k of size n_k , the entropy of this cluster is defined as:

$$E(C_k) = -\frac{1}{\log Q} \sum_{q=1}^Q \frac{n_k^q}{n_k} \log \frac{n_k^q}{n_k} \quad (6.1)$$

where Q is the number of classes in the dataset, n_k^q is the number of samples in the q -th class assigned to the k -th cluster. The entropy of the entire clustering solution is then defined as the sum of the individual cluster entropies weighted by the cluster size. That is :

$$Entropy = \sum_{k=1}^K \frac{n_k}{n} E(C_k) \quad (6.2)$$

A perfect clustering solution is the one which has clusters containing samples from only a single class, thus, the entropy equals zero. In general, the smaller the entropy, the better the clustering solution.

The cluster purity is the fraction of maximal n_k^q to the size n_k of cluster k . It is defined as :

$$P(C_k) = \frac{1}{n_k} \max_k (n_k^q) \quad (6.3)$$

The overall purity of the clustering is obtained as the weighted sum of the individual cluster purities and is given by :

$$Purity = \sum_{k=1}^K \frac{n_k}{n} P(C_k) \quad (6.4)$$

The larger purity, the better the clustering solution.

6.2 Between-, within- cluster criteria

In this section we show clustering validity criteria based on calculating distances for clustered data. As we mentioned above we do not consider external indexes therefore we do not test our data for randomness. We are sure that we process satellite images as data with potential clusters (a variety of surfaces). Moreover, we suppose that algorithms of feature extraction provide reliable information. For hierarchical data clustering validity criteria are based on pairwise distance matrix, while for the partitional clustering algorithm these criteria are calculated for each cluster of a given clustering.

Validity criteria for hierarchical clustering

Indices to validate hierarchical clustering show how good a hierarchical tree fits data. One of the indexes to verify this criterion is the Cophenetic Correlation Coefficient (CPCC) [Jain & Dubes, 1988]. This coefficient is based on the cophenetic matrix. The cophenetic proximity measure d_C on I samples is the level in the dendrogram of a particular hierarchical clustering at which samples X_v and X_u are first in the same cluster [Jain & Dubes, 1988]. The lower the difference between the cophenetic matrix and the matrix of similarities (or distances) the better the hierarchy fits the data set. The Cophenetic Correlation Coefficient is:

$$CPCC = \frac{1/M \sum d(v, u) d_C(v, u) - m_D m_C}{1/M \sqrt{(\sum d^2(u, v) - m_D)(\sum d_C^2(u, v) - m_C)}}, \quad (6.5)$$

where $m_D = 1/M \sum d(u, v)$, $m_C = 1/M \sum d_C(u, v)$, and $1 \leq u < v \leq I$. This coefficient takes values in the range from -1 to 1 . When fitting is not very good then the value of the coefficient tends to -1 , while a good data fitting of data by hierarchy tends to 1 . The cophenetic matrix represents an ultrametric and satisfies to its conditions [Jain & Dubes, 1988]. The matrix depends on the hierarchical methods. Applying any hierarchical algorithm to the ultrametric matrix produces the same clustering whatever the algorithm.

Validity criteria for partitional clustering

Some of earlier criteria for cluster validity may be found in [Jain & Dubes, 1988]. A very popular criterion is based on calculating between- and within-cluster distances. Partitional clustering is partition data into groups (clusters). A good partitional clustering is such that it reduces the distance among points in the same cluster and at the same time it increases distances among different clusters. Based on this idea a set of criteria have been calculated [Coleman & Andrews, 1979]. Within cluster distance is based on the distances

among points in the cluster. Such distance can be calculated using a square Euclidean distance and represented as the covariance or scatter matrix.

$$S_w = \frac{1}{K} \sum_{k=1}^K \Sigma_k, \quad (6.6)$$

where Σ_k is the covariance matrix of cluster k as in Eq. (5.31). Between-cluster distance is calculated between clusters and usually using the Euclidean distance. This distance may be considered as the covariance or scatter matrix of cluster means.

$$S_b = \frac{1}{K} \sum_{k=1}^K (\mu_k - \mu_0)^T (\mu_k - \mu_0), \quad (6.7)$$

where μ_0 is the mean of data and is written as

$$\mu_0 = \frac{1}{I} \sum_{i=1}^I X_i, \quad (6.8)$$

Several criteria have been derived from these within-, between-cluster distances which are based on cluster separability [Coleman & Andrews, 1979]:

$$\beta_1 = \text{tr}(S_w^{-1} S_b), \quad (6.9)$$

$$\beta_2 = \ln(|S_w + S_b| / |S_w|), \quad (6.10)$$

$$\beta_3 = \text{tr} S_b / \text{tr} S_w, \quad (6.11)$$

$$\beta_4 = \text{tr} S_b \cdot \text{tr} S_w, \quad (6.12)$$

where $\text{tr}(\cdot)$ denotes matrix trace (sum of the diagonal elements of a matrix), and $|\cdot|$ the determinant of the matrix. We should note that β_1 and β_2 are invariant under any non-singular linear transformation, while β_3 depends on the coordinate system. To determine the number of clusters which gives the best separability of data one should cluster data for different numbers of clusters for one of the parameters β . The maximal value indicates the optimal number of clusters. It is easy to verify, that when we have one (i) cluster in data and (ii) as many clusters as the number of samples, parameter β_4 becomes equal to 0. The maximum of β_4 lies between these two limit cases and will indicate the appropriate number of clusters. It has been shown in [Coleman & Andrews, 1979] that when β_3 achieves value 1 then β_4 achieves its maximum value. This relation may be useful to avoid unnecessary calculations for further number of clusters.

Maximum or minimum of these criteria shows the optimal number of clusters. Sometimes, when data have a complex structure, the curve of these criteria may exhibit several local minima or maxima. It is the case when data can be clustered in different numbers of clusters dependently on the scale under which data are seen. For this example, several well separated clusters can be considered each of them containing small well grouped subclusters. As a real example for satellite image processing, large clusters as city and forest can be well separated, but each of them will contain subclusters as suburb, downtown, etc., for city and different kind of forest for forest class.

6.3 Information measure

The above given criteria for partitional clustering were proposed in the earlier stage of data mining. Nowadays, there are criteria which are based on information theory and are often more effective. Under a set of such criteria we consider *information-theoretical measures*. We present a set of the measures applied to a probabilistic model. In addition we derive a new criterion based on a simplification of the probabilistic model of clustering.

Bayesian information criterion

Usually data are estimated by two stage inference procedure. At the first stage assuming that data obey a given model, we estimate parameters of the model. Such estimation is done for each model. At the second stage using the found parameters we compare the models and select the best of them. This two stage procedure is argued by the fact that the more complex the model the better it fits data. We should find a tradeoff between model fitting and model complexity. The first stage of modelling has been considered in the previous Chapter 5 via Maximum likelihood estimation of the Gaussian Mixture Model. Here we pay attention to the second stage of the model selection.

Model selection can be done with respect to a theoretical approach based on the Bayes' theorem [Mackay, 2002]. Having two models \mathcal{M}_1 and \mathcal{M}_2 we aim to select the one which best fits data \mathcal{D} . Using prior probability $P(\mathcal{M}_1)$ of the model \mathcal{M}_1 and the probability $P(\mathcal{M}_1 | \mathcal{D})$ of model \mathcal{M}_1 given data \mathcal{D} (and the same probabilities for \mathcal{M}_2). We apply Bayesian formula and compute the ratio:

$$\frac{P(\mathcal{M}_1 | \mathcal{D})}{P(\mathcal{M}_2 | \mathcal{D})} = \frac{P(\mathcal{M}_1) P(\mathcal{D} | \mathcal{M}_1)}{P(\mathcal{M}_2) P(\mathcal{D} | \mathcal{M}_2)}, \quad (6.13)$$

where the ratio $\frac{P(\mathcal{M}_1)}{P(\mathcal{M}_2)}$ express the prior preference of model \mathcal{M}_1 with respect to model \mathcal{M}_2 . Therefore, when two models \mathcal{M}_1 and \mathcal{M}_2 have the same prediction power we prefer simpler model to complex ones. If the ratio in Eq. (6.13) is greater than 1 we select model \mathcal{M}_1 as the best one. Model estimation (parameter estimation Θ) can be done through the Bayesian theorem Eq. (5.26). The posterior probability of Θ is given by:

$$P(\Theta | \mathcal{D}, \mathcal{M}_i) = \frac{P(\mathcal{D} | \Theta, \mathcal{M}_i) P(\Theta | \mathcal{M}_i)}{P(\mathcal{D} | \Theta)} \quad (6.14)$$

The posterior probability of each model is:

$$P(\mathcal{M}_i | \mathcal{D}) \propto P(\mathcal{D} | \mathcal{M}_i) P(\mathcal{M}_i), \quad (6.15)$$

where the normalised constant $P(\mathcal{D}) = \sum_i P(\mathcal{D} | \mathcal{M}_i) P(\mathcal{M}_i)$ has been omitted. Here the main difficulty in applying this formula is to calculate the evidence $P(\mathcal{D} | \mathcal{M}_i)$. This can be done by parametric or non-parametric model estimation for classification or clustering. Then the evidence is the normalisation in Eq. (5.26)

$$P(\mathcal{D} | \mathcal{M}_i) = \int P(\mathcal{D} | \Theta, \mathcal{M}_i) P(\Theta | \mathcal{M}_i) d\Theta. \quad (6.16)$$

Usually, $P(\Theta | \mathcal{D}, \mathcal{M}_i) = P(\mathcal{D} | \Theta, \mathcal{M}_i) P(\Theta | \mathcal{M}_i)$ has a peak at the most probable parameter $\hat{\Theta}$. In this way the evidence may be approximated, using Laplacian method [Ripley,

1996], by the height of the peak. When the posterior is approximated by the Gaussian, the Occam factor is obtained from the determinant of the corresponding covariance matrix:

$$\begin{aligned} P(\mathcal{D} \mid \mathcal{M}_i) &\simeq P(\mathcal{D} \mid \hat{\Theta}, \mathcal{M}_i) \times P(\hat{\Theta} \mid \mathcal{M}_i) \det^{(-1/2)}(A/2\pi), \\ \text{Evidence} &\simeq \text{Likelihood} \times \text{Occam factor} \end{aligned} \quad (6.17)$$

where $A = -\nabla\nabla \ln P(\hat{\Theta} \mid \mathcal{D}, \mathcal{M}_i)$ is the Hessian evaluated for the optimal parameter $\hat{\Theta}$. As we may see from the Bayesian model selection, the evidence is obtained by multiplying the best fit likelihood by Occam factor. Occam factor may be estimated for I samples fitted by a parametric model with J degrees of freedom. With some calculations we obtain:

$$\log P(\hat{\Theta} \mid \mathcal{M}) - \frac{J}{2} \log(I/2\pi) - \log \det^{(-1/2)}(\mathcal{I}), \quad (6.18)$$

where $\det(\mathcal{I})$ is the determinant of \mathcal{I} the Fisher matrix evaluated at $\hat{\Theta}$ with elements:

$$\mathcal{I}_{u,v} = E \frac{\partial \log P(X \mid \Theta)}{\partial \Theta_u \partial \Theta_v}. \quad (6.19)$$

To select the order of the model [Schwarz, 1978] used approximation of Eq. (6.18) without a determinant of the Fisher matrix. Using this approximation the logarithm of $P(\mathcal{D} \mid \mathcal{M}_i)$ Eq. (6.17) leads to the *Bayesian information criterion* (BIC):

$$BIC = -\log P(\mathcal{D} \mid \hat{\Theta}) + \frac{J}{2} \log(I). \quad (6.20)$$

The minimisation of this criterion shows the optimal order of the model.

Akaike information criterion

Akaike has proposed the Akaike Information Criterion (AIC) which is similar to BIC, but was derived on a different theoretical basis.

$$AIC = -2 \log P(\mathcal{D} \mid \hat{\Theta}) + 2J \log(I). \quad (6.21)$$

BIC often proposes to select simpler models because it has a larger penalty term than AIC [Friedman et al., 2001]. AIC tends to overfit a model, *i.e.*, means to select more complex models. That is why AIC can overestimate the number of mixture components [Mclachlan & Peel, 2000]. The survey of different information criteria and classification criteria to select the order of a model as well as their empirical comparison is given in [Mclachlan & Peel, 2000].

Minimum description length criterion

Stochastic complexity

As we are working with a finite set of discrete data modelled by density functions we can consider their negative logarithm. This logarithm is an integer code. Then for such models \mathcal{M} the code length $L(X \mid \mathcal{M})$ may be written as [Rissanen, 1995]:

$$L(X \mid \mathcal{M}) = -\log P(X \mid \hat{\Theta}) + \frac{J}{2} \log \frac{I}{2\pi} + \log \int \sqrt{|\mathcal{I}(\Theta)|} d\Theta + o(1). \quad (6.22)$$

This code length is also called Stochastic Information Complexity (SIC) of the model. The optimisation of this criterion leads to the selection of the best model fitting data.

2-parts description length

Model selection may also be viewed through the code length of the model. The model provides a probability of data fitting. Using Shannon entropy we may calculate how much information the model contains. The measure of this information is expressed in bits. It says how many bits in average we should take to code our model. The more bits are used to represent information the more complex information and, consequently, the more complex the model is. There exists a universal coding proposed in [Rissanen, 1984] which consists in two parts:

Part 1 Model description length $-L(\mathcal{M})$ describes the model \mathcal{M} and its parameters $\hat{\Theta}$.

Part 2 Data description length $-L(X | \mathcal{M})$ describes data X knowing the model \mathcal{M} and its parameters $\hat{\Theta}$.

Then the universal coding is the sum of two parts:

$$L_{2P} = -L(\mathcal{M}) - L(X | \mathcal{M}). \quad (6.23)$$

In the literature the first term is usually represented as:

$$-L(\mathcal{M}) = \frac{J}{2} \log(I) + o(1), \quad (6.24)$$

while the second term is:

$$-L(X | \mathcal{M}) = -\log P(X | \hat{\Theta}, \mathcal{M}). \quad (6.25)$$

Then the two part universal coding becomes:

$$L_{2P} = -\log P(X | \hat{\Theta}, \mathcal{M}) + \frac{J}{2} \log(I) + o(1). \quad (6.26)$$

From equation (6.26) we see that universal coding L_{2P} is the same as BIC Eq. (6.20). It has been also shown that the two part description code L_{2P} Eq. (6.26) is the approximation of the SIC Eq. (6.22).

6.4 MDL for the Gaussian Mixture Model

In this section we consider MDL to determine the optimal number of clusters. Clustering is obtained by Gaussian mixture model and EM-algorithm which estimates parameters of GMM. In addition, we write MDL criterion for hard clustering when each sample belongs only to one cluster. For this we introduce a complete-likelihood of GMM taking an additional variable which indicates the hard clustering. This consideration leads to some interesting simplifications of MDL criterion. The simplified MDL may be extended and applied to other algorithms. In addition, new hierarchical algorithms may be derived.

With the assumption that the data instances X_i are independently distributed, the joint data probability (probability of observing data set X or likelihood function) is the product of the individual instance probabilities:

$$P(X | \Theta) = \prod_{i=1}^I \sum_{k=1}^K \alpha_j P_k(X_i | \Theta_k). \quad (6.27)$$

The Expectation-Maximisation (EM) algorithm [Mclachlan & Peel, 2000; Mackay, 2002] can be used to estimate the optimal parameters Θ_k of GMM. Without loss of generality we say that the k^{th} component of GMM models the k^{th} cluster.

The purpose of clustering data is to simplify their representation in the feature space by replacing each sample by a generic class which is likely to express all the properties of the samples. However, when substituting a sample by its model, an error is introduced. The more complex the model, the less the error. The "model complexity" is well expressed by the number of parameters needed to build the model. In the mixture of Gaussians case where every cluster is given by its mean (5.30) and its covariance matrix (5.31), the more clusters are used, the more complex the model, and the less the error between data and model. A method to choose the optimal number of clusters consists in selecting the number that most efficiently codes the data, i.e. which provides the shortest description of the models. This method, called Minimum Description Length (MDL), has been proposed by Rissanen [Rissanen, 1978, 1984; Barron et al., 1998]. MDL is defined as [Rissanen, 1984]:

$$\min_{\mathbb{k}, \Theta} -\log(P(X|\Theta)) + \frac{1}{2}\mathbb{k}\log(I), \quad (6.28)$$

where $\log(P(X | \Theta))$ is the log-likelihood of the mixture model (6.27) and $\frac{1}{2}\mathbb{k}\log(I)$ is a penalty function with \mathbb{k} parameters.

MDL for the Complete Log-likelihood of GMM

Let see the log-likelihood for the mixture of Gaussian distributions in details. To complete the likelihood $P(X|\Theta)$ Eq. (6.27) of the finite mixture expressed by Eq. (5.27), we should introduce the hidden variable z which attributes any sample to a class: $z = \{z_1, \dots, z_i, \dots, z_I\}$ [Figueiredo, 2002] [Govaert, 2003]. Label z_i is coded as a binary vector $z_i = [z_{i1}, \dots, z_{ik}, \dots, z_{iK}]$, where $z_{ik} = 1$ if sample i belongs to cluster k , or 0 if not. Using Eq. (6.27), the complete log-likelihood $\log(P(X, z|\Theta))$ becomes [Figueiredo, 2002; Govaert, 2003]:

$$\begin{aligned} \log(P(X, z | \Theta)) &= \log \left(\prod_{i=1}^I \sum_{k=1}^K z_{ik} \alpha_k P_k(X_i | \Theta_k) \right) = \\ &= \sum_{i=1}^I \sum_{k=1}^K z_{ik} \log(\alpha_k P_k(X_i | \Theta_k)). \end{aligned} \quad (6.29)$$

We derive in more details simplification of the complete log-likelihood $\log(P(X, z|\Theta))$ Eq. 6.29 in Appendix C.

In the right part of the MDL definition Eq. (6.28), \mathbb{k} is the model free parameters number. In case of Gaussian mixture model free parameters are:

- ★ $K - 1$ parameters for K weights α_k (since $\sum \alpha_k = 1$);
- ★ J parameters for each mean μ_k ;
- ★ $J(J + 1)/2$ parameters for each covariance matrix Σ_k .

Therefore, the number of free parameters is:

$$\mathbb{k} = K - 1 + K(J + J(J + 1)/2) = K(J^2 + 3J + 2)/2 - 1. \quad (6.30)$$

Using the complete log-likelihood Eq. (C.5) and the free parameter number of Eq. (6.30), the description length Eq. (6.28) of Gaussian mixture model with K clusters is:

$$-\frac{1}{2} \sum_{k=1}^K n_k \log \left(\frac{\alpha_k^2}{|\Sigma_k|} \right) + (K(J^2 + 3J + 2)/2 - 1) \log(I)/2 + \text{const}. \quad (6.31)$$

The *const* term having no influence on MDL for different cluster numbers and as $\alpha_k = n_k/I$, we may minimise:

$$\Lambda = - \sum_{k=1}^K n_k \log \left(\frac{n_k^2}{|\Sigma_k|} \right) + K(J^2 + 3J + 2) \log(I)/2. \quad (6.32)$$

Equation Eq. (6.32) shows that a quality of clustering only depends on the weighted determinants of the covariance matrices which express the square errors between data and model. Estimating the covariance matrices Σ_k and the populations of each cluster n_k , we can draw the MDL curve Λ as a function of the cluster number K . The minimum on this curve indicates the optimal description of the data set X , i.e. the minimum error with the minimum model complexity.

The MDL criterion Eq. (6.32) may be applied to any clustering method: to EM, which, as said before, provides the best clustering, given a number of clusters, or to simpler algorithms - like K-means which may be seen as a simplified version of EM [Mackay, 2002], or Kernel K-means, which is an extension of K-means. Based on this remark, we propose to define an MDL optimisation of Kernel K-means in Section 6.4.

Graph of MDL to determine the number of clusters

We demonstrate our approach on synthetic data before applying it for real data such as satellite images. The simplest and often used example of synthetic data is Gaussian distributions where each distribution is a cluster. In the literature on data mining the number of distributions is usually around ten. But in our case when we work with satellite images we may have several tens or even hundreds of clusters. That is why we demonstrate experiments with 24 and 49 Gaussians to determine the number of clusters by MDL criteria. We use the MDL criterion Eq.(6.32) for our experiments. The examples of 24 and 49 clusters are presented in Figure (6.1b) and Figure (6.1d) respectively. The number of points for each Gaussian is 100.

We run *K-means* algorithm for a fixed number of clusters (starting from 2 clusters) with the random selection of cluster centres. Initialising 5 times clustering centres we select the best clustering with the minimal MDL criteria. Then we change the number of clusters and repeat clustering. The curve of MDL for 24 clusters is shown in Figure (6.1a). We see that this curve has a well defined global minimum, which shows the optimal number of clusters. The same experiments are done for 49 Gaussians Figure (6.1d). The corresponding MDL curve is shown in Figure (6.1c). As in the previous case, it has a global minimum that indicates the true number of clusters.

For this experiment MDL has a one well defined global optimum, Figures 6.1a and 6.1c. Thus, we can apply an optimisation method (e.g., a dichotomy search) to determine the optimal number of clusters on MDL curve instead of computing MDL values for each of the number of clusters. It can therefore reduce computational time and machine resources.

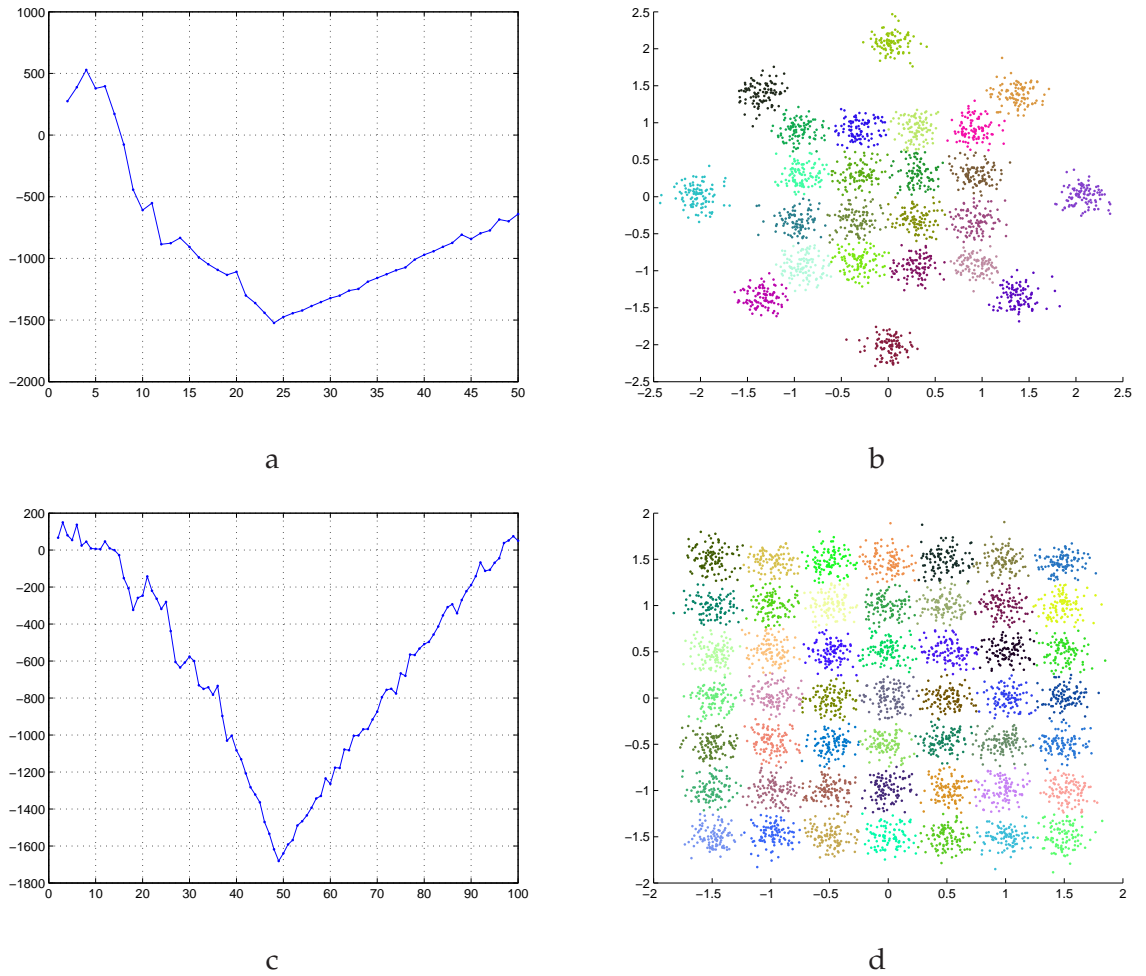


Figure 6.1: MDL curve to determine the optimal number of clusters. a - MDL for 24 Gaussian clusters, b - the optimal clustering of 24 Gaussians, c - MDL for 49 Gaussian clusters, d - the optimal clustering of 49 Gaussians,

During our experiments with MDL we experimentally observe that it is better to start from a high value of the cluster number and analyse MDL by decreasing this value. In this way MDL curve is not so noisy and we can obtain the optimal value very quickly (several steps).

The optimal number of clusters and features

Here a simple example to determine simultaneously the optimal number of clusters and the optimal number of features on synthetical data is considered. As we discussed at the beginning of Section 6.4 MDL criterion estimates the optimal model parameters for data clustering. These parameters include the number of clusters and the number of data dimensions. Therefore we can estimate both these parameters to obtain the optimal data clustering in the sense of MDL.

The estimation is based on the simplified MDL criterion Eq.(6.32). We generate 9 Gaussian clusters in 3 dimension space. Clusters have random sizes: 51, 93, 135, 165, 86, 48, 34, 65, 29. Data are presented in Figure 6.2 a. We concatenate the same three dimensions of data to have correlations. In addition, we concatenate three dimensions of data perturbed by Gaussian noise. At the end we concatenate two dimensions which have Gaussian noise. In the total we have 11 correlated and noisy features of data.

The experiment has following steps: K-means algorithm is run to cluster data of 11 dimensions. For the number of clusters from 2 to 15 the algorithm is run three times with random initialisations, the best clustering for 3 initialisations is selected by MDL criterion. Then the last dimension is deleted from data and clustering is repeated. The best clustering results is selected by MDL criterion. Figure 6.2 b represents MDL criterion for different number of clusters and different number of features. The optimal number of clusters equals 9 and the optimal number of features equals 3, which corresponds to the minimum of MDL criteria. The third dimension of features for MDL criteria shows that the optimal number of clusters as 9, see Figure 6.2 c. The ninth dimension of clusters for MDL criteria indicates the optimal number of features as 3, see Figure 6.2 d. A critical point in this experiment is that features are arranged and removed one by one estimating MDL criterion. In practice, we do not know very often which feature should be removed first. The simplest solution may be in estimating MDL criterion foreach removed feature to decide which of them should be removed first (if it is necessary). Another technique of features selection can be considered via wrapping or filtering methods [Campedel et al., 2005].

The idea of this example was to show that MDL criteria can be used to select the best data features and the best number of clusters in data.

Kernel MDL

In this Section we propose to derive kernel MDL criterion, from the formulation of simplified MDL Eq.(6.32). From Eq.(6.32) it can be seen that the simplified MDL depends on determinants of matrices $|\Sigma_k|$, which describe the model of data error. This error may be calculated in the original space X , as well as in the transformed space using a kernel. Therefore, we propose to define a general MDL, similar to Eq.(6.32), as:

$$-\sum_{k=1}^K n_k \log \left(\frac{n_k^2}{\text{Dist}(X_u, X_v | u, v \subseteq k)} \right) + P(K, J, I) \quad (6.33)$$

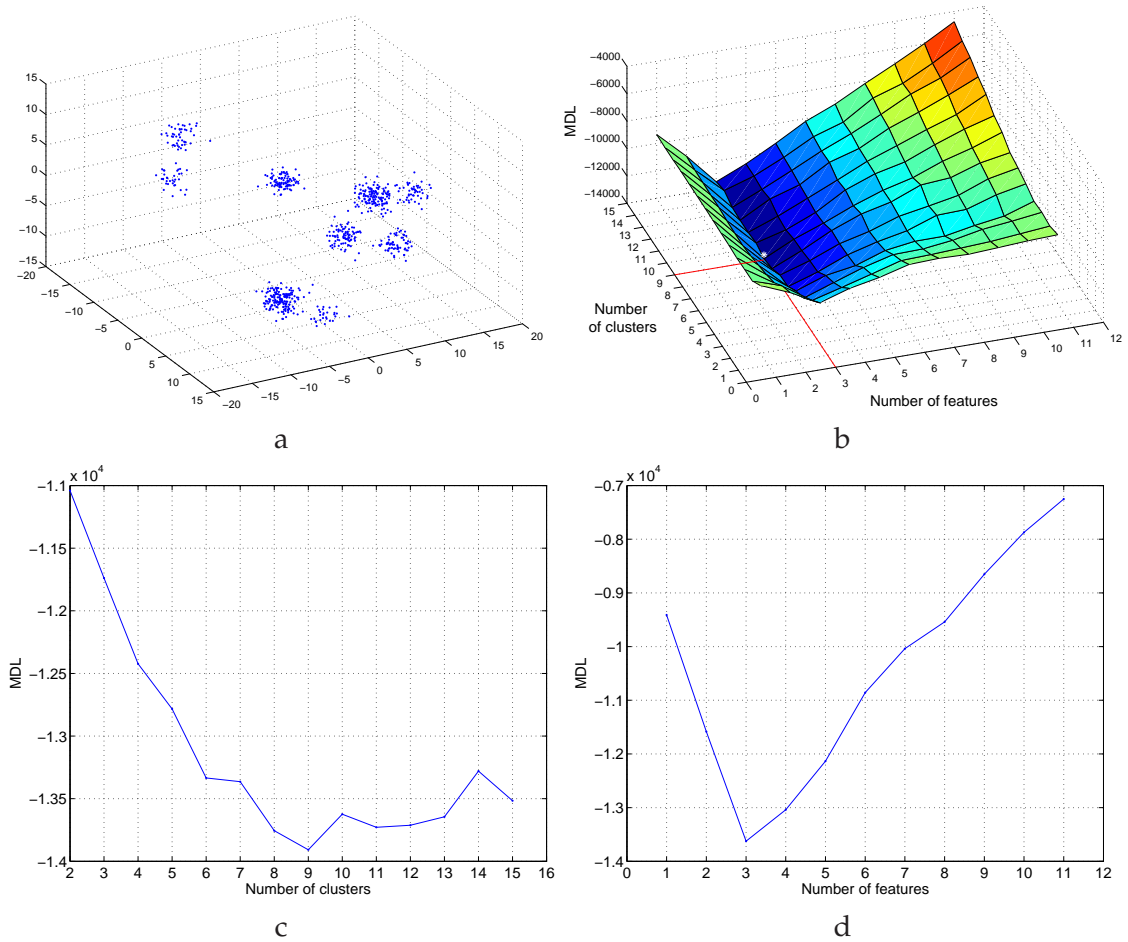


Figure 6.2: MDL criteria to select the optimal number of clusters and features for correlated and noisy data: a - a toy example of 9 Gaussian clusters. Three first dimensions are displayed, dimensions 4 – 6 are the same data, dimensions 7 – 9 are noisy original data and dimensions 10 – 11 have Gaussian noise. b - MDL criteria for different number of clusters and different number of features (data clustered by K-means algorithm). c - the third dimension of features for MDL criteria (the optimal number of clusters is 9), d - the ninth dimension of clusters for MDL criteria (the optimal number of features is 3).

where $Dist(X_u, X_v | u, v \subseteq k)$ is the error function for samples X_u and X_v being represented by the k^{th} cluster (for instance, the distance between X_u and the mean of cluster k) and $P(K, J, I)$ is a penalty function.

The simplest error function is the Euclidean distance which may be calculated using the kernel \mathcal{K} (5.17). The sum-squares distances from patterns to their corresponding k^{th} cluster centroid has been presented in [Shawe-Taylor & Cristianini, 2004] as the optimisation function for Kernel K-means:

$$S_k = \frac{1}{n_k J} \sum_{u \subseteq k} \left(\mathcal{K}(X_u, X_u) - \frac{1}{n_k} \sum_{v \subseteq k} \mathcal{K}(X_u, X_v) \right). \quad (6.34)$$

In the case when \mathcal{K} is a linear kernel, S equals the variance in the original space X as expressed by (5.18). Therefore, assuming that the variances of each cluster are equal for each dimension, we may rewrite the determinant of covariance matrix Σ_k as:

$$|\Sigma_k| = S_k^J. \quad (6.35)$$

As the error S_k (6.34) may be derived for any kernel, *e.g.* Gaussian (5.19), we may substitute the determinant (6.35) in the MDL expression (6.32) to obtain the kernel MDL:

$$\text{KMDL} = - \sum_{k=1}^K n_k \log \left(\frac{n_k^2}{S_k^J} \right) + K(J^2 + 3J + 2) \log(I)/2. \quad (6.36)$$

For the following experiments the same penalty function as in (6.32) have been used. The derivation of an alternative penalty is not addressed in this paper. One of the main advantages of this formulation is that the explicit mean of a cluster k is not needed. This point is important when this mean has no physical meaning, as it is often the case for non-convex clusters. To calculate MDL criterion for the mixture of Gaussians in the original space X the distance between samples and the nearest cluster centroid must be calculated. Problems may appear in case when data are distributed on clusters with holes as in Figure 6.3-d.

Experiments with synthetic data

In this section we test kernel MDL on synthetic data before applying it to real data such as satellite images. The simplest and often used example of synthetic data consists in using Gaussian distributions where each distribution is a cluster. When working on satellite images, we expect to have a large number of clusters because of the great variety of possible scenes. Therefore, we demonstrate the potential of the method with a rather large number of clusters, larger than in the literature [Jain & Dubes, 1988].

We make use of 20 Gaussian distributions with 100 samples per cluster as presented in Figure 6.3-a. EM algorithm run 20 times for each cluster number, with a different random initialisation. Two curves are presented in Figure 6.3-b, showing the results of clustering using either MDL (6.32) or KMDL (6.36) with Gaussian kernel and parameter $\sigma = 2$. For all curves of KMDL a constant is added to better visualise with MDL. As expected, both curves exhibit a well defined minimum, with an optimal number of clusters equals to 20.

The same experiments were done for another toy example having clusters with a complex structure. Points of this cluster are distributed on a circle. Here again, EM-algorithm and Kernel K-means with Gaussian kernel ($\sigma = 0.5$) have been used. We should say that

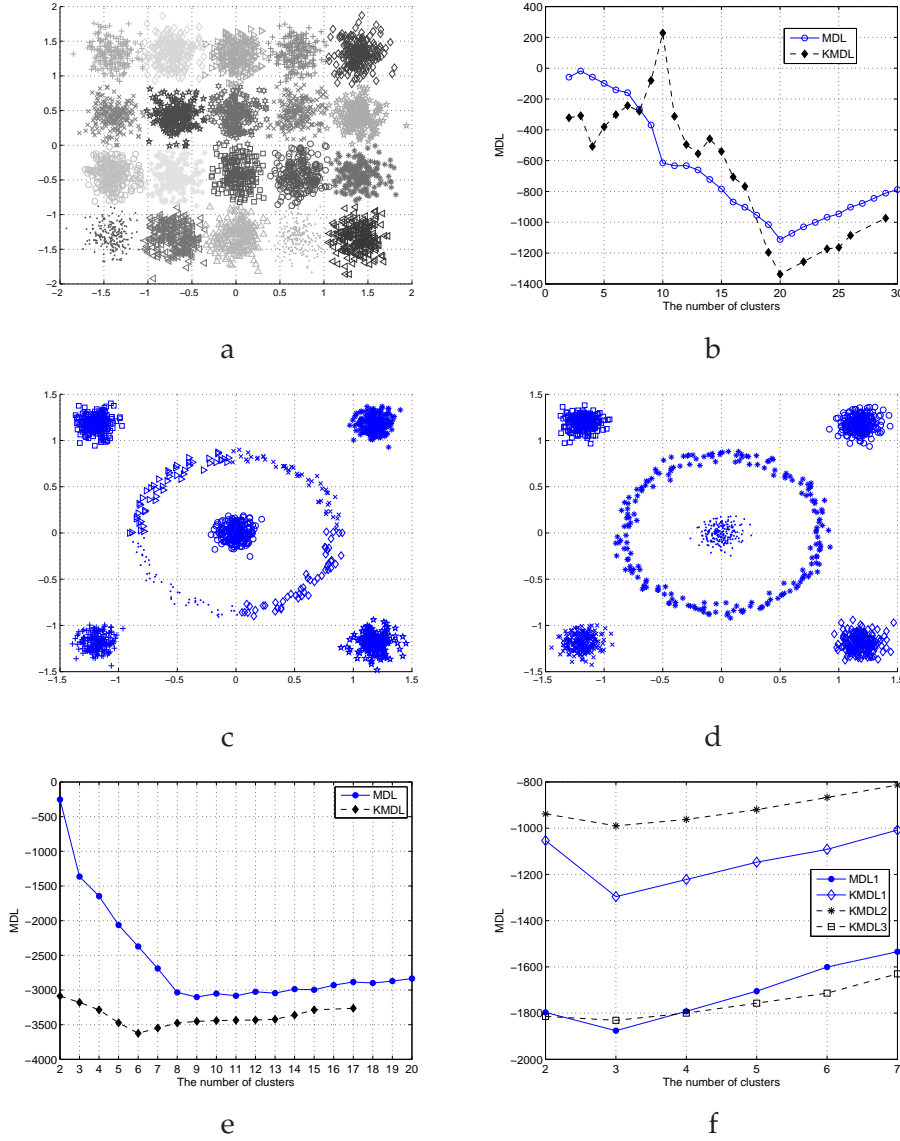


Figure 6.3: Synthetic examples. In a: synthetic example 1 with 20 clusters. In b: results on clustering example 1. Detection of the optimal number of clusters by MDL (6.32) (solid line) and by KMDL (6.36) (dashed line). In c: example 2 with a circular cluster as clustered by EM. In d: the same as clustered by Kernel K-means. In e: curves drawn for example 2. In f: Optimal number of clusters for Thyroid and Iris data. MDL (6.32) (solid line with points) and KMDL1 (6.36) with $\sigma = 5$ (solid line with diamonds) propose 3 as an optimal number of clusters for Thyroid data set. KMDL2 (6.36) with (5.18) (dashed line with stars) and KMDL3 (6.36) with (5.19) $\sigma = 4$ (dashed line with squares) propose 3 as an optimal number of clusters for Iris data set.

the choice of the optimal value of σ is not considered in this thesis. Optimal results are presented in Figure 6.3-c and Figure 6.3-d. From Figure 6.3-e, it may be observed that EM with MDL detects more clusters than expected because of the difficulty to linearly separate a cluster with a complex structure (also seen in Figure 6.3-c where the circle is split into 4 clusters). On the contrary Kernel K-means with the Gaussian kernel optimally separates the mixture in Figure 6.3-d, and KMDL determines the true number of clusters.

The last experiment concerns two real world data sets Iris and Thyroid taken from the UCI machine learning repository. Iris data contain 3 classes, 50 samples per class and 4 features per sample. The minimum of KMDL (6.36) with the linear kernel (5.18) and the Gaussian kernel (5.19) determines the true number of clusters equals 3 Figure 6.3-f. Thyroid data have 3 classes: 150, 35 and 30 samples per class, respectively, and 5 features per sample. Both criteria KMDL (6.36) with the Gaussian kernel (5.19) and MDL (6.32) determine the true number of clusters as 3 Figure 6.3-f.

From this set of experiments, several practical rules have been observed. At first, as in the previous Section, it seems that it is better to start from high values of cluster number to progressively reduce it in order to have a less chaotic behaviour of the curve. Then we observe that the MDL is often unequivocal, allowing to use speeding search techniques like dichotomy for instance.

Experiments with real data: satellite images

The experiment

In the framework of the CNES-DLR Competence Centre we are interested in information extraction and image understanding for Earth observation with high resolution images ¹. In order to reduce the amount of information carried by an image, we propose to categorise satellite images. To avoid bias and omissions due to human expertise, we investigate unsupervised image category extraction. In this scope we consider each cluster as a category. The optimal number of clusters obtained from a given set of images is therefore an important clue which cannot be arbitrarily fixed. The previous approach (based on simplified MDL (6.32) and KMDL (6.36)) will be our guideline to determine this number.

We are working with images from the SPOT 5 satellite, they are panchromatic images with a ground resolution of 5m per pixel. Each original image is very large (12000×12000 pixels) and quite complex; therefore we extract smaller images (1024×1024 pixels) with rather homogeneous content on urban areas. These (1024×1024) images will, from now on, be named "the images" since the original large images will no longer be used in the rest of this document. The images represent 6 cities: Copenhagen (Denmark), Istanbul (Turkey), Los Angeles (USA), La Paz (Mexico), Madrid (Spain), Paris (France). We assume that, because of geography, culture and history each image has different surface textures. Sub-samples of images are presented in Figure 6.4. From these images, we form a database of samples by cutting each image into 400 samples, each of size 64×64 pixels. Samples overlap by 13 pixels. The composed database contained 2400 samples, 6 cities and 400 samples per city. From each sample, 202 features have been extracted: statistics issued from Quadratic Mirror Filters filtering, statistics from Gabor filters, statistics from Haralick co-occurrence matrix descriptors and geometrical features. 15 features

¹<http://www.coc.enst.fr/>

were automatically selected from the initial features using unsupervised feature extraction [Campedel et al., 2005].

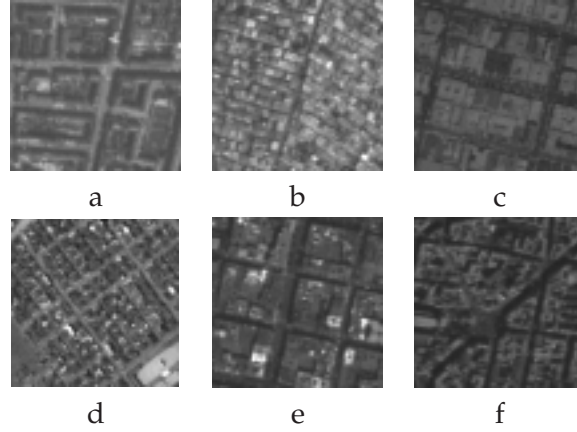


Figure 6.4: Samples of SPOT5 images (64×64 pixels per sample) : a - Copenhagen (Denmark), b - Istanbul (Turkey), c - Los Angeles (USA), d - La Paz (Mexique), e - Madrid (Spain), f - Paris (France). ©Copyright CNES

The data matrix of size 2400×15 is clustered with two algorithms: EM-algorithm [Mclachlan & Peel, 2000; Mackay, 2002] with GMM and Kernel K-means with the Gaussian kernel Eq.(5.19) [Shawe-Taylor & Cristianini, 2004] and parameter $\sigma = 15$ which equals the dimension of normalised data. We compare two clustering algorithms in order to compare their clustering criteria. 50 random initialisations were performed and the best clustering was chosen. In our experiments the data were normalised in such a way that their mean equals 0 and the standard deviation of each column is 1, so that the weight of each feature is the same:

$$\mu_j = \frac{1}{I} \sum_{i=1}^I X_{ij}, \quad (6.37)$$

$$\sigma_j = \sqrt{\frac{1}{I} \sum_{i=1}^I (X_{ij} - \mu_j)^2}, \quad (6.38)$$

$$\tilde{X}_{ij} = \frac{X_{ij} - \mu_j}{\sigma_j} \quad (6.39)$$

Setting in Eq.(5.19) σ as the data dimension ($\sigma = J$), we obtain the curves shown in Figure 6.5 for MDL and for KMDL Eq.(6.36). For EM-algorithm the optimal number of clusters is 9 whereas for Kernel K-means it is 11. We may present these optimal clusterings as distribution matrices (as in Tables 6.1 and 6.2, respectively), where each column corresponds to a city in the same order as in Figure 6.4, and each line represents a cluster.

Discussion

In the ideal case, where all the cities would be perfectly different, we could consider that the clustering is good if each cluster consists of one city only. From the classification

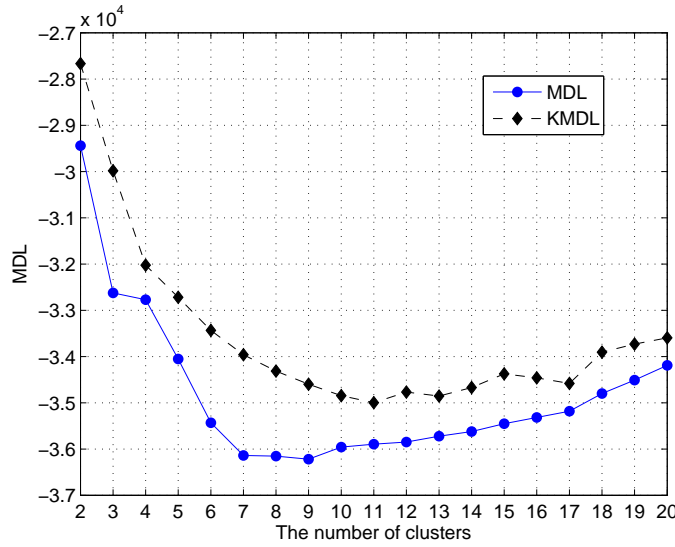


Figure 6.5: Detection of the optimal number of clusters by MDL (solid line) and KMDL (dashed line) criteria for SPOT 5 image textures.

matrices Tables 6.1 and 6.2 we can see that the EM-algorithm and Kernel K-means give almost the same clusters. But EM-algorithm finds cluster 4 as a mixture of two cities (Los Angeles and Paris), although these cities exhibit rather different structures Figure 6.4. The classification matrix of Kernel K-means (Table 6.2) shows that these two cities are separated (clusters 3 and 8). Even if we set the number of clusters to 12 for the EM-algorithm the confusion between these cities remains. This confusion disappears when the number of clusters is 15, but it will not be an optimal clustering in terms of MDL. We consider that Kernel K-means better clusters data than EM-algorithm because clusters better correspond to cities. Some texture examples of clustered cities (4 textures per cluster) by Kernel K-means are presented in Tables 6.3 and 6.4. The samples closest from the centre of the corresponding clusters have been chosen. Each row of Table 6.3 has 4 texture examples for clusters from 1 to 6 and Table 6.4 for clusters from 7 to 11. We analyse visually these examples using classification matrix in Table 6.2. The first and sixth rows of Table 6.3 correspond to 4 textures of La Paz. These clusters show two different surfaces for this city. The second row has samples from every city and corresponds to large places which are likely to be similar almost everywhere around the world. The third column is a typical examples of Paris city blocks and we see from the classification matrix in Table 6.2 that cluster 3 collects nearly all samples of this city. Cluster 4 has mixed samples from Istanbul and Copenhagen with a domination of Istanbul (see cluster 4 in Table 6.2). These textures represent both urban and rural areas. Cluster 5 has also similar urban textures from these cities but with a domination of Copenhagen. Cluster 7 in Table 6.4 has mainly textures from Madrid but also from other cities. Los Angeles is represented by cluster 8 with its typical square streets. Half textures of Madrid are represented by cluster 9. Dense areas of Istanbul correspond to cluster 10. Cluster 11 has textures which contain wide roads. From this early interpretation of classification results, we are quite satisfied by the way the textures have been grouped and by the homogeneity of the obtained classes. Results of clusterings in Tables 6.1 and 6.2 show that several clusters have redundant information. It means that for different clusterings there are clusters which have the

Table 6.1: Clustering matrix for 6 cities with EM-algorithm

Clusters	Cities						Σ
	Copenhagen	Istanbul	Los Angeles	La Paz	Madrid	Paris	
1	2	3	2	4	155	6	172
2	117	14	0	0	0	0	131
3	86	131	1	0	5	6	229
4	6	3	253	20	24	251	557
5	131	221	0	0	0	0	352
6	0	0	5	256	7	32	300
7	28	11	7	20	32	48	146
8	30	17	132	4	177	56	416
9	0	0	0	96	0	1	97
	400	400	400	400	400	400	

Table 6.2: Clustering matrix for 6 cities with Kernel K-means algorithm

Clusters	Cities						Σ
	Copenhagen	Istanbul	Los Angeles	La Paz	Madrid	Paris	
1	0	0	0	94	0	1	95
2	28	10	6	22	31	49	146
3	0	0	19	24	9	259	311
4	67	123	1	0	4	6	201
5	112	27	0	0	1	0	140
6	0	0	4	252	5	28	289
7	20	16	72	4	172	34	318
8	13	2	296	0	35	19	365
9	2	2	2	4	142	4	156
10	114	208	0	0	1	0	323
11	44	12	0	0	0	0	56
	400	400	400	400	400	400	

same samples. It will be useful for data mining to combine samples that always belong to common clusters that may reduce redundant information and find some interesting particular clusters in data [Kyrgyzov et al., 2005].

Conclusions

In this Section a new criterion called Kernel MDL (KMDL) to estimate the optimal number of clusters for the Kernel K-means algorithm has been proposed. This criterion is derived from a simplified formulation of the classical MDL for the Gaussian Mixture Model. Both KMDL and the simplified MDL allow determining the optimal number of clusters using simply the error function between the data and the model of clusters. To adapt the criterion to the Kernel K-means algorithm we defined this error function as the corresponding optimised criterion.

The error can be calculated on the kernel function with the Kernel K-means algorithm. The advantage of this approach is that Kernel K-means can linearly separate data which are nonlinearly separable in the original space. As we can see from experimental results the two criteria MDL and KMDL work well and give optimal numbers of clusters each for its own algorithm. Kernel K-means algorithm with KMDL shows superior results than EM with MDL for synthetic data as well as real data. Kernel K-means algorithm with KMDL is able to detect clusters with non globular shapes contrary to EM with MDL. Both approaches give different data clusterings.

6.5 An unsupervised hierarchical clustering based on KMDL

In this Section we develop two new hierarchical clustering algorithms. The first uses the MDL criterion Eq.(6.32) and the second KMDL Eq.(6.36), presented in the previous Section.

Let us first formulate a general unsupervised hierarchical algorithm which optimises GMDL Eq.(6.33). Our proposition is similar to the one presented in [Heas & Datcu, 2005] but differs by the used criterion. In [Heas & Datcu, 2005] the authors propose a hierarchical algorithm optimising MDL by combining two clusters at each step (a level of hierarchy). The idea of this approach is to cluster data into large number of "small" clusters and then optimise hierarchically MDL criteria to find the optimal number of data clusters. Instead of calculating MDL for each model they consider a hierarchy of models and analyse MDL. We use the similar approach but for the proposed GMDL criterion Eq.(6.33) and we also extend this algorithm for kernels.

The choice of an optimal data representation at each level of the hierarchical model is described by GMDL criterion Eq.(6.33). Instead of the direct calculation of this criterion at each level of hierarchy and the search of its optimum it is better to consider its gradient. The minimum value of the gradient shows the optimum as well as a direction in which this optimum may be found. Moreover, the gradient reduces the computation time as well as the volume of stored and processed data.

Firstly, we define the gradient of GMDL Eq.(6.33). Let $GMDL_{K+1}$ be a GMDL Eq.(6.33) with $K + 1$ clusters and $GMDL_K$ with K clusters, respectively. At step K let us combine

Table 6.3: Texture examples of clusters, Kernel K-means

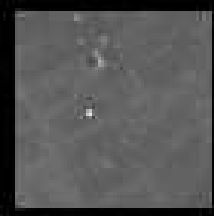
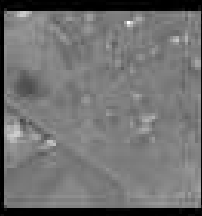
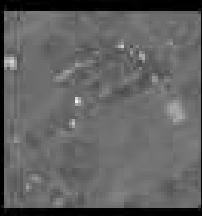
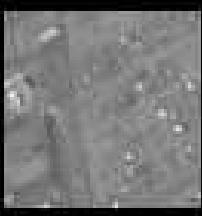



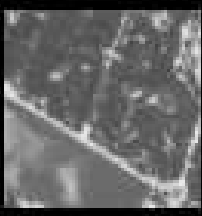
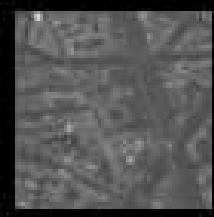
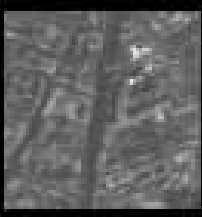
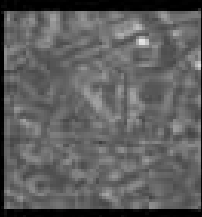
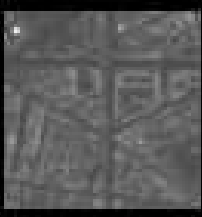
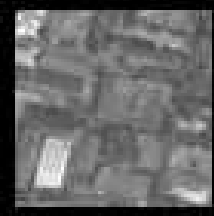
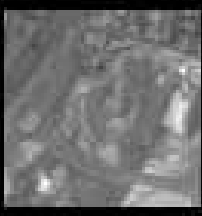
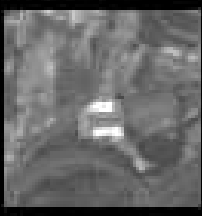
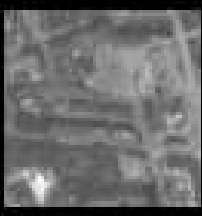
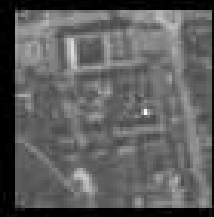
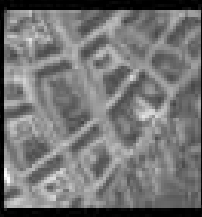
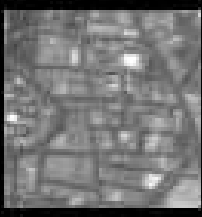
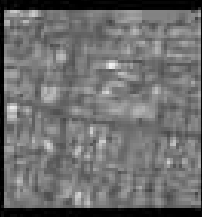
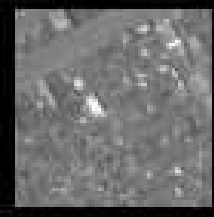
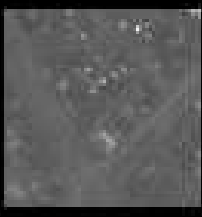
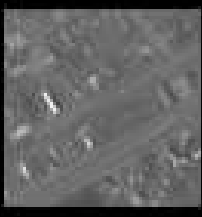
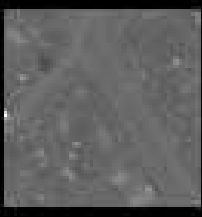
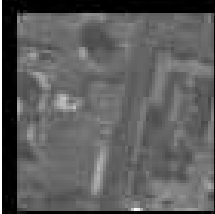
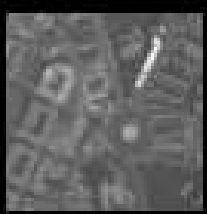
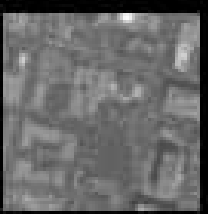

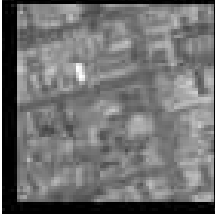
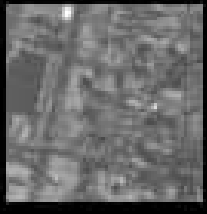
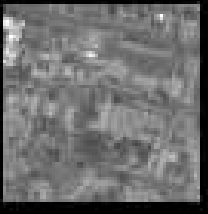
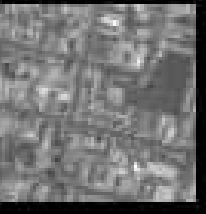

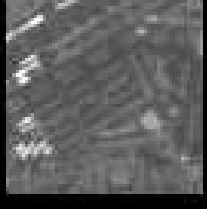
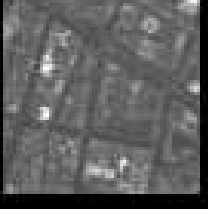
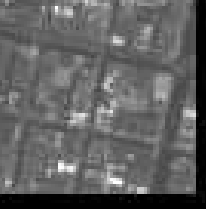
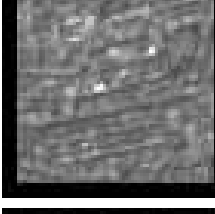
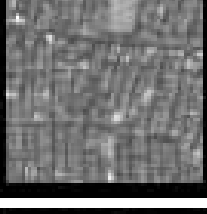
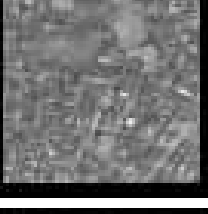
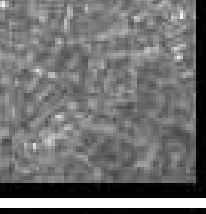
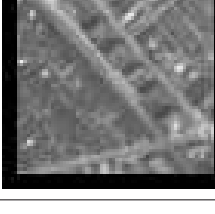

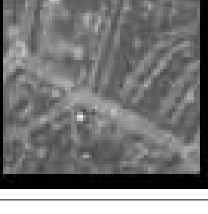

Clusters	Texture examples			
1				
2				
3				
4				
5				
6				

Table 6.4: Texture examples of clusters, Kernel K-means

Clusters	Texture examples			
7				
8				
9				
10				
11				

two clusters k and k' . The gradient of GMDL Eq.(6.33) is expressed as:

$$\begin{aligned}
 \Gamma_{k \cup k'} &= \frac{\partial \text{GMDL}}{\partial K} = \text{GMDL}_K - \text{GMDL}_{K+1} = \\
 &= - \sum_{k=1}^K n_k \log \left(\frac{n_k^2}{\text{Dist}(X_u, X_v | u, v \subseteq k)} \right) + P(K, J, I) + \\
 &= \sum_{k'=1}^{K+1} n_{k'} \log \left(\frac{n_{k'}^2}{\text{Dist}(X_u, X_v | u, v \subseteq k')} \right) - P(K+1, J, I) = \\
 &= n_k \log \left(\frac{\text{Dist}(X_u, X_v | u, v \subseteq k \cup k')}{\text{Dist}(X_u, X_v | u, v \subseteq k)} \right) + n_{k'} \log \left(\frac{\text{Dist}(X_u, X_v | u, v \subseteq u \cup v')}{\text{Dist}(X_u, X_v | u, v \subseteq k')} \right) + \\
 &= 2n_k \log(n_k) + 2n_{k'} \log(n_{k'}) - 2(n_k + n_{k'}) \log(n_k + n_{k'}) - \frac{\partial P(K, J, I)}{\partial K}.
 \end{aligned} \tag{6.40}$$

The same function $P(J, D, I)$ is used as in Eq.(6.30). The gradient of this function is:

$$C = \frac{\partial P(K, J, I)}{\partial K} = (J^2 + 3J + 2) \log(I) / 2 = \text{const} \tag{6.41}$$

Let

$$F_{kk'} = 2n_k \log(n_k) + 2n_{k'} \log(n_{k'}) - 2(n_k + n_{k'}) \log(n_k + n_{k'}) - C, \tag{6.42}$$

be a function of sizes of two merged clusters. Then, for simplicity, we write the gradient Γ of GMDL Eq.(6.40) as:

$$\begin{aligned}
 \Gamma_{k \cup k'} &= n_k \log \left(\frac{\text{Dist}(X_u, X_v | u, v \subseteq k \cup k')}{\text{Dist}(X_u, X_v | u, v \subseteq k)} \right) + \\
 &= n_{k'} \log \left(\frac{\text{Dist}(X_u, X_v | u, v \subseteq k \cup k')}{\text{Dist}(X_u, X_v | u, v \subseteq k')} \right) + F_{kk'}.
 \end{aligned} \tag{6.43}$$

To combine two clusters k and k' the next condition should be satisfied:

$$(k, k') = \arg \min_{uv} \{ \Gamma_{u \cup v} : \Gamma_{u \cup v} \leq 0; u, v = 1, \dots, I, u \neq v \}. \tag{6.44}$$

As we see from Eq.(6.41) the term of complexity does not depend on the number of clusters K . This term has no influence on the best choice of two clusters to be combined. The interpretation of the penalty function Eq.(6.41) is that, it provides a threshold on the hierarchical tree indicating where it should be cut to give the optimal data clustering according to GMDL Eq.(6.33). One of the possible questions to be considered could be the automatic determination of the threshold C during the optimisation procedure. We did not address this problem in our paper.

We may propose now the General Unsupervised Hierarchical Clustering (GUHC) algorithm for GMDL.

If we suppose that $C = 0$ in (6.41) and at step **Step 3** we choose the minimum of $\Gamma_{p \cup q}$ without the condition of negativity, then we can build an optimal clustering tree.

The *GMDL* criterion is calculated at each optimisation step K of *GHUC* algorithm as:

$$\text{GMDL}_K = \Gamma^{(K)} + \text{GMDL}_{K+1}. \tag{6.45}$$

Algorithm GUHC-algorithm

-
- 1: Initiate with K clusters given by any optimal method or consider each sample as a cluster.
 - 2: Compute $Dist(X_u, X_v|u, v \subseteq p \cup q)$ and $\Gamma_{p \cup q}$, where $p, q = 1, \dots, K$.
 - 3: Find $(k, k') = \min_{pq} \{\Gamma_{p \cup q} : \Gamma_{p \cup q} \leq 0; p, q = 1, \dots, K, p \neq q\}$ and $\Gamma^{(K)} = \Gamma_{k \cup k'}$.
 - 4: If no $\Gamma^{(K)}$, the optimal clustering is obtained, stop.
 - 5: Set $Dist(X_u, X_v|u, v \subseteq k) \leftarrow Dist(X_u, X_v|u, v \subseteq k \cup k')$.
 - 6: Reestimate $Dist(X_u, X_v|u, v \subseteq k \cup p)$ and $\Gamma_{k \cup p}$.
 - 6.1: Set $Dist(X_u, X_v|u, v \subseteq p \cup k) \leftarrow Dist(X_u, X_v|u, v \subseteq k \cup p), p = 1, \dots, K$.
 - 7: Delete row $Dist(X_u, X_v|u, v \subseteq k' \cup p), p = 1, \dots, K$.
 - 8: Delete column $Dist(X_u, X_v|u, v \subseteq p \cup k'), p = 1, \dots, K$.
 - 9: Go to **Step 3**.
-

An unsupervised hierarchical clustering algorithm, MDL

In the case when we use GMDL criterion Eq. (6.33) as the optimality criterion Eq. (6.32) we have

$$Dist(X_u, X_v|u, v \subseteq k) = |\Sigma_k|, \quad (6.46)$$

$$Dist(X_u, X_v|u, v \subseteq k \cup k') = |\Sigma_{k \cup k'}|.$$

Then the hierarchical algorithm based on MDL minimises the gradient $\Gamma_{k \cup k'}$ Eq.(6.43) which has the form:

$$\Gamma_{k \cup k'} = \frac{\partial \Lambda}{\partial K} = n_k \log \left(\frac{|\Sigma_{k \cup k'}|}{|\Sigma_k|} \right) + n_{k'} \log \left(\frac{|\Sigma_{k \cup k'}|}{|\Sigma_{k'}|} \right) + F_{kk'}. \quad (6.47)$$

This algorithm finds the optimal number of Gaussian clusters for data X and constructs their hierarchical tree.

An unsupervised hierarchical clustering algorithm, KMDL**Direct error computation**

When GMDL Eq. (6.33) is used as the criterion Eq. (6.36) we obtain:

$$Dist(X_u, X_v|u, v \subseteq k) = S_k^J, \quad (6.48)$$

$$Dist(X_u, X_v|u, v \subseteq k \cup k') = S_{k \cup k'}^J, .$$

Then the hierarchical algorithm based on KMDL minimises the gradient $\Gamma_{k \cup k'}$ (6.43) which has a form:

$$\Gamma_{k \cup k'} = \frac{\partial \text{KMDL}}{\partial K} = J n_k \log \left(\frac{S_{k \cup k'}}{S_k} \right) + J n_{k'} \log \left(\frac{S_{k \cup k'}}{S_{k'}} \right) + F_{kk'}. \quad (6.49)$$

We give a description of the hierarchical algorithm for KMDL criterion. Let us note some of its properties. The proposed error S_k Eq. (6.34) has the good property that its calculation may be done by precalculated data, since the error $S_{k \cup k'}$ may be computed using

errors S_k and $S_{k'}$. It allows avoiding the storage and processing of a kernel matrix \mathcal{K} Eq. (5.17) at each minimisation step of eq. (6.49). Moreover, the calculation of the kernel matrix \mathcal{K} is not needed, but only the computation of errors S_k , $S_{k'}$ and $S_{k \cup k'}$. This significantly decreases the need in memory. If we compute the kernel matrix \mathcal{K} it makes difficult to apply the algorithm for real applications such as images or large database, because of the dimensional issue of matrix \mathcal{K} . In image processing we want to cluster an image of size $n \times n$ on a pixel basis thus with n^2 samples, providing a matrix \mathcal{K} of size $n^2 \times n^2$, i.e. with n^4 terms. It produces a huge volume of data for large n and can not be processed in a reasonable time for our experiments. We write S_k (6.34) as two terms:

$$S_k = \frac{1}{n_k J} \sum_{u \subseteq k} \mathcal{K}(X_u, X_u) - \frac{1}{n_k^2 J} \sum_{u, v \subseteq k} \mathcal{K}(X_u, X_v) = \frac{1}{n_k J} Sd_k - \frac{1}{n_k^2 J} Ss_{kk}, \quad (6.50)$$

where Sd_k and Ss_{kk} is the sum of corresponding diagonal elements and the sum of elements of the kernel matrix \mathcal{K} , respectively. Let Ss be a square matrix where each element $Ss_{kk'}$ is the sum of elements of the kernel matrix \mathcal{K} :

$$Ss_{kk'} = \sum_{u \subseteq k, v \subseteq k'} \mathcal{K}(X_u, X_v). \quad (6.51)$$

The error $S_{u \cup v'}$ of combination of two clusters S_u and $S_{v'}$ can be written as:

$$S_{u \cup v'} = \frac{1}{J} \left(\frac{Sd_k + Sd_{k'}}{n_k + n_{k'}} - \frac{Ss_{kk} + Ss_{k'k'} + 2Ss_{kk'}}{(n_k + n_{k'})^2} \right) \quad (6.52)$$

Matrix Ss may be calculated once or at every optimisation step. To overcome memory complexity for large data sets we propose to initialise the clustering by a high number of clusters which is much lower than the number of samples. The initial clustering may be done by any simple algorithm such as K-means. Note, that if the kernel \mathcal{K} is linear Eq.(5.18), then the algorithm in this Section is equivalent to the algorithm in Sec. 6.5 with the spherical covariance matrix Eq.(6.35).

The proposed hierarchical algorithm has a hypothesis that clusters are spherical. Therefore, calculations are very fast. The modification of this algorithm for the case when there is no prior information on the form of clusters is presented in the following section.

Eigen values for error computation

Here we propose to calculate the gradient Eq.(6.40) of the hierarchical algorithm based on kernel eigen values. Let GMDL criterion Eq.(6.33) be Eq.(6.32) and λ_u is the u^{th} eigen value of Σ_k Eq.(5.31), where $u = 1, \dots, n_k$ and $\lambda_1 \geq \dots \geq \lambda_u \geq \dots \geq \lambda_{n_k}$. Then we write the determinant Eq.(6.35) as:

$$|\Sigma_k| = \prod_{u=1}^{n_k} \lambda_u \quad (6.53)$$

It is interesting to note, that such eigen values can be obtained from the kernel \mathcal{K} defined in Eq. (5.17). For example, eigen values of the covariance matrix Σ_k Eq. (5.31) of data X are equivalent to the eigen values of the linear kernel Eq.(5.18). To compute the covariance matrix Σ_k Eq.(5.31) we normalise data X so that the mean μ_k Eq. (5.30) is equal to zero. So, in a similar way, we compute eigen values of the kernel, but normalising it

such that the mean of this kernel equals to zero. This operation is called kernel centring [Shawe-Taylor & Cristianini, 2004]. A new feature map is given by:

$$\hat{\phi}(X) = \phi(X) - \frac{1}{n_k} \sum_{u=1}^{n_k} \phi(X_u) \quad (6.54)$$

Let X_u and X_v be the same vectors with notations \mathbf{x} and \mathbf{z} , respectively. Then the new centred kernel $\hat{\mathcal{K}}$ is:

$$\hat{\mathcal{K}}(\mathbf{x}, \mathbf{z}) = \mathcal{K}(\mathbf{x}, \mathbf{z}) - \frac{1}{n_k} \sum_{v=1}^{n_k} \mathcal{K}(\mathbf{x}, \mathbf{z}_v) - \frac{1}{n_k} \sum_{u=1}^{n_k} \mathcal{K}(\mathbf{x}_u, \mathbf{z}) + \frac{1}{n_k^2} \sum_{u,v=1}^{n_k} \mathcal{K}(\mathbf{x}_u, \mathbf{z}_v) \quad (6.55)$$

Let, λ_u be eigen values of the centred kernel $\hat{\mathcal{K}}$ Eq.(6.55). Then errors in Eq.(6.43) are:

$$\begin{aligned} Dist(X_u, X_v | u, v \subseteq k) &= \prod_{u=1}^{n_k} \lambda_u, \\ Dist(X_u, X_v | u, v \subseteq k \cup k') &= \prod_{u=1}^{n_k + n_{k'}} \lambda_u. \end{aligned} \quad (6.56)$$

The hierarchical algorithm based on KMDL minimises the gradient $\Gamma_{k \cup k'}$ Eq.(6.43) which has a form:

$$\Gamma_{k \cup k'} = J n_k \log \left(\frac{\prod_{u=1}^{n_k + n_{k'}} \lambda_u}{\prod_{u=1}^{n_k} \lambda_u} \right) + J n_{k'} \log \left(\frac{\prod_{u=1}^{n_k + n_{k'}} \lambda_u}{\prod_{u=1}^{n_{k'}} \lambda_u} \right) + F_{kk'}. \quad (6.57)$$

This approach is more effective because it may calculate not only the Gaussian clusters but clusters of any forms. Unfortunately, such an error calculation is more computational expensive than the direct error computation presented above.

We give a simple example of clustering synthetical data. Two data sets are generated: (i) 15 Gaussians with random covariances Figure 6.6a and (ii) 6 clusters (two Gaussians, two cigars and two circle clusters) Figure 6.6b.

Results of the optimal data clustering by *GHUC*-algorithm with linear kernel and Γ (6.57) are shown in Figures 6.6a and 6.6b. We see in Figure 6.6a that the algorithm with a linear kernel detects correctly Gaussians clusters, but is not appropriate to recover circular clusters in Figure 6.6b. MDL curves showing the optimal number of clusters for data in Figure 6.6 are shown in Figure 6.7. MDL curve for Gaussians (Figure 6.6a) and Γ Eq.(6.47) is presented in Figure 6.7a. It indicates correctly the optimal number of clusters as 15.

For these data Kernel MDL (with Gaussian kernel \mathcal{K} Eq.(5.19)) and Γ Eq.(6.57) is given in Figure 6.7b. The MDL curve with points corresponds to parameter $D = 15$ in C Eq.(6.41) and shows the correct number of clusters 15. The MDL curve with diamonds has $D = 2$ in C Eq.(6.41) and goes down indicating the number of clusters as 35. However, we should say that clustering results are the same for different values of D in kernel MDL. We have shown that gradient of the penalty function Eq.6.41 does not influence on the hierarchical clustering. Finally, KMDL curve for the linear kernel and data in

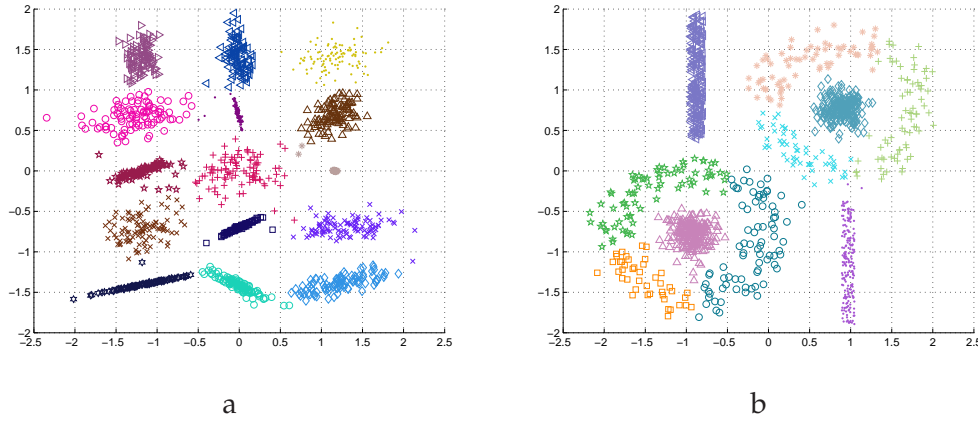


Figure 6.6: Data clustering by *GHUC*-algorithm with linear kernel and Γ (6.57): a - the optimal clustering of 15 Gaussian. d - the optimal clustering of 2 circular, 2 Gaussian and 2 cigar clusters.

Figure 6.6b is demonstrated in Figure 6.7c. The optimal number of clusters detected by this criterion is 10.

Now we cluster data in Figure 6.6b by *GHUC*-algorithm with Gaussian kernel. The optimal clustering is presented in Figure 6.8a. It shows that the hierarchical algorithm detects correctly all 6 clusters (clusters are nonlinearly separated). Two KMDL curves (Γ Eq.(6.57) with Gaussian kernel K Eq.(5.19)) are presented in Figure 6.8b: (i) the curve with points for $D = 15$ in C Eq.(6.41) and (ii) the curve with diamonds for $D = 2$. Here again, clustering results are the same for different values of D in kernel MDL. The optimal number of clusters is 6 for the curve with points in Figure 6.8b.

It is very interesting to note that the proposed kernel hierarchical algorithm based on *KMDL* is robust to the choice of the kernel parameter. For Gaussian kernel K Eq.(5.19) the algorithms gives the same clusters for varying the parameter σ from 10^{-1} to 10^7 .

From the experimental results we note that the function C Eq.(6.41) describing the model complexity is not appropriate when we use, Gaussian kernel Eq.(5.19). But as we previously said this function has no influence on the hierarchical clustering tree and only specifies where we should cut the tree. Thereby the clustering tree is the same for any choice of C Eq.(6.41).

6.6 Conclusions

In this Chapter the problem of model selection has been considered. Different criteria have been shown for hierarchical, partitional and probabilistic clustering algorithms. For hierarchical clustering a criterion based on the cophenetic matrix has been presented, while for partitional clustering within- and between-clustering criteria have been discussed. For probabilistic models as Gaussian mixture model information theoretic criteria as AIC, BIC, SIC and MDL have been revised. A similarity between these criteria has been shown. Data clustering has been considered via GMM with the estimation of its parameters by EM-algorithm. The simplification of MDL for hard clustering and GMM has been proposed. Simplified MDL criterion can be applied for simpler clustering algorithm such as K-means algorithm as well as for its modification as kernel K-means or spectral

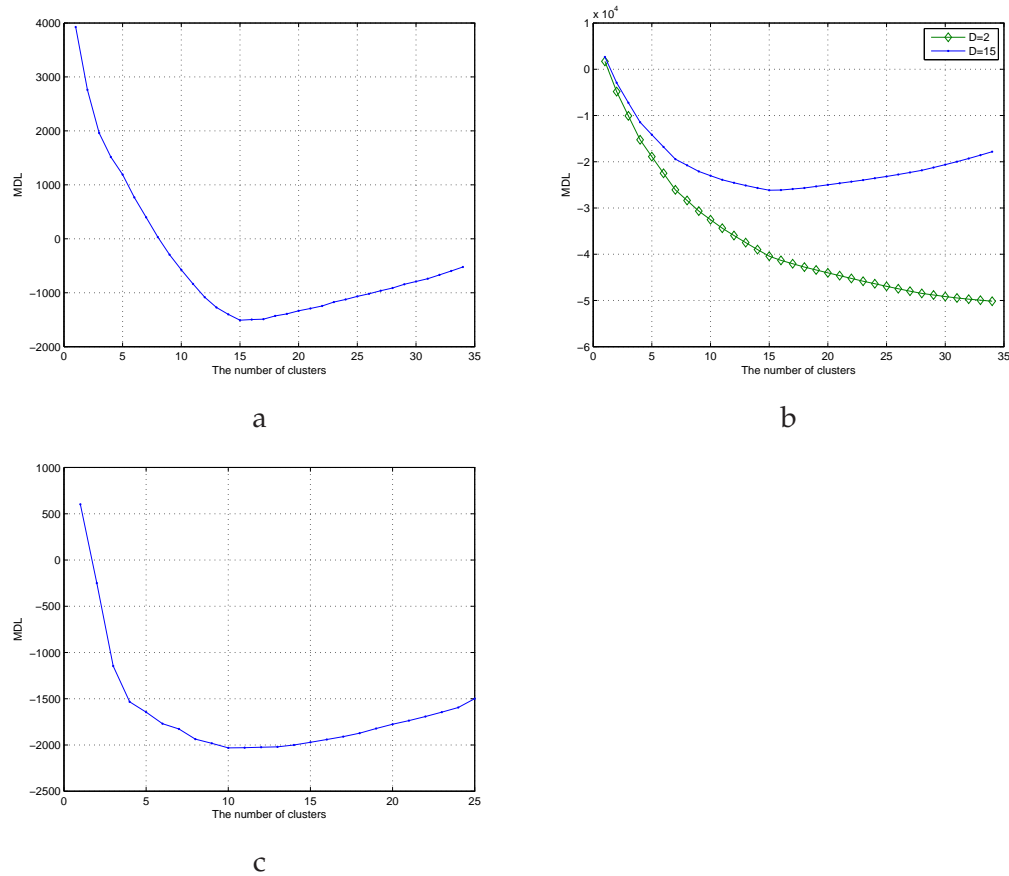


Figure 6.7: MDL curves for clustering data in Figure 6.6. a - MDL curve for clustered Gaussians by Γ (6.47). The optimal number of clusters is 15. b - KMDL curve for clustered Gaussians by Γ (6.57) with Gaussian kernel K (5.19). A line with points for $D = 15$ in C (6.41) and a line with diamonds for $D = 2$. Clustering results are the same for the same number of clusters. The optimal number of clusters is 15. c - KMDL curve for clustered data in Figure 6.6b. The optimal number of clusters is 10.

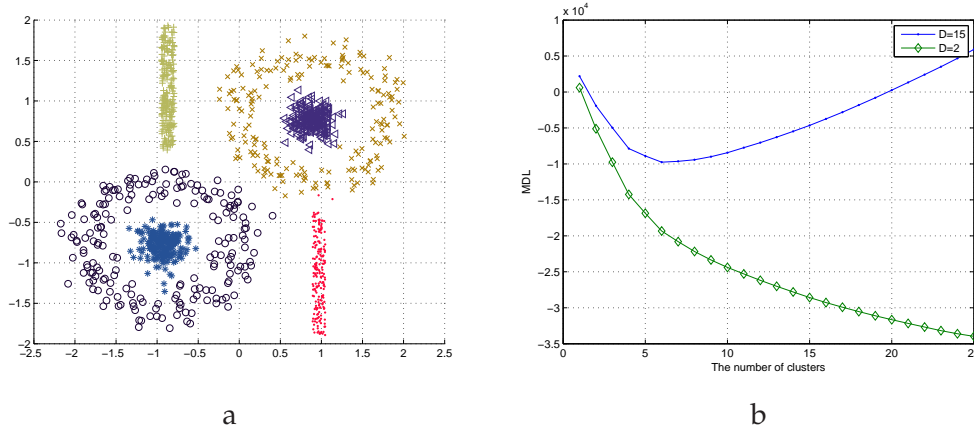


Figure 6.8: Synthetic examples. In a: Result of optimal clustering of example 2 in Figure 6.6-d by *GHUC*-algorithm with Gaussian kernel. in b: KMDL curve for clustered example 2 by Γ (6.57) with Gaussian kernel K (5.19). A line with points for $D = 15$ in C (6.41) and a line with diamonds for $D = 2$. Clustering results are the same for the same number of clusters. Optimal number of clusters is 6.

K-means. The advantage of kernel K-means is that it can separate clusters which are not linearly separated.

A hierarchical algorithm based on the simplified MDL has been proposed in this Chapter. The idea of this algorithm consists in optimisation of MDL criteria. The optimisation is done via the gradient descend method. Each step of optimisation is performed by constructing a hierarchical tree. At each level of the tree we combine two clusters. The order to combine clusters indicated by gradient of MDL criteria. This hierarchical clustering is performed in the unsupervised way and determine the optimal number of clusters.

This algorithm was extended to formulate kernel hierarchical clustering based on the kernel MDL. The advantage of this algorithm is that it is able to cluster data in unsupervised way, to find the optimal number of clusters and to separate clusters which are not separable linearly.

Chapter 7

Combination of clustering results

A survey of basic and recent methods for combining clustering results are presented in this chapter. This survey covers a wide set of approaches : from well formulated with theoretical bases to empirical approaches. Here combination of clusterings is considered as an unsupervised task since we aim at avoiding user interaction, either because it may take a lot of time for a user to analyse clusterings, or because it is very difficult to interpret them. We propose to combine clusterings using clustering algorithms. We will see that each approach has its advantages and disadvantages. Disadvantages motivate us to state the problem of combination in a new way and discard them. After problem formulation we propose two unsupervised methods and algorithms to combine different clusterings. The first method, although efficient, has no clear proof about its convergence to a unique global optimal solution. Concerning the second method the same problem is reformulated, that leads a global solution. In addition, an algorithm is proposed to find this solution. A proof of the convergence of this algorithm to the global unique solution is derived. Advantages of the proposed methods are discussed. Results on synthetical as well as real data sets are provided at the end of this Chapter. Examples of combination will be illustrated in the following Chapter.

7.1 Introduction

We introduce the problem of clustering combination with application to the satellite image analysis. In the recent years many different imaging sensors for Remote Sensing have appeared delivering a huge amount of digital images. This kind of data of Earth Observation (EO) is used by the experts of various domains (ecology, agriculture, defence, etc.). Along with the image, the expert gets a description in terms of sensor type, geographical coordinates, time of reception, spectral bands. This information gives a rough description of the image, but it characterises a whole image and can not give answers about the precise content of the image. Numerous experiments on image processing have shown that a small part of any satellite image may be captured in a vector of measures which expresses the main properties : the multi spectral or radiometric content, the textural properties, the structural properties, etc. [Pratt, 2001; Barnes, 2007]. Such information would describe the exact content in terms of imagery, may extremely improve managing image database, facilitate understanding and discovering new classes from images. What we call "content of the image" is the answer to the question "What is in the image?", e.g., which structures, which scattering properties, objects, etc. Experts from different do-

mains give different answers to this question because they operate with various concepts and have various applications in mind. They also have a different knowledge about the way to describe the image. It means that an expert has his own ontology i.e., a specification of a conceptualisation of a knowledge domain. Under the word "a concept" we consider a group of similar objects. One of the approaches for automatic discovering of concepts in the data is the clustering. In this case, concepts are clusters. There is a variety of clustering algorithms (Chapter 5). Each of them has its own advantages and disadvantages. Some algorithms are robust and correctly cluster data even in the case of heavy noise but they may be not sensitive to data with a complex structure. On the contrary, other algorithms are able to find true clusters in the data with complex structures but the influence of noise on their clustering results is very strong. Thus, the trade off is often to choose the best clustering algorithm.

Clustering algorithms are basic components of pattern recognition. They are used for data mining tasks when there is no or few prior information about data to be analysed [Jain & Dubes, 1988]. The profit of clustering is to obtain groups of samples which will be seen as similar for further data exploration or retrieval in supervised or semi-supervised classification [Duda et al., 2000].

It is a common practice when several clustering steps are made in parallel, either because different algorithms, or, because different parameters of the same algorithm. Different clusterings provide complementary results from which we may benefit [Fred & Jain, 2005; Strehl & Ghosh, 2002; Topchy et al., 2004a; Ayad & Kamel, 2005; Boulis & Ostendorf, 2004; Li et al., 2004; Y. Qian, 2000; Topchy et al., 2004b]. At this point the main difficulty is to determine a judicious criterion to combine elementary clusterings to obtain a final clustering solution. Then a remaining problem may arrive to efficiently implement the chosen method in the case of very large databases. The contribution of this Chapter is to address these two problems.

Many different methods may be used to fuse information issued from different clustering [Diday, 1979; Michaud & Marcotorchino, 1979; Marcotorchino & Michaud, 1982; Fred & Jain, 2005; Strehl & Ghosh, 2002; Topchy et al., 2004a; Kuncheva, 2004]. We consider two approaches in this Chapter: (i) probabilistic and (ii) algebraic clustering combination. The probabilistic approach analyses clusterings as nominal data and combination is performed via unsupervised clustering. The algebraic approach operates with matrix representation of clusterings. Algebraic methods are based on the property of two samples to belong or not to the same cluster, depending on the type of clustering. A review of methods is given in Section 7.2. Probabilistic combination with different methods of nominal data clustering is presented in Section 7.3. We formulate the combination criterion for the algebraic approach with some mathematical developments in Section 7.4. Section 7.4 describes the proposed combination algorithm along with an improvement to efficiently process real data. In Section 7.5 a global optimum of the proposed combination criterion is shown to be found exactly by an iterative mean shift algorithm. Combination results on both synthetical and real data are presented and discussed in Section 7.5. Finally, estimation of clustering stability is discussed in Section 7.6.

7.2 Related works

Combination of different inferences has found many applications relative to supervised classification. There are many works which establish the usefulness of such an approach

[Xu et al., 1992; Huang & Suen., 1995; Al-Ani & Deriche, 2002; Kang & Kim, 1997]. However clustering combination is not yet deeply studied [Kuncheva, 2004]. The goal of combination of classifiers (respectively clusterers) is to combine their outputs to improve the final classification (respectively clustering). The difference between combining classifications and clusterings is that for combining of classifications a finite number of classes is used and the classes of the different classifiers are the same. On the contrary, clusterings give different sets of clusters, different numbers of clusters and clusters usually do not correspond from one clustering to the other. Therefore, the combination of clusterings can provide a new set of clusters.

There exist many different methods to aggregate information pieces issued from different clustering techniques. One of the most attractive is based on the use of a co-association matrix [Diday, 1979; Michaud & Marcotorchino, 1979; Marcotorchino & Michaud, 1982]. A co-association matrix reflects the number of occurrences of two samples to be in the same cluster depending on the classification algorithm. An element of this symmetric square matrix with size equal to the number of samples, may be interpreted as the frequency of two samples to be in the same cluster. The co-association matrix will be introduced in Section 7.4.

In [Fred & Jain, 2005], the authors propose a methodology which is related to an algebraic approach of clustering combination described in [Marcotorchino & Michaud, 1982; Diday, 1979]. A number of clusterings are obtained by *K-means* algorithm with random initialisations and a random number of clusters. Collecting their results allows building the co-association matrix. This problem has been considered as the detection of "des formes fortes" proposed in [Diday, 1979]. Several approaches have been proposed to issue a "consensus" clustering from the set of given clusterings [Michaud & Marcotorchino, 1979; Marcotorchino & Michaud, 1982]. In [Michaud & Marcotorchino, 1979], the authors propose several measures to find the "consensus clustering" via aggregation approach. Clustered data samples are presented by binary matrices and aggregated in order to obtain the consensus clustering. The "consensus" clustering is also coded as a binary matrix and approximates the set of the clusterings.

Another approach to find the "consensus" clustering is based on the linear programming [Marcotorchino & Michaud, 1982]. The objective function firstly proposed in work [Michaud & Marcotorchino, 1979] is shown to have a linear form [Marcotorchino & Michaud, 1982]. Finally, the exact solution to find the "consensus" clustering is proposed. An important drawback of the proposed approach is a square memory complexity. It makes difficult its application for very large data sets.

An approach of clustering combination is based on a hierarchical classification with a single-link method. This algorithm is applied to the co-association matrix to group samples which appear the most frequently together [Marcotorchino & Michaud, 1982]. In [Fred & Jain, 2005], the final number of classes is taken either as the one that corresponds to the longest lifetime on the dendrogram of the hierarchical algorithm or as the one which provides the highest mutual information measure between the initial clusters and successive classifications. In this case, normalised mutual information (denoted *NMI* in [Fred & Jain, 2005]) is the objective criterion of the method. It expresses a global quality of the final partition. This method, only based on the frequency of association of different samples to the same class, is interesting for the user who does not need to care about the elementary clustering methods. It makes no assumption on the reasons for which samples have been grouped and does not question about the pertinence of the initial clustering stage. However it suffers from several limitations: (i) it requires some

prior knowledge on the approximate number of clusters, (ii) it does not guarantee any optimality of the final classification and (iii) it may face storage and computational problems when dealing with very large sample sets.

The first limitation comes from the initial clustering stage. If the number of initial clusters is sequentially increased from 2 to the number of samples, the co-association matrix tends to be a near diagonal matrix with small values out of diagonal. Therefore, the more clusters used to build the co-association matrix, the more clusters result from the combination. To limit this trend, following the method presented in [Fred & Jain, 2005], one should constrain the initial K parameter of the *K-means* to values close from the targeted number of classes.

The third limitation is due to the single-link algorithm used to extract the final classes (or similarly to the complete-link or to the average-link algorithms which are proposed as alternatives in [Fred & Jain, 2005]). This algorithm requires the storage of the complete co-association matrix (or at least its upper-part). In case of thousands of samples, this may create storage and computational difficulties.

To address the second limitation, the method proposed by Topchy *et al.* may be used [Topchy *et al.*, 2004a]. In order to optimise the final classification, Topchy *et al.* consider the clustering combination in the framework of finite mixture models of clustering ensembles and solve it according to the maximum likelihood criterion with the Expectation-Maximisation (EM) algorithm. Another solution to overcome this second limitation may be found in Strehl and Ghosh [Strehl & Ghosh, 2002] using also the mutual information as an objective function as in [Fred & Jain, 2005], but optimising it with a greedy combinatorial algorithm. Unfortunately, its complexity is exponential with the number of samples. Both Topchy *et al.* [Topchy *et al.*, 2004a] and Strehl and Ghosh [Strehl & Ghosh, 2002] methods require a predetermined number of final classes. We will propose below a way to overpass this constraint.

In [Y. Qian, 2000] Qian & Suen propose to combine clustering labels jointly with a feature space of data. We do not consider such an approach here because very often it is impossible to combine unambiguously criteria of different clustering algorithms. In addition, for their approach, several prior parameters should be tuned to combine clustering results. Ayad and Kamel [Ayad & Kamel, 2005] combine clusterings generated by *K-means* algorithm with the same predetermined number of clusters. Authors argue that the representation of clustering labels by a co-association matrix is cumbersome and propose to analyse a matrix of pairwise distances between clusters, instead. They find the correspondence between clusters from different clustering. Then a group-average hierarchical clustering is applied to group elements of this matrix, in such a way that they always combine clusterings with the same number of clusters. Authors do not provide any objective function to estimate the combination quality and the number of clusters after combination.

In [Boulis & Ostendorf, 2004], a matrix of sample associations is used to represent different clusterings. Then combination of clusterings is obtained by clustering this matrix. In this approach, the final number of clusters should be a priori known. Lange and Buhmann [Lange & Buhmann, 2005] make use of a probabilistic model of the co-association matrix. The *EM*-algorithm optimises model parameters. It requires $O(I^2)$ operations for each iteration, where I is the number of data samples making it difficult to apply this approach to a high volume of data.

Clustering combination is a recent interesting topic in data mining but it appears up to now weakly exploited. A recent survey [Kuncheva, 2004] only reviews the few methods

of clustering combination which we also present in this chapter.

As we have seen many methods need a priori information about data to combine clusterings or to manually set parameters for the combination scheme. This motivates us to state the problem in a form which will not depend on any parameter and prior knowledge.

Firstly, we propose to consider combination as nominal data clustering. Several algorithms can be applied: from K-means to EM-algorithm with multinomial mixture models. The optimal combination for these algorithms can be selected by MDL criteria presented in Chapter 6. Secondly, we formulate combination using the co-association matrix. It allows to process large volumes of data as well as large numbers of final classes without using the co-association matrix explicitly. We propose an objective function and two algorithms to combine different clusterings. The first algorithm uses a hierarchical approach and shows competitive performances compared to existing ones. It combines clusterings in an unsupervised way for a large volume of data. Unfortunately, there is no proof that it always achieves a global optimum. The second algorithm is a fast iterative combination algorithm for which we prove the convergence to a global optimum of the proposed objective function. It outperforms experimentally proposed combination approaches.

7.3 Nominal data clustering

Combination of clusterings may be seen as nominal (or categorical) data grouping, where labels of clusterings are nominal data. Standard clustering algorithms may be applied to find a solution of clustering combination, e.g., via "hard clustering" as K-means algorithm [Diday, 1979] or probabilistic modelling with EM-algorithm [Mclachlan & Peel, 2000; Hardle et al., 2003; Bishop, 2006]. In Chapter 5 K-means, spectral K-means, kernel K-means algorithms have been considered. They cluster continuous data for which Euclidean or other (kernel) distance is determined [Shawe-Taylor & Cristianini, 2004]. Nominal data should be transformed to a binary data set in order to apply distances and clustering algorithms as for continuous data [Diday, 1979; Mclachlan & Peel, 2000; Bishop, 2006]. This question is considered in the following subsection. From the other hand, the probabilistic approach with EM-algorithm may be applied directly to nominal data (but for convenience, we also transform data to show clearly calculation of probabilities), Sections 7.3 and 7.3.

Partitional clustering

Partitional clustering algorithms such as K-means can be chosen to cluster nominal data, e.g., labels or names. In our case, nominal data are cluster labels.

Let us consider one clustering. Intuitively, there is no distance between different labels (since there is no metrics in the label space) and no order among labels. But samples belonging to the same cluster considered as equivalent. On the contrary, samples with different labels have all the same difference. Let this difference take value 1.

Let a data set be clustered into a given number of clusters. Then let assume that there are several clusterings and each of them having its own number of clusters. Let I be the number of samples and P the number of clusterings. Each clustering (denoted with index $p, p = 1, \dots, P$) associates each sample i with one and only one cluster.

We may describe the p^{th} clustering by a binary rectangular matrix B^p with I rows and

J_p columns, where J_p equals the number of clusters in the p^{th} clustering, so that:

$$B_{ik}^p = \begin{cases} 1, & \text{if sample } i \in k, \\ 0, & \text{otherwise.} \end{cases} \quad (7.1)$$

where $i = 1, \dots, I, k = 1, \dots, J_p$. B^p is called a partition matrix. Let us note some properties of matrix B^p :

1. all columns of the matrix B^p are orthogonal, it means that the vector product of any two different columns equals zero;

$$2. \sum_{k=1}^{J_p} B_{ik}^p = 1;$$

3. if samples i and l are from the same cluster (have the same label), then

$$\sum_{k=1}^{J_p} B_{ik}^p B_{lk}^p = 1, \text{ otherwise, if } i \text{ and } l \text{ are from different clusters } \sum_{k=1}^{J_p} B_{ik}^p B_{lk}^p = 0.$$

We can conclude that the vector product of rows of B^p can be used to state the distance between nominal samples presented by binary matrix B^p as 1 - vector product. Now, let binary matrix B be the concatenation of matrices B^p such that

$$B = [B^1, \dots, B^p, \dots, B^P]. \quad (7.2)$$

Here again the distance between samples i and l can be stated via the vector product of B_i and B_l . If samples i and l are always in the same cluster for different clusterings p (clusters may have different labels in different clusterings) then the vector product $B_i B_l'$ equals P . When samples i and l have no chance to be in the same cluster for any clustering p then the vector product $B_i B_l'$ equals 0.

The vector product :

$$B_i B_l' = \sum_{k=1}^{\sum_{p=1}^P J_p} B_{ik} B_{lk} \quad (7.3)$$

has two important properties:

1. it is limited by 0 from the bottom,
2. it is limited by P from the top.

After normalising Eq.(7.3) by P we obtain: $0 \leq B_i B_l' \leq 1$. Then the distance between two samples i and l can be stated via the vector product as:

$$d(B_i B_l') = 1 - \sum_{k=1}^{\sum_{p=1}^P J_p} B_{ik} B_{lk} \quad (7.4)$$

When the distance is introduced for nominal data presented by binary matrix B then a clustering algorithm can be applied, e.g. K-means algorithm.

Let us pay attention at direct clustering. As it was mentioned in Section 5.2 clustering can be done by verifying all possible clustering solutions (combinatorial search).

Combinatorial search

Let the combination of clustering be expressed by the binary matrix B^s . Combinatorial search for the optimal clustering B^s has two bounds:

1. if B^s is the identity matrix, then each cluster has exactly one sample,
2. if B^s has only one row of ones, then only one cluster contains all samples.

The number of possible solutions between these two extreme cases is given by Stirling number Eq. (5.1) to obtain all possible clusterings or variants of B^s [Jain & Dubes, 1988]. This number is too high for most of practical cases (up to 10^{155} for 100 samples). The direct search of B^s could be applied only to a set with a very small number of samples.

One of the ways for direct search may be in computing some statistical quantities on B^s to restrict the search range. For example, to compute the lower and higher bounds for possible number of clusters. Others links on this topic for a few samples could be found in [Jain & Dubes, 1988].

Partitional algorithms

Since distance Eq. (7.4) is introduced, we can apply K-means algorithm to cluster binary matrix B Eq. (7.2). The result of this clustering is the combination of clusterings. Analogously, spectral and kernel K-means algorithms may be applied to cluster matrix B . In the case of kernel K-means presented in Section 5.4 the linear kernel \mathcal{K} Eq.(5.18) for the data B is simply covariance matrix $\mathcal{K} = BB'$, where $'$ denotes the matrix transposition operation. We should note, that for kernel K-means it has been proven in [Shawe-Taylor & Cristianini, 2004] that the first K eigen vectors of kernel \mathcal{K} contain a global optimal solution for data clustering into K clusters, but there is no explicit solution how to obtain this clustering.

As clusterings can be presented by binary matrices it allows applying clustering algorithms to binary data. One of such approach can be found in [Govaert & Nadif, 2007], where K-means algorithm and Expectation-Maximisation algorithm with multinomial mixture model are compared. That work is not considered for combination of clusterings, however it has many common points with this problem. The comparison of K-means and EM-algorithm is given in Chapter 8.

We have shown the distance between samples i and l via the vector product Eq. (7.4). Let have a look at the Euclidean distance $d_E(B_i, B_l)$ between points B_i and B_l :

$$\begin{aligned}
 d_E(B_i, B_l) &= (B_i - B_l)^2 = \sum_{k=1}^{\sum_{p=1}^P J_p} (B_{ik} - B_{lk})^2 \\
 &= \sum_{k=1}^{\sum_{p=1}^P J_p} B_{ik}^2 - 2 \sum_{k=1}^{\sum_{p=1}^P J_p} B_{ik} B_{lk} + \sum_{k=1}^{\sum_{p=1}^P J_p} B_{lk}^2 \\
 &= 2P - 2 \sum_{k=1}^{\sum_{p=1}^P J_p} B_{ik} B_{lk} = 2P - 2B_i B_l'.
 \end{aligned} \tag{7.5}$$

We see that Euclidean distance $d_E(B_i, B_l)$ Eq. (7.5) is equivalent to the vector product distance $d(B_i, B_l)$ Eq. (7.4) and differs only by a constant $2P$. The normalised by $2P$ Euclidean distance have the same properties: it is bounded $0 \leq d_E(B_i, B_l) \leq 1$. In addition, the Euclidean distance $d_E(B_i, B_l)$ is the same as the Hamming distance multiplied by a constant:

$$d_H(B_i, B_l) = \frac{1}{2}d_E(B_i, B_l) = P - B_i B_l'. \quad (7.6)$$

The Hamming distance shows how many bits are different between two binary strings. We see that different distances $d(B_i B_l')$ Eq. (7.4), $d_E(B_i, B_l)$ Eq. (7.5) and $d_H(B_i, B_l)$ Eq. (7.6) are the same for combination of clusterings (they differ by a coefficient).

The main problems of the approach presented in this Subsection concern K-means like algorithm:

1. local optimal solution,
2. the number of clusters is not known.

The first problem can be solved by running several times the algorithm and selecting the "best" clustering, e.g., in the sense of the square error. For the second problem simplified MDL (Chapter 6) can be used to determine the optimal number of clusters. But in this case the number of model parameters for simplified MDL should be changed: the dimension of data B should equals P and not $\sum_{p=1}^P J_p$. A comparison of algorithms to cluster binary data can also be found in [Govaert & Nadif, 2007].

In the following Section we demonstrate how a probabilistic approach can be applied to combine different clusterings. Probabilistic models and the estimation of its parameters by EM-algorithm will also be done.

Binomial distribution

Combination of clusterings may be considered as clustering of nominal data via a probabilistic modelling. In this case we do not apply a distance as, e.g., Euclidean or other (Section 7.3), but we can cluster clusterings using probabilistic classification. In this Chapter we propose to survey Bernoulli probabilistic model that gives a passage to a multinomial model which is more general.

Here again clusterings are presented as binary matrix. We use the same notations as previous, where each clustering p , $p = 1, \dots, P$ is presented by binary matrix B^p Eq. (7.1), and P is the total number of clusterings. We can concatenate matrices B^p into one matrix B . Instead of element notations B_{ij_p} , $i = 1, \dots, I$, $j_p = 1, \dots, J_p$, let use simply notation B_{ij} , where $j = 1, \dots, \sum_{p=1}^P J_p$.

Let b be a single random binary variable (or one column of the matrix B). Then we may introduce the probability that variable b takes value 1 as:

$$P(b = 1 \mid \mu) = \mu, \quad (7.7)$$

where $0 \leq \mu \leq 1$. Then the probability that b takes 0 is $P(b = 0 \mid \mu) = 1 - \mu$. The probability of the distribution b is called Bernoulli distribution:

$$Bern(b \mid \mu) = \mu^b (1 - \mu)^{(1-b)}, \quad (7.8)$$

where its mean is $\mathbb{E}[b] = \mu$, its variance is $\text{var}[b] = \mu(1 - \mu)$. Then we may write a likelihood function for binary data B , with assumption that data are *i.i.d* (independently and identically distributed) from $P(b | \mu)$ as:

$$P(B | \mu) = \prod_{i=1}^I P(b_i | \mu) = \prod_{i=1}^I \mu^{b_i} (1 - \mu)^{(1-b_i)}. \quad (7.9)$$

Then the logarithm of likelihood function may be written as:

$$\log P(B | \mu) = \sum_{i=1}^I (b_i \log \mu + (1 - b_i) \log(1 - \mu)). \quad (7.10)$$

Setting the derivative of $\log P(B | \mu)$ to zero with respect to μ we obtain a maximum likelihood estimation for μ :

$$\mu = \frac{1}{I} \sum_{i=1}^I b_i, \quad (7.11)$$

or if the number of 1 in the column b of B is m then:

$$\mu = \frac{m}{I}. \quad (7.12)$$

The distribution of m , given the data B is called Binomial distribution:

$$\text{Bin}(m | I, \mu) = \binom{I}{m} \mu^m (1 - \mu)^{(I-m)}, \quad (7.13)$$

where

$$\binom{I}{m} = \frac{I!}{(I-m)!m!}, \quad (7.14)$$

is the number of ways of choosing m objects out of I . The mean of this distribution is $\mathbb{E}[m] = I\mu$ and the variance $\text{var}[m] = I\mu(1 - \mu)$.

Bernoulli mixture model

Our goal in this section is to model binary data and to combine them into clusters. This modelling is based on a mixture of Bernoulli distribution and estimating mixture parameters. Each mixture component corresponds to a cluster of nominal data. Below we give the mixture model for Bernoulli distributions and EM-algorithm which is used to estimate parameters of the mixture.

Bernoulli distribution is also known as latent class analysis. In the previous section a single variable b of B has been considered, let now see the set of variables or matrix B , where $B_{ij} \in \{0, 1\}$, $i = 1, \dots, I$, $j = 1, \dots, J$ and $J = \sum_{p=1}^P J_p$. When all variables (column of B) have the Bernoulli distribution with parameter μ_j , then:

$$P(B | \mu) = \prod_{j=1}^J \mu_j^{B_j} (1 - \mu_j)^{(1-B_j)}, \quad (7.15)$$

where $\mu = \{\mu_1, \dots, \mu_j, \dots, \mu_J\}$. The mean value of this distribution is $\mathbb{E}[B] = \mu$ and the covariance matrix is $\text{cov}[B] = \text{diag}\{\mu_j(1 - \mu_j)\}$.

The mixture of K components of Bernoulli distributions for sample B_i has the same general form as for GMM Eq. (5.27):

$$P(B_i | \boldsymbol{\mu}, \boldsymbol{\alpha}) = \sum_{k=1}^K \alpha_k P_k(B_i | \mu_k), \quad (7.16)$$

where $\boldsymbol{\mu} = \{\mu_{1j}, \dots, \mu_{kj}, \dots, \mu_{Kj}\}$ and the probability of B_i given μ_k is

$$P_k(B_i | \mu_k) = \prod_{j=1}^J \mu_{kj}^{B_{ij}} (1 - \mu_{kj})^{(1-B_{ij})} \quad (7.17)$$

Assuming that data B are *i.i.d.* the logarithm of the joint data probability or likelihood function is the logarithm of the product of probabilities of B given $\boldsymbol{\mu}$ and $\boldsymbol{\alpha}$:

$$\log P(B | \boldsymbol{\mu}, \boldsymbol{\alpha}) = \log \prod_{i=1}^I P(B_i | \boldsymbol{\mu}, \boldsymbol{\alpha}) = \sum_{i=1}^I \log \left\{ \sum_{k=1}^K \alpha_k P_k(B_i | \mu_k) \right\} \quad (7.18)$$

Introducing a complete likelihood and using Bayes' theorem with some transformations [Bishop, 2006] we may obtain the weight w_{ik} or the conditional probability that the sample B_i belongs to the class k :

$$w_{ik} = \frac{\alpha_k P_k(B_i | \mu_k)}{\sum_{l=1}^K \alpha_l P_l(B_i | \mu_l)}. \quad (7.19)$$

EM-algorithm for Gaussian mixture model have been proposed in Section 5.5 . We can apply this algorithm to the Bernoulli distribution as well. Weights w_{ik} are calculated on the E step of EM-algorithm. Parameters $\boldsymbol{\mu}$ and $\boldsymbol{\alpha}$ are calculated on the M step and maximise the expected complete likelihood of the data. The expected number of points N_k of the component k is:

$$N_k = \sum_{i=1}^I w_{ik}. \quad (7.20)$$

The mean value μ_k of this component is:

$$\mu_{kj} = \frac{1}{N_k} \sum_{i=1}^I w_{ik} B_{ij}. \quad (7.21)$$

The weights of mixture model is:

$$\alpha_k = \frac{N_k}{N}. \quad (7.22)$$

In addition, next constraints should be satisfied:

$$\sum_{k=1}^K \alpha_k = 1, \sum_{j=1}^J \mu_{kj} = 1, 0 \leq \alpha_k \leq 1 \text{ and } 0 \leq \mu_k \leq 1.$$

Now the EM-algorithm for the Bernoulli mixture model (BMM) can be given.

We have seen that combination of clusterings may be done through the probabilistic approach. This approach includes

1. representation of clusterings in a binary matrix,

Algorithm 7.3 Pseudo code of EM-algorithm for the Bernoulli mixture model

1: Initialise K means μ_k and α_k .

2: **E-step**

2.1: Calculate probabilities $P_k(B_i | \mu_k)$ as in Eq. (7.17):

$$P_k(B_i | \mu_k) = \prod_{j=1}^J \mu_{kj}^{B_{ij}} (1 - \mu_{kj})^{(1-B_{ij})}$$

2.2: Calculate weights w_{ik} as in Eq. (7.19):

$$w_{ik} = \frac{\alpha_k P_k(B_i | \mu_k)}{\sum_{l=1}^K \alpha_l P_l(B_i | \mu_l)}$$

4: **M-step** Re-estimate parameters N_k Eq. (7.20), μ_k Eq. (7.21) and α_k Eq. (7.22)

4.1:
$$N_k = \sum_{i=1}^I w_{ik}.$$

4.2:
$$\mu_{kj} = \frac{1}{N_k} \sum_{i=1}^I w_{ik} B_{ij}.$$

4.3:
$$\alpha_k = \frac{N_k}{N}.$$

5: Evaluate the log-likelihood function:

$$\log P(B | \mu, \alpha) = \sum_{i=1}^I \log \left\{ \sum_{k=1}^K \alpha_k P_k(B_i | \mu_k) \right\}$$

6: **If** log-likelihood is converged,

then stop,

else go to **Step 2**.

2. modelling binary data via the Bernoulli mixture model,

3. applying EM-algorithm to find groups of clusterings,

4. information theoretical measures discussed in the Chapter 6 may be applied to select the best model.

Examples and comparison between different approaches of clustering combination are given in Chapter 8. We should note several disadvantages of such modelling:

1. it is well known that EM-algorithm gives a locally optimal result,

2. classification result of this algorithm depends on the initialisation. There is a little chance to have a good initialisation for large data set, e.g., for image processing.

3. it may be difficult to apply this algorithm when many (hundreds) clusters and clusterings (some tens) are used: during classification by EM-algorithm (**Algorithm 7.3**), the multiplication in $P_k(B_i | \mu_k)$ Eq. (7.17) will go to zero).

To avoid these problems the following solution may be proposed: select the best classification and mixture model by MDL Eq. (6.28) criterion for different initialisations and the number of mixture components.

Unsupervised classification of nominal data by Bernoulli distributions supposes that each variable takes one of two binary values. But in the case of clustering combination each variable of B takes a set of mutually exclusive values. We may see this property in revising matrix B^p . Columns of B^p are mutually exclusive and orthogonal. In this case, instead of Bernoulli distribution data should be modelled by a multinomial distribution [Bishop, 2006]. This is explained below as well as EM-algorithm to estimate parameters of the multinomial model.

Multinomial mixture model

Labels of one clustering result may be coded as binary matrix B^p . Columns of this matrix are orthogonal and mutually exclusive. The Bernoulli distribution for binary data suppose that binary variables are independent but not mutually exclusive. For the last case binary data B^p as well as concatenated matrix B of B^p should be modelled by multinomial distribution.

Groups in clusterings may be found by EM-algorithm with a probabilistic modelling. Under probabilistic modelling we consider a mixture model of multinomial distributions. Parameters of this model are estimated by EM-algorithms. Without loss of generality we further consider that the obtained unsupervised classification is a combination of clusterings and found classes represent clusters of clusterings.

We should make a difference between binomial and multinomial distributions. The multinomial models generalise binomial distribution, however both models may be applied to nominal data and very often give the same results. Although we should note that multinomial mixture model with EM-algorithm may process the higher number of clusters than the binomial mixture, in regard to the problem of multiplication of probabilities $P_k(B_i | \mu_k)$ Eq. (7.17).

We have p different clusterings, where each clustering have K^p clusters, $p = 1, \dots, P$. Let $j_p = 1, \dots, K^p$ be the index for cluster j_p in clustering P . Without loss of generality let binary matrix B be the concatenation of binary matrices B^p as in Eq.(7.2). Then an element B_{ij_p} means that sample i belongs to cluster j_p in clustering p , where $i = 1, \dots, I$. Matrix B has a property that the sum over each sample i and each cluster j_p is $\sum_{j_p=1}^{J_p} B_{ij_p} = 1$, $\forall i, p$. Denoting the probability that $B_{ij_p} = 1$ via parameter μ_{ij_p} we express it as:

$$P(B_{ij_p}) = \prod_{j_p=1}^{J_p} \mu_{ij_p}, \quad (7.23)$$

where parameter $\mu_{ij_p} \geq 0$ and $\sum_{j_p=1}^{J_p} \mu_{ij_p} = 1$, $\forall i, p$ because it is a probability. Then we may write the multinomial distribution of data B over μ

$$P(B | \mu) = \prod_{p=1}^P \prod_{j_p=1}^{J_p} \mu_{j_p}^{B_{j_p}}, \quad (7.24)$$

where $\mu = \{\mu_1, \dots, \mu_{j_p}, \dots, \mu_{J_p}\}$. It is easy to see that this distribution Eq. (7.24) is the generalisation of Eq. (7.15).

Here again as in previous cases we write mixture model of K components of multinomial distributions for sample B_i :

$$P(B_i | \mu, \alpha) = \sum_{k=1}^K \alpha_k P_k(B_i | \mu_k), \quad (7.25)$$

where $\mu = \{\mu_{1j}, \dots, \mu_{kj}, \dots, \mu_{Kj}\}$ and the probability of B_i given μ_k is

$$P_k(B_i | \mu_k) = \prod_{p=1}^P \prod_{j_p=1}^{J_p} \mu_{kj_p}^{B_{ij_p}}. \quad (7.26)$$

The logarithm of the likelihood function (under assumption that B are *i.i.d.*):

$$\log P(B | \mu, \alpha) = \log \prod_{i=1}^I P(B_i | \mu, \alpha) = \sum_{i=1}^I \log \left\{ \sum_{k=1}^K \alpha_k P_k(B_i | \mu_k) \right\} \quad (7.27)$$

Parameters of multinomial distribution Eq. (7.26) are estimated through the maximisation of its likelihood function [Bishop, 2006].

As before weight w_{ik} or the conditional probability that the sample B_i belongs to class k is:

$$w_{ik} = \frac{\alpha_k P_k(B_i | \mu_k)}{\sum_{l=1}^K \alpha_l P_l(B_i | \mu_l)}. \quad (7.28)$$

EM-algorithm is applied as in previous sections but to mixture of multinomial distributions. The expected number of points N_k of the component k is:

$$N_k = \sum_{i=1}^I w_{ik}, \quad (7.29)$$

and weights of mixture model is:

$$\alpha_k = \frac{N_k}{N}. \quad (7.30)$$

Mean value μ_{kj_p} of mixture component k is:

$$\mu_{kj_p} = \frac{1}{N_k} \sum_{i=1}^I B_{ij_p}. \quad (7.31)$$

Then the EM-algorithm for the Multinomial mixture model (MMM) is given in **Algorithm 7.3**.

Here again, the same problems arise as in its previous section when EM-algorithm is used: local optimal results, dependence of on initialisation, selection of the best model and estimation of the number of mixture components. They can be solved via model estimation by MDL criterion Eq. (6.28).

If we do not have a priori information about parameters of the multinomial mixture model μ_k Eq. (7.31) and α_k Eq. (7.30) and K we may estimate them several times (for different initialisations) and select the optimal values in the sense of MDL Eq. (6.28). At the beginning we should fix the number of mixture components K . One of the simplest way to initialise α_k Eq. (7.30) is $\alpha_k = 1/K$ or randomly with respect to conditions. To obtain stable classification results we may run EM-algorithm several times and each time we randomly initialise parameters μ_k Eq. (7.31) and/or α_k Eq. (7.30). The best selected model has the lowest value of log-likelihood function of MMM.

Changing the number of components K for MMM and estimating parameters of MMM for each run we can estimate K or the complexity of the model by any information criterion, e.g., MDL Eq. (6.28). We obtain the curve with maxima (or minima, it

Algorithm 7.3 Pseudo code of EM-algorithm for the multinomial mixture model

1: Initialise K means μ_k and α_k .

2: **E-step**

2.1: Calculate probabilities $P_k(B_i | \mu_k)$ as in Eq. (7.26):

$$P_k(B_i | \mu_k) = \prod_{p=1}^P \prod_{j_p=1}^{J_p} \mu_{kj_p}^{B_{ij_p}}$$

2.2: Calculate weights w_{ik} as in Eq. (7.28):

$$w_{ik} = \frac{\alpha_k P_k(B_i | \mu_k)}{\sum_{l=1}^K \alpha_l P_l(B_i | \mu_l)}$$

4: **M-step** Re-estimate parameters N_k Eq. (7.29), μ_k Eq. (7.31) and α_k Eq. (7.30)

4.1:
$$N_k = \sum_{i=1}^I w_{ik}.$$

4.2:
$$\mu_{kj_p} = \frac{1}{N_k} \sum_{i=1}^I B_{ij_p}.$$

4.3:
$$\alpha_k = \frac{N_k}{N}.$$

5: Evaluate the log-likelihood function:

$$\log P(B | \mu, \alpha) = \sum_{i=1}^I \log \left\{ \sum_{k=1}^K \alpha_k P_k(B_i | \mu_k) \right\}$$

6: **If** log-likelihood is converged,

then stop,

else go to **Step 2**.

depends on the sign) indicating the optimal K . Here again we should properly select the number of free parameters to correctly penalise the model complexity in MDL Eq. (6.28): the dimension of data should be equal to P .

From a practical point of view, the combination of clustering results by the mixture model of Bernoulli distributions or multinomial distributions give very often the same results. But we should differ BMM from MMM, because MMM is generalisation of BMM. Thus, MMM is preferred to BMM for clustering combination.

7.4 Combination using a co-association matrix

In this section we propose to survey some solutions for clustering combination using a co-association matrix. We also present new methods for clustering combination to avoid disadvantages of existing approaches. The idea of the proposed combination is to group samples which are in the same cluster in most cases. Firstly, we propose an objective function to combine different clustering results. Then we develop a hierarchical algorithm to optimise the objective function. Such an algorithm is competitive compared to any other algorithms of combination but in spite of its very good results it does not guarantee the

convergence to a global solution. After analysis of the objective function we propose an improved method which gives a global solution. Moreover we describe conditions for such a convergence.

It is interesting to note, that such a grouping may be expressed as the minimisation of the square error between samples presented by labels of clusterings. We prove in this section that the global solution of the minimum square error may be found using the gradient estimation of a density function. All the locally optimal modes of the density form groups of samples and consequently constitute a global solution of the combination. One of the advantages of such a method is that the proposed algorithm has a fast convergence and a near linear complexity. It is an important advantage when a great amount of data is to be processed as in the case of satellite image processing. The combination of clusterings is performed on synthetic and real databases. The effectiveness of the proposed method and its superiority with respect to other combination approaches are demonstrated.

Problem statement

Let us consider the case where we have a large set of samples and different clustering methods, each of them providing a partition of the sample set into a specific number of clusters. As before let I be the number of samples and P the number of clusterings. Each clustering (denoted with index $p, p = 1, \dots, P$) associates each sample u with one and only one cluster.

The elementary co-association matrix A^p collects information on which sample v belongs to the same cluster as u :

$$A_{uv}^p = \begin{cases} 1, & \text{if } u \text{ and } v \text{ are in the same cluster,} \\ 0, & \text{otherwise.} \end{cases} \quad (7.32)$$

Therefore A^p is a binary symmetric square matrix of size I .

We may similarly describe the p^{th} clustering by partition matrix B^p defined in Eq. (7.1) with I rows and J_p columns, where J_p equals the number of clusters in the p^{th} clustering, $u = 1, \dots, I, j = 1, \dots, J_p$.

We verify that:

$$A^p = B^p B^{p'}, \quad (7.33)$$

where $'$ denotes the matrix transposition.

For the P clusterings, we can compute the average matrix A as:

$$A = \frac{1}{P} \sum_{p=1}^P A^p = \frac{1}{P} \sum_{p=1}^P B^p B^{p'}. \quad (7.34)$$

A is the global co-association matrix or, in short, the co-association matrix. For large P , we may say that two elements u and v have a probability A_{uv} to belong to the same cluster.

Let us denote B^s the consensus partition, *i.e.* a partition of the samples which reflects at best the point of view of every clustering. Our goal is to obtain such a consensus partition B^s from the co-association matrix A . From B^s , we may compute a square matrix D of size I as:

$$D = B^s B^{s'}. \quad (7.35)$$

Such a matrix D would be the binary co-association matrix corresponding to the consensus classification. For any problem, where P different clusterings are performed, we may observe one matrix A , but the consensus partition B^s is unknown as well as D . The purpose of the clustering combination is to derive these unknown matrices.

Several different matrices D could be obtained, depending on the criterion chosen to derive D from A . For instance, in [Fred & Jain, 2005] the matrix D is obtained from A by maximising an information theoretic criterion (normalised mutual information NMI criterion). We propose here to formulate the solution as the one which minimises the square error between D and A :

$$E = \|D - A\|^2, \quad (7.36)$$

which may be rewritten (since D is binary) as:

$$E = \sum_{u=1}^I \sum_{v=1}^I \left(\sum_{r=1}^I (B_{ur}^s B_{rv}^{s'}) - A_{uv} \right)^2 = \sum_{u=1}^I \sum_{v=1}^I D_{uv} (1 - 2A_{uv}) + \sum_{u=1}^I \sum_{v=1}^I A_{uv}^2, \quad (7.37)$$

subject to $B^{s'} B^s = \mathbf{I}$, with $\sum_i \mathbf{I}_{ii} = I$ and $B_{uv}^s \in \{0, 1\}$,

where \mathbf{I} is a diagonal matrix of size I with diagonal elements equal to the cluster sizes. The proposed quadratic objective function Eq. (7.37) has a convex form for all possible consensus clusterings, contrary to a mutual information criterion as proposed in [Fred & Jain, 2005; Strehl & Ghosh, 2002]. Therefore it may be solved exactly by efficient methods.

Eigen vector decomposition

It can be noted that In Equation (7.37), $\sum_u \sum_v A_{uv}^2$ is a constant, which plays no role in the minimisation.

Let Q denote a square matrix with element $Q_{uv} = (1 - 2A_{uv})$. As $D = B^s \cdot B^{s'}$ using a matrix trace and its cyclic property, from (7.37) we have now to minimise:

$$\sum_u \sum_v D_{uv} Q_{uv} = \text{Tr}(D \cdot Q) = \text{Tr}(B^s \cdot B^{s'} \cdot Q) = \text{Tr}(B^{s'} \cdot Q \cdot B^s) \quad (7.38)$$

Let $\bar{Q} = -Q$, then the best clustering, which minimises Eq. (7.37) is given by:

$$\max \text{Tr}(B^{s'} \cdot \bar{Q} \cdot B^s) \quad (7.39)$$

with a constraint on the norm of B^s : $\text{Tr}(B^s \cdot B^{s'}) = I$.

The relaxed solution of this problem could be derived from a projection matrix $U = V \cdot \sqrt{|\Lambda|}$, where $\bar{Q} = V \cdot \Lambda \cdot V'$ is the eigen decomposition of the matrix $\bar{Q} = 2A_{uv} - 1$, V is a diagonal matrix of eigen values and Λ is a matrix with eigen values. The matrix B^s has 1 for maximal element of each row of U and 0 for the rest of them. Also using the same substitution the matrix B^s can be obtained from the eigen decomposition of A . With the same notations $A = V \cdot \Lambda \cdot V' = (V \cdot \sqrt{\Lambda}) \cdot (V \cdot \sqrt{\Lambda})'$, where $B^s = V \cdot \sqrt{\Lambda}$.

Discussion: The theoretical solution proposed here is a good solution when matrix Q reflects a perfect clustering. In this case, we just have U different column vectors for Q . The eigen-space is therefore of the same dimension as U , each eigen-vector of B^s having as a multiplicity order the number of samples in the corresponding clustering. Matrix B^s (resulting from the eigen-vector decomposition) is of size P .

When Q is not perfect, the decomposition has more than P non-zero eigen vectors. We may expect the P largest eigen values correspond to optimal solution and the following ones correspond to noise. Therefore we will just discard the $I - P$ smallest eigen-vectors. But nothing guarantees that the obtained B^s has binary vectors, and the decision issue (which sample belongs to one of the vector of B^s) may be not true. Moreover it is not guaranteed that it is positive, a necessary condition to reflect the membership of a sample to the corresponding cluster. For these reasons, there is no warranty that eigen vectors will produce a desired solution; this method may lead to wrong results. This approach is discussed in [Shawe-Taylor & Cristianini, 2004].

Bounds of square error E

The low bound of square error E Eq.(7.37) is 0 and it is achievable when the "consensus" clustering is one of the given clusterings and all they are equal. When clusterings are different then the lower bound is not equal to 0. The bound of error E (7.37) can be calculated [Marcotorchino & El ayoubi, 1991; Benhadda & Marcotorchino, 1998]. Let us analyse the theorem of Hoffman-Wielandt:

If two symmetrical matrices A Eq.(7.34) and D Eq.(7.35) of size $I \times I$ have ordered eigen values $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_I$ and $\gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_I$, respectively, then:

$$\|A - D\|_F^2 \geq \sum_{i=1}^I (\lambda_i - \gamma_i)^2, \quad (7.40)$$

where $\|A - D\|_F^2 = \sum_{u=1}^I \sum_{v=1}^I (A_{uv} - D_{uv})^2$ is a Forbenius norm and $u, v = 1, \dots, I$.

Then using Eq. (7.37) we write inequality (7.40) as:

$$\sum_{u=1}^I \sum_{v=1}^I A_{uv}^2 - 2 \sum_{u=1}^I \sum_{v=1}^I D_{uv} A_{uv} + \sum_{u=1}^I \sum_{v=1}^I D_{uv}^2 \geq \sum_{i=1}^I \lambda_i^2 - 2 \lambda_i \gamma_i + \gamma_i^2. \quad (7.41)$$

As $\sum_{u=1}^I \sum_{v=1}^I A_{uv}^2 = \sum_{i=1}^I \lambda_i^2$ and $\sum_{u=1}^I \sum_{v=1}^I D_{uv}^2 = \sum_{i=1}^I \gamma_i^2$ then we obtain:

$$\sum_{u=1}^I \sum_{v=1}^I D_{uv} A_{uv} \leq \sum_{i=1}^I \lambda_i \gamma_i. \quad (7.42)$$

Let C_j be the j^{th} consensus cluster having n_j samples, where $\sum_{j=1}^J n_j = I$ and $j = 1, \dots, J$. Then we may rewrite inequality 7.42 as:

$$\sum_{u=1, u \in C_j}^I \sum_{v=1, v \in C_j}^I A_{uv} \leq \sum_{j=1}^J \lambda_j \gamma_j \quad (7.43)$$

or

$$\frac{1}{n_j} \sum_{u=1, u \in C_j}^I \sum_{v=1, v \in C_j}^I A_{uv} \leq \sum_{j=1}^J \lambda_j, \quad (7.44)$$

because $\gamma_i = n_j$ for $i, j = 1, \dots, J$ and $\gamma_i = 0$ for $i = J + 1, \dots, I$.

We see that inequality (7.40) indicates the lower bound of the error E (7.37). More over inequality (7.44) shows how good is consensus clustering $\{C_j\}$.

Cholesky decomposition

Another side of this problem can be viewed from the properties of matrix B^s . A desired solution for the square matrix B^s is a lower triangular matrix. It is clear that column permutation of B^s does not change the matrix D . We use this property to represent B^s in the form of a lower triangular matrix:

1. We permute the first column in B^s with a column, which has a "one" in the first row.
2. Then we search the first zero in the first column and a position of one in such a row.
3. After we permute the second column of B^s with a column in which we have found the one. We repeat this procedure by searching the first zero in a current column and permuting the next column with a column that has a one in the row that corresponds to the found zero.

An example is shown in Figure (7.1). The matrix B^s of size 8×4 corresponds to the assigning of 8 elements to 4 clusters.

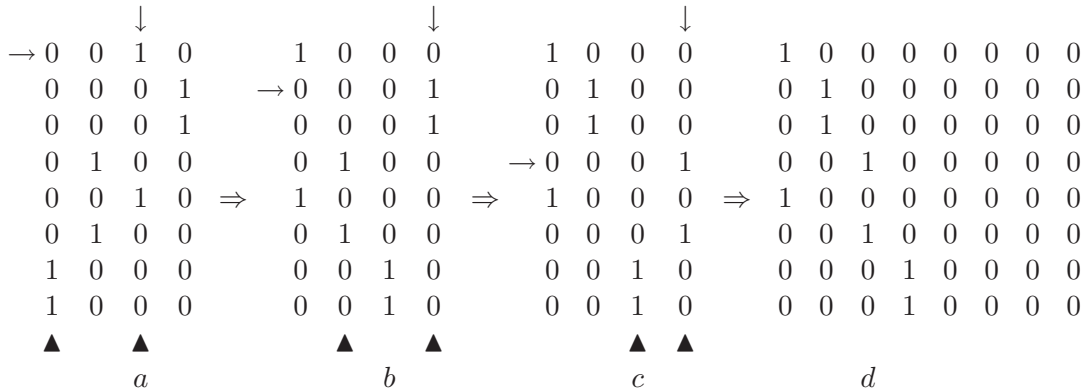


Figure 7.1: Matrix permutations to obtain a Choleski matrix. a - permutation of first and third columns, b - permutation of second and fourth columns, c - permutation of third and fourth columns, d - permuted lower triangular matrix B^s with added zeros.

We see that the first row of a left matrix in Figure 7.1 has 1 in a third column. We permute first and third columns Figure (7.1a). Then in the first column the first 0 is at the second row in which 1 is at the fourth position. We permute second and fourth columns Figure (7.1b). At the next step the second column has first zero below a diagonal at the

forth position and in this row 1 is at the forth position, so we permute third and forth columns Figure 7.1c. A final result of permutation is in Figure (7.1d). We see that our matrix B^s can be presented as the lower diagonal matrix.

An example of a co-association matrix is in Figure 7.2

$$A^1 = B^s \cdot B^s = \begin{matrix} & \begin{matrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{matrix} \end{matrix}$$

Figure 7.2: Co-association matrix A associated to the partition matrix B^s in Figure 7.1

Cholesky decomposition of a square matrix A solves the equation $A = B \cdot B^{s'}$ with a minimal L_2 error that corresponds to $\min E$, under the condition that A is positive definite or semi-definite. But, for the problem at hand, as for the previous case, this solution does not ensure that B^s has positive values. At last, the result of Cholesky decomposition of the matrix A is very dependent on lines and columns permutations.

Quadratic programming

Problem 1 When looking for B^s as an upper triangular matrix, we need to search only $I(I+1)/2$ elements of this matrix. If all these elements are arranged in single column vector b^s [Pratt, 2001] the problem (7.38) could be rewritten as a constrained quadratic problem:

$$\begin{aligned} \min & b^{s'} \cdot Q^* \cdot b^s \\ \text{subject to} & b^{s'} \cdot b^s = I \text{ and } 0 \leq b^s \leq 1 \end{aligned} \tag{7.45}$$

where the Q^* matrix is constructed by block diagonal concatenation of the Q matrix and by deleting rows and columns which correspond to zero elements of the upper triangular part of B^s .

Unfortunately, for real applications, e.g., image classification, the complexity issue becomes dominant: it is very cumbersome to work with a matrix Q^* of size $(I(I+1)/2)^2 = O(I^4)$, when I is of the order of 10^3 .

Problem 2 As matrix A_{uv} is symmetric, then $A_{uv} = A'_{uv}$ and $Q = Q'$. With these constraints our problem becomes:

$$\begin{aligned}
& \min \quad \text{Tr}(B^{s'} \cdot Q \cdot B^s) \\
& \text{subject to } u \cdot B^s \leq v, \quad v = (I, \dots, 1), \\
& \quad \quad \quad B^s \cdot u' = u', \quad u = (1, \dots, 1), \\
& \quad \quad \quad B^s = \{0, 1\} \text{ or } 0 \leq B^s \leq 1
\end{aligned} \tag{7.46}$$

The Lagrange function for 7.46 is:

$$L(B^s, \mu, \eta) = \text{Tr}(B^{s'} \cdot Q \cdot B^s) + (u \cdot B^s - v) \cdot \mu' + \eta \cdot (B^s \cdot u' - u') \tag{7.47}$$

where μ and η are vectors of Lagrange multipliers.

The Karush-Kuhn-Tucker conditions for (7.46) to have a minimum are:

$$\begin{aligned}
\frac{\partial L}{\partial B^s} &= 2Q \cdot B^s + u' \mu + \eta' u = 0, \\
\frac{\partial L}{\partial \mu} &= u \cdot B^s - v = 0, \\
\frac{\partial L}{\partial \eta} &= B^s \cdot u' - u' = 0, \\
B^s, \mu, \eta &\geq 0
\end{aligned} \tag{7.48}$$

Discussion: Problems 1 and 2 given above have non-convex quadratic formulation because matrices Q^* and Q are nondefinite (they have positive and negative eigen-values). Non-convex quadratic problems is very difficult to solve, however there exists methods of quadratic optimisation designed to find a local minimum of (7.45) [Floudas & Visweswaran, 1994].

Proposed solution

Combination algorithm

In order to combine clusterings and find B^s that minimises E Eq. (7.37) we propose to use a single-link merging algorithm Jain & Dubes [1988]. This algorithm has been experimentally shown to give very good results when compared to other hierarchical algorithms such as average-link, Ward, complete-link, etc., Fred & Jain [2005]. The motivation of using single-link algorithm is based on the previous remark that the general term A_{uv} of matrix A may be considered as the probability of two samples to belong to the same cluster. Of course we do not know the memberships of u and v and the actual number of clusters J , but it is reasonable to group in the same cluster elements of A that have the highest probability of coassociation, that is the way single-link works Jain & Dubes [1988]. We propose the Least Square Error Combination (LSEC) algorithm for solving Eq. (7.37) (see Algorithm 1). The optimal number of clusters J is found when the error E in Eq. (7.37) is minimum. At the first step we initialise B^s as the identity matrix supposing that each cluster has only one sample. Error $E^{(1)} = I^2$ is initialised to have its

Algorithm 7.1: Pseudo code of *LSEC*-algorithm

-
- 1: Set B^s as the identity matrix, $J \leftarrow I$, $i \leftarrow 1$ and $E^{(i)} \leftarrow I^2$.
 - 2: Find clusters' indexes $(j, k) = \arg \max_{u \in j, v \in k} A_{uv}$; $j, k = 1, \dots, J$, $j \neq k$.
 - 3: Set $B^* \leftarrow B^s$.
 - 4: Merge two clusters j and k by $B_{uj}^s \leftarrow (B_{uj}^s + B_{uk}^s)$.
 - 5: Remove column k from matrix B^s .
 - 6: $E^{(i+1)} \leftarrow \sum_{u=1}^I \sum_{v=1}^I \left(\sum_{j=1}^J (B_{uj}^s B_{vj}^s) - A_{uv} \right)^2$.
 - 7: **if** $E^{(i+1)} \leq E^{(i)}$, **then**
 - 8: $i \leftarrow i + 1$,
 - 9: $J \leftarrow J - 1$,
 - 10: go to **Step 2**;
 - 11: **else** $B^s \leftarrow B^*$, B^s is the optimal partition, stop.
-

maximal value. A partition presented by matrix B^s is stored to matrix B^* before merging two clusters. Merging is continued till minimising error $E^{(i)}$.

Simulated example

In order to demonstrate the efficiency of this algorithm, it has been experimented on synthetic noisy data. The experiment was carried out on a data set of 100 samples with $J_p = 5$ classes each of which has 20 samples. 25% of class labels were randomly changed according to a uniform noise. Matrix B^p Eq. (7.1) is constructed for each of $P = 100$ noisy clusterings. Matrix A Eq. (7.34) was estimated by B^p , $p = 1, \dots, P$. The accuracy of combination is the mean accuracy of each class in percentage. The accuracy of each class is the relative number of points of a cluster with the maximal size obtained after the combination. Figure 7.3 shows comparison of *LSEC*-algorithm with *NMI* criterion for *single – link* algorithm [Fred & Jain, 2005]. For 100 noisy clusterings, *LSEC*-algorithm's

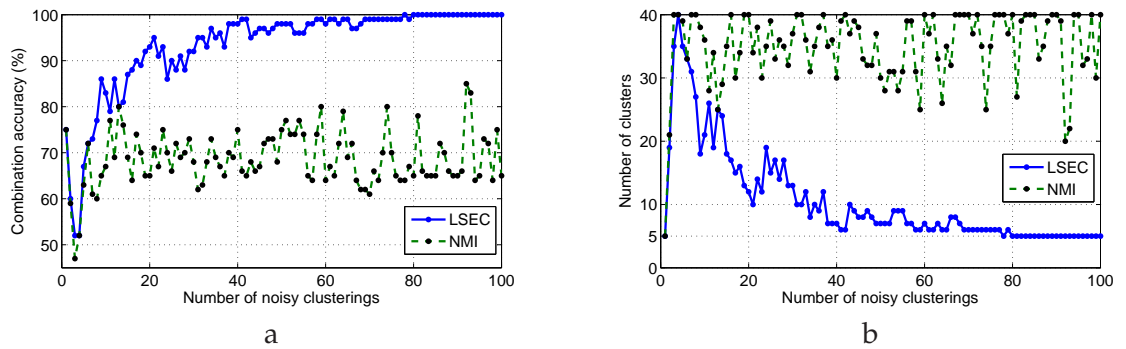


Figure 7.3: Combination comparison of *LSEC* algorithms and *NMI* criterion with *single – link* algorithms. a - Combination accuracy for number of noisy clusterings, b - Estimated number of clusters as a function of the number of noisy clusterings

accuracy is 100%, contrary to *NMI* criterion with about 70% of the accuracy. in Figure

7.3a for the first clustering the accuracy of combination is 75% because it considers one clustering with 25% of noise. Then accuracy falls down for three noisy clusterings and after grows up for *LSEC* but for *NMI* criterion it remains at most the same. Figure 7.3b shows that the accuracy also concerns the estimation of the cluster number. We see that for a large number of noisy clusterings the accuracy of *LSEC*-algorithm to determine the cluster number is good when it fails for the *NMI* criterion.

Matrix A is computed in $I(I - 1)/2$ iterations. To combine clusters, I iterations are needed and error E is calculated in $I(I - 1)/2$ iterations for each combination. The time complexity of such an algorithm is therefore in $O(I^3)$. It is not appropriate for high volume of data. To overcome this problem we propose an efficient initialisation procedure (in Section 7.4) as well as an optimisation of the algorithm adapted for high volume of data (in Section 7.4).

Approximate solution. Initialisation

One of the simplest ways to go towards a minimum of Equation (7.37) is a gradient like method, which starts from a good initialisation and iteratively modifies B^s that improves (7.37). A better initialisation is likely to accelerate the convergence. A good initialisation may be the eigen vector decomposition limited to the K first eigen vectors. Another initialisation may be the one B^p , among all the clusterings, which were used to build matrix A , provides the clustering with the minimum error:

$$B^s = \min_{B^p} \left\{ \sum_{u=1}^I \sum_{v=1}^I \left(\sum_{r=1}^I (B_{ur}^p B_{rv}^{rp}) - A_{uv} \right)^2, p = 1, \dots, P \right\} \quad (7.49)$$

A gradient like method which iteratively modifies B^s and minimises the error E Eq. (7.37) will also be considered. Suppose an elementary step of optimisation consists in allocating sample q to cluster j instead of its initial cluster j_0 . Let B^{j_0} and B^j be the partition matrices before and after this allocation. The variation of E Eq. (7.37) is given by:

$$\Delta E(q|j_0 \rightarrow j) = \sum_{u=1}^I \sum_{v=1}^I (D_{uv}^j - D_{uv}^{j_0})(1 - 2A_{uv}), \quad (7.50)$$

where $D^i = B^i B^{i'}$ and $D^{j_0} = B^{j_0} B^{j_0'}$ as in Eq. (7.35).

The change is accepted if and only if $\Delta E(q|j_0 \rightarrow j)$ is not positive, and the process is iterated until no change improves E .

Let us consider two partitions B^{j_0} and B^j where the second partition B^j is obtained by affectation of sample q of B^{j_0} to the cluster j .

Then the variation of E depends only on the difference between D_{uv}^j and $D_{uv}^{j_0}$, Figure 7.5.

j_0										j									
1	0		1	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
1	0		1	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
\rightarrow	0	1	0	0	1	1	1	1	1	1	1	\rightarrow	1	0	1	1	1	0	0
0	1		0	0	1	1	1	1	1	1	1	0	1		0	0	0	1	1
$B^{j_0} =$	0	1	$D_{uv}^{j_0} =$	0	0	1	1	1	1	1	1	$B^j =$	0	1	$D_{uv}^j =$	0	0	0	1
0	1		0	0	1	1	1	1	1	1	1	0	1		0	0	0	1	1
0	1		0	0	1	1	1	1	1	1	1	0	1		0	0	0	1	1
0	1		0	0	1	1	1	1	1	1	1	0	1		0	0	0	1	1
0	1		0	0	1	1	1	1	1	1	1	0	1		0	0	0	1	1
a			b							c		d							

Figure 7.4: Influence of affectation. a - a partition matrix B^{j_0} , b - a co-association matrix D^{j_0} , c - a partition matrix B^j , d - a co-association matrix D^j .

$$D_{uv}^j - D_{uv}^{j_0} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 7.5: The difference $D_{uv}^j - D_{uv}^{j_0}$ for the variation of E

We can see from Figure 7.5, ΔE Equation (7.50) could be rewritten as:

$$\begin{aligned} \Delta E(q|j_0 \rightarrow j) &= \sum_{u=1}^I \sum_{v=1}^I (D_{uv}^j - D_{uv}^{j_0})(1 - 2A_{uv}) = \\ &= \sum_{u=1}^I \sum_{v=1}^I D_{uv}^j(1 - 2A_{uv}) - \sum_{u=1}^I \sum_{v=1}^I D_{uv}^{j_0}(1 - 2A_{uv}) = \\ &= 2 \sum_k (1 - 2A_{qk}) - 2 \sum_l (1 - 2A_{ql}) \end{aligned} \quad (7.51)$$

where index k denotes samples in cluster j except sample q , and l denotes samples of cluster j_0 except sample q .

The looked for cluster j having one sample k should minimise $2(1 - 2A_{qk})$ but it is equivalent to finding the maximum of A_{qk} for every possible cluster $j = 1, \dots, J$. All diagonal elements of matrix A are maxima and have 1, that is why we find maximum of A_{qk} for each q excepting diagonal elements. Using nonpositiveness condition for the error variation $\Delta E(q|j_0 \rightarrow j) \leq 0$ Eq. (7.51) we write the necessary condition to examine

points A_{qk} :

$$\begin{aligned}\Delta E(q|j_0 \rightarrow j) &\leq 0 \\ 2(1 - 2A_{qk}) &\leq 0 \\ 0.5 &\leq A_{qk}\end{aligned}\tag{7.52}$$

The Condition (7.52) means that two points could be combined if they are in the same cluster more than in a half cases. This optimisation procedure is equivalent to building nearest-neighbour sub graphs. It avoids the storage of the square matrix A . It is very important when processing a large amount of data. Points belonging to each sub graph are assigned to the same cluster. Such clusters will form now on the initialisation matrix B^s for LSEC-algorithm (instead of the identity matrix) resulting in a noticeable gain of computation time. We note that this combination is a local optimum for the general criteria Equation (7.37). An alternative solution is to use a simulated annealing strategy which will accept a change of cluster for a sample, even if E is not decaying, but according to a probability which slowly goes to zero when iterating the process. It would be, of course, far more expensive in the sense of computational time.

Gradient descent optimisation and storage reduction

In proposed **Algorithm 7.1** matrix A should be computed at **Step 4**. This step may be difficult for real applications such as images or large database clustering, because of the dimension of matrix A . For instance, when processing images, we often have to deal with thousands of pixels. For an image of size $n \times n$ (thus with n^2 samples), we have to build a matrix A of size $n^2 \times n^2$, i.e. with n^4 terms. For example, with a small image of 200x200 pixels we should construct a co-association matrix that has 1.6×10^9 elements; with 1 byte per term we should process about 1.49 gigabytes at each combination step. This huge volume of data can not be processed in a reasonable time. However, we can find the solution for this problem in analysing the error of combination (Equation 7.37).

Instead of calculating the error at each step of the optimisation procedure, we suggest using the optimisation error gradient as proposed in Eq. (7.50), and follow a descending approach as an optimisation strategy. The error gradient reduces the computation time as well as the volume of stored and processed data.

Let k and l be indexes of samples belonging to two clusters j_0 and j respectively with n_{j_0} and n_j samples each. Let $D^{j_0} = j_0 j_0' + j j'$ be the binary co-association matrix before combination and $D^j = (j_0 + j)(j_0 + j)'$ after combination. All elements of D^j either are equal to 1 or to 0. Matrices D^{j_0} and D^j are displayed respectively in Figure 7.6. Their difference $D_{uv}^j - D_{uv}^{j_0}$ is in Figure 7.7.

Let look first at a simple example. Suppose we want to find the error gradient after a combination of two clusters j_0 and j . Let $D^{j_0} = j_0 j_0' + j j'$ and $D^j = (j_0 + j)(j_0 + j)'$ be the co-association matrices before and after the combination. They are displayed in Figure 7.6, respectively. Their difference $D_{uv}^j - D_{uv}^{j_0}$ is in Figure 7.7.

Let E^{j_0} and E^j be errors as in Eq. (7.37) before and after combination.

We obtain the difference ΔE between errors E^j and E^{j_0} by substituting matrices D^{j_0}

j_0	j									$j_0 + j$							
1	0		1	1	1	0	0	0	0	0	1		1	1	1	1	1
1	0		1	1	1	0	0	0	0	0	1		1	1	1	1	1
\rightarrow 1	0		1	1	1	0	0	0	0	0	\rightarrow 1		1	1	1	1	1
0	1		0	0	0	1	1	1	1	1	1		1	1	1	1	1
$B^{j_0} = 0$	1	$D_{uv}^{j_0} =$	0	0	0	1	1	1	1	1	$B^j = 1$	$D_{uv}^j =$	1	1	1	1	1
0	1		0	0	0	1	1	1	1	1	1		1	1	1	1	1
0	1		0	0	0	1	1	1	1	1	1		1	1	1	1	1
0	1		0	0	0	1	1	1	1	1	1		1	1	1	1	1
a						b					c				d		

Figure 7.6: Influence of affectation. a and b - a partition matrix B^{j_0} , and its co-association matrix $D_{uv}^{j_0}$, c and d - a partition matrix B^j , after having merged the two clusters j_0 and j , and its co-association matrix D^j .

$$D_{uv}^i - D_{uv}^{j_0} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 7.7: The difference $D_{uv}^j - D_{uv}^{j_0}$ for the variation of E due to the merging of the two clusters j_0 and j

and D^j in Eq. (7.50):

$$\begin{aligned}
E^{j_0} &= \sum_u^I \sum_v^I D_{uv}^{j_0} Q_{uv}, & E^j &= \sum_u^I \sum_v^I D_{uv}^j Q_{uv}, \\
\Delta E &= E^j - E^{j_0} = \sum_u^I \sum_v^I D_{uv}^j Q_{uv} - \sum_u^I \sum_v^I D_{uv}^{j_0} Q_{uv} \\
&= \sum_u^I \sum_v^I (D_{uv}^j - D_{uv}^{j_0}) Q_{uv} = \sum_u^I \sum_v^I ((j_0 + j)(j_0 + j)' - (j_0 j_0' + j j')) Q_{uv} \quad (7.53) \\
&= \sum_u^I \sum_v^I (j_0 j_0' + j_0 j' + j j_0' + j j - j_0 j_0' - j j') Q_{uv} \\
&= \sum_u^I \sum_v^I (2j_0 j') Q_{uv} = 2n_{j_0} n_j - 4 \sum_k^{n_{j_0}} \sum_l^{n_j} A_{kl}.
\end{aligned}$$

A new condition for subcluster combination is obtained from the condition that the

error gradient is non positive $\Delta E \leq 0$ and $Q_{uv} = 1 - 2A_{uv}$:

$$\begin{aligned} \Delta E &= 2n_{j_0}n_j - 4 \sum_k^{n_{j_0}} \sum_l^{n_j} A_{kl} \leq 0 \\ n_{j_0}n_j &\leq 2 \sum_k^{n_{j_0}} \sum_l^{n_j} A_{kl} \\ 0.5 &\leq \frac{\sum_u^I \sum_v^I A_{uv}}{n_{j_0}n_j} \end{aligned} \quad (7.54)$$

Property (7.54) states that two subclusters j_0 and j are combined if the sum of their connection probabilities is greater than a half of all possible connections of their points. We say that the normalised sum of their connections is greater than 0.5. The last term in the gradient ΔE Eq. (7.53) allows us to calculate a double sum without storage of whole matrix A .

A complete iterative algorithm

Now let use the results presented in Section 7.4 which provide a good initialisation of the algorithm by an initial clustering based on nearest neighbour graphs. Let J^g be the number of these initial clusters. From J^g , a binary matrix B^g is built according to Eq. (7.1) and a single matrix $\mathbf{B} = [B^1, \dots, B^p]$ as a concatenation of B^p . A is derived from Eq. (7.34) as:

$$A = \frac{1}{P} \mathbf{B} \mathbf{B}'. \quad (7.55)$$

Matrix S of size $J^g \times J^g$ can be computed as the sum of connections between all pairs of J^g clusters:

$$S = B^{g'} A B^g = \left(\frac{B^{g'} \mathbf{B}}{\sqrt{P}} \right) \left(\frac{B^{g'} \mathbf{B}}{\sqrt{P}} \right)'. \quad (7.56)$$

Let each element N_{kl} of a matrix N correspond to the number of all possible connections of two clusters k and l :

$$N_{kl} = n_k n_l, \quad (7.57)$$

where $k, l = 1, \dots, J^g$ and n_k, n_l are the numbers of samples in clusters k and l , respectively. The normalised sum of connections between two clusters k and l allows building matrix \bar{S} where each element \bar{S}_{kl} is expressed as:

$$\bar{S}_{kl} = S_{kl} / N_{kl}, \quad (7.58)$$

with $0 \leq \bar{S}_{kl} \leq 1$. From matrix \bar{S} we may propose a generalisation of condition (7.54): if $\bar{S}_{kl} \geq 0.5$, clusters k and l should be combined to reduce the error E Eq. (7.37) for *LSEC*-algorithm. Ranking \bar{S}_{kl} elements in descending order indicates the order in which two clusters should be grouped at **Step 2**. This algorithm, called *DLSEC* (differential *LSEC*) significantly reduces computations and may be applied to large volumes of data.

We compare calculation cost for a direct search presented in Section 7.4 with an optimised search proposed in Sections 7.4 to 7.4 in Figure 7.8. An ideal clustering with $J_p = 6$ clusters is taken as an example. Random changes of labels are performed on 20% of the samples. By repeating this procedure 100 times we construct the matrix B^c . Then we

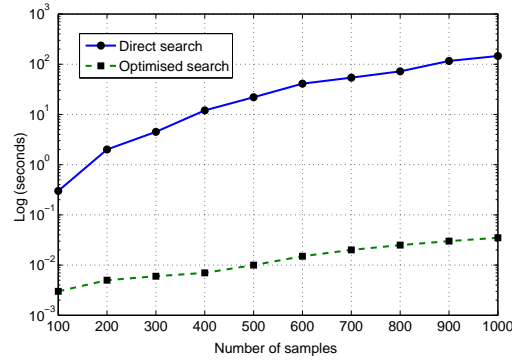


Figure 7.8: Time computation for direct and optimised search vs. the number of samples, in the case of synthetic data.

measure the time needed for the combination by direct and optimised search. As can be seen from Figure 7.8, the proposed optimised search decreases significantly the computational time. Moreover, after combination of noisy clusterings, a perfect clustering with 6 classes is always obtained. The bootstrapping method [Kuncheva, 2004] may be one of the possible applications of the *DLSEC*-algorithm. For the experiment, we set randomly 60% of samples with initial clustering labels and 40% as unclassified labels for which we attributed the same label. After 100 steps of boosting the combination returns the initial clustering. It could be one of the issues for a parallel clustering of huge amounts of data or for improving clustering.

To compute J^g clusters of the nearest neighbour graph for the initialisation of *DLSEC* algorithm as described in Section 7.4 $I(I-1)/2$ operations at most are needed. The combination of these clusters as presented in Section 7.4 needs $J^g - 1$ operations, where $J^g \ll I$. The time complexity of optimised *DLSEC*-algorithm is approximately $O(I^2 + J^g)$. Note, that *DLSEC* method only needs about $O(I^2)$ operations at most for the complete optimisation compared to the method in [Lange & Buhmann, 2005] which requires $O(I^2)$ operations at each step of the optimisation process. Moreover, *DLSEC*-algorithm can have a linear complexity if we consider, for a sample, the classes of its nearest-neighbours as in image processing applications. We introduced the objective function and the optimised hierarchical algorithm to find the optimal consensus clustering. Unfortunately there is no clear proof that the hierarchical algorithm may achieve a global optimum of the objective function. To overcome this limitation we reformulate the optimisation process as well as the optimality conditions and propose an exact algorithm to find the global optimum for E Eq. (7.37).

Examples of combining

Eigen decomposition, standard hierarchical methods In this Section we demonstrate a toy example of clustering combination. We test methods of combination on to producing the correct number of classes and the correct classes from the only examination of the co-association matrix.

We will make use of a *smooth* co-association matrix A for experiments. Smoothness for A means that this matrix differs from a perfect clustering which only have zeroes and ones, i.e. square internal blocks. Co-association matrix A should be symmetric with elements satisfying $0 \leq A_{uv} \leq 1$ and $A_{uu} = 1$. Moreover, matrix A is

formed by a matrix B , see Eq. (7.33). To create smooth matrix A we use smooth matrix B where each column equal to a one dimensional function of Gaussian distribution (Figure 7.9). To ensure having terms between 0 and 1, we normalise matrix A as suggested in [Shawe-Taylor & Cristianini, 2004]:

$$A_{uv} = \frac{A_{uv}}{\sqrt{A_{uu}A_{vv}}} \quad (7.59)$$

Therefore, the obtained matrix A verifies all the previous requirements.

An example of columns of matrix B and corresponding matrix A are presented in Figures 7.9a and 7.9b, respectively. We apply our algorithm to find binary matrix B^s that

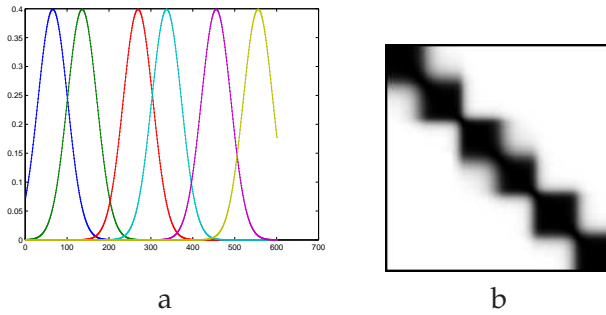


Figure 7.9: A matrix B made of 6 clusters and about 600 samples. a - columns of matrix B : each column has a Gaussian distribution, b - the normalised matrix $A = BB'$

gives the minimal error E Eq. 7.37 as well as others hierarchical algorithms: complete-link algorithm, Ward algorithm and unweighted average distance (UPGMA) algorithm (Chapter 5). For the last 3 algorithms we analyse the error E Eq.(7.37) at every level of the clustering tree. For each algorithm and each level of hierarchy we build E as a function depending on the number of clusters (Figure 7.10).

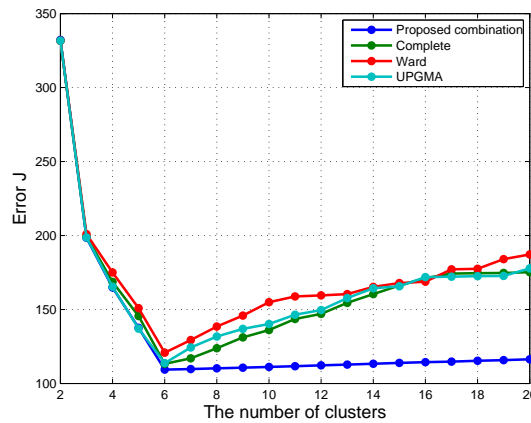


Figure 7.10: Error E as a function of the number of clusters for several clustering algorithms. The exact number of clusters (6) is found by all algorithms.

All algorithms find the correct number of clusters (6) in Figure 7.10. Confusion matrices of each hierarchical clustering are presented in Table (7.2). We see from the confusion

matrix in Table 7.2a that the proposed combination method has no error: each cluster corresponds to the true class. On the contrary, all the other algorithms have some confusions.

Classes						
1	2	3	4	5	6	
101	0	0	0	0	0	101
0	101	0	0	0	0	101
0	0	101	0	0	0	101
0	0	0	93	0	0	93
0	0	0	0	109	0	109
0	0	0	0	0	96	96
101	101	101	93	109	96	601

a

Classes						
1	2	3	4	5	6	
101	7	0	0	0	0	108
0	94	2	0	0	0	96
0	0	92	0	0	0	92
0	0	7	93	5	0	105
0	0	0	0	104	4	108
0	0	0	0	0	92	92
101	101	101	93	109	96	601

b

Classes						
1	2	3	4	5	6	
88	0	0	0	0	0	88
13	101	2	0	0	0	116
0	0	99	1	0	0	100
0	0	0	92	9	0	101
0	0	0	0	88	0	88
0	0	0	0	12	96	108
101	101	101	93	109	96	601

c

Classes						
1	2	3	4	5	6	
100	0	0	0	0	0	100
1	101	2	0	0	0	103
0	0	84	0	0	0	84
0	0	15	93	0	0	108
0	0	0	0	109	4	113
0	0	0	0	0	92	92
101	101	101	93	109	96	601

d

Table 7.2: Combining of co-association matrix A by different methods. a - The proposed combination, b - Complete link algorithm, c - Ward algorithm, c - Unweighted average distance (UPGMA) algorithm. Method (a) is only one to provide a clustering without error.

Eigen vectors and values In Section 7.4 we show possible approaches to solve Equation (7.37) using eigen vector decomposition. In this Section we present examples for these methods. Let use the same co-association matrix A as in previous experiment. Then the relaxed solution for A is a projection matrix $P = V \cdot \sqrt{|\Lambda|}$, where V and λ are the eigen vectors and eigen values of either $\bar{Q} = 2A - 1$ or A .

Let see eigen values for the matrix $\bar{Q} = 2A - 1$: $\Lambda_{1,1} = 257.8$, $\Lambda_{2,2} = 232.9$, $\Lambda_{3,3} = 170.4$, $\Lambda_{4,4} = 149.1$, $\Lambda_{5,5} = 119.9$, $\Lambda_{6,6} = 5.7$, $\Lambda_{601,601} = -334.8$ and for the matrix A : $\Lambda_{1,1} = 139$, $\Lambda_{2,2} = 128.9$, $\Lambda_{3,3} = 114.9$, $\Lambda_{4,4} = 84.5$, $\Lambda_{5,5} = 73.7$, $\Lambda_{6,6} = 60$. We see that the sum of eigen values equals to the number of elements $\sum_i \Lambda_{ii} = 601$ and the number of positive values corresponds to the number of clusters. It can be supposed that each positive value Λ_{ii} corresponds to the number of points in each cluster. Then using this information it could be possible to find clusters. Let B^s be a partition of data. We get matrix B^s by setting 1 for maximal element of each row of P and 0 for the rest of them. Examples of projection P for matrices $\bar{Q} = 2A_{uv} - 1$ and A are presented in Figure (7.11a) and (7.11b), respectively.

We build matrix B using two projections in Figure 7.11. The confusion matrices for different partitions are in Tables 7.3a and 7.3b, respectively. Confusion matrices expected

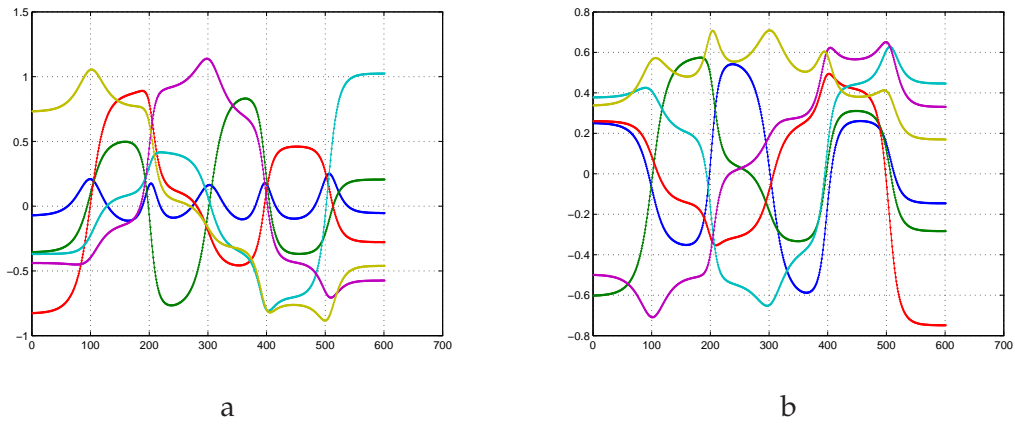


Figure 7.11: Projection on eigen spaces. 6 first vectors of the matrix P . a - for matrix $2A - 1$, b - for matrix A

<i>Classes</i>						
1	2	3	4	5	6	
101	45	0	0	0	0	146
0	56	3	0	101	0	160
0	0	98	39	0	0	137
0	0	0	54	5	0	59
0	0	0	0	3	1	4
0	0	0	0	0	95	95
101	101	101	93	109	96	601

a

<i>Classes</i>						
1	2	3	4	5	6	
0	50	0	0	0	0	50
67	0	0	0	0	95	162
0	0	0	0	108	1	109
34	51	101	93	1	0	280
101	101	101	93	109	96	601

b

Table 7.3: Confusion matrices for the matrix P . a - eigen solution for $2A - 1$, b - eigen solution for A .

to be diagonal as in Tables 7.2. Diagonal form means a correct combination. But we observe in Tables 7.3a and 7.3b, high confusion that means clusters are not well separated. As we can see eigen values do not coincide to the number of samples. Moreover, positive Λ_{ii} could decrease slowly that will make difficult to determine the correct number of clusters. These obstacles influence on a result of combination and can lead to unpredictable results. Therefore, we do not consider direct application of eigen decomposition as an effective method.

7.5 Proposed Mean Shift combination

In this section P clusterings are considered as labels coded by p binary matrices B^p Eq. (7.1), where $p = 1, \dots, P$. The matrices are concatenated into a single matrix \mathbf{B} and form space \mathbb{R}^d , where $d = \sum_{p=1}^P J_p$. We propose to search a consensus clustering which, as previously, minimises the square error E Eq. (7.37). We prove in this section that this minimisation is equivalent to the minimisation of the square error among samples b_u , where b_u is a row of \mathbf{B} and $u = 1, \dots, I$.

All samples $\{b_u\}$ are located on a hyper circle, since they simultaneously satisfy a hyper plane equation $\sum_{j=1}^d b_{uj} = d = \text{const}$ and a hyper sphere equation $\sum_{j=1}^d b_{uj}^2 = d = \text{const}$. Therefore vectors $\{b_u\}$ may be normalised by a constant $\sqrt{\sum_p J_p}$ such that their square norm is 1. Let us write the minimisation of square error E Eq. (7.37) as:

$$\begin{aligned} \min_{B^s} E &= \min_{B^s} \sum_{u=1}^I \sum_{v=1}^I D_{uv} (1 - 2A_{uv}) = \min_{J, C_j} \sum_{j=1}^J \sum_{u \in C_j} \sum_{v \in C_j} (1 - 2A_{uv}) = \\ &= \min_{J, C_j} \sum_{j=1}^J n_j^2 \left(1 - \frac{2}{n_j^2} \sum_{u \in C_j} \sum_{v \in C_j} A_{uv}\right) \end{aligned} \quad (7.60)$$

where unknown consensus clusters C_j has corresponding binary matrix B^s , where $j = 1, \dots, J$ and J is the unknown number of consensus clusters. Unknown cluster C_j has the unknown number of samples n_j .

The matrix B^s has a size of $I \times J$. As all elements verify $0 \leq A_{kl} \leq 1$ and $A_{uv} = b_u b'_v$ we may derive a condition to guarantee that the error Eq. (7.60) is always minimised:

$$\|b_u - b_v\|^2 < 1 : u, v \in C_j \Rightarrow \frac{1}{n_j^2} \sum_{u \in C_j} \sum_{v \in C_j} A_{uv} > 0.5. \quad (7.61)$$

This condition shows the expression in the parenthesis of the last part of Eq. (7.60) is always negative. We may also say that if during the estimation of consensus clusters C_j the condition (7.61) is hold and the number of samples n_j is growing then error E Eq. (7.60) is always minimised.

Proving convergence with mean shift

Let μ_j be the mean vector of cluster C_j , $\mu_j = \sum_v b_v / n_j$, $v \in C_j$. The square norm of μ_j is:

$$\|\mu_j\|^2 = \frac{1}{n_j^2} \left\| \sum_v b_v \right\|^2 = \frac{1}{n_j^2} \sum_v (\|b_v\|^2 + 2 \sum_u b_v b'_u) = \sum_{u \in C_j} \sum_{v \in C_j} A_{uv} / n_j^2. \quad (7.62)$$

The square error σ_j^2 of cluster C_j with mean μ_j is:

$$\sigma_j^2 = \frac{1}{n_j} \sum_{u=1}^{n_j} \|b_u\|^2 - \left\| \frac{1}{n_j} \sum_{u=1}^{n_j} b_u \right\|^2 = 1 - \|\mu_j\|^2. \quad (7.63)$$

where $\|b_u\|^2 = 1$. Minimising the last term in Eq. (7.60) is equal to maximising both μ_j and the number of samples n_j in any cluster j .

Proposition 7.5.1. *A global minimum of the error E Eq. (7.60) is achieved by an optimisation algorithm which maximises the norms of local mean vectors μ_j Eq. (7.62) or/and minimises square errors σ_j^2 Eq. (7.63) jointly with maximising the number of samples n_j in clusters:*

$$\begin{aligned} \min E &= \min_j n_j^2 (1 - 2\|\mu_j\|^2) = \min_j n_j^2 (2\sigma_j^2 - 1), \\ \text{under conditions } \|\mu_j\|^2 &> 0.5, \sigma_j^2 < 0.5. \end{aligned} \quad (7.64)$$

A nonparametric approach to find a solution is the goal of near all information processing tasks. The base of such an approach in regard to the pattern recognition is the nonparametric density estimation by its gradient [Fukunaga, 1990], [Comaniciu & Meer, 2002], so-called the density estimation by mean shift vectors.

The multivariate kernel density estimation with kernel $K(b)$ and window radius h , computed in the point b has a form [Fukunaga, 1990]:

$$\hat{f}(b) = (Ih^d)^{-1} \sum_{u=1}^I K(h^{-1}(b - b_u)) \quad (7.65)$$

For such an estimation an appropriate kernel should be selected to approximate the density and if the kernel has unknown parameters they should also be estimated. One of the popular kernels is the Gaussian kernel with the width of the kernel window [Comaniciu, 2003] as parameter. This kernel is not appropriate for the problem at hand because it makes the assumption that the more data are available the denser the distribution. In the case of normalised samples $\{b\}$ the higher the number of samples does not guarantee the higher density. We aim to group samples $\{b\}$ which are located on the different positive axes.

We propose to use the multivariate Epanechnikov kernel which yields the minimisation of the average global error between the estimate and the true density [Comaniciu et al., 2000]:

$$K_E(\mathbf{b}) = \begin{cases} \frac{(d+2)}{2c_d} (1 - \|\mathbf{b}\|^2), & \text{if } \|\mathbf{b}\| \leq 1, \\ 0, & \text{otherwise.} \end{cases} \quad (7.66)$$

where c_d is the volume of the unit d -dimensional sphere of radius 1 and $\mathbf{b} = \{b\}$. The profile of kernel K_E is the function $k_E: [0, \infty) \rightarrow R$ such that $K(\mathbf{b}) = k(\|\mathbf{b}\|)$:

$$k_E(b) = \begin{cases} \frac{(d+2)}{2c_d} (1 - b), & \text{if } b \leq 1, \\ 0, & \text{otherwise.} \end{cases} \quad (7.67)$$

The density estimation Eq. (7.65) is obtained via its gradient [Comaniciu & Meer, 2002]:

$$\hat{\nabla} f_{h,K}(b) = \frac{2c_{k,d}}{Ih^{d+2}} \left[\sum_{i=1}^I k \left(\left\| \frac{b-b_i}{h} \right\|^2 \right) \right] \left[\frac{\sum_{i=1}^I b_i k \left(\left\| \frac{b-b_i}{h} \right\|^2 \right)}{\sum_{i=1}^I k \left(\left\| \frac{b-b_i}{h} \right\|^2 \right)} - b \right]. \quad (7.68)$$

The second term in Eq. (7.68) is the mean shift:

$$\mathbf{m}_{h,k}(b) = \frac{\sum_{i=1}^I b_i k \left(\left\| \frac{b-b_i}{h} \right\|^2 \right)}{\sum_{i=1}^I k \left(\left\| \frac{b-b_i}{h} \right\|^2 \right)} - b, \quad (7.69)$$

which expresses the difference between point b and the mean of the samples weighted by kernel $k(\cdot)$. It also shows the direction in which the density is increasing and where the weighted mean value should be replaced. The *mean shift* estimation always converges and proceeds in two steps [Comaniciu & Meer, 2002]: (i) compute the mean shift vector $\mathbf{m}_{h,k}$; (ii) move kernel $k(b)$ by $\mathbf{m}_{h,k}$.

Let us note two very important properties of mean shift algorithm applied to data $\{b\}$.

Property 1. All $\{b\}$ vectors have positive values, consequently the cosine between successive mean shift vectors always remains positive [Comaniciu & Meer, 2002], guaranteeing a fast and good rate of convergence and we have never a chaotic descent.

Property 2. As the mean shift algorithm converges [Comaniciu & Meer, 2002] and all data $\{b\}$ have values from a finite set, the mean shift estimation of μ_j is obtained in a finite number of iterations. In practice, the iteration number for convergence is very small (some units).

Condition (7.61) to achieve a global minimum of error E Eq. (7.64) shows that the maximal distance among samples $\{b_u\}$ is less than 1. From this condition, the distance from mean vector μ_j to any point of cluster j is less than 1. The Epanichnekov kernel is differentiable in a sphere of radius 1; therefore optimisation converges to a global optimum [Comaniciu et al., 2000]. We demonstrate a theorem which asserts the global optimality of Epanechnikov kernel to minimise E Eq. (7.64).

Theorem 7.5.1. Epanechnikov kernel is the optimal kernel to find a global minimum for error E Eq. (7.64) by the mean shift algorithm.

Proof. See Appendix D. □

We also mention that reclustering labels by K-means algorithm after mean shift combination does not decrease error E Eq. (7.64) of combination and even may increase it.

Optimal adaptive radius for mean shift combination

We proved in Appendix D that the mean shift combination with the Epanechnikov kernel finds an optimal solution for error E Eq. (7.60). Because starting point is a data sample $\mu_j = b_i$ the threshold is chosen as 1 (7.61) satisfying the condition of the Epanechnikov kernel with a radius 1. As $\|\mu_j\|^2$ is changed during the search, an optimal radius should be estimated. Condition (7.61) shows that the optimal solution of the error Eq. (7.60) is found when $A_{uv} > 0.5$. In such a case using the square norm of mean vector μ_j Eq. (7.62)

calculated on n_j samples and the worst case when $A_{uv} = b_u b'_v = 0.5 : u \neq v; A_{uu} = 1$, then $\|\mu_j\|^2 = (0.5n_j(n_j-1) + n_j)/n_j^2 = 0.5(1 + 1/n_j)$. To optimise the error E Eq. (7.60) the optimal adaptive radius r_j (or similarly the minimal distance from any sample $b_u : u \notin j$ to the mean vector μ_j) should be:

$$r_j = \sqrt{\|b_u - \mu_j\|^2} = \sqrt{1 - 2b_u \sum_{v \in j} b_v/n_j + \|\mu_j\|^2} = \sqrt{\|\mu_j\|^2} = \sqrt{0.5(1 + 1/n_j)} \quad (7.70)$$

This formula shows when $\mu_j = b_v$ then $r_j = 1$ that satisfies Eq. (7.61) and :

$$\lim_{n_j \rightarrow \infty} r_j = \lim_{n_j \rightarrow \infty} \sqrt{\|\mu_j\|^2} = \lim_{n_j \rightarrow \infty} \sqrt{0.5(1 + 1/n_j)} = \sqrt{0.5} \approx 0.7071. \quad (7.71)$$

From this limit we obtain a low bound for the square norm of the mean vector $\mu_j : 0.5 < \|\mu_j\|^2$. This value always guarantees the minimisation of error E Eq. (7.60). We may now present the algorithm of the mean shift combination (MSC) with Epanechnikov kernel and adaptive radius r_j .

MSC-algorithm

Initialise $j = 1, l = 1, c_i = 0 : i = 1, \dots, I$

Step 1 Initialise $r_j = 1, k = 1, y_k = b_j$

Step 2 Compute $y_{k+1} = \frac{1}{n_k} \sum_{b_i \in W(y_k, r)} b_i$,
 $r_k = \sqrt{0.5(1 + 1/n_k)}$,
 $k \leftarrow k + 1$ till convergence.

Step 3 Assign $r_j = r_k, c_i = l, \forall i : \sqrt{\|b_i - y_{conv}\|^2} < r_j, l = l + 1; j : c_j \equiv 0$. Go to **Step 1**.

where n_j is the number of points in the window $W(y_k, r_j)$ of radius r_j with centre y_k and c_i has labels after the combination.

Practical aspects of mean shift

In this section we give some notes on practical application of the mean shift algorithm for clustering combination. The following aspects are discussed:

1. accelerating the mean shift via appropriate initialisation,
2. assigning samples to mean vectors,
3. computation of error E ,
4. merging mean shift vectors.

Accelerating the mean shift via appropriate initialisation

A classical version of mean shift algorithm consists in application of the iterative procedure by starting from every point of data set. When the algorithm is applied to a large data set, e.g., image samples, then the direct application may be time consuming. There are several ways to avoid examining of all starting points:

1. after the mean shift has converged, then we may select points from the radius of the converged mean vector and do not consider them for further computation. If any point converge to the converged radius then it belongs to the converged mode.
2. Another way to reduce computation time may be in excluding points from the radius of converged mean shift. After the mean shift has converged, then run it on the reduce data set.

In addition, we start mean shift from samples potentially belonging to large clusters. This allow us to find quickly a large cluster, eliminate it from data set and process run algorithm on the reduced data set. The simplest way to select potentially large cluster is to find a cluster with the largest size in all clusterings. Another way to select the large cluster may be in using the entropy as a measure of clustering consistency. The entropy is calculated for each clustering:

$$H_p = - \sum_{k=1}^{J_p} \frac{n_k}{I} \log \frac{n_k}{I}, \quad (7.72)$$

where $p = 1, \dots, P$. Clustering p which has the minimal value of H_p can be used to find the largest cluster. Then, a sample from the largest cluster is taken as starting point for mean shift algorithm.

The next way to select a starting point from a potentially large cluster is: find all unique clusters as intersection of clusterings. A sample from the largest unique cluster is a good candidate for initialisation. This approach is interesting when clusterings are not very different. However, a procedure of finding all intersections can have a quadratic computation complexity when clusterings very different (e.g., noisy or bootstrapped).

Assigning of samples to mean vectors

Some practical problem emerge when the mean shift algorithm is applied on the nominal data (clusterings or segmentations) and when data are not really randomly and independently distributed. Sometimes there are two different converged points with overlapped radiuses. The problem is the following: how to decide which points belong to which mean vector? In the case of the continuous kernel, e.g., the Gaussian kernel, we may compute the probability of points belonging to every converged mode and assign points via the Bayes classification rule. In the case of a truncated kernel, e.g., the Epanechnikov kernel, the probability is calculated not for all samples of given modes. There are several solutions to assign a sample to one of the mean vectors:

1. combine into one cluster those samples which share some samples belonging to different modes (it gives robust clustering combination result);
2. assign samples to the nearest neighbour mode;

Computation of error E

To improve combination by the mean shift algorithm we propose to consider separately the estimation of parameters (means vectors, in our case) and assigning labels to samples. This influences directly the quality of combination calculated using square error E . It means that the error is calculated using the square norms of the estimated mean vector and the number of samples assigned to this vector, contrary to direct computation

of square error E , when only samples from a spherical window are used. This aspect comes from the fact that the estimated mean vector may go far from the original starting point. Therefore, we consider all starting points converged to the same mean vector as belonging to the same combined cluster.

Merging of mean shift vectors

Another very important aspect of the mean shift algorithm presented in the thesis is that during the convergence there are mean shift vectors which are very near from each other. We propose to combine these neighbour vectors using the next simple rule:

1. if the converged mean vector contains in its radius another converged mean vector and
2. if after their combination the norm of the new vector is greater then 0.5,

then these two vectors are combined and the new mean vector is reestimated. This combination guarantees to minimise the square error E .

Results

Synthetic clustering combination

In this Section we present different clustering combination criteria and algorithms on synthetic data. To generate simulations we take one clustering and exchange randomly samples from true clusters to false ones. From these clusterings, several classes are extracted by different methods: the hierarchical *single-link*, *Ward* and *average-link* algorithms [Jain & Dubes, 1988] with the average normalised mutual information *NMI* [Fred & Jain, 2005]; our *LSEC*-algorithm and *MSC*-algorithm for square error E Eq. (7.37) as presented in this paper, and *AUTOCLASS* clustering system [Cheeseman & Stutz, 1996] that cluster labels by mixtures of multinomial models with Expectation-Maximisation algorithm.

The first example is made for 2 classes each of size 50 samples. 25% randomly selected samples are changed to other random value of labels. Each labelling is represented as binary matrix B Eq. (7.1). We collect 100 of such noise labelings and construct co-association matrix A Eq. (7.32). Figure 7.12 shows two criteria to determine the optimal number of clusters: for *NMI* [Fred & Jain, 2005] the optimal number has highest value and for square error E Eq. (7.37) the lowest value indicates the optimal number of clusters. For such an elementary example *NMI* criterion grows up with a growing number of clusters for all hierarchical algorithms, contrary to error E Eq. (7.37) which provides always the true number of clusters in all cases. *AUTOCLASS* gives the true solution, but for this system we should indicate a priory number of clusters and set a large number (near 100) of restarting to find a good solution. It is well known that all clustering systems that are based on *EM*-algorithm do not guarantee a global solution and the best solution is selected using restarting (*e.g.*, with random parameter initialisation). Proposed *LSEC*-algorithm as well as (*MSC*)-algorithm gives 2 clusters without errors. For experiments with a high number of clusters and noisy labels error E Eq. (7.37) indicates more precisely the true number of clusters than average *NMI*.

UCI data

We perform experiments with the clustering combination on real datasets taken from

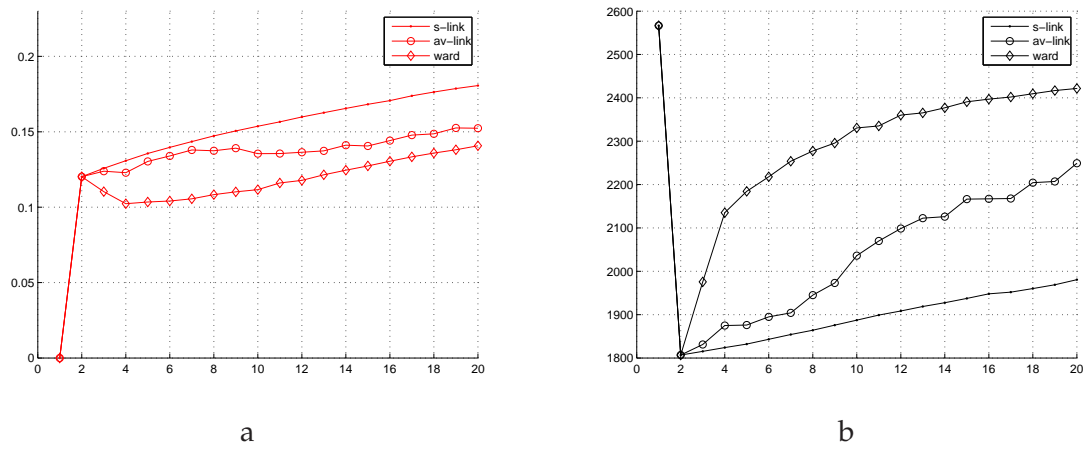


Figure 7.12: Combination of clusterings with 30% of noised labels by single-link (points), average-link (circles) and Ward (diamonds) hierarchical algorithms: a - NMI criterion, b- square error E Eq. (7.37).

the UCI machine learning repository and compare results with the work of [Fred & Jain, 2005] where the normalised *NMI* criteria is studied. The goal of these experiments is to show that the proposed combination algorithms are competitive and may even outperform averaged *NMI* criterion in [Jain & Dubes, 1988].

Real data from UCI repository are the same as in [Fred & Jain, 2005]: 1. Iris data (150 samples, 4D); 2. Breast Cancer (683 samples); 3. Optical Digits (3823 samples); 4. Log yeast (384 samples); 5. Std Yeast (384 samples).

To obtain clusterings of data we use *K-means* algorithm for fixed and random number of clusters. The fixed number of clusters k^* is the "natural" known number of classes and the random number is chosen randomly near k^* . After the combination we estimate its quality as the percentage of misclassified samples. The largest number of samples in a combined class was set as the true and all other samples in this class are set as misclassified. The minimum value of this error is used to indicate the best clustering for 100 random initialisations of *K-means* algorithm.

LSEC-algorithm, *MSC*-algorithm and *AUTOCLASS* (AC) were used to combine different clusterings and their results are compared to the best Evidence Accumulation Clustering (EAC) with single-link or average-link approaches (EAC-SL, EAC-CL), Table 3 and Table 2 in [Fred & Jain, 2005] for fixed and random k^* , respectively. Here again for *AUTOCLASS* combination we should always set a priori number of clusters and a large number of restartings to obtain a good solution. Results of combination of clusterings is presented in Table 7.4 as error rates of classification (in percentage).

From Table 7.4 we see that the best individual clustering obtained by *K-means* algorithm (column 3) in most cases leads to less errors comparing to combined results of *NMI* [Fred & Jain, 2005] (column 4). Also *LSEC* and *MSC* algorithms have lower errors (columns 6,7) in most cases compared to *NMI* criterion. *AUTOCLASS* (column 5) justifies good performance of *LSEC* and *MSC* algorithms with near the same error. The same values of error for Iris and Breast Cancer data explained by the fact that such data have small size that is why clusterings as well as combinations are the same.

In addition *MSC*-algorithm outperforms *LSEC*-algorithm. We observe that in several

Table 7.4: Error (in percentage) of the clustering combination

Data set	Fixed k^*						Variable k^*			
	k^*	KM(min)	Jain	AC	LSEC	MS	Jain	AC	LSEC	MS
Iris	3	10.7	11.1	10.7	10.7	10.7	10.0	10.0	10.0	10.0
Brest Cancer	2	3.9	4.0	3.9	3.9	3.9	2.9	2.9	2.9	2.9
Optical Digits	10	13.1	23.2	17.3	17.1	15.7	21.0	11.8	11.1	10.5
Log Yeast	5	58.6	66.6	59.4	58.8	58.8	59.0	49.2	52.8	50.3
Std Yeast	5	26.1	31.8	31.2	32.8	32.5	33.0	26.5	26.8	26.3

cases MSC-algorithm has significantly lower values of the clustering errors than NMI [Fred & Jain, 2005] (less than 7.8% for Log Yeast and 7.5% for Optical Digits). An effect that the best $K - means$ clustering error less than several combinations for fixed k^* is explained by the fact of presence of many low quality clusterings.

The second set of experiments was done with a varying number of clusters, where K-means was initialised randomly. Columns 8, 9, 10 and 11 of Table 7.4 show clustering errors after the combination by Jain, *AUTOCLASS*, *LSEC* and *MSC* algorithms, respectively. In such a kind of experiments with the combination we find "stable" clusters instead of the natural clusters. That is why estimated numbers of clusters k' may differ from a priory known k^* . Here again, we see that performance of proposed combination algorithms (columns 10 and 11) is still very good and better than EAC-SL or EAC-AL (column 7, Table 2) in [Fred & Jain, 2005] Table 2. Interesting, that in [Fred & Jain, 2005] there is no definitive decision about what algorithm of combination is the best. Experiments on synthetic examples as well as real data bases show better performance of our combination algorithms than in [Fred & Jain, 2005]. In addition, proposed approaches have near linear complexity and may process a huge volume of data.

Discussions

In Sections 7.4 and 7.5, two efficient optimisation algorithms for the combination of optimal clusterings have been proposed. They avoid the use of any parameter, does not depend on initialisation, determines the number of clusters in an unsupervised way and significantly reduces redundant information. We showed the objective function and conditions for its optimisation. The first method uses single-link algorithm to find an optimal solution. This algorithm has been chosen experimentally because of its good results compared to other hierarchical algorithms. But it does not guarantee the convergence to a global optimum. To avoid this problem a new combination approach is proposed based on the mean shift procedure. It has been proved in Section 7.5 that mean shift minimises the square error between clusterings, achieves a global optimum and has a linear complexity. These methods are able to process very large sets of samples, without facing problems of memory or time complexity. The combination of different clusterings is able to improve unsupervised data mining and infer new information about data.

Clustering combination makes possible using different clustering algorithms. In this way we can compare and process different metrics which are not comparable. If several algorithms are used to analyse data, then it is better to estimate the optimal number of clusters for each algorithm, taking into account its metric.

The combination may be used for many different applications of data mining tasks:

clustering of nominal data (e.g. text documents), combination of different clusterings or segmentations of the same scene (e.g. by clustering different groups of features or clustering time-series images), video clustering, motion detection, etc. It also may stabilise clustering result for an algorithm which depends on the choice of initial parameters.

7.6 Measure of clustering stability, stable patterns

One of the important and interesting questions for clustering algorithm is a *measure of clustering stability*. Some discussions about clustering stability can be found in [Kuncheva, 2004]. Under *the stability of clustering* we consider the measure which defines how samples share the same cluster with others samples. The notion of *clustering stability* includes *the stability related to one data* and *the stability of the whole clustering data set*. *The stability related to one data* describes the quantitative ability of one sample to share the same cluster with others samples. We define this measure as S_i for the i^{th} sample using the matrix A Eq.7.34:

$$S_i = \frac{2}{I} \sum_{v=1}^I |A_{iv} - 0.5| \quad (7.73)$$

The stability of clustering of data is the mean value of stability of all data points:

$$S = \frac{1}{I} \sum_{i=1}^I S_i \quad (7.74)$$

Measures S_i and S are positive, real and limited from above by 1 and from below by $1/I$. If we do not take into account diagonal elements of A for S_u and normalise them by $I - 1$ in the Eq.(7.73), then S_i and S is in the range of $[0, 1]$. Measures S_i and S do not depend on any metric of algorithm and even do not depend on an algorithm used for clustering. They are based only on the clustering labels. Using the measure of *clustering stability* we can estimate and select stable points for several clusterings of the same algorithm. The more higher value of S_i the more stable point i . Also it is possible to compare different clustering algorithms and select the one which gives more stable clustering results.

Measures S_i and S considers the stability of points to be in the same cluster as well as the stability of points from other clusters. We can calculate how stable a sample is within its cluster. Let have one clustering and a set of clusterings presented by matrix A . With little modification of Equations (7.73) we define measures for sample i in cluster k as S'_i :

$$S'_i = \frac{1}{\#k} \sum_{v \in k} A_{iv} \quad (7.75)$$

where $\#k$ is the number of samples in cluster k . Then *the clustering stability S' of data* is:

$$S' = \frac{1}{I} \sum_{i=1}^I S'_i. \quad (7.76)$$

As it has been shown in this Chapter clustering combination corresponds to minimisation of square distances E Eq. (7.63) for every combined cluster k . This distance depends on the square norm of the mean vector μ_k Eq. (7.62). Therefore, the index of stability can be used directly from the equation (7.63). It has the following interpretation:

the lower the square distance of cluster k Eq. (7.63) the stable cluster k . If all clusterings are the same in cluster k , then σ_k equals to zero. For unstable clusterings the limit of stability is $\sigma_k = 1 - \frac{1}{I}$. If the number of samples I is very high then the upper bound equals to 1. Therefore, stability is limited as: $0 \leq \sigma_k < 1$.

Examples of stable patterns and clustering stability

We perform experiments on clustering stability using one clustering coded as binary matrix B . The clustering has 6 classes 100 examples per class. The matrix A is obtained using matrices B_i , $i = 1, \dots, 100$. Each matrix B_i is the noised B . The noise is a random changing a cluster label of a given sample. The intensity of noise is measured in percents, e.g. 5% of noise change randomly 5% of sample.

Examples of A for 10% and 30% of noise are in Figure 7.14a and in Figure 7.14b, respectively.

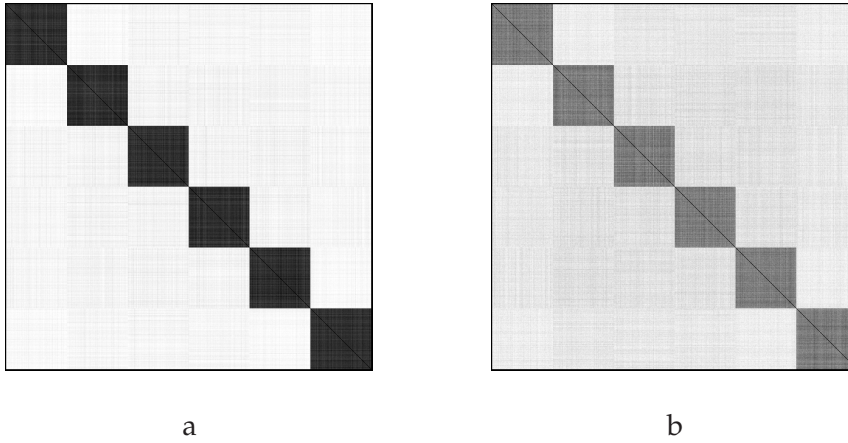


Figure 7.13: Matrix A for different noise intensity. a - 10% of noise, b - 30% of noise

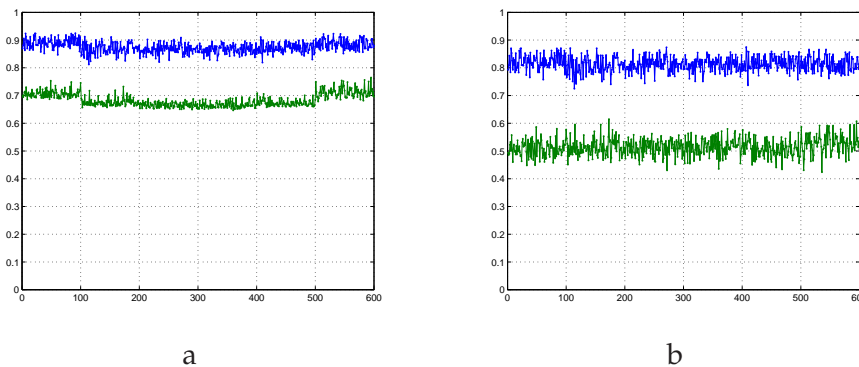


Figure 7.14: Clustering stability. a - measure S_i for 10% of noise (green) and S_i for 30% of noise (blue), b - measure S'_i for 10% of noise (green) and S'_i for 30% of noise (blue). Stability of data clustering for 10% of noise is $S = 0.87$ and for 30% of noise is $S = 0.68$. Stability of data clustering within clusters for 10% of noise is $S' = 0.81$ and for 30% of noise is $S' = 0.51$

Stable patterns have maximal values of S_i or S'_i . We can use this stability for the tasks of image analysis and data mining.

Let us to have maps and corresponding satellite or/and aerial images. Maps have been made by interpretation of the Earth surface. Image features give supplementary information about the surface. We can cluster images such that the clustering looks like a map. From the maps and the clustering we can select stable clusters (in the sense of measures S' Eq.(7.76) or E Eq.(7.37)). Moreover, changing the number of clusters in the clustering algorithm we could build a curve of S' or E . The optimal value on these curves indicate the clustering which is near to the maps.

Here are several advantages of this schema compared to supervised classification. Firstly, new clusters could be found in an unsupervised way. Secondly, there is no need to calculate any distance between the image feature space and maps. In addition, the optimal number of clusters is estimated using only clustering labels. Various examples of clustering combination are presented in Chapter 8.

Self-optimising effect

In this section we show results of experiments about a *self-optimising effect* of the proposed combination method. Under the *self-optimising effect* we mean optimisation by the method of combination a parameter which is not supposed to be optimised at the beginning. This parameter is the optimal number of clusters of data. It has been observed from experiments that this number after combination tends to the true number of data clusters.

Experiments have been performed on data shown in Figure 7.15a. Data have 16 Gaussians with the same covariance matrix where each Gaussian has 100 points. This is the simplest illustrative example.

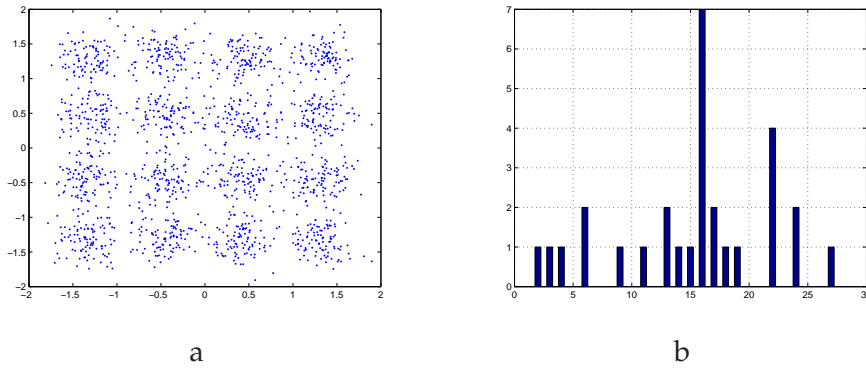


Figure 7.15: Matrix A for different noise intensity. a - 10% of noise, b - 30% of noise

We run the clustering algorithm, e.g., *K-means*, for every number of clusters from 2 to 30. Then we build a histogram as a function of the number of clusters to show how many clusters have been obtained after combination.

The histogram is in Figure 7.15b. We see that the maximum value of the histogram corresponds to the true number of clusters 16. Number 7 shows how many times 16 clusters have been estimated by combination. This illustrative example shows an interesting property of combination: it is able to estimate true clustering.

7.7 Conclusions

In this Chapter the problem of clustering combination has been considered. Previous works on this problem have been reviewed. Several recent algorithms of the clustering combination need some parameter tuning. Clustering combination has been presented through the clustering of clusterings. The simplest combination has been obtained by K-means algorithm applied to binary representation of clusterings. Equivalence of different measures has been illustrated (vector product, Euclidean and Hamming distances).

More complex modelling of combination based on the probabilistic approach has also been considered. In this case clusterings are considered as nominal data and are modelled either by mixtures of Bernoulli or multinomial models. Estimation of model parameters is done by EM-algorithm. The best model for mixtures can be chosen by MDL criteria. It has been noted that probabilistic mixtures suffer from random initialisations of the model parameters which yields to different result of clustering combination.

Their disadvantages motivated us to state the problem of combination in an unsupervised way. The problem statement is based on the co-association matrix. The measure of square distances between a consensus clustering and given clusterings has been used. Two algorithms to optimise this criterion have been proposed. The first algorithm is a hierarchical one and the second one is iterative. Despite of the good performance of the hierarchical algorithm there is no proof that it may achieve the global optimal solution of the proposed criterion. On the contrary, it has been proven in Section 7.5, Theorem 7.5.1 that the iterative algorithm finds the global and unique optimal solution of the clustering combination. Practical aspects of iterative algorithm application have been discussed.

Finally, several measures estimating clustering stability have been proposed. They are able to indicate stable samples, clusters and clustering and compare different clusterings.

In the following Chapter we demonstrate application of clustering combination.

Chapter 8

Clustering combination and image analysis

In this chapter we demonstrate various examples of applications exploiting clustering combination. At the beginning, we give a short list of proposed applications with brief explanations. Then we compare performances of unsupervised clustering combination algorithms to demonstrate effectiveness of proposed method (Chapter 7). Different criteria to evaluate combination are given: supervised and unsupervised. A supervised criterion is used only to compare results of combination to original clustering. Unsupervised criteria are used to estimate the optimal combination without previous knowledge of original clustering. Comparison results are discussed. Possible applications are demonstrated mostly for images.

We list now some applications of combination which are demonstrated in this Chapter:

1. Comparing clustering combination methods. The performances of different combination algorithms and objective functions are compared.
2. Combining via reclustering. A schema of combination via reclustering is given.
3. Combining of satellite image segmentations. An example of unsupervised combination of segmented images is presented.
4. Combining of images with artefacts. First we demonstrate a synthetical example on how to remove artefacts from images, then segmented satellite images with clouds are used.
5. Determining the optimal number of clusters for image series.
6. Combining for image deblurring. Brief discussions on image deblurring are given.
7. Clustering of nominal data. Combining is viewed as grouping of nominal data.
8. Combining for feature selection. A method for unsupervised feature selection is presented.

Other possible applications of clustering combination may be considered. Different maps of the same scene (touristic, agricultural, industrial etc.) can be combined to find

some common areas of ground occupancy. Combination of clustered or segmented images of the same scene can be helpful for supervised, semi-supervised or unsupervised image mining. These images may be segmentations of multi- or hyperspectral images captured at once or in different times (time series images) or of images obtained from different satellites.

We propose to compare methods of combination.

8.1 Comparing clustering combination methods

Different approaches can be used to combine clustering results, e.g., single-link hierarchical clustering, K-means clustering, multinomial mixture models with EM-algorithm and mean shift combination. We propose to compare these methods through their performances with several criteria:

1. NMI-criterion [Fred & Jain, 2005],
2. square distance between clusterings E (as defined in Eq. (7.60), Chapter 7),
3. MDL criterion for binary data Eq. (6.32),
4. MDL criterion for probabilistic models of binary data Eq. (6.28),
5. clustering error E_c , as described below.

We will show that each of these criteria is adapted for a particular clustering algorithm.

Clustering error E_c

In this subsection we would like to estimate the error of clustering combination compared to the true clustering. The true clustering is used only to validate clustering combination. After combination, clustering error may be estimated as the relative number of misclustered samples to the total number of samples. We call this error as "compactness" because it characterises the compactness of clustering results. Let the true clustering have K clusters and the combined clustering have P clusters. Then "compactness" is computed as follows:

$$\text{compactness} = \frac{1}{I} \sum_{p=1}^P \left(\#p - \max_k \#(p \cup k) \right) = 1 - \frac{1}{I} \sum_{p=1}^P \max_k \#(p \cup k), \quad (8.1)$$

where I is the number of samples and $\#p - \max_k \#(p \cup k)$ is the number of misclustered samples for cluster p . The expression $\#(p \cup k)$ is the general term of the confusion matrix: the columns correspond to true clusters, and rows to combined clusters.

When the number of combined clusters is greater than K (the true cluster number), the clustering error ("compactness") Eq. (8.1) tends to zero as P increases. But it is not a "good" clustering because clusters are small and not "compact". To overcome this problem it must be penalised by another error called "sparsity" of clustering. "Sparsity" is estimated in the following way: for each true cluster we take the ratio of size of the

marginal cluster to the size of the true cluster. Then the mean value is computed on all true clusters. Finally, the "sparsity" is computed as one minus the mean value:

$$\text{sparsity} = 1 - \frac{1}{K} \sum_{k=1}^K \frac{\max_p \#(p \cup k)}{\#k}, \quad (8.2)$$

where $p = 1, \dots, P$, $\#k$ is the number of samples in cluster k and $\max_p \#(p \cup k)$ is the number of samples of marginal cluster p with class k .

The trade between the two errors Eq. (8.1) and Eq. (8.2) can be derived as their sum. It will characterise the error of clustering combination in regards to the true clustering:

$$E_c = \text{sparsity} + \text{compactness}. \quad (8.3)$$

The error E_c is limited by 0 and 1: $0 \leq E_c \leq 1$. When the combined clustering is the same as the true clustering then error E_c equals zero. This error E_c Eq. (8.3) is used only to estimate the quality of combination and is not used in any combination algorithm. A similar criterion has been used in [Le Hegarat-Masclé et al., 1997].

Synthetic data

We perform a comparison of combination results on synthetical data which are noisy clusterings. For that $I = 200$ samples have been chosen, distributed in K clusters with equal populations each. We simulate noisy clusterings by allowing uniform random allocation of labels to a wrong cluster. A noise level of 20% means that we change 20% of the samples, randomly chosen from their original cluster to another one.

We collect 100 noisy clusterings for a given number of clusters and a given level of noise. Noisy clusterings are generated for 2, 5, 10 and 20 clusters and 20%, 25% and 30% levels of noise. Thus, to evaluate combination algorithms we build 12 data sets each of which has 100 noisy clusterings. This experiment is similar to the one described in Section 7.4 but realised on more data sets.

We propose to examine different clustering algorithms for different combination criteria. We have chosen the following algorithms and criteria:

1. hierarchical single-link applied to the co-association matrix with
 - (a) NMI criterion as in [Fred & Jain, 2005],
 - (b) square error E Eq. (7.60),
2. K-means applied to binary representation of clusterings with
 - (a) square error E Eq. (7.60),
 - (b) simplified MDL Eq. (6.32),
3. Multinomial mixture model (MMM) and EM-algorithm with
 - (a) error E Eq. (7.60),
 - (b) MDL criterion Eq. (6.28).

We note, that when MDL criterion Eq. (6.28) is used for evaluation we should correctly determine the number of free parameters. This number depends on the data dimension. Since we represent clusterings by binary matrices Eq. (7.1) or Eq. (7.2), the dimension which determines the degree of freedom of model (either for K-means or for multinomial model, Chapter 7) equals to the number of clusterings P and not the number of columns in Eq. (7.2). Considering dimension equals to P we can apply MDL criterion either as in Eq. (6.32) or as in Eq. (6.28). We show in Sections 8.1 and 8.1 which criterion to use for which combination algorithm. Results of clustering combination for three combination algorithms are presented in Figures 8.1, 8.2 and 8.3.

S-link combination: NMI and error E

We begin the comparison of combinations for single-link algorithm with two unsupervised criteria: NMI Figure 8.1 a-d and error E Eq. (7.60) Figure 8.1 e-h. Combination has been tested on twelve data sets described above. As we remember s-link hierarchical clustering algorithm is applied to co-association matrix A Eq. (7.34) which represents different clusterings. Maximal value of NMI criterion indicates the optimal number of combined clusters while error E Eq. (7.60) and E_c Eq. (8.3) have to be minimised.

From Figure 8.1 a-d we see that NMI criterion can determine the true number of clusters, i.e., 2, 5, 10 and 20 only for a weakly noisy clusterings with 20% of noise (black line of Figure 8.1 a-d) and gives false numbers of clusters by growing up for clusterings with 25% and 30% of noise (Figure 8.1 a-d, blue and red lines). On the contrary, square error E Eq. (7.60) allows to determine the true number of clusters, i.e., 2, 5, 10 and 20 for all clusterings (Figure 8.1 e - h, black, blue and red lines). Sharp peaks indicating the true number of clusters are observed for the number of cluster 2, Figure 8.1 a, e, i, while with an increasing number of clusters and noise, error E Eq. (7.60) has flat peaks. Figures 8.1 i-l show error of clustering combination E_c for different numbers of clusters. All minimum of supervised error E_c , Eq. (8.3) equal to 0. It shows experimentally that single-link combination may find true numbers of clusters for synthetical data using a good objective function as E Eq. (7.60).

K-means combination: error E and MDL

Figures 8.2 a-d illustrate square error E Eq. (7.60) for K-means algorithm which is used to combine clusterings. Here again, true clustering is determined for different numbers of clusters, i.e., 2, 5, 10 and 20 and all noise levels (20%, 25%, 30%). With an increasing number of clusters and noise level error E Eq. (7.60) tends to have flat behaviour.

On the contrary, MDL criterion Eq. (6.32) has sharp peaks on the true number of clusters Figures 8.2 e - h and as well successfully combines clustering. We note that K-means algorithm uses different initialisations and gives different clustering result. This results are considered as combination of clusterings. K-mean is run 50 times for a given number of clusters and MDL Eq. (6.32) curve is plotted for the number of clusters from 2 to 30. Points above curves, in Figures 8.2 e - h are values of MDL Eq. (6.32) for different initialisations. Only minimal values (diamonds) are considered as optimal and are connected by lines (black, blue and red). We see that MDL criterion with K-means has very sharp and clear minimum of MDL for different numbers of clusters and different levels of noise. On all Figures 8.2 a - h error E and MDL have black lines (20% of noise) under blue ones (25% of noise) and blue lines under red lines (30% of noise). For MDL criteria

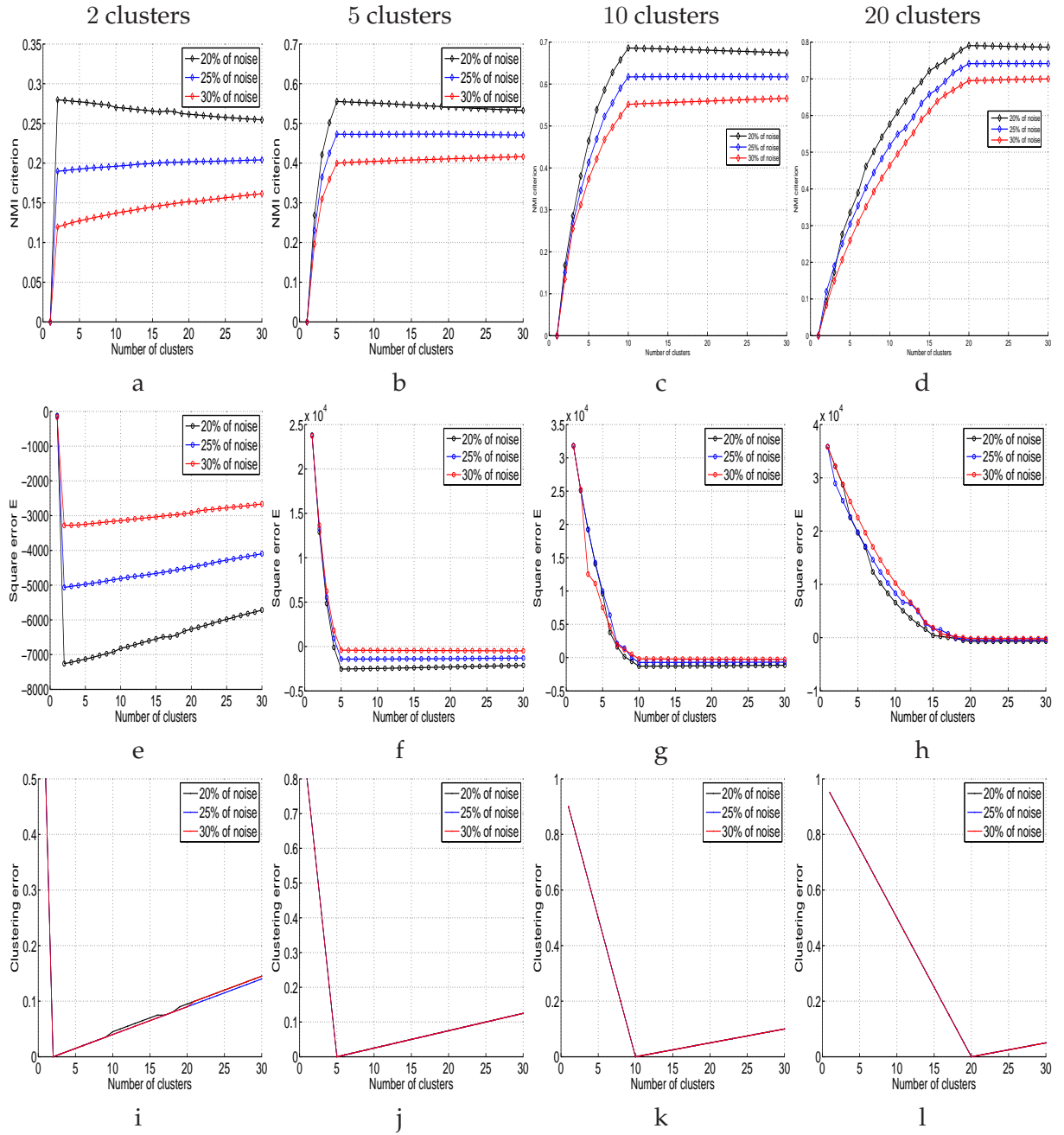


Figure 8.1: Comparison of combinations of single-link algorithm with NMI criterion and error E Eq. (7.60). Different numbers of clusters are tested: 2, 5, 10 and 20 (from left to right). From each "true" clustering 100 noisy clusterings with 20%, 25% and 30% of noise are generated. Combinations : a - d - single-link with NMI criterion, e - h - single-link with error E Eq. (7.60), i - l - clustering error for single-link algorithm E_c Eq. (8.3).

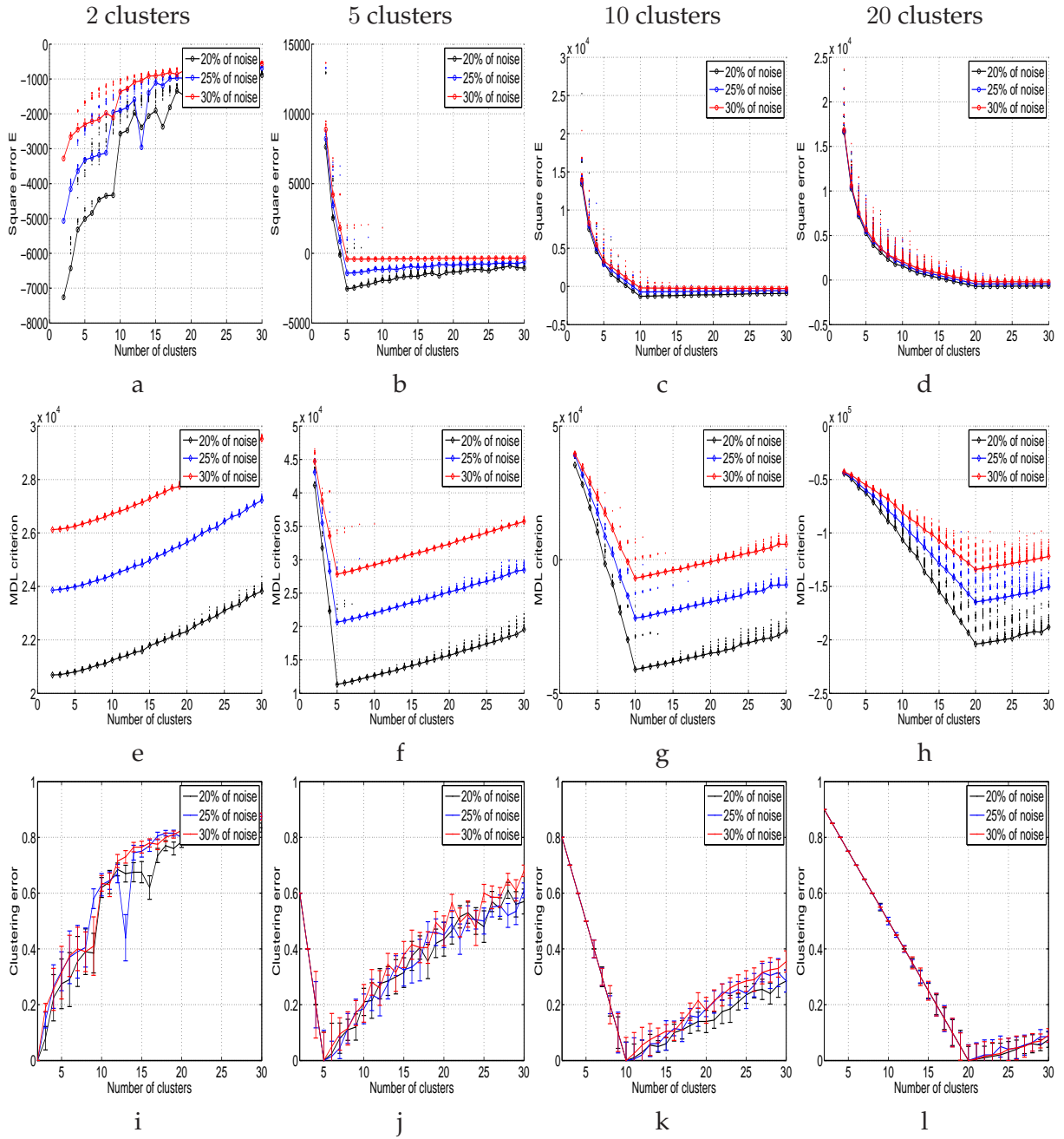


Figure 8.2: Comparison of combinations based on K-means algorithm with MDL criteria and error E . Different numbers of clusters are tested: 2, 5, 10 and 20 (from left to right). From each "true" clustering 100 noisy clusterings with 20%, 25% and 30% of noise are generated. Combinations : a - d - K-means with error E Eq. (7.60), e - h - K-means with MDL criterion Eq. (6.32), i - l - clustering error for K-means algorithm E_c Eq. (8.3).

this is explained that less noisy data have less complexity. The explanation for error E Eq. (7.60) is that less noisy data have less square error.

Here we should note that K-means uses random initialisations and it cannot guarantee a global optimum. In the case of a high number of clusters, e.g., some tens, results may be very different and the number of runs necessary to achieve the optimum may be very high (it may be not useful for practical applications). However, the initial solution can be taken as one of the given clusterings. A good initialisation is the closest clustering to all other clusterings, e.g., in the sense of the Euclidean distance. But again, this clustering may be far from the optimal one and K-means may not achieve the optimal value. We see that a stable method of combination with guaranteeing optimal solution should be used.

Clustering error E_c (Figures 8.2 i - l) shows that K-means algorithm may achieve optimal combination. The error E_c has zeroes at the optimal combinations for 2, 5, 10 and 20 (from left to right) clusters and all noisy levels. We also remark that standard deviation of the error E_c Eq. (8.3) grows with growing numbers of clusters. This effect is explained by the fact that K-means has more degrees of liberty and gives many different clusterings.

MMM with EM-algorithm combination: error E and MDL

Figures 8.3 a-h show the comparison of error E Eq. (7.60) and MDL Eq. (6.28) criterion for clustering combination. Parameters of the mixture of the multinomial model (MMM) have been estimated by EM-algorithm. Figures 8.3 a-d correspond to error E . Minimal values of error E indicates the optimal number of clusters for combination. As we see clusters 2, 5, 10 and 20 are correctly estimated by error E for different levels of noise.

MDL criterion estimated with MMM and EM-algorithm is shown in Figures 8.3 e-h. The optimal number of clusters is determined as the minimum values of MDL curve. We see in Figures 8.3 e-h that with growing number of clusters MDL criteria becomes to have rather chaotic behaviour. This effect appears because clustering results are very different for different initialisations and it is possible to have many local optima for a high number of clusters.

The probability of obtaining a good clustering is decreasing with a growing number of clusters. This effect is observed through the chaotic behaviour of supervised error E_c Eq. (8.3), which shows the quality of the clustering, Figures 8.3 i-l. Standard deviation of error E_c for 2 clusters is much smaller than for E_c with 10 clusters Figure 8.3 k and 20 clusters Figure 8.3 l. It expresses the fact that combination of clusterings by MMM with EM is very unstable. In the same time the supervised error of the clustering combination E_c in Figures 8.3 i-l shows that the optimal combinations are achieved. These values of E_c equal zero and are observed on all Figures 8.3 i-l. This proves experimentally that MMM with EM algorithm can be used to combine clustering results, however for large numbers of clusters the global optimum may never be achieved.

Discussion

We give a short summary of experiments for combination of synthetical data. Different algorithms of clustering combination have been compared in this Section. We have seen that the square error E Eq. (7.60) and MDL criterion Eq. (6.28) show the optimal number of clusters in contrast to NMI criterion [Fred & Jain, 2005] which fails in some cases. In addition, we should note that direct application of hierarchical single-link algorithm may

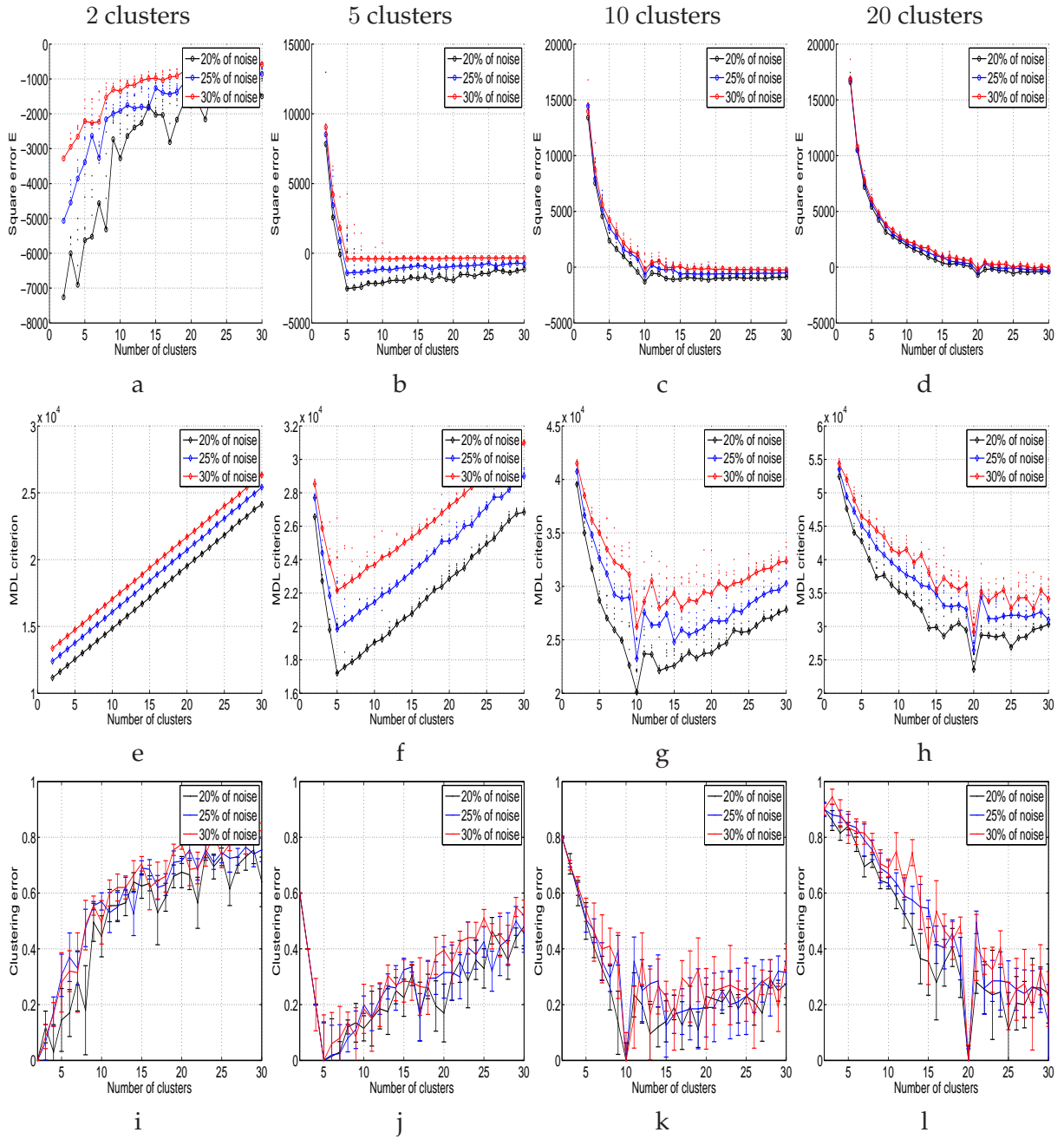


Figure 8.3: Comparison of combinations of Multinomial mixture model estimated by EM-algorithm with error E and MDL criteria. Different numbers of clusters are tested: 2, 5, 10 and 20 (from left to right). From every clustering 100 noisy clusterings with 20%, 25% and 30% of noise are generated. Combinations : a - d - MMM with EM and error E Eq. (7.60), e - h - MMM with EM and MDL criterion Eq. (6.28), i - l - clustering error E_c for MMM with EM algorithm Eq. (8.3).

be time and memory consuming for large data sets because of the square complexity. On the contrary, K-means or EM-algorithm have linear time and memory complexity and may be applied to quickly test combinations. But they still suffer from depending on initialisations and may not give the optimal clustering in the case of a large number of clusters.

Clustering combination performed by mean shift for all synthetical cases given above returns the exact true clustering. It means that each noisy clustering set after combination by MSC have been recovered as the true clustering with error E_c Eq. (8.3) equals zero.

8.2 Combining via reclustering

In this Section we propose another approach to combine different clusterings. In previous Section we have shown how to combine clusterings by single-link algorithm, K-means or multinomial mixture model with EM-algorithm. The choice of MDL criterion to select the optimal number of clusters has also been studied. We remind that to select the optimal clustering combination we run EM-algorithm with different random initialisations and then select the best model. Here we propose to apply K-means, BMM or MMM with EM-algorithm (with random initialisations) and then to recluster again these results. The schema of such a clustering combination is given in Figure 8.4 This schema can be used

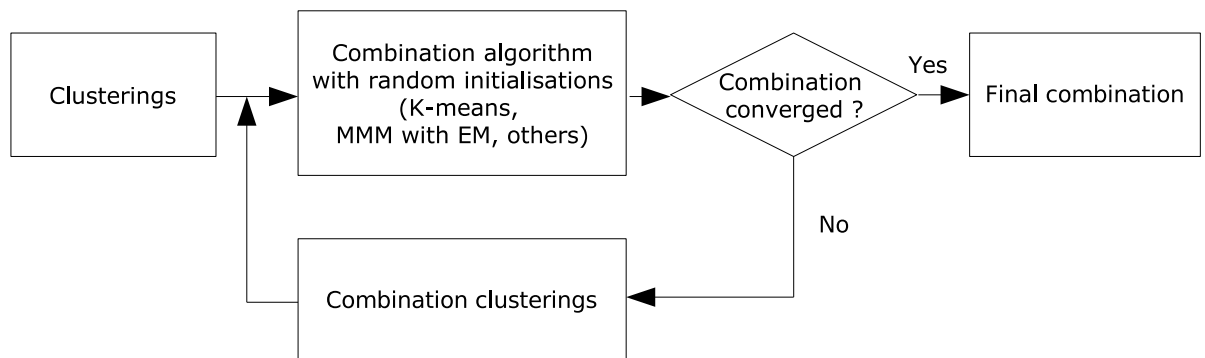


Figure 8.4: Schema for clustering combination using label reclustering with random initialisations. Convergence of combination can be estimated on how many clusterings are the same: if more then a half, then combination is converged, else take results of combination and recombine them. Alternatively, either the square distance E Eq. (7.60) or the stability of clusterings can be used as indication of convergence.

to find a stable clustering combination for a given set of algorithms. In this way we can loop the process until convergence is achieved. There are several ways to estimate convergence:

1. evaluate MDL criterion and check the difference between actual and previous values;
2. measure the stability of clustering (described in Section 7.6) and stop when a certain level is achieved (this level can be fixed or computed);

3. take one of the clusterings for a given iteration, then stop if this clustering is the same as one half of clusterings.
4. the distance between clusterings may be estimated as the square distance E Eq. (7.60) and if it equals zero, then stop.

8.3 Combining of satellite image segmentations

In this section we demonstrate the ability of our combination approach in the context of satellite image segmentation. Data are considered as the image pixels; segmented regions provide clusterings. We do not pay attention here to the segmentation algorithm. Segmentations have been obtained with the same algorithm but with different parameters (e.g., watershed segmentation with different thresholds).

As mentioned before, combination can be performed either using K-means algorithm or MMM with EM-algorithm. It is clear that when we have data as image segmentations which have hundreds of clusters it will be time consuming to analyse MDL curve to find the optimal number of clusters, as discussed in Section 8.1. As initial solution for combination by clustering algorithm, one of the segmentations results can be selected. This initial segmentation should be nearest to all segmentations expressed, e.g., in the sense of square distance E Eq. (7.60). But, as explained in Section 8.1, the selection of one of the clusterings may not lead to the optimal combination. However, sometimes it can lead to a good enough solution.

Below we give six segmentations to be combined, see Figure 8.5 a-f. Different methods of segmentation combining are applied: (i) [Giros, July 31 2006-Aug. 4 2006], (ii) a multinomial mixture model with EM-algorithm to estimate MMM parameters (Chapter 7), and (iii) mean shift combination MSC (Chapter 7).

The result of the segmentation combination presented in Figure 8.5 g is performed by the method described in [Giros, July 31 2006-Aug. 4 2006] using mutual information between segmentations. In Figure 8.5h the combination by MMM with EM-algorithm is presented. As seen from previous sections MMM with EM needs a good initialisation, otherwise it is very time consuming to initialise it randomly to obtain the best combination.

The second segmentation has been selected as the initial solution for MMM. The distance from each segmentation to all others is computed as the square error E Eq. (7.60), see Figure 8.6. The minimum value of square error E corresponds to the second segmentation. We compute parameters of MMM (mean values) given the second segmentation and run EM for MMM to obtain a combination of segmentations, see Figure 8.5 h.

The result of combination by MSC algorithm is given in Figure 8.5 i. We see from Figure 8.5 i that the error E Eq. (7.60) has the lowest value for combination by MSC algorithm. It means that combination by MSC produces the segmentation which is closer to others comparing to combination by other methods.

8.4 Combining of images with artefacts

One of the main problems in satellite imagery is that the Earth surface is covered with clouds that may provide a lot of partially useless images. For several satellite images of the same scene it is possible to determine common background supposing that clouds

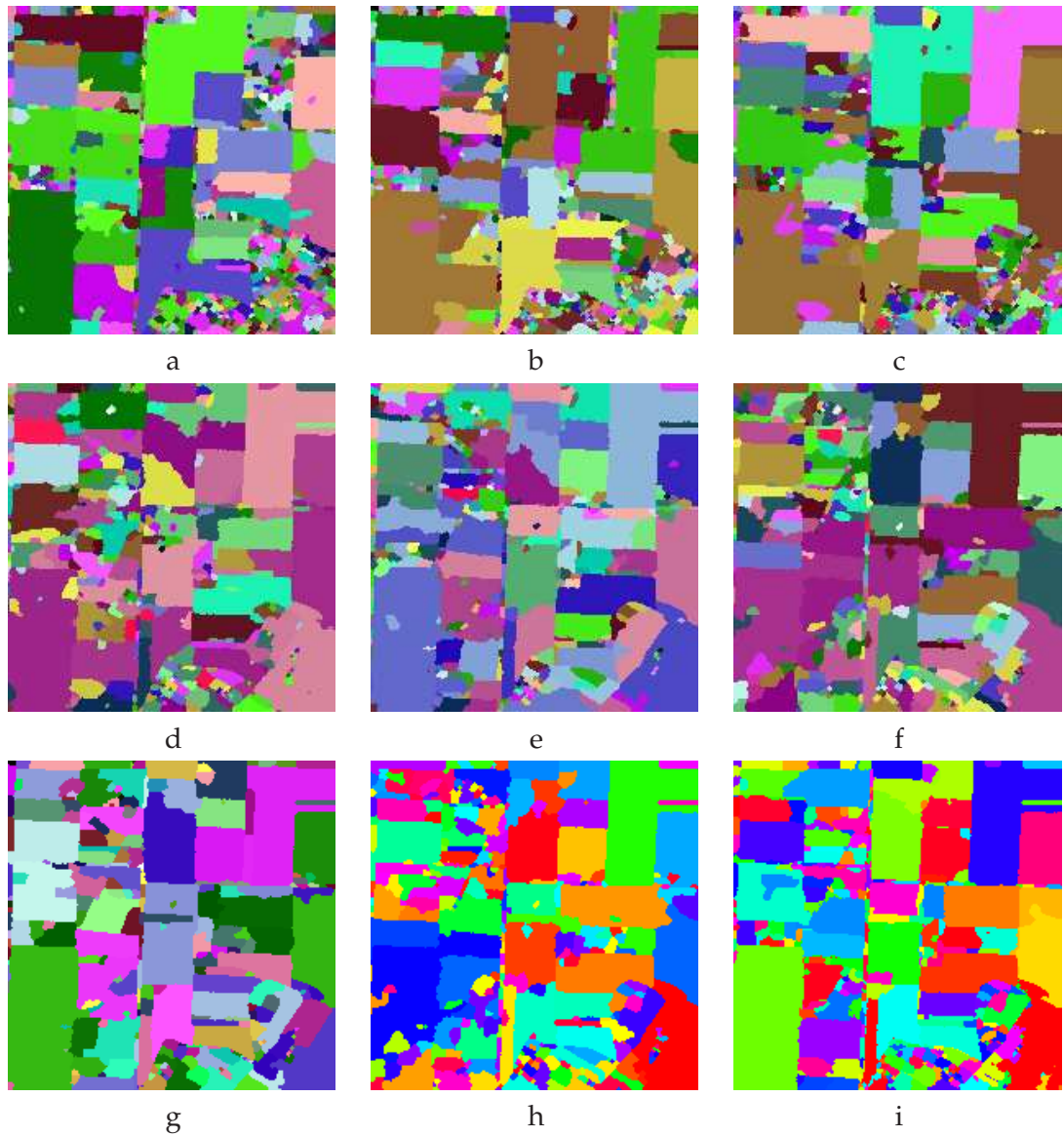


Figure 8.5: Combination of segmentations. a - f - 6 segmentations to be combined. The square distance E Eq. (7.60) is used to evaluate the quality of combination. g - combination obtained by Giros $E = 17.2 \cdot 10^6$, h - combination obtained by MMM with EM $E = 18.8 \cdot 10^6$, i - combination obtained by MSC $E = 3.5 \cdot 10^6$. The error is negative because a constant term is not taken into account in Eq. (7.60).

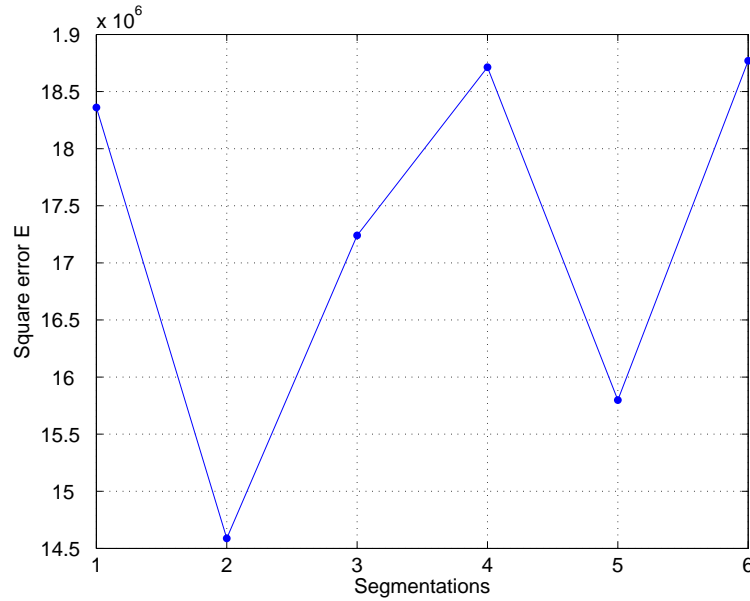


Figure 8.6: Square distance E Eq. (7.60) from 6 segmentations to their common representation. The distance from combination obtained by MSC algorithm $3.5 \cdot 10^6$ is three times lower than distances from any of the 6 segmentations.

randomly cover the Earth surface with many "open" areas. Then it is possible to segment these images, combine different segmentations and reconstruct the original scene. Here we propose two examples of combination using:

1. synthetical image segments with artefact,
2. real satellite images with clouds.

Synthetic segmentations

We propose to generate synthetical segmentations with simulated "clouds" as an example. We generate 25 regions of size 20×20 pixels with at most 10 different labels. Thus we have a simulated image of 100×100 pixels where several regions may have the same label. In addition, we simulate the presence of "clouds" on images. The "cloud" is a circular segment (however, it may have any shape) with a random position and a random radius. 10 examples of such segmentations are proposed, Figure 8.7 a-j.

Examples of simulated segments in Figure 8.7 a-j contain twice as less different labels than the number of regions. We illustrate it to show that combination of segmentations may provide useful information. The combination of segmentations is performed by MSC algorithm which finds in an unsupervised way the number of correct segments which equals 25. The result of combination of the 10 segmentations is shown in Figure 8.8. We see from Figure 8.8 that the combination allows reconstructing original background image of segmentations. All 25 regions have been detected even if each segmentation has no more than 10 labels. It proves practically that combination may derive new and interesting information from data with different points of view (segmentations, clusterings, classifications, maps, etc.).

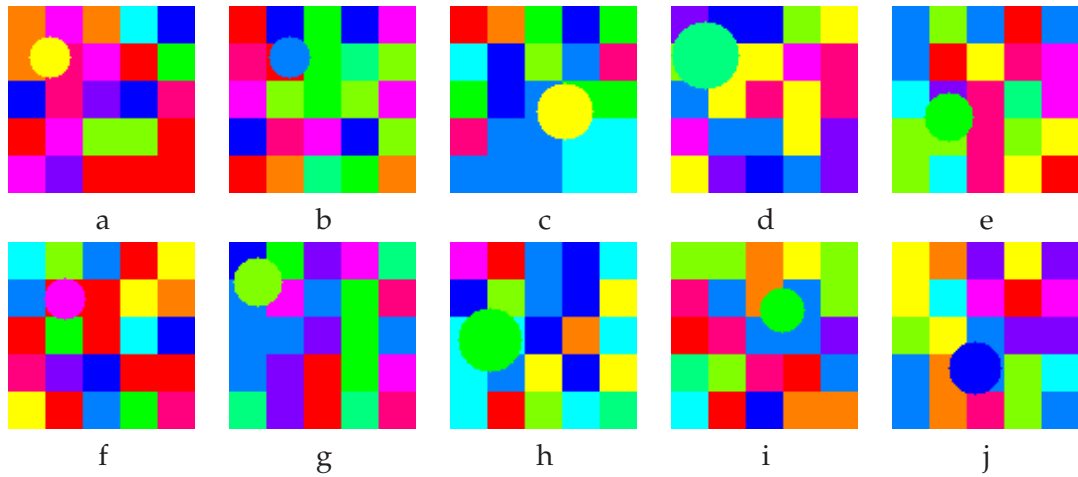


Figure 8.7: Simulated segmentations with "clouds". Each image has 25 regions with at most 10 labels. Circular regions on each image simulate "clouds".

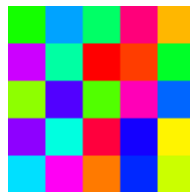


Figure 8.8: Result of combination by MSC algorithm of simulated segmentations given in Figure 8.7. After combination 25 segments have been detected, "clouds" have disappeared.

Combination of clustered images with clouds

In this section we give a combination example for satellite image segmentations. Here we propose to combine image segmentations of the same scene but captured at different times. Five images are SPOT5 time series of the same scene. Each image has size of 200×200 pixels. Examples of satellite images are presented in Figure 8.9.

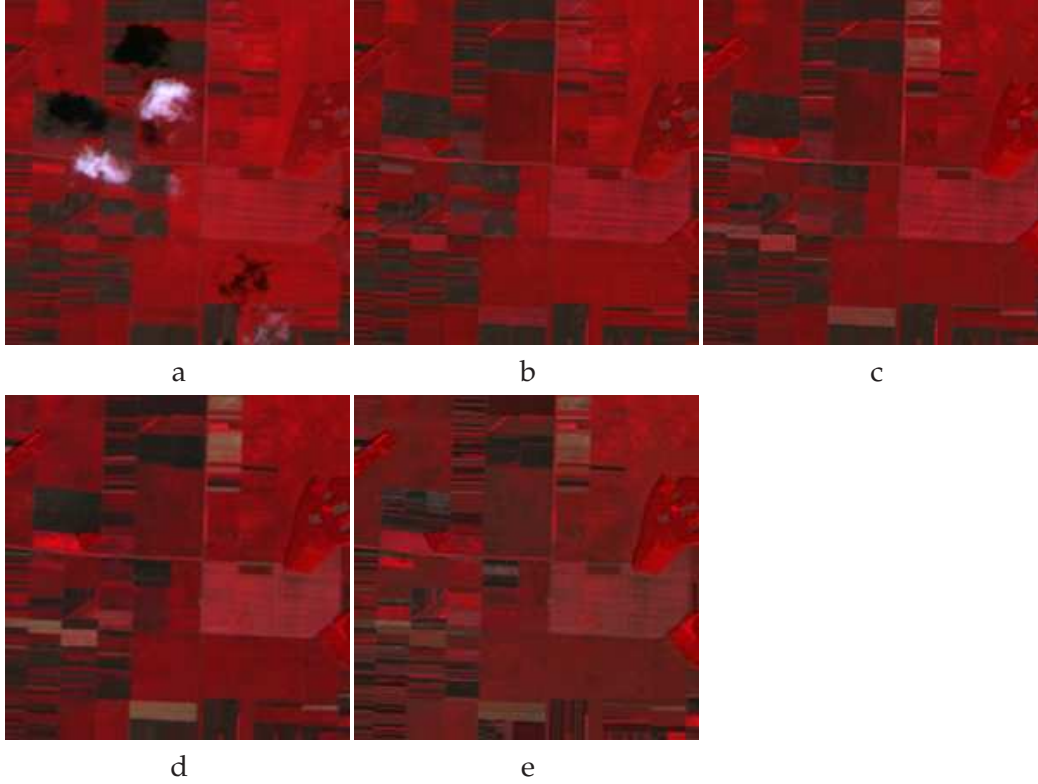


Figure 8.9: SPOT5 time series images of the same scene, ©CNES.

The first image in Figure 8.9 a contains clouds. We want to segment images and combine their segmentations. The considered segmentations are obtained by K-means clustering algorithm with a fixed number of clusters equal to 3. We have not considered the estimation of the optimal number of clusters and only show combination results. Image segmentations are shown in Figure 8.10.

The first image, see Figure 8.10a shows segments of clouds. The result of combination Figure 8.10f does not have the cloud.

To evaluate the quality of segmentation combining illustrated in Figure 8.10 we calculate distances in Table 8.1. E_1 shows distances between each segmentation and five segmentations, while E_2 shows distances between each segmentation and their combination.

Maximum values of errors E_1 and E_2 for the first segmentation in Table 8.1 means that, this segmentation (image with "clouds", Figure 8.10a) is far from both all segmentations and their combination in Figure 8.10f. Hence, we may detect an image with artefacts as clouds, smog, etc. Moreover, in this case clouds do not influence combination of segmentations.

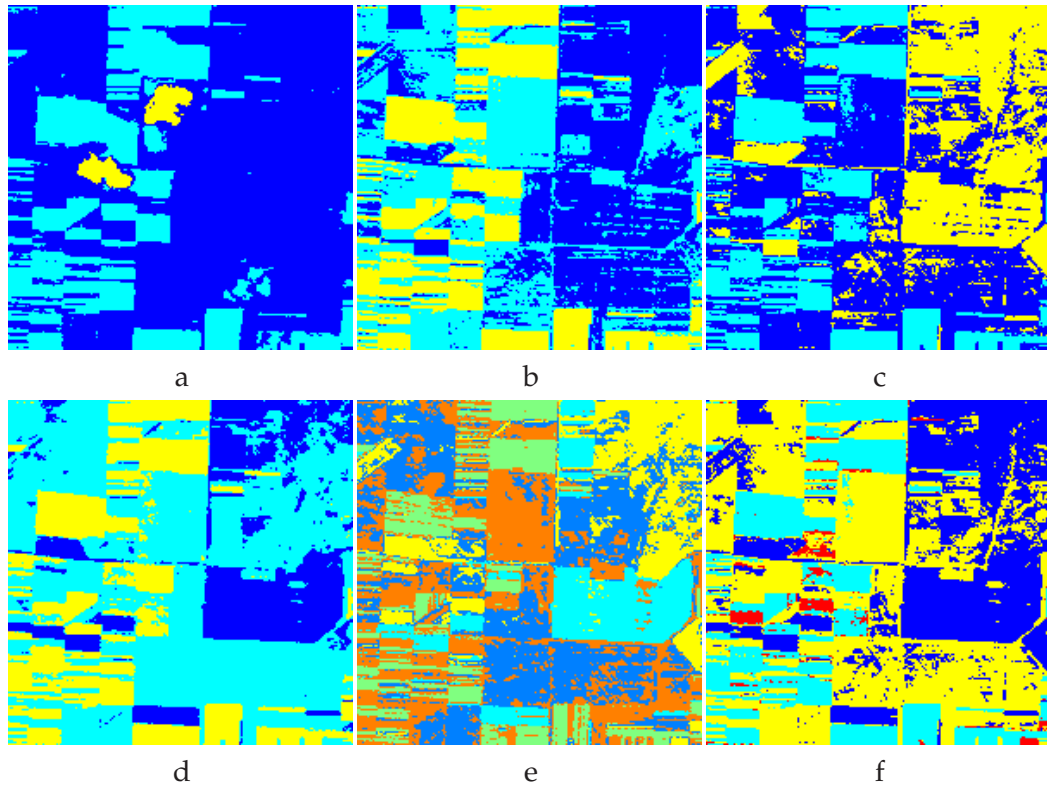


Figure 8.10: Combination of clustered images: a-e - image clusterings (Figure 8.9). f - combination of clusterings.

Table 8.1: Distances for combined segmentations in Figure 8.10

Error	Segmentations				
	1	2	3	4	5
E_1	$1.8e + 08$	$1.2e + 08$	$1.0e + 08$	$1.1e + 08$	$1.6e + 08$
E_2	$23.6e + 07$	$11.3e + 07$	$1.7e + 07$	$9.3e + 07$	$19.5e + 07$

For real applications we may further substitute segments from combination by original parts of images. This allows "reconstructing" the image disturbed by clouds.

8.5 Determining the optimal number of clusters for image series

The combination method proposed in this thesis may combine clusterings with different numbers of clusters. Here we propose to consider this number as a parameter to be estimated. Let us consider as instance clustered images of the same scene (like in the example in the previous Section) Figure 8.9 a-e. We want to cluster a new image of this scene "similarly" to existing clusterings. The problem is how to estimate the number of clusters for the new image.

Let the first image Figure 8.9a be a "new image". We aim at comparing it to the four clusterings of Figure 8.10 b-e, where each clustering has 3 clusters. We would like to estimate the number of clusters for the first image Figure 8.9a. In addition, clustering of the first image should be similar to the four clusterings in Figure 8.10 b-e.

To estimate the number of clusters for the first image we cluster it with an increasing number of clusters from 2 to 15 and draw the distance E Eq. (7.60) from the clustering to the four others in Figure 8.10 b-e. This procedure is repeated for the number of clusters from 2 to 15. In Figure 8.11 the minimum value of the distance E indicates that the first image with 4 clusters is the closest to clusterings in Figure 8.10 b-e.

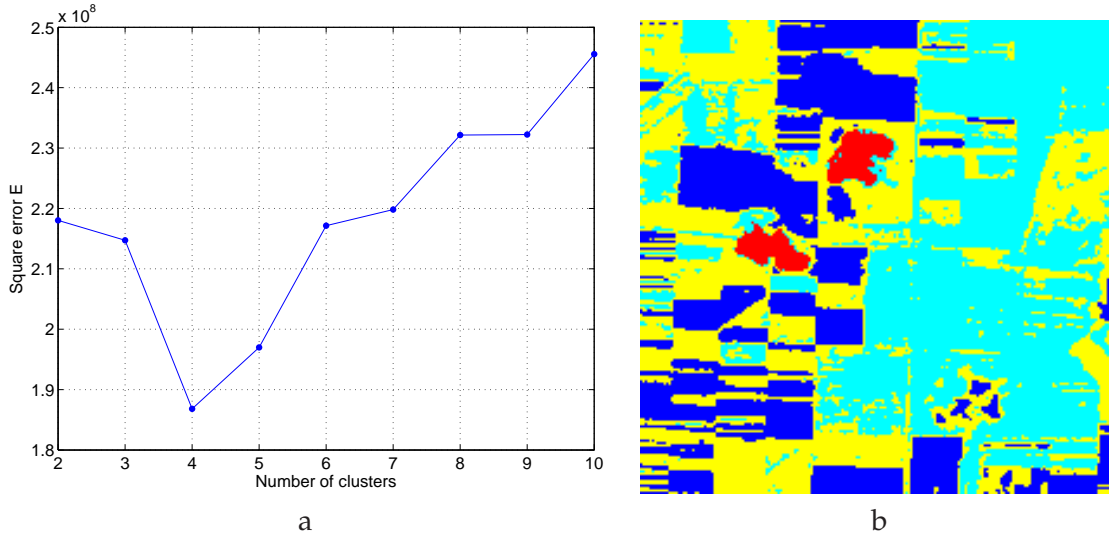


Figure 8.11: Optimal clustering compared to four given clusterings. a - the distance E Eq. (7.60) between the first clustered image in Figure 8.9a and the four clustered images in Figure 8.10 b-e. b - the result of clustering with the optimal number of clusters which equals 4.

We see from the optimal clustering of Figure 8.11 b that one cluster corresponds to clouds. We may conclude that if a new image has more clusters than given images then it contains some new information.

Other approaches may be considered via combination. For example, when the number of segments depends on a parameter of the segmentation algorithm (e.g., a threshold

for watershed segmentation algorithm), then the optimal value of this parameter may be determined as the parameter for which segmentations are similar.

An example in video processing is motion estimation with a fixed camera observing a scene. It is possible to detect the background and analyse movements. We could also imagine detecting movements, based on the square error E Eq. (7.60) between segmented frames. A frame with movements can be detected as one which is far from the combination of several frames. It is also possible to separate movements in the video from the background. For scenes with more movements error E will be larger than for the rare movements on the scene.

8.6 Combining for image deblurring

In this section, we illustrate briefly an idea observed from practical experiments on combination concerning for image deblurring.

Let us generate an image with 7 gray levels, Figure 8.12a. Then let shift this image in the direction of its two spatial axis. Suppose that this shift has random Gaussian noise with 2 pixels standard deviation. If we take the mean of 50 shifted images we obtain a blurred image as in Figure 8.12b.

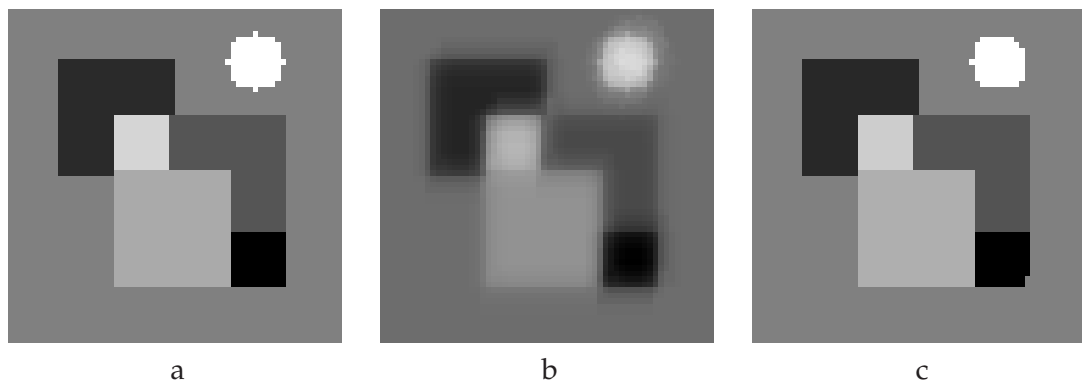


Figure 8.12: The square error between the original image (a) and blurred image (b) is 148.44, while the square error between the reconstructed image (c) and the blurred (b) image is 12.35. Reconstructed image differs from the original by several pixels, e.g., in the area of white circle.

Since each image has only 7 gray levels we can unwrap it into a vector and represent this vector by a binary matrix as in Eq. (7.1). Then we can concatenate shifted binary images into one matrix B as in Eq. (7.2). Finally we propose to combine the concatenated binary matrix B by the mean shift combination proposed in Chapter 7. Under image deblurring we consider an image obtained after combination. The result of combination is shown in Figure 8.12c. We see from Figure 8.12c that globally the square error 12.35 between the reconstructed image (c) and the blurred (b) image is much lower than 148.44 between the original image (a) and blurred image (b).

It shows that combination gives more accurate result from the set of shifted images.

Figure 8.13 shows lines from the original image, from the mean of shifted images and from reconstructed image.

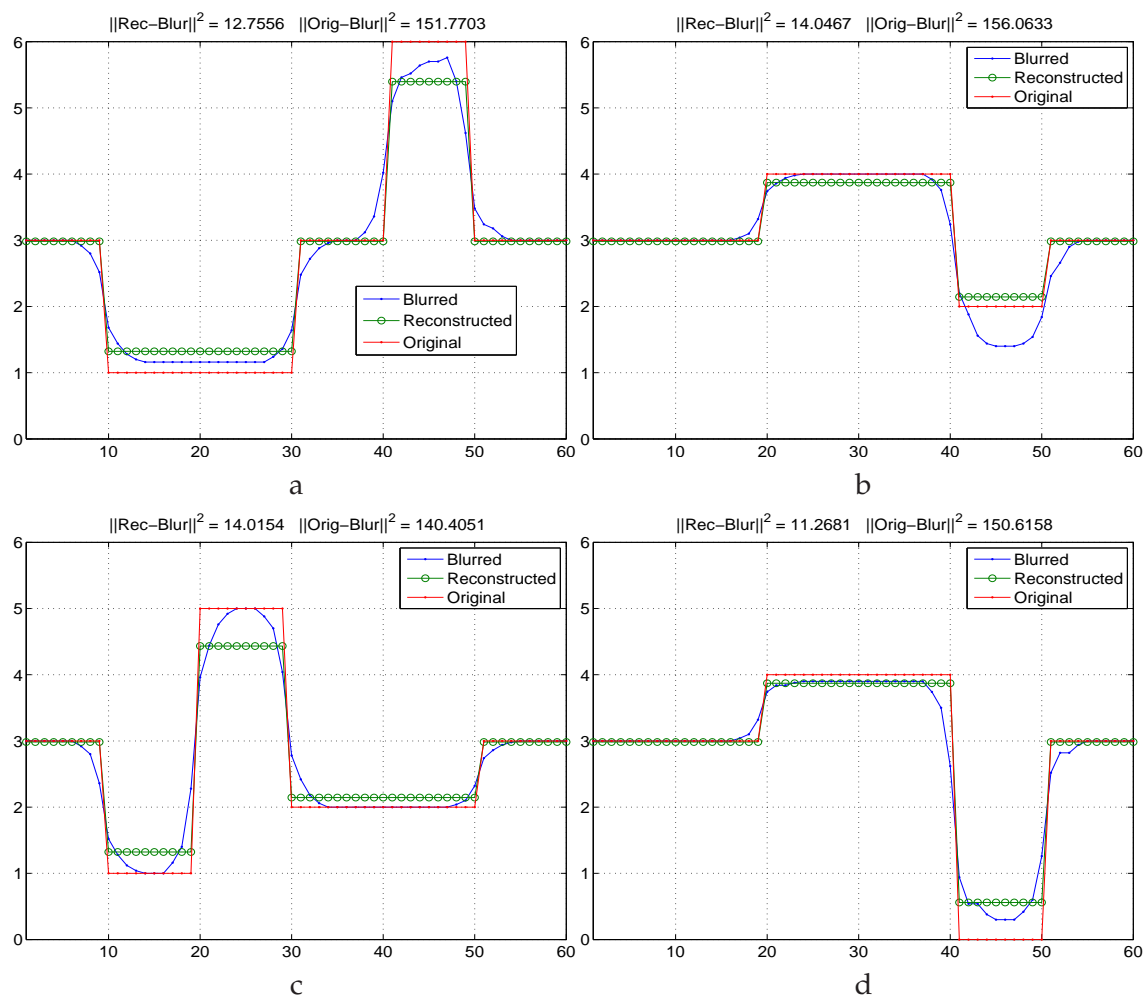


Figure 8.13: Lines of the original image, mean of shifted images and reconstructed image (Figure 8.12 a-c).

As we see from Figure 8.13 a-d, the reconstruction is better for each line than the original line of the image.

We observe that combination of randomly shifted binary images returns a good reconstructed image in the sense of the minimum of the square error. We may consider this process as image deblurring.

Here proposed combination for image deblurring remains still open because it has been presented only on simulated data. To apply it on real data we should obtain matrices B^p Eq. (7.1) from the blurred image. How to obtain these matrices is not considered here. But deblurring may be considered as an additive model where blurred image is the sum of shifted perfect images. As the mean of shifted images does not depend on their order of summation, we may construct the matrices B^p by suppressing images from the mean image. But there is the problem of image shifting. The shifting of an image may be avoided via Fourier transform, because the amplitude of Fourier transform does not depend on shift. This question needs to be more deeply analysed.

8.7 Clustering of nominal data

In this sub section we give a small example of nominal data combination. This data set has each variable as a set of nominal data, e.g., a set of categories. Every variable can be coded as binary matrix B Eq. (7.1) and all of them concatenated into binary matrix B as in Eq. (7.2). Then instead of clustering data B by MMM with EM-algorithm we may combine binary data by mean shift combination (MSC), Chapter 7.

Here we consider nominal data taken from UCI repository ¹.

The first experiment is performed on a data set of votes. It includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes. There are nine different types of votes: voted for, paired for, and announced for, voted against, paired against, and announced against, voted present, voted present to avoid conflict of interest, and did not vote or otherwise make a position known. We represent labels of votes as binary matrices and then all matrices are concatenated into binary matrix B . For the missing labels which have a character "?" we put zero in B . MSC algorithm is applied on matrix B . After combination 2 clusters have been found. Three samples have been detected as outliers. The first sample has an unknown vote and the two second have 15 unknown votes among 16, that is why they have been clustered as single clusters. The error of combination is 11.4% (the percentage of misclassified samples) which is comparable to 11% classification in [Gionis et al., 2005].

The second data set is Mushroom Data Set with 8124 instances and 22 categorical attributes. This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms. Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. The latter class has been combined with the poisonous one. The error after combination by MSC algorithm is 10.6% and is better than 10.9% in the work [Gionis et al., 2005].

MSC-algorithm has been applied to obtain combination of nominal data. Here again, we do not need random initialisations and MSC - algorithm find automatically the optimal number of clusters.

¹[http://www.ics.uci.edu/\\$\sim\\$mllearn/{MLR}epository.html](http://www.ics.uci.edu/\simmllearn/{MLR}epository.html)

8.8 Unsupervised feature selection algorithm

One of the attractive topics of data mining is feature selection. This problem is very important because it consists in eliminating noisy, correlated and dependent features. This procedure allows to improve supervised, semi-supervised and unsupervised data analysis and to significantly reduce time computation and memory volume of data to be processed. This is a very actual problem for large volumes as well as for small volumes of data. Unsupervised feature selection is desired when no a priori information on feature preference is available.

In this section we propose an original method of unsupervised feature selection. As we suppose that data features may be correlated and dependent, we propose to group similar features and then select one feature which represents each group. This grouping can be considered as a clustering process applied to the feature set (it corresponds to clustering transposed data matrix).

How many clusters should be used? One approach to estimate the number of clusters is to use, for example, MDL approach. But in the case of feature clustering MDL criteria will have very high penalty and there is no sense in using MDL in this case.

For unsupervised feature selection we propose to cluster features by K-means algorithm which is simple and fast. We run it for the number of clusters from 2 to the number of features. For each number of clusters we initialise K-means with ten random initialisations. Then we group features by mean shift combination (MSC) in order to find the optimal cluster number for features. When we run K-means for a high number of clusters, combination of clustering results becomes a critical issue. Despite of it, this feature selection algorithm gives very good results in practice, see the following chapter.

After combination of feature clusterings we select from each combined cluster a feature which is the most stable. Stability is estimated by stability criteria S Eq.(7.74) in Section 7.6. Data analysis (clustering, classification, etc.) is then performed on the selected features set.

This idea has been published in [Campedel et al., 2007]. Results of this approach are compared to supervised feature selection and supervised classification. Classification has been performed by SVM classifier which achieves generally very high performances in practical tasks. In the paper [Campedel et al., 2007] we demonstrate that unsupervised feature selection provides very good results compared supervised selection.

8.9 Conclusions

In this chapter different and new examples of clustering combination have been considered. Comparison of different algorithms for combination is performed. We conclude, that the proposed objective function and the proposed MSC algorithm prove practically their superiority compared with other objective functions and combination algorithms. Effectiveness of combination is shown via (i) clustering and classification errors, (ii) stability of solutions and (iii) the fast implementation. One of the main advantages of the proposed combination is that it gives the same combination for the same set of clusterings.

The following applications to image analysis have been considered: combination of different clustering results and segmentations, parameter estimation, artefact detection, estimation of movements. Combination of time series images with artefacts may reconstruct a scene, e.g., in the case of images with clouds it is possible to recognize and sub

stitute clouds by "stable" image segments. Some ideas about image deblurring by combination have been mentioned.

For data analysis in general, it allows to combine nominal data values, estimate and find stable patterns, analyse and characterise stability of clusters and clusterings.

An important application of combination consists in unsupervised feature selection which shows very good practical results proving by the way the interest of the proposed method.

Another experiments are given in the following Chapter where unsupervised mining approaches are applied to multimedia and satellite images.

Part III

Semantic construction

Chapter 9

Semantic construction for images

In this chapter we address a problem of semantic construction for images. Semantic can be viewed as a set of concepts and relations among them [Suykens & Horvath, 2002]. This representation helps to show a variety of knowledge about images (concepts-relations) in a compact form. Moreover, semantic may be used in managing images (classifications, clustering, querying, etc.).

Two types of images are considered for experiments in this chapter: (i) multimedia and (ii) satellite images. We begin to construct semantic for multimedia images. This experiment has been carried out partially supervised (obtaining different classifications) and unsupervised (combining classifications). Its description is presented in Section 9.3. The aim of this experiment is to verify the proposed approach for semantic construction. Multimedia images have been used because of their easy interpretation by users. This experiment being successful is applied to satellite images in a fully unsupervised way, see Section 9.4. But, at first, a brief introduction to semantic is presented.

One of the earliest works on semantic construction for computer sciences may be found in [Gotlieb & Kumar, 1968]. The authors propose to analyse indexed vocabulary, where each index expresses words, collections of words, or phrases. The idea of this work is to establish semantic associations among indices depending on context, when indexing terms may have different connections. A power of this approach is that it uses a known vocabulary; therefore associations among terms are derived. From the other hand, it has a drawback because of using a priori knowledge about semantic relationships among terms. Finally, indexed terms are grouped into clusters (concepts). We suppose that this approach can be extended via automatic data analysis and applied to images. The problem here is that for images there are neither vocabulary nor concepts and they should be detected in an unsupervised way.

Recently, semantic construction for images became very popular [Kuhn et al., 2007; Carneiro et al., March 2007]. It is a rather difficult task.

In [Kuhn et al., 2007], a semantic clustering is proposed, based on latent semantic indexing along with clustering of textual items which share similar vocabulary. Clusters represent semantic topics with links between them, and visualised on a 2D map.

A survey of high-level semantics for content-based image retrieval is given in [Liu et al., 2007]. The authors consider semantic based image retrieval to support data mining instead of improving low-level feature extraction algorithms. Five state-of-the-art techniques are revised: (i) object ontology, (ii) machine learning level, (iii) relevance-feedback, (iv) semantic templates (v) fusing of text and visual content of multimedia images.

The semantic construction for textual data is similar to semantic construction for im-

ages. But problems emerge for images: (i) there is no image vocabulary (indexes), (ii) no a priori knowledge on how indexes are related and how they group to form concepts, (iii) there is no explicit semantic relation among concepts. Despite of the lack of information there exist two assumptions: (i) images are tractable data and can be interpreted, and (ii) images reflect useful information. Indeed, when we are looking at multimedia images we are able to detect objects, types of textures, colours and therefore, to categorise images into different groups. Moreover, it is possible to describe images by words. All these assumptions allows applying unsupervised methods of image processing to extract image terms (indexes), image concepts and relations among concepts. This will be demonstrated in Section 9.4. A representation of clustering results is discussed in the following Section.

9.1 Visualisation of clusterings

One of the goals of unsupervised data analysis is pattern detection. The idea of this approach has been discussed in Chapter 5. It consists in clustering data and identifying their semantic content. When we cluster a large volume of complex data it probably results many clusters (tens or hundreds). To navigate in the results becomes a rather difficult task. Therefore, there is a need to automatise the analysis of clusterings. For that, we should extract new information from the obtained clustering, e.g., estimate parameters of clusters, relations between them, degrees of connections, etc. Different distances can be considered as relations between clusters, e.g., the Euclidean distance or any other. For visualisation, clusters may be considered as concepts and for simplicity represented as nodes, while relations between them can be regarded as edges which connect nodes. Two representations are possible: trees and graphs structures.

What does a tree of clusters represent? A tree is an undirected graph with a single node at the top, leaves at the bottom and no loop. A tree may generalise clusters in one concept. Analysing nodes from top to bottom levels we are able to extract new pieces of information, e. g., which concepts preferably grouped and which concepts are grouped only at the top of the tree. Navigation using a tree is simple and fast and the user concentrates its attention on the current level of tree. But the tree has some disadvantages compared with graph representation. At each level of the tree we have only information about sub trees and no relation to the other nodes of the tree.

What does the graph of clusters represent? A graph is a set of nodes and edges, where undirected edges connect nodes. This representation is very useful to analyse how every cluster (node) is related to other clusters (nodes) and to measure the degree of this relation. This representation may be helpful in the search of patterns represented by one cluster or by different clusters. Cluster connections are not limited by the levels as for the tree and allows analysing more deeply clusters and their relations. This may be very useful when searching and constructing new concepts.

Trees and graphs may be extracted from the matrix of relations with relations being distances, similarities and dissimilarities. Examples of representations of clustering results as well as their analysis are given in the following sections.

9.2 Extraction of relations among concepts

In this Section we introduce relations among concepts represented by trees and graphs. We concentrate our attention on the case where data are clustered by different unsupervised algorithms. All discovered clusters reflect the original feature space, but have been discovered by different distances or models. The clusterings can be combined to obtain general (consensus) clustering. It is sometimes very difficult or even impossible to combine them via original feature space. That is why we propose to combine clusterings by co-association approach, see Chapter 7, avoiding by the way the concordance of different methods.

Relations between clusters in the coassociation matrix are exploited in this section. They reflect relations between different clustering results. The mean shift algorithm MSC of the combination of clusterings finds the optimal combination, as shown in Section 7.5. The degree of relations between combined clusters may be considered as the Euclidean distance between clusters in the space of clusterings B . If instead of the Euclidean distance we calculate the normalised sum of relations between combined clusters in space B , then it corresponds exactly to the vector product of two mean vectors of the combined clusters in space B . The vector product is related to the Euclidean distance as it has been shown in Section 7.3.

In the literature a very popular relation is exploited by the single-link algorithm [Fred & Jain, 2005] selecting two nearest neighbour clusters. The formulation of the coassociation matrix A Eq. (7.32) allows finding them without explicit calculation of the whole matrix. A single-link tree reflects the relations between clusters.

To construct the graph of relations, a vector product between means of clusters calculated on matrix B is used. Clusters are equally important and no preference for clusters is considered (there is no order). Clusters are represented as points in the 2 dimensional space situated on a circle with equal distances between neighbour points. Relations among clusters are presented by edges. The importance of relations is displayed through the thickness of the edge: the more important the relation the thicker the edge. Moreover, we can display only relations (edges) of a given cluster or to display the most important relations after thresholding.

Examples of clustering representations by trees and graphs for multimedia images are given in the following Section.

9.3 Semantic construction for multimedia images

In this section semantic of multimedia images is addressed. Examples of the analysis are also given. The idea behind this experiment is to ask several observers to classify a finite set of images, then, to exploit the combination of the classifications to derive semantic concepts for the images. This experiment, if successful, will support the idea that semantic may emerge from a consensual clustering.

The experiment has been made at TélécomParisTech, among the large processing research group and have been realised via webpage interface written in html, php and java scripts. An example of the webpage is shown in Figure 9.1. It runs on my home page ¹. For the experiment 45 multimedia images containing a variety of subjects have been selected. They are displayed in a random order in Figure 9.2.

¹<http://www.tsi.enst.fr/~kyrgyzov/webclass/>

Now we give the experiment protocol. Each user is asked to classify 45 images according to its own personal "best criterion" (as if organising his (her) own image directory in his (her) computer archive). Each user may choose as many (or as few!) classes as he (she) wants. All the classes will be at the same level (no hierarchy among classes). Moreover, the user is asked to give a name to each class, and at the end to annotate with a free vocabulary the class content. The interest of this experiment is to have independent classifications from different users each classification being as pertinent and "good" as any other. The protocol is the following:

Step 1. **"OBSERVING"** (5 minutes).

The user examines all images in the left part of the web page (can click on an image to enlarge it), defines the classes which are the best adapted to sort these images and choose the name for each class.

Step 2. **"CLASSIFICATION"** (5 minutes).

Clicking on the line under each image the user creates the class (by selecting "Add new class") selects an existing class. At this stage the user can change the name of classes at any time. After giving names to all images the user clicks on a button "OK".

Step 3. **"DESCRIPTION"** (5 minutes).

Names of the given classes are available on the right bottom part of the page. The user is asked to give several words (nouns) for each class in order to precise and annotate the class content. These words are freely chosen by the user, without the system. At the end the user should click on the "Classification is done" button.

In the total 50 different users participated to the experiment providing 50 different and independent classifications of 45 images. Two experiments have been conducted. In the first experiment the different classifications have been combined. In the second experiment the literal descriptions of classes have been combined. Below we explain more precisely the experiments.

Combining of classifications

The goal of this experiment is to find a consensus classification among the 50 different ones given by the users. The classifications reflect independent points of view and in the same time they have some common information. Combining is able to find consensus clusters which reflect groups of classes given by users, as it has been shown in Chapter 7. We hope that new clusters may be extracted from the combination result and they will better represent images than any single user's classification.

The first problem of combining of classifications is that each of them has its own number of classes. Moreover, classes from different classifications have no correspondence between each other because they have been obtained from different users independently. The second problem is that we do not a priori how many clusters the combination of classifications should have. The combination of classifications is performed by the unsupervised MSC algorithm, see Section 7.5. When applying this algorithm on the 50 given classifications of 45 images, see Figure 9.2, the number of estimated clusters by MSC equals 8.

The misclassification square error E as defined in Eq. (7.60) is calculated to estimate how far each classification is from all others. The error E is shown in Figure 9.3.

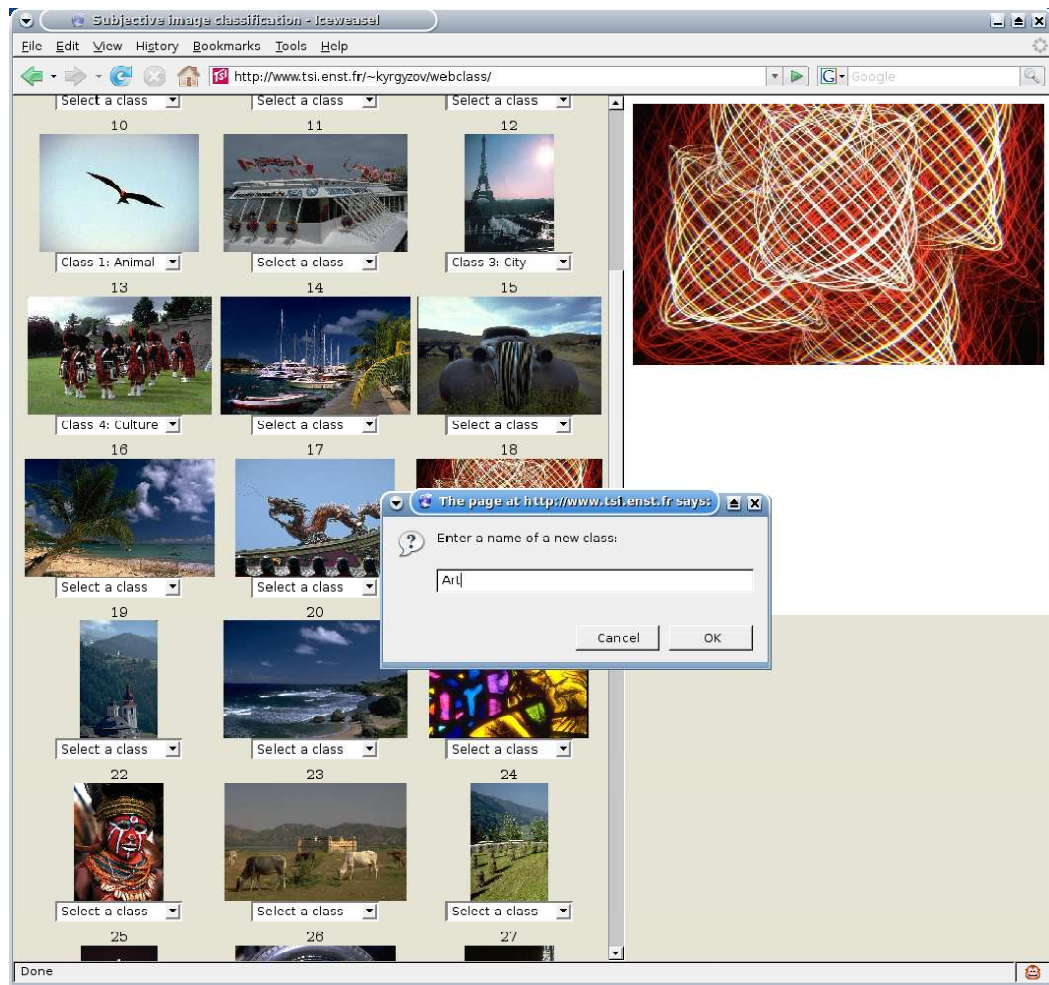


Figure 9.1: Web interface for description of multimedia images.

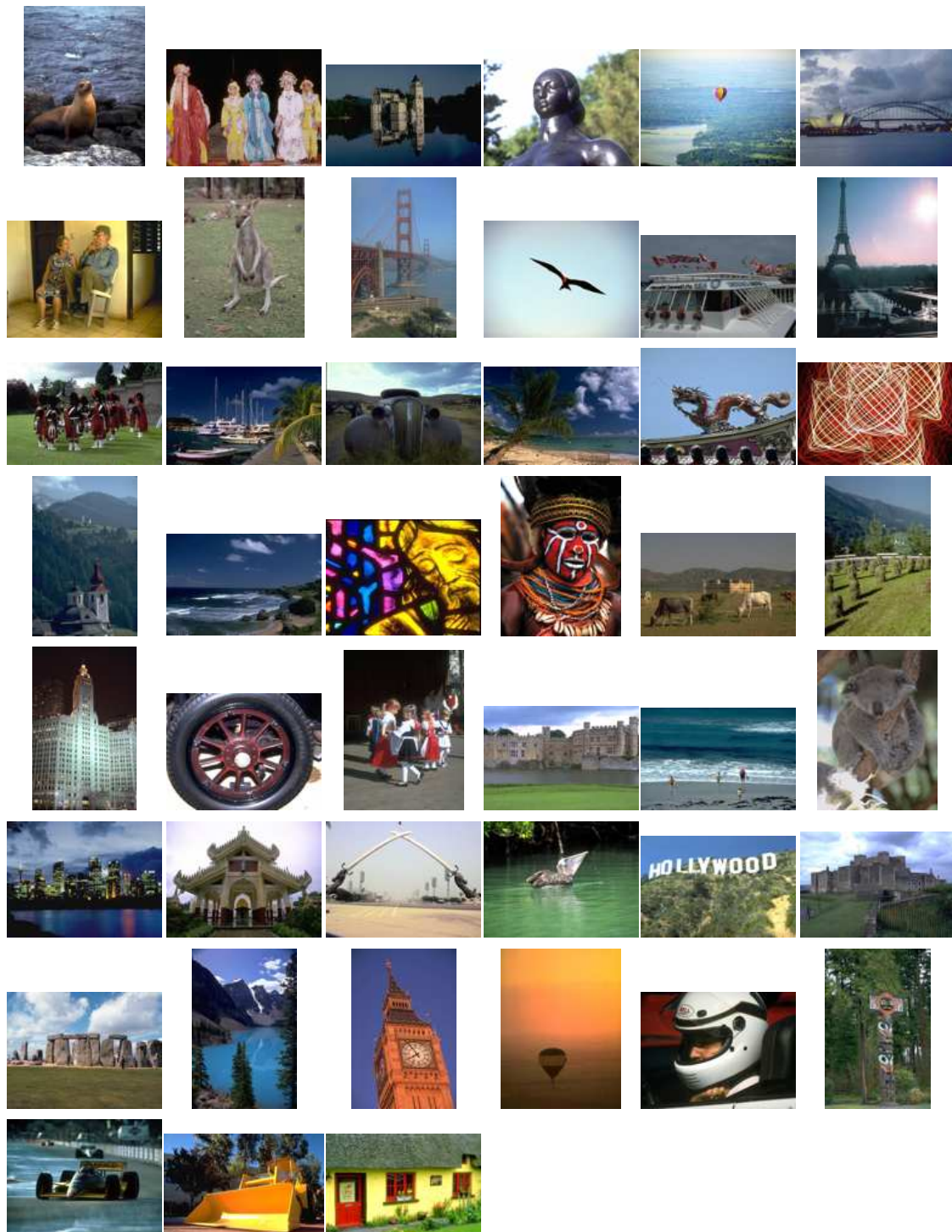


Figure 9.2: The 45 multimedia images which have been used for the experiment, Figure 9.1. Images presented in a random order issued from Corel Photo Library.

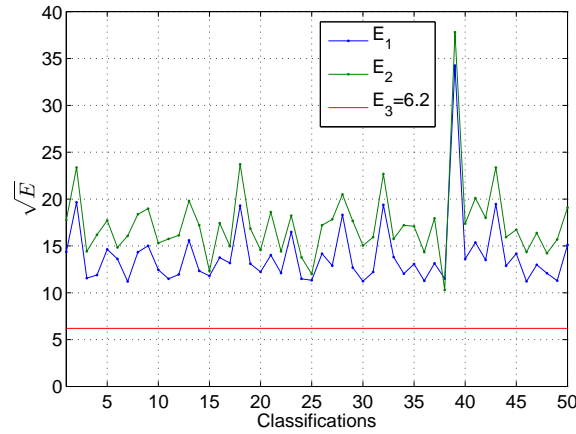


Figure 9.3: Error \sqrt{E} Eq. (7.60) between classifications and combined clustering. $E_1 = \sqrt{\|B^p(B^p)^T - A\|^2}$ - distance E from 50 classifications to each of them ($\min E_1 = 11.2$), $E_2 = \sqrt{\|BB^T - B^p(B^p)^T\|^2}$ - distance E from consensus clustering to each of 50 classifications ($\min E_2 = 10.2$), $E_3 = \sqrt{\|BB^T - A\|^2}$ - distance E from consensus clustering to 50 classifications ($E_3 = 6.2$),

We see from Figure 9.3 that none of the 50 classifications has zero error with all 50 classifications. It means that the 50 classifications are different. We also see from Figure 9.3 that the combined clustering has the lowest error ($\sqrt{E} = 6.2$) that says the combination is situated "at the middle" of classifications. Error \sqrt{E} can be interpreted as a consensus with mean error of 6 images, while other classifications give the minimal errors of 11 and 10 images. Combined clustering has 8 clusters and differs from any other given classification. Images of 8 clusters of combination are given in Figure 9.4. We observe from Figure 9.4 that images are grouped corresponding to some common sense or in the other words semantic context: scenes, objects, actions, etc.

Now let us represent relations among clusters issued from co-association matrix A as discussed in the previous section. These relations are graph and tree connections among clusters, see Figure 9.5a.

The degree of connections for the graph is reflected by the width of the edge: the wider the edge, the more important the connection. We see from Figures 9.4 and 9.5a that clusters which have close semantic meanings have more important degrees of connections. For example, cluster 8 (where we observe cows, mountains, sky and a building) has strong connection to clusters 1 (the cluster of animals), 3 (the cluster of urban landscape) and 5 (the cluster of payages). Another example, cluster 6 represented by boats is linked to cluster 5 and to cluster 7 which groups vehicles. It is meaningful links among discovered clusters. Also we observe the important link between cluster 3 (architecture) and clusters 4 (viewing as art creation). Here again this link is logical because there is no clear distinction between art and architectural creations.

The representation of relations between combined clusters by the tree is illustrated in Figure 9.5b. This figure proves experimentally the importance of connections among clusters illustrated in Figure 9.4. The tree in Figure 9.5b generalise different concepts into one single cluster at the highest level of tree. We note that in the case of several groups of clusters which have no connections it will produce several trees, each of which has its own meaning. Data represented by these trees will have no common sense provided by

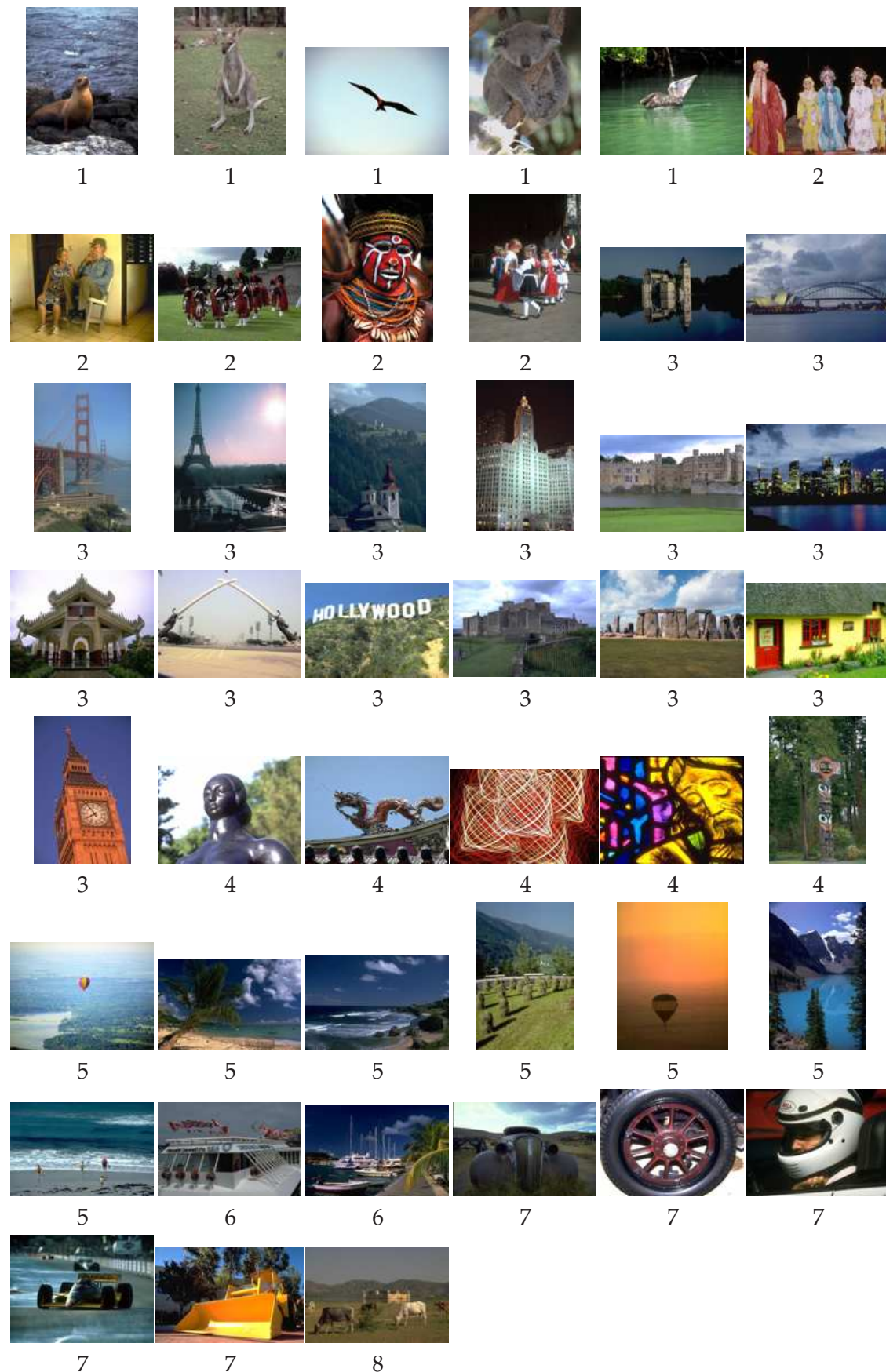


Figure 9.4: Combination of classifications. 8 clusters have been detected. Corresponding cluster label is shown under each of 45 images.

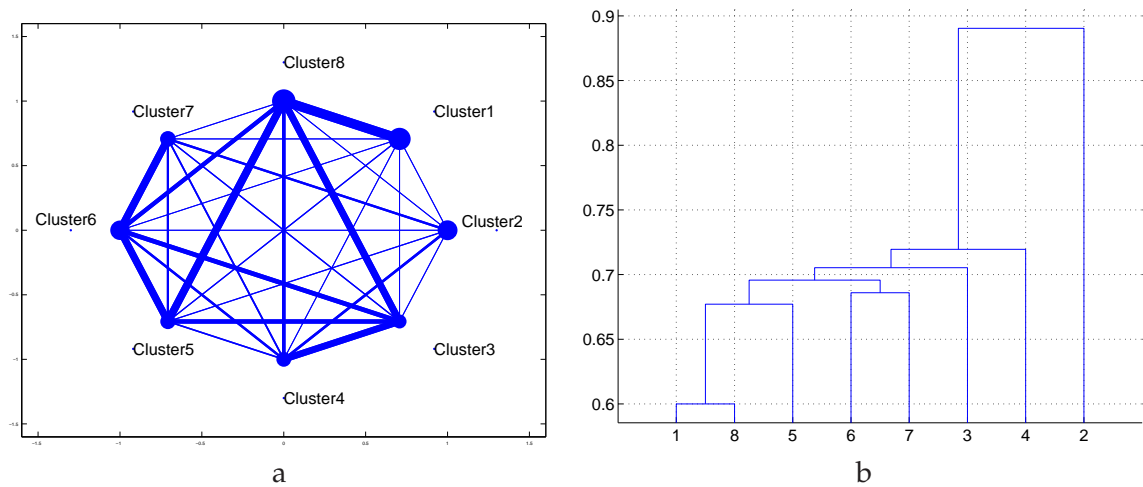


Figure 9.5: Connections among clusters of the combined classifications: a - graph, b - tree. Connections show semantic relations among clusters illustrated in Figure 9.4.

users.

Combining descriptions of classifications

In this part we explore the classifications of the multimedia images, but instead of combining labels of classifications we analyse the textual descriptions associated with these classifications. Remind that each user after having classified images with its own number of classes and its own classes, was asked to describe the meaning of each class with a set of words.

The goal of this experiment is to analyse and combine descriptions obtained from different users. As before we assume the result of combination provides semantic of images. In addition, we compare combination of classifications and combination of descriptions.

Combination of words is very similar to the text mining (clustering) and includes the following steps:

1. Text preprocessing. Elimination of articles, words with mistakes, coding words by labels, etc.
2. Text clustering. Choosing models and algorithms to cluster text data.
3. Representation of clustering results.

Below more precisely these three steps are discussed in order to combine descriptions of images.

Text processing

Here we operate with the same 50 classifications which have been analysed in the previous section. Words attributed only by the user are used to characterise classes. Descriptions of image classifications have been done mainly with English words. Mainly nouns have been selected while articles and endings have been removed manually to avoid mistakes, presence or absence of comas, etc. For a larger set of data, a programme of linguistic processing would probably be necessary written.

In the total we obtained 157 words (or group of words) to describe image classes. The extracted dictionary of image class descriptions is given in Appendix E.

Text clustering The textual descriptions are devoted to the clustering of the image database, using the words as descriptors.

As seen in Chapter 7 there are several methods to cluster nominal data. We propose to apply mean shift combination algorithm (see Section 7.5) which has shown promising results in Chapter 8. For that purpose we construct matrix B Eq. 7.1 of size $I \times J$ where $I = 45$ images and $J = 157$ words. We calculate how many times each word j is used in the 50 different descriptions of each image i . If image i contains word j then $B_{ij} = B_{ij} + 1$. If image i is never described by word j then $B_{ij} = 0$. This process is done for all 50 descriptions. To satisfy the definition conditions of coassociation matrix $A = B \cdot B'$ Eq. (7.33) we normalise matrix B such that every row B_i has a square norm equals 1. After data preprocessing matrix B is clustered by MSC - algorithm which finds 5 clusters. The result of clustering is presented in Figure 9.6.

Visually we observe that images in each cluster are semantically related. For example, cluster 1 contains all animals, cluster 2 represents persons. Cluster 3 shows architecture and art objects, while cluster 4 corresponds to landscapes. Finally, vehicles are grouped in cluster 5. Below we discuss in details results of word clustering.

Representation of results

We observe that the combination of words (Figure 9.6) produces almost the same result that the combination of visual classifications (Figure 9.4). It is an interesting and important conclusion because it shows the correspondence between the language representation and the visual appreciation of the images. It sustains the reliability of the experiment and of the proposed method of combination. However, the number of clusters of the text clustering equals 5 it is lower than when the combining visual classifications which provided 8 clusters. This may be explained by the fact that the same words describe different concepts and therefore reduce the variability of groups. For instance the clusters "architecture" and "art" from visual experiment are combined in the same cluster in word clustering. The extracted word description of each cluster is presented in Table E.1. From Table E.1 we see that cluster 3 mainly represents two categories "architecture" and "art". Words of this cluster are very similar by their sense.

In addition we propose to analyse clustering results represented by tree and graph in Figure 9.7.

We observe that cluster 3 has strong relation with cluster 4 in Figure 9.7a. It is also illustrated in the tree Figure 9.7b because these clusters are connected first. Connections between cluster 3 and 4 are justified by similar meaning of words in Table E.1.

Here, as in previous section where combining of classifications have been demonstrated, we observe on graphs and trees almost the same connections between discovered clusters. It shows that visual and text information reflect the same semantic representation or in other words the same meaning. It also confirms pertinence of the proposed unsupervised analysis.

Discussions

An experiment on combination of visual classifications and words descriptions of images has been presented in this Section. It opens new directions in data analysis. It also shows that the task of data mining can be solved by different approaches and illustrates concordance of data mining results. The interesting part of the experiment is that the analysis

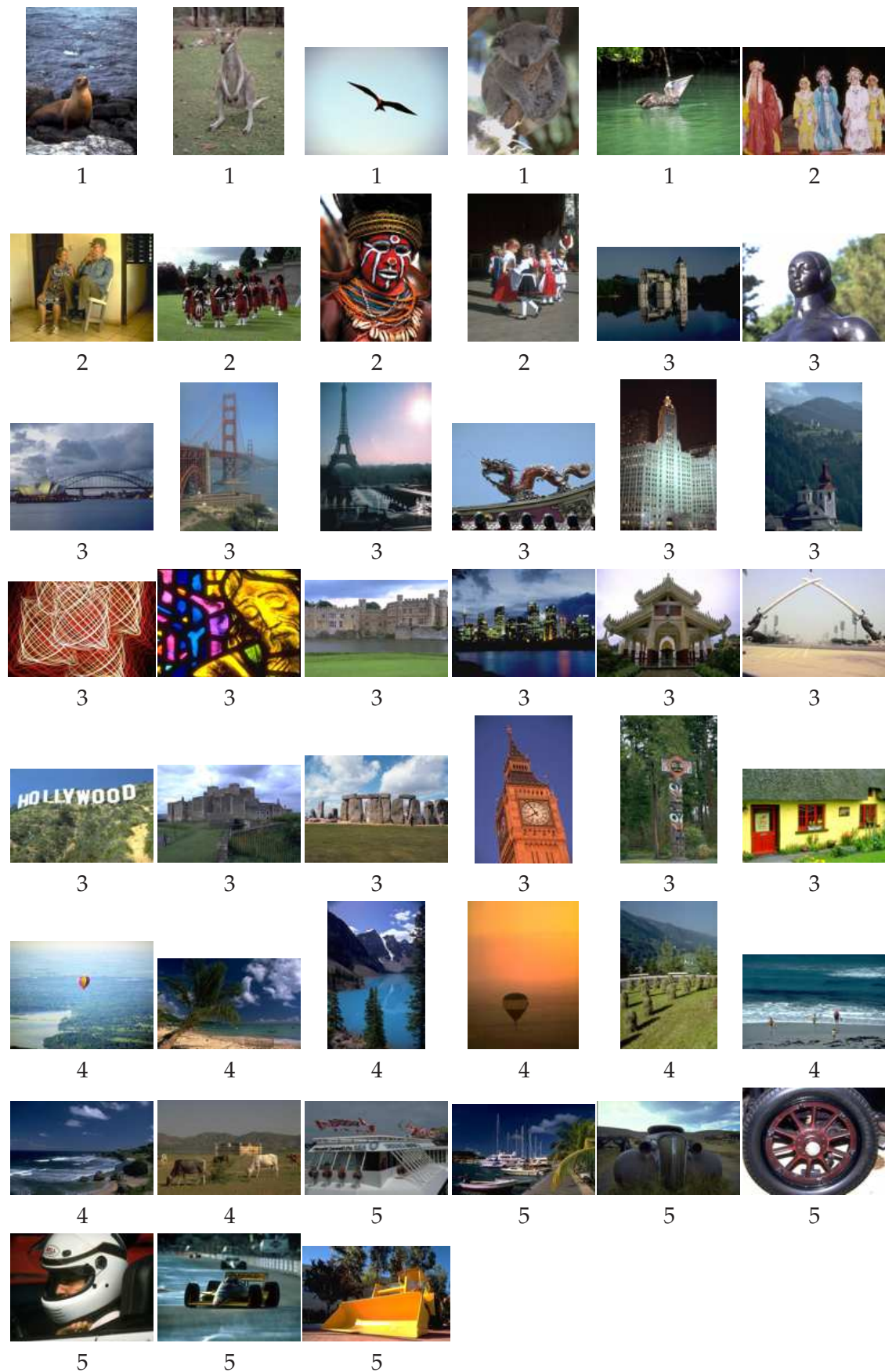


Figure 9.6: Clusterings combining of words of images. Ordered 45 multimedia images corresponding to labels of 5 optimal clusters.

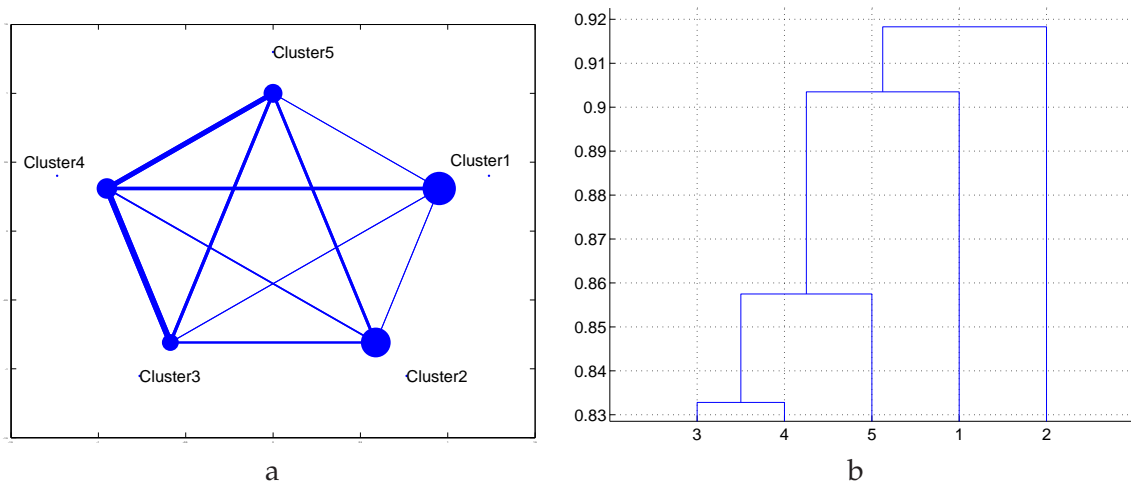


Figure 9.7: Connections among clusters of the combined classifications for the text representation: a - graph, b - tree.

of descriptions has been done in fully unsupervised way. We do not have a priori information on how many clusters exist and how many images are distributed in clusters.

We present a correspondence matrix for two clusterings : one is a combination of visual classifications, the other is a combination of textual descriptions, Table 9.1. 8 visual clusters and 5 word clusters are obtained in Sections 9.3 and 9.3, respectively. This matrix shows how many samples (images) are shared by any two clusters.

Table 9.1: Correspondence matrix of combined classifications

Word clusters	Visual clusters								Σ
	1	2	3	4	5	6	7	8	
1	5	0	0	0	0	0	0	0	5
2	0	5	0	0	0	0	0	0	5
3	0	0	15	5	0	0	0	0	20
4	0	0	0	0	7	0	0	1	8
5	0	0	0	0	0	2	5	0	7
Σ	5	5	15	5	7	2	5	1	45

We see from Table 9.1 that word cluster 3 contains samples from visual clusters 3 and 4 which corresponds to two semantic concepts of "architecture" and "art". Word cluster 4 has samples from cluster 5 and 8 which represents images of "paysage". Last word cluster 5 ("transport") merge visual clusters 6 ("boats") and 7 ("vehicles"). Table 9.1 shows that visual clusters are included into word clusters. It illustrates pertinence of the realised experiment and of the MSC combination method.

This approach may be tested on large volumes of multimedia images, e.g., taken from internet. Here clusters have been considered as concepts. Semantic data representation is able to indicate connections between concepts. It may be further exploited for more accurate searching or mining of data bases. Finally, a user may construct its own semantic classes via the analysis of cluster connections by the graph or tree representation. We

should note, that this kind of experiment is not limited to images and can be applied to different types of data.

9.4 Semantic construction for satellite images

In the previous Section we have successfully demonstrated how semantic may be constructed on the example of multimedia images. This experiment has been carried out partially in supervised way (manual image classifications) and unsupervised way (deriving semantic from classifications). A limited number of images has been considered (45) and different users have classified these images in different classes. From the combination of these classifications semantic grouping of images have been emerged (trees and graphs). In this section we propose to go more deeply in unsupervised image mining and to apply it on the large and mostly unknown data set of satellite images, in order to emerge similar semantic grouping.

For this experiment the participation of users is not involved and all operations are made in a fully unsupervised way. We give now the essential steps of the carried out experiment:

1. Feature extraction from satellite images.
2. Unsupervised feature selection.
3. Unsupervised data clustering by different clustering algorithms.
4. Unsupervised selection of the number of clusters for each algorithm.
5. Unsupervised combination of different clustering results.
6. Unsupervised building of satellite image semantic via representation of clustering combination.

We begin experiments by an illustrative example of unsupervised combination of satellite image clusterings. Then unsupervised satellite image clustering is demonstrated. Finally, several experiments of building of the satellite image semantic are proposed.

Combining of samples of satellite images

In this Section we demonstrate applications of different clustering algorithms to analyse satellite images in urban areas. Results of clustering combination are presented here. The goal of these experiments is to show that different points of view on the data can produce generalised results as compact clusters which may have semantic interpretation.

We perform experiments on 6 different SPOT5 satellite images at a resolution of 5 meters per pixel. Each of the 6 images has a size 1024×1024 pixels. They represent 6 world cities: Paris (France), Copenhagen (Denmark), La Paz (Mexico), Los Angeles (USA), Istanbul (Turkey), Madrid (Spain). We expect that each image exhibit different textures which reflect different historical, cultural and architectural configurations. Indeed each image is not homogeneous and each city exhibits different textures depending on the place in the city. We are interested in discovering similarities between cities, and to discover which part of a given city looks alike a part of another city. Those similarities will

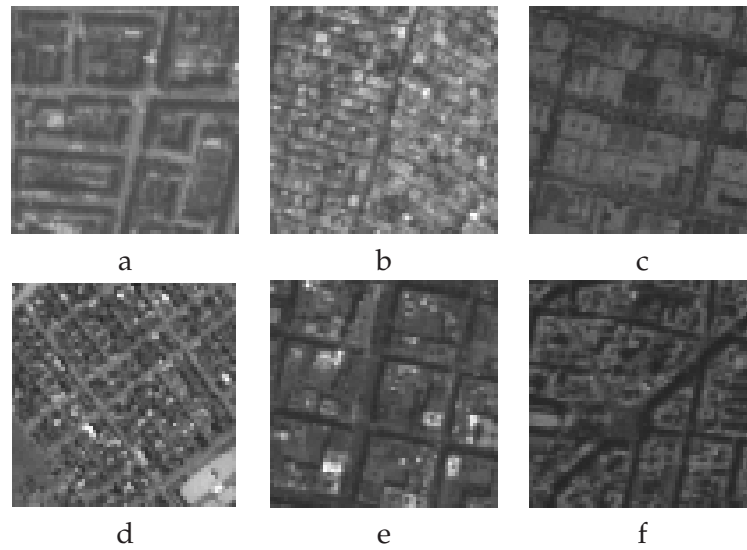


Figure 9.8: Textures of SPOT5 images: a - Copenhagen (Denmark), b - Istanbul (Turkey), c - Los Angeles (USA), d - La Paz (Mexico), e - Madrid (Spain), f - Paris (France). ©CNES

Classes						
1	2	3	4	5	6	Σ
30	12	335	0	138	20	535
8	1	2	314	9	6	340
273	134	0	0	3	0	410
89	246	1	0	5	5	346
0	0	42	81	19	339	481
0	7	20	5	226	30	288
400	400	400	400	400	400	

a

Classes						
1	2	3	4	5	6	Σ
0	0	31	68	16	328	443
11	1	3	327	9	11	362
0	9	36	5	260	46	356
132	266	1	0	7	6	412
237	116	0	0	3	0	356
20	8	329	0	105	9	471
400	400	400	400	400	400	

b

Classes						
1	2	3	4	5	6	Σ
139	269	1	0	7	6	422
17	8	333	0	101	2	461
0	9	39	5	262	35	350
11	1	4	333	9	12	370
233	113	0	0	3	0	349
0	0	23	62	18	345	448
400	400	400	400	400	400	

c

Classes						
1	2	3	4	5	6	Σ
297	317	0	0	0	0	614
0	0	296	0	7	254	557
1	6	88	71	76	130	372
0	0	1	325	0	3	329
0	0	12	0	299	0	311
102	77	3	4	18	13	217
400	400	400	400	400	400	

d

Classes						
1	2	3	4	5	6	Σ
37	8	311	88	104	363	911
136	91	0	0	0	0	227
22	66	1	0	4	6	99
0	12	86	5	281	27	411
205	223	0	0	3	1	432
0	0	2	307	8	3	320
400	400	400	400	400	400	

e

Classes						
1	2	3	4	5	6	Σ
240	116	0	0	3	0	359
132	266	1	0	5	6	410
0	9	36	5	262	27	339
17	8	336	0	107	14	482
11	1	2	333	9	10	366
0	0	25	62	14	343	444
400	400	400	400	400	400	

f

Table 9.2: Confusion matrices and clustering errors for 6 classes. a - *K - means* algorithm 28%, b - *Spectral K - mean* algorithm 27%, c - *Kernel K - means* algorithm 26%, d - *EM - algorithm* 38%, e - *Ward's hierarchical clustering* algorithm 42%, f - proposed combination algorithm for clusterings 26%

emerge from clusterings as presented in the previous chapters. Examples of images are presented in Figure 9.8.

To cluster these images we build a database of samples. For that we cut an image in 400 samples of size 64×64 pixels. Windows overlap by 13 pixels. As a result, we obtain a database of 2400 samples, grouped in 6 classes of 400 samples each. From each texture several features have been extracted. Features are: statistics of Quadratic Mirror Filters, statistics of Gabor filters and Haralick features, see Chapter 3. 10 features were selected from 185 [Campedel et al., 2004], see Section 8.8. We apply different unsupervised clustering algorithms to cluster matrix data of size 2400×10 : a classical *K-means* algorithm [Jain & Dubes, 1988], Spectral *K-means* algorithm [Ng et al., 2002], Kernel *K-means* algorithm [Shawe-Taylor & Cristianini, 2004], Ward's hierarchical clustering algorithm [Jain & Dubes, 1988] and *Expectation-Maximisation* algorithm with a Gaussian mixture model, see Chapter 5. To cluster data we set the fixed numbers of clusters to 6 since we know that 6 cities are represented. Clustering results are presented as confusion matrices in Tables 9.2 a-e.

In this Section we set the number of clusters equal for every algorithm. But in following subsections the optimal number of clusters is estimated. Here, estimation of clustering quality may be given by the percentage of samples which are wrongly clustered in the wrong class and all others samples in this cluster are set as misclassified.

From the confusion matrices in Tables 9.2 a-e we see that for some classes different algorithms give different clustering solutions. All clusterings have redundant information but at the same time their intersections can generate new informative clusters. To analyse intersections between all clusters is a very difficult task. In Table 9.2 f we perform the consensus combination as presented in Chapter 7 to generate a common result. We see from Table 9.2 f that this consensus combination provides results as good as the best single classification (26% error). It confirms that clustering combining produces reasonable results.

Unsupervised image clustering of urban content (QuickBird)

In this Section unsupervised clustering of a high resolution image is demonstrated. We have presented above experiments made on SPOT5 images. Nowadays, new satellites produce images of very high quality and resolution. One of that satellites is QuickBird from which we obtained images with a resolution of $0.6m$ per pixel and a size of 24000×24000 pixels. An small piece of such an image is presented in Figure 9.9. It is a QuickBird image of Las Vegas of size 3000×3000 pixels.

This image corresponds to a quarter of private houses. Visually there are houses with different roof structures and colours. We aim to show that a fully unsupervised clustering may determine different quarters with houses which have similar roofs.

We cut the original image into small images of size 64×64 pixels and extract features, see Chapter 3. Unsupervised feature selection expressed in Section 8.8 is applied and selects the best set of 32 features. We only show results of clustering of the image by K-means and EM algorithm with the Gaussian mixture model. MDL criterion is used to estimate the optimal number of clusters, see Figure 9.10.

MDL curve presented in Figure 9.10b shows that EM-algorithm determines less clusters than K-means algorithm, see Figure 9.10a. It is reasonable because EM algorithm with the GMM model is more adaptive to model data and consequently has more simpler model than the model of K-means algorithm.



Figure 9.9: QuickBird satellite image of city Las Vegas with a resolution $0.6m$ per pixel and size 3000×3000 pixels. ©CNES

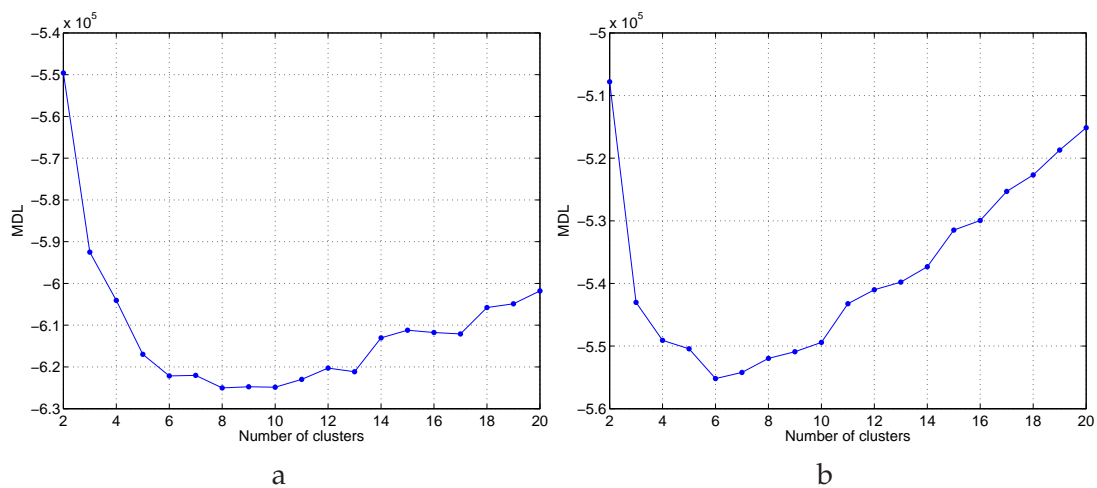


Figure 9.10: MDL curve to determine the optimal number of clusters for image in Figure 9.9. a - MDL criterion for K-means algorithm, the optimal number of cluster equals 8; b - MDL criterion for EM-algorithm with GMM, the optimal number of cluster equals 6.

Result of image clustering is shown in Figure 9.11. Clusters are displayed with different colours. As the size of clustered image is smaller than the size of the image we interpolate labels to have the same size as the original images.

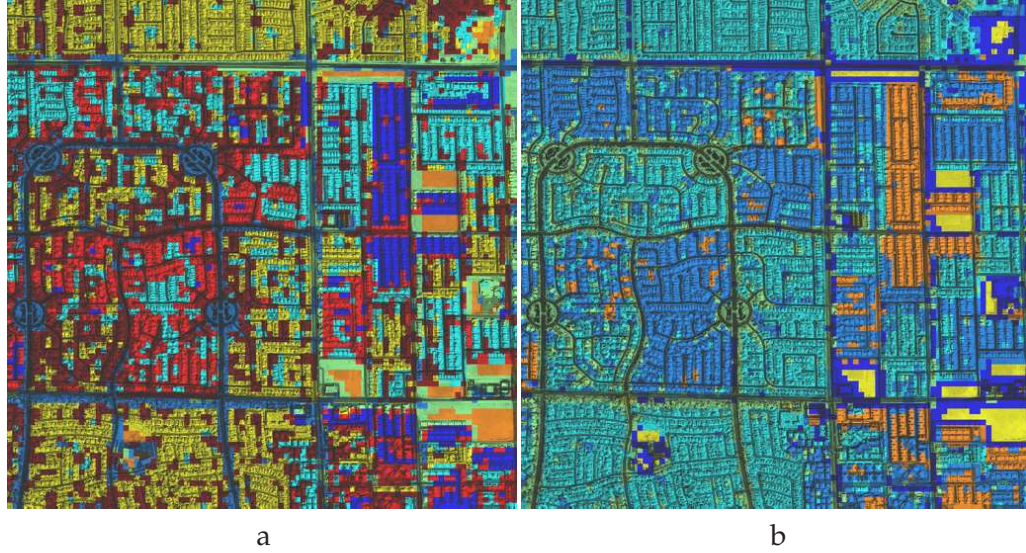


Figure 9.11: Optimal clustering of image in Figure 9.9: a - Kmeans algorithm with 8 clusters, MDL is in Figure 9.10a. b - EM-algorithm and GMM with 6 clusters, MDL is in Figure 9.10b.

From the Figures 9.11a and 9.11b we see that we have detected clusters which correspond to different urban squares with different forms of houses. Interesting, that for the high resolution image we detect a main road as a separate cluster. K-means clustering gives a little bit perturbed clustering, in Figure 9.11a, which is comparable with EM-algorithm, in Figure 9.11b. It demonstrates that EM-algorithm with GMM model better clusters than K-means algorithm.

Examples of textures of each clusters for K-means and EM-algorithm with GMM are presented in Figures 9.12 and 9.13, respectively.

Figure 9.13 represent textures of 6 clusters detected by EM-algorithm with GMM and estimated by MDL criteria.

Taking into account that there are many different algorithms each of which gives different clustering we are interested in combining different results to obtain a general clustering. In addition, we aim to get a semantic representation of satellite images via clustering combining. Below we demonstrate two examples of unsupervised semantic construction for satellite images with general and urban content.

Satellite image of general content (SPOT 5)

Here we propose to analyse a SPOT5 satellite image of Bezier city located in the South of France. This image presented in Figure 9.14 has a resolution of 5 meters per pixel and a size of 3000×3000 pixels.

As we see from Figure 9.14 half of the image is covered with sea and the other half is covered with a continental part. The continental part has city regions and wide surfaces

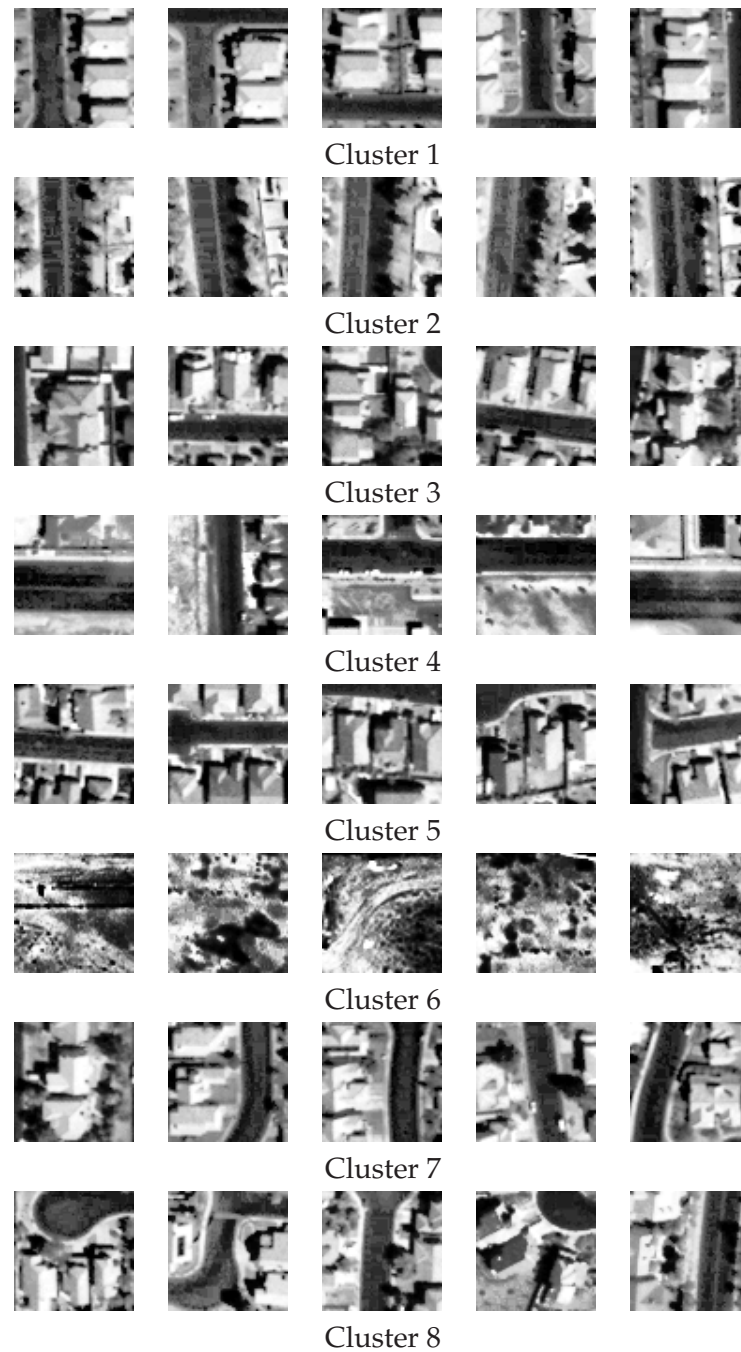


Figure 9.12: Examples of texture clusters of Las Vegas clustered by K-means algorithm, see Figure 9.11a. The optimal number of clusters detected by MDL criterion equals 8, see Figure 9.10a. ©CNES

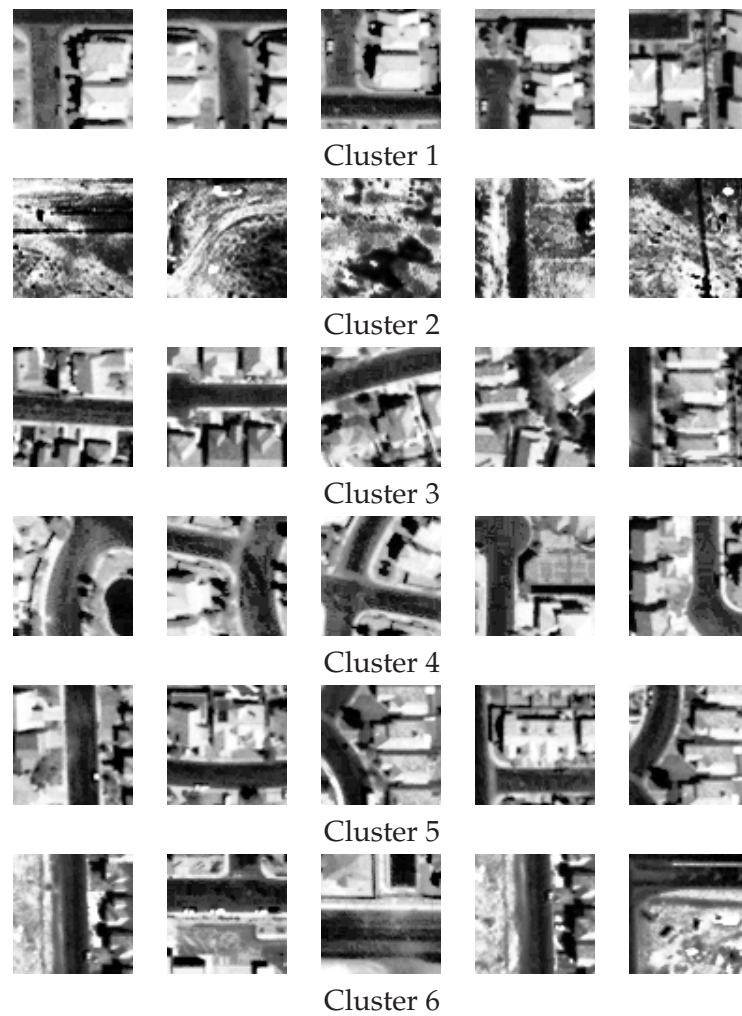


Figure 9.13: Examples of texture clusters of Las Vegas clustered by EM-algorithm with GMM, see Figure 9.11b. The optimal number of clusters detected by MDL criterion equals 6, see Figure 9.10b. ©CNES



Figure 9.14: SPOT 5 satellite image of city Bezier, South of France (©CNES).

of fields.

In previous sections the semantic has been extracted from multimedia images via different supervised classifications. Here instead of classifications we propose to process unsupervised clusterings. They can be obtained by three strategies:

1. from one clustering algorithm with different conditions (number of clusters, initialisations, etc.);
2. from different clustering algorithms (each has the optimal clusterings);
3. from different descriptions of the same data with one or different clustering algorithms.

The first approach has been considered in the work of [Fred & Jain, 2005], as the direct combination of clusterings for different numbers of clusters and initialisations. A drawback of this approach is that it may result as many clusters as is used for clustering: the more clusters are used to cluster data the more clusters are after combining. Moreover, there was no indication how to estimate the number of clusters for a clustering algorithm.

We advocate for the second approach and propose to select the best optimal model, i.e., the best clustering for each algorithm. In that way we are sure to combine optimal clusterings. This approach mainly concerns the quality of clustering algorithm and the selection of the best clustering result. MDL criteria estimates the optimal number of clusters, the best initialisation and the best clustering for each algorithm.

The third approach is interesting but is not studied in the thesis. On the contrary, in our experiment, all features are involved at once to cluster data. We suppose that a

full set of features better describes data than any isolated subpart. However, to eliminate redundant features a feature selection procedure can be applied.

Now we propose a schema to extract semantics from data (satellite images). It includes the following steps:

1. image clustering by different algorithms, each of them giving the optimal clustering,
2. combination of clustering results,
3. representation of image semantic by a graph and a tree.

We follow the mining schema presented in the previous Section, but with unsupervised clusterings instead of supervised classifications. All unsupervised clusterings are obtained by different clustering algorithms.

Now we detail how images are clustered. The first step is the data representation. This means cutting a big satellite image into small subimages called samples. Each sample has a size of 64×64 pixels, and a window which cut samples is sliding with a step of 32 pixels from left to right and from top to bottom of the image. In the total there is 8464 subimages for the given size of window, image and step. It corresponds to 92×92 samples.

The second step is the feature extraction from 8464 images of size 64×64 pixels. These features have been described in Chapter 3 and constitute the following groups of features:

1. Gabor features,
2. geometrical features,
3. Haralick features,
4. QMF features.

In the total 134 features have been extracted. At this step we have obtained a dataset to be clustered. Let this data be noted as matrix X of size 8464×134 .

The dimension of the space equals 134 that is very high for data clustering. This problem of the "curse of dimensionality" is explained in Section 4.2. There is a need to select features. We emphasise that features are selected in an unsupervised objective way to avoid subjective interpretation of data. It is also done to avoid changing of clustering results from one user to another.

An approach of unsupervised feature selection procedure is proposed in the work of [Campedel et al., 2007], and explained in Section 8.8. In the work [Campedel et al., 2007] the number of clusters for a feature clustering has been chosen from 2 to the size of features. Then the LSEC-algorithm (Section 7.4) has been applied to different clusterings to find a consensus clustering and the optimal number of feature clusters. Here we propose a slightly modified approach: we run K-means for the number of clusters changed from 2 to the half of the total number of features. As we have seen from Section 7.6, the combination approach has a tendency of self organising. It means that if we cluster data with smaller number of clusters but the true number of clusters is higher than used for clustering, then clustering combination algorithm will tend to detect the number of clusters which is near the true number. As we will see later the number of clusters detected after

the combination of clustered features is near the half of the maximal size of clusters (the half of the number of features). That is why we have run K-mean for a number of clusters from 2 to one half of the number of features.

We cluster 134 features with K-means algorithm and the number of clusters from 2 to 67 with 3 random initialisations. Then MSC-algorithm of clustering combination is applied to find the consensus clustering. Stable features are selected as representative. The stability is computed as S Eq.(7.75). As the result 31 stable features have been selected.

Next step is data clustering, starting with 8464 samples and 31 features. There is two critical problems : (i) the choice of the number of clusters and (ii) the initialisation process. To chose the optimal number of clusters we propose to use MDL criterion, which also solves the problem of the initialisation by selecting the best model with the minimal value of MDL. MDL criterion is computed for different number of clusters from 2 to 20 and for 5 random initialisations. For a given number of clusters, the minimal MDL value corresponds to the optimal clustering among 5 ones obtained with random initialisation. Finally, the best clustering has minimal MDL and indicates the best number of clusters.

The figure of MDL estimating the number of clusters for 7 different algorithms is presented in Figure 9.15.

MDL (Figure 9.15) exhibit a regular curve (without a noisy behaviour) and shows that the optimal number of clusters is: 8 for K-means algorithm, 5 for EM-algorithm with GMM, 6 for spectral K-means algorithm, 4 for kernel K-means algorithm, 6 for Ward's hierarchical algorithm, 5 for the complete-link hierarchical algorithm and 7 for the average-link hierarchical algorithm. The analysis of the number of clusters corresponds to expected results: simpler clustering algorithms have higher number of clusters. For example, MDL for EM and GMM Figure 9.15b shows lower number of clusters than MDL obtained by K-means algorithm Figure 9.15a. It is explained by the fact that more complex models tend to fit data with simpler models of clusters. As a conclusion: the complex hypothesis provides the simplest model (GMM fits data simpler than K-means). For each algorithm we select the best clustering which corresponds to the minimum value of MDL criterion. Further we process these 7 optimal clusterings only.

To visualise clustering results we interpolate each clustering (labels) to the size of the original image and superpose these two images. The original image has a size of 3000×3000 pixels while the size of samples is 8464 which corresponds to a size 92×92 pixels. We use a symbolic interpolation that is the repetition several times of the same label in horizontal and vertical directions. Examples of optimal clusterings is shown in Figures 9.16 and 9.17.

Clustering results presented in Figure 9.16 and 9.17 provide very good results from the point of view image interpretation. We easily discover the cluster of fields, different zones of urban areas, different zones of sea and a coast line. In Figures 9.16 and 9.17 the cluster of city is clearly separated from the cluster of fields. Another very interesting area is two detected clusters on the sea area: cluster of the sea and cluster of the shallow water enveloped by the coast, see Figures 9.16a-d and Figure 9.17a. In the internal part of the water, which is enveloped by the coast there is a cluster of sea activities: fishing farms. A sample of this cluster is presented by the first image of cluster 3 in Figure 9.18. It is very important aspect of unsupervised image clustering, because it is not visible on the original image in Figure 9.14 but is detected by the clustering.

The next step of unsupervised image mining is combination of clustering results. The optimal combination of 7 clusterings has been performed by MSC algorithm and found an optimal clustering with 6 clusters. Results of the combination which have been in-

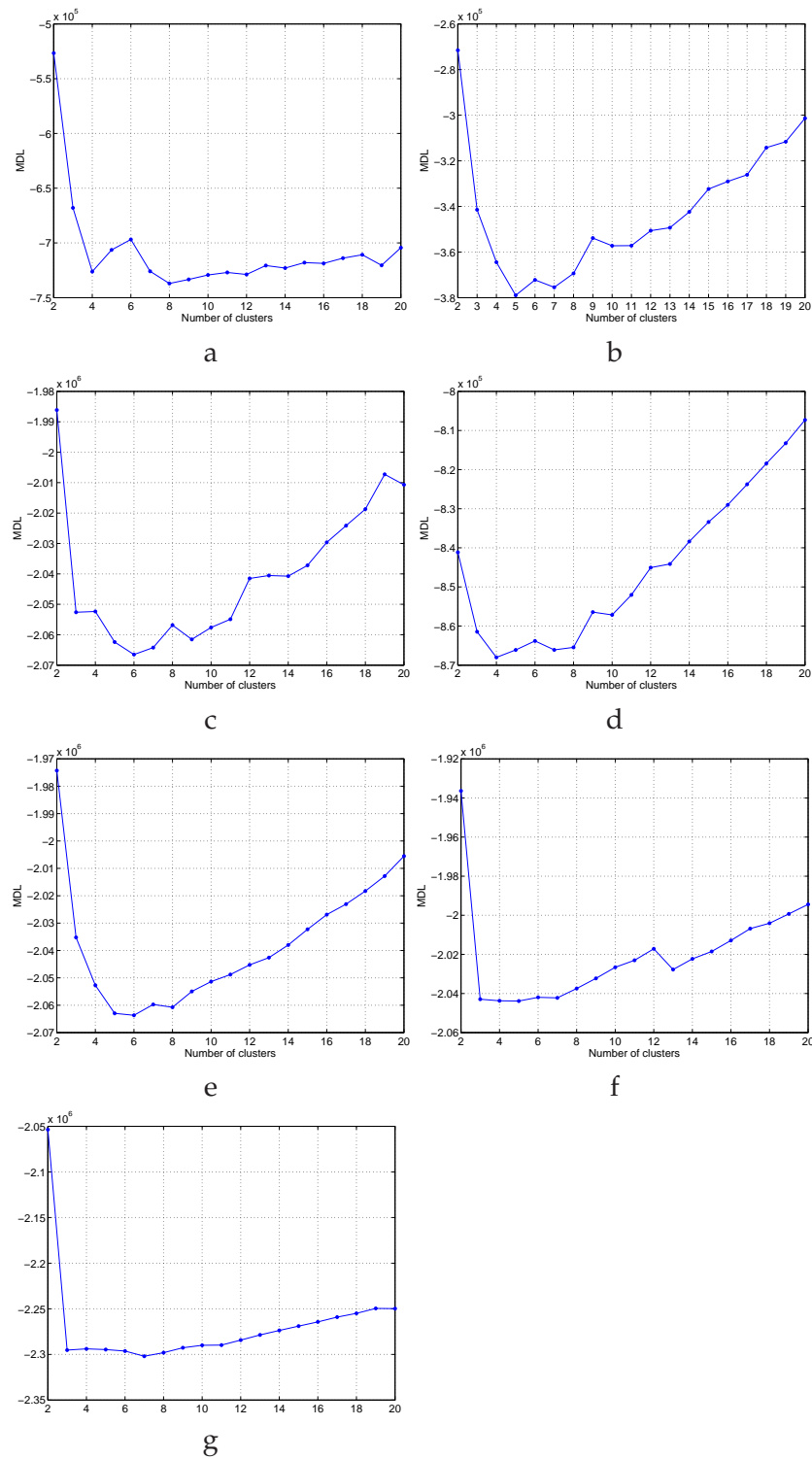


Figure 9.15: MDL curve to determine the optimal number of clusters: a - K-means algorithm 8 clusters, b - EM-algorithm with GMM 5 clusters, c - Spectral K-means algorithm 6 clusters, d - Kernel K-means algorithm 4 clusters, e - Ward's hierarchical algorithm 6 clusters, f - complete-link hierarchical algorithm 5 clusters, g - average-link hierarchical algorithm 7 clusters.

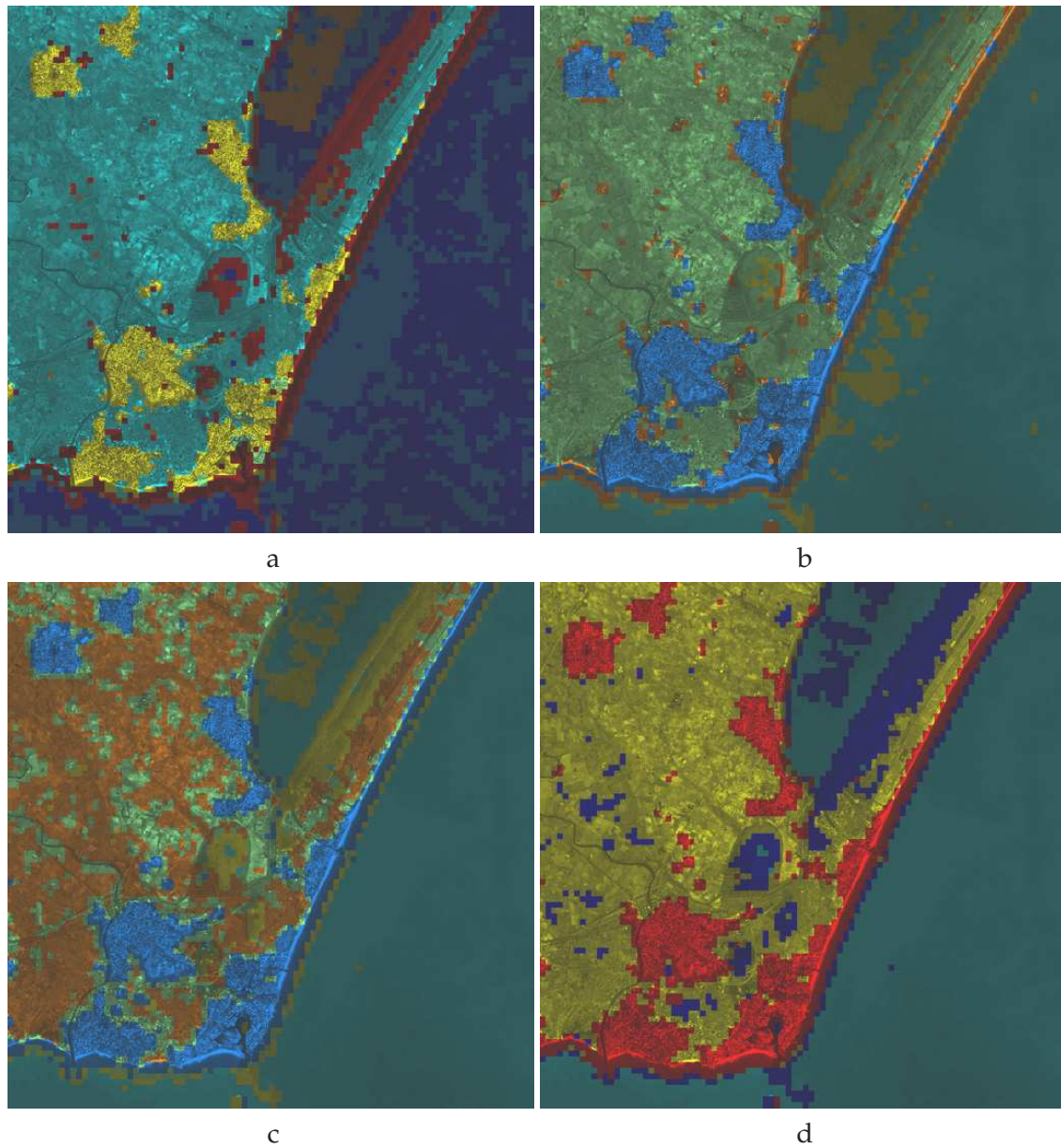


Figure 9.16: Optimal clusterings of SPOT5 image of Bezier: a - K-means algorithm 8 clusters, b - EM-algorithm with GMM 5 clusters, c - Spectral K-means algorithm 6 clusters, d - Kernel K-means algorithm 4 clusters,

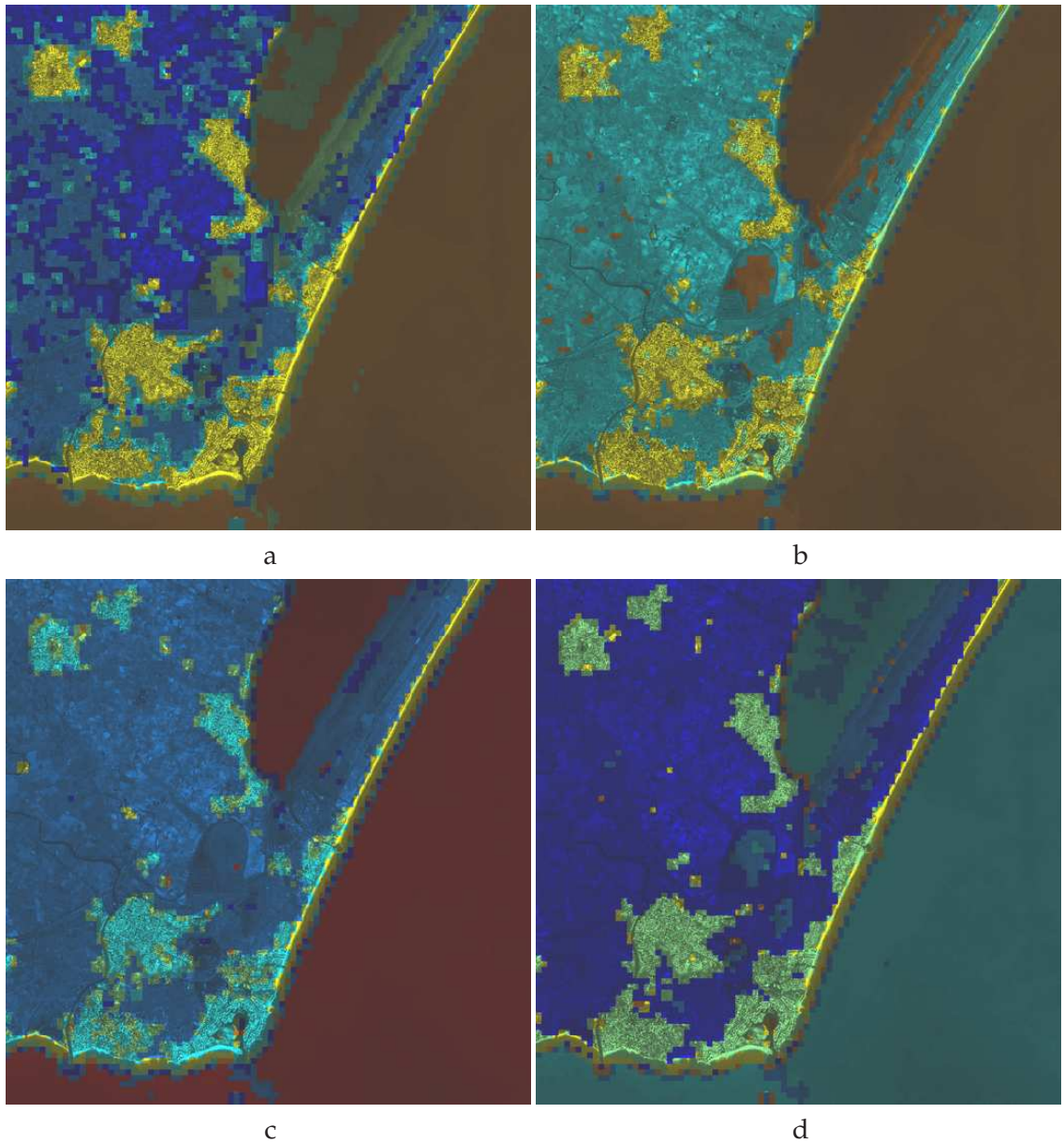


Figure 9.17: Optimal clusterings of SPOT5 image of Bezier: a - Ward's hierarchical algorithm 6 clusters, b - complete-link hierarchical algorithm 5 clusters, c - average-link hierarchical algorithm 7 clusters. d - combination of 7 optimal clusterings by MSC algorithm (6 clusters have been detected).

terpolated as in previous case by symbolic interpolation is presented in Figure 9.17d. Examples of each cluster are presented in Figure 9.18.

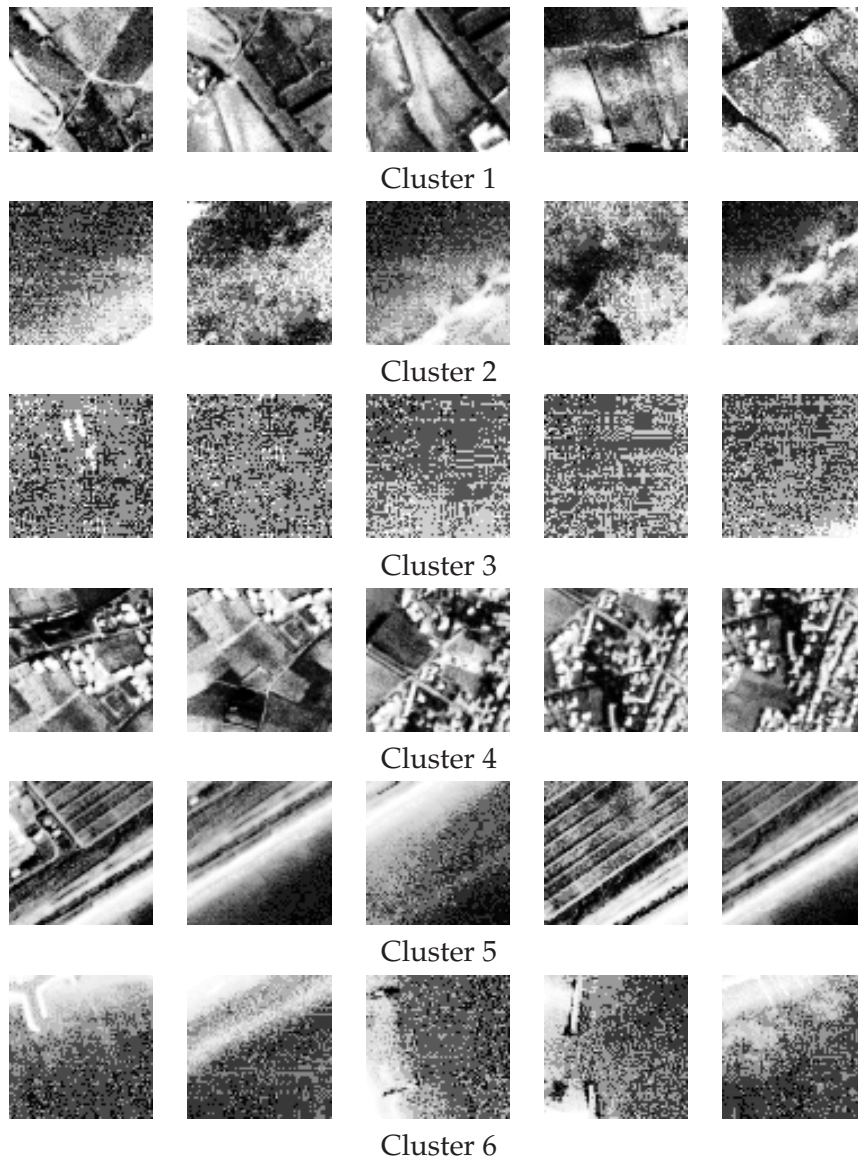


Figure 9.18: 6 clusters detected by clustering combination of Bezier image, see Figure 9.17d.

The combination of clusterings see Figure 9.17d is similar to 7 optimal clusterings see Figure 9.16a-d and 9.17a-c.

The next step is the construction of the semantic links among concepts where each concept corresponds to a cluster after combination. Links are presented by tree and graph structures and displayed in Figure 9.19.

It can be observed from Figure 9.19 that clusters after combination represent semantic concepts and relations among these concepts show logical links. Tree representation helps to show generalisation of concepts up to one single meaning, see Figure 9.19a, while the graph structure shows all possible semantic connections among concepts, see Figure

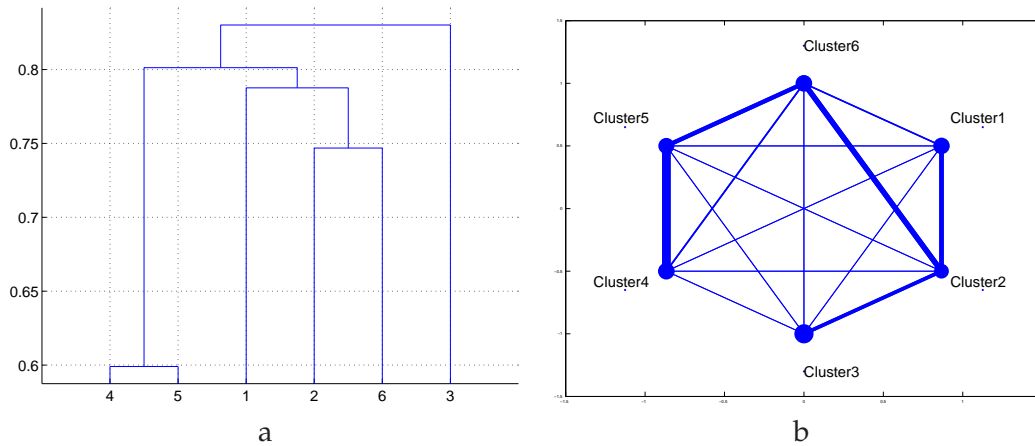


Figure 9.19: Semantic connections among concepts (clusters of combination presented in Figure 9.17d): a - tree connections. b - graph connections.

9.19b. These two different representation help the user to find and construct new clusters or even to understand some hidden phenomena of surface organisation.

Unsupervised construction of the semantic for satellite image with common content has been demonstrated in this Section. The image has a variety of surfaces which have been separated by unsupervised clustering: city, field, sea, etc.. In the next subsection an image with more complex content is considered. This content represents urban zones which are difficult to distinguish in a semantic sense.

Satellite image of urban areas (SPOT 5)

In this Section as in the previous we construct semantic for a satellite image. Here we consider the image of Paris which represent a complex urban content. The image is issued from SPOT5 at a resolution of 5 meters per pixel and with a size of 3000×3000 pixels, see Figure 9.20.

As in previous cases we use the same protocol for satellite image mining (Section 9.4). Firstly, we divide the image into samples. Then 134 features are extracted from each sample and 32 most important of them are selected in the unsupervised way. Then different algorithms are applied to cluster data: K-means, EM-algorithm, spectral K-means, hierarchical algorithms (Ward, complete-link and average-link). For iterative clustering algorithms different random initialisations are used. MDL criterion is used as before to select the best clustering for a given number of clusters. This criterion also determines the optimal number of clusters as shown in Figure 9.21a-g.

It can be observed from Figure 9.21a and b that the number of clusters for EM-algorithm is less than for K-means algorithm.

For the sake of visualisation, clusterings are interpolated and superimposed with the original images. Results of image clustering with different algorithms are presented in Figures 9.22 and 9.23.

We may note from Figures 9.22 and 9.23 that some clusters are found by every algorithms, e.g., forest, clouds. But each algorithm gives different partitions of urban areas (downtown and suburbs). It is explained that urban areas is more complex and a different algorithms finds different partitions of urban zones.

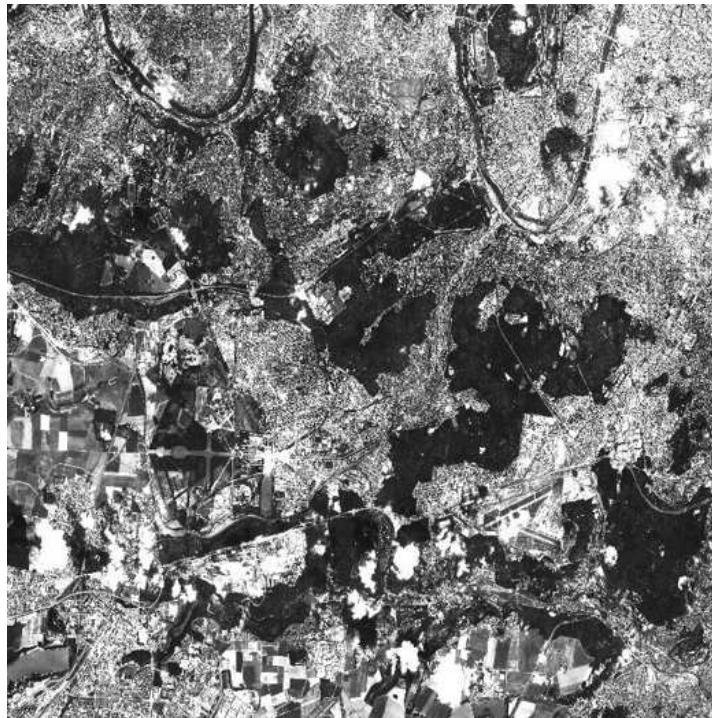


Figure 9.20: SPOT 5 satellite image of city Paris ©CNES.

The optimal combination of clusterings has been found by MSC algorithm and contains 16 clusters, see Figure 9.23d.

Examples of each cluster are presented in Figures 9.24 and 9.25. We observe from Figure 9.24 that cluster 1 represents images of river in urban area, while cluster 4 has samples of river in rural area. Cluster 2 shows examples of areas with private houses, and cluster 3 corresponds to downtown of Paris. Cluster 5 has open surface with trees, and cluster 8 has samples with trees which are more dense. Cluster 6 has samples with a half of white and a half of black parts which may be classified as fields, however we may note samples with clouds. Cluster 7 has rather commercial zones with buildings and wide roads.

From Figure 9.25 we see clearly that cluster 15 and 16 represent clouds (a histogram equalisation has been used to visualise samples, that sometimes may confuse visual observation). Cluster 10 has samples of forest, cluster 14 has less trees in the urban zone, and clusters 13 has some part of trees. Cluster 11 has images of fields with strong straight lines. Cluster 12 corresponds to urban area as well as cluster 9 with some artefacts as clouds.

The semantic representation of combined satellite image clusterings is given in Figures 9.26a-b. Tree and graph representations of the combination of different clusterings reflect the common information among clusters, see Figures 9.25 and 9.26. For example, clusters 2 in Figure 9.24 and 14 in Figure 9.25 are first connected by tree structure in Figure 9.26a. Clusters 15 and 16 in Figure 9.25 are connected by graph in Figure 9.26b and by tree in Figure 9.26a. We may conclude that clusters have reasonable connections for the tree as well as for the graph structure.

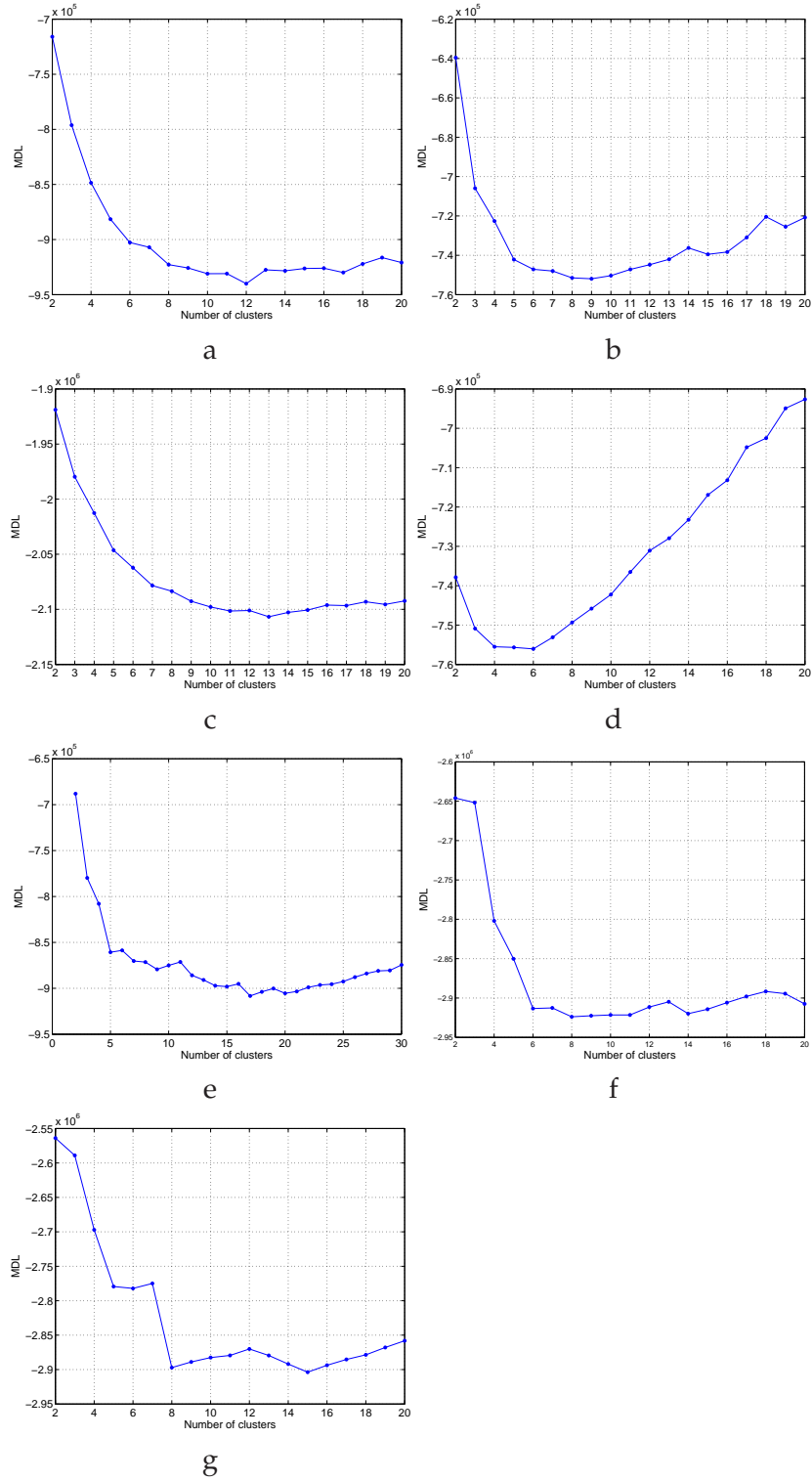


Figure 9.21: MDL curve to determine the optimal number of clusters: a - K-means algorithm 12 clusters, b - EM-algorithm with GMM 9 clusters, c - Spectral K-means algorithm 13 clusters, d - Kernel K-means algorithm 6 clusters, e - Ward's hierarchical algorithm 17 clusters, f - complete-link hierarchical algorithm 8 clusters, g - average-link hierarchical algorithm 15 clusters.

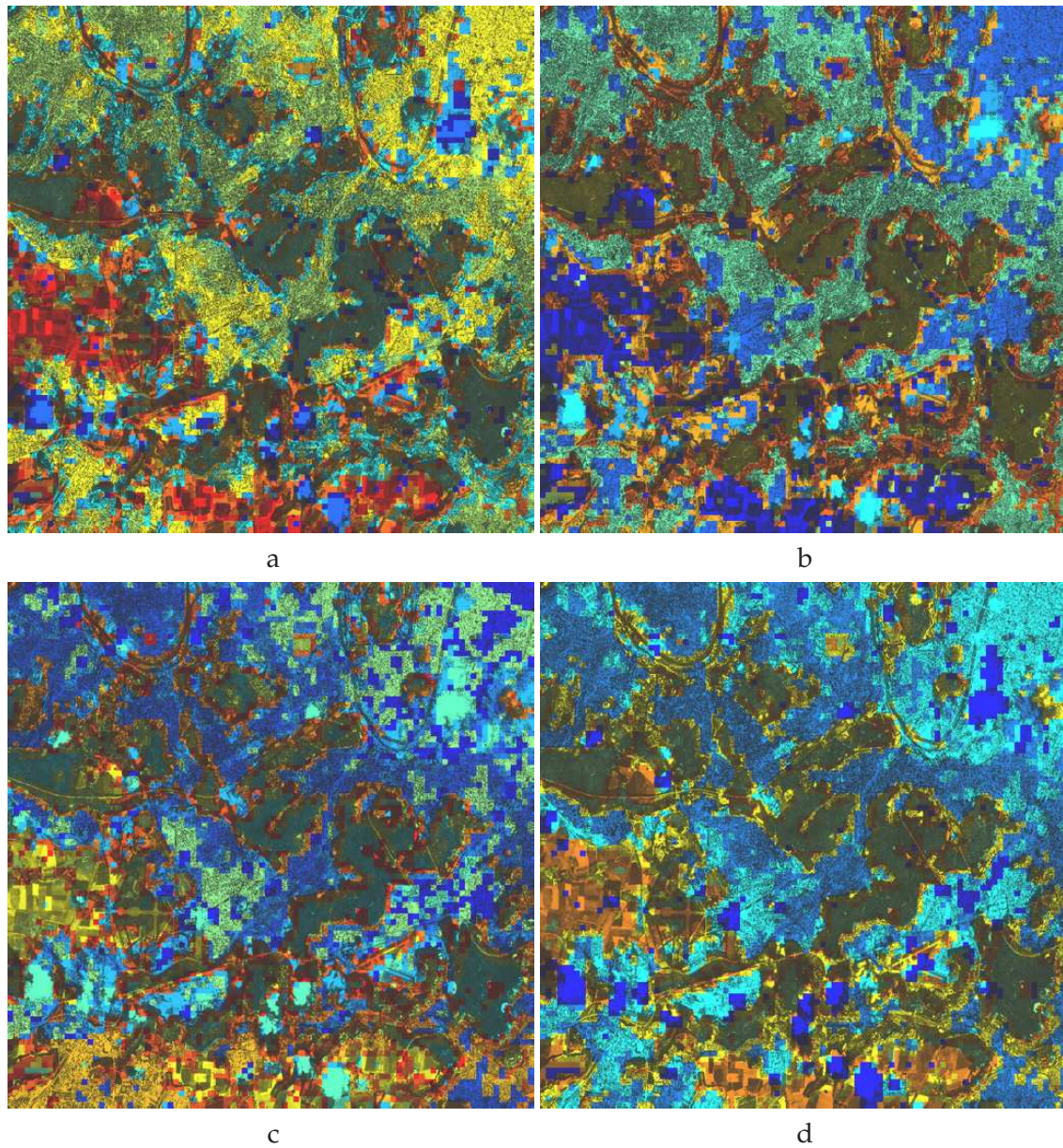


Figure 9.22: Optimal clusterings of SPOT5 image of Paris: a - K-means algorithm 12 clusters, b - EM-algorithm with GMM 9 clusters, c - Spectral K-means algorithm 13 clusters, d - Kernel K-means algorithm 6 clusters

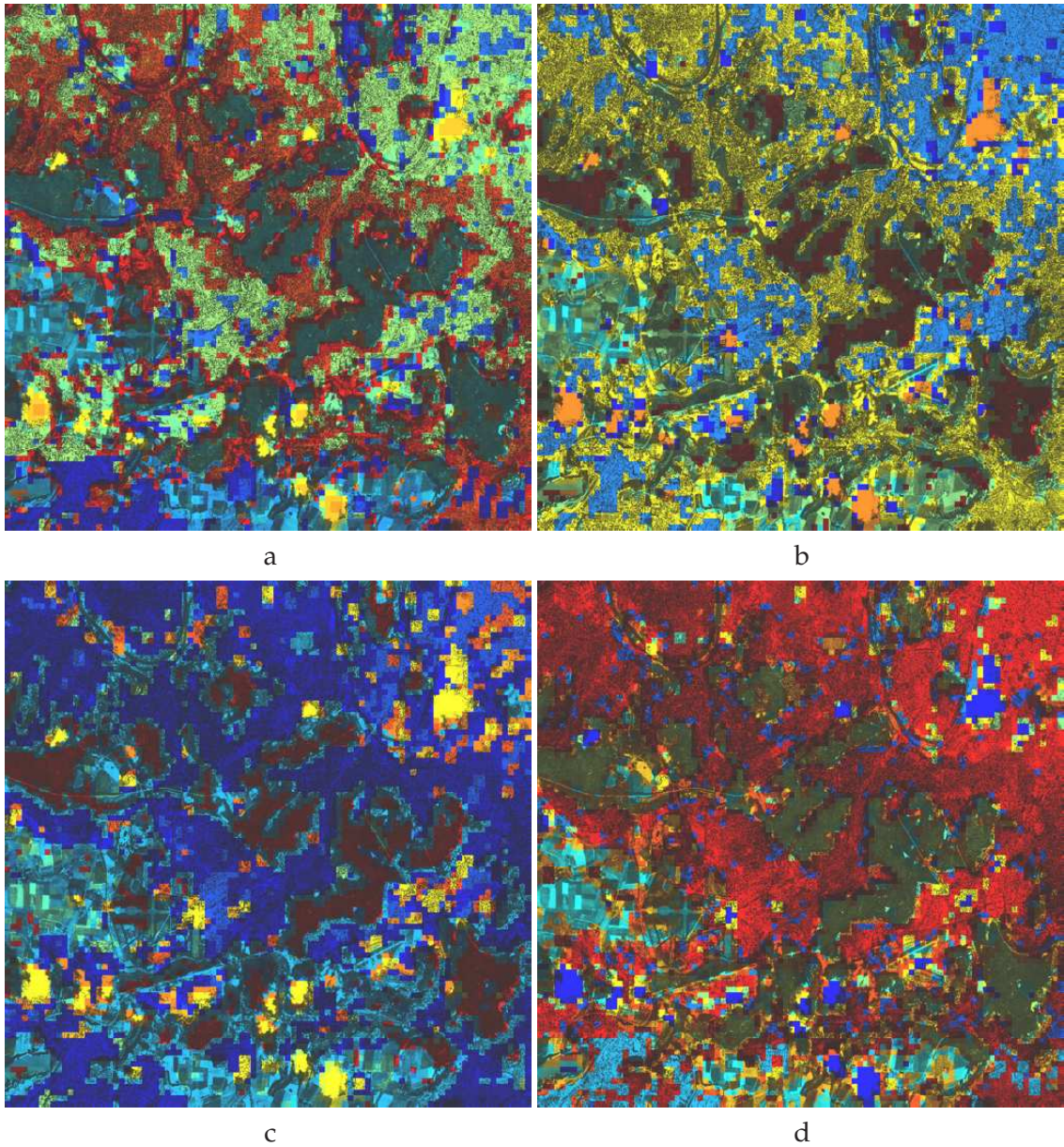


Figure 9.23: Optimal clusterings of SPOT5 image of Paris: a - Ward's hierarchical algorithm 17 clusters, b - complete-link hierarchical algorithm 8 clusters, c - average-link hierarchical algorithm 15 clusters. d - combination of different optimal clusterings by MSC algorithm (16 clusters have been detected).

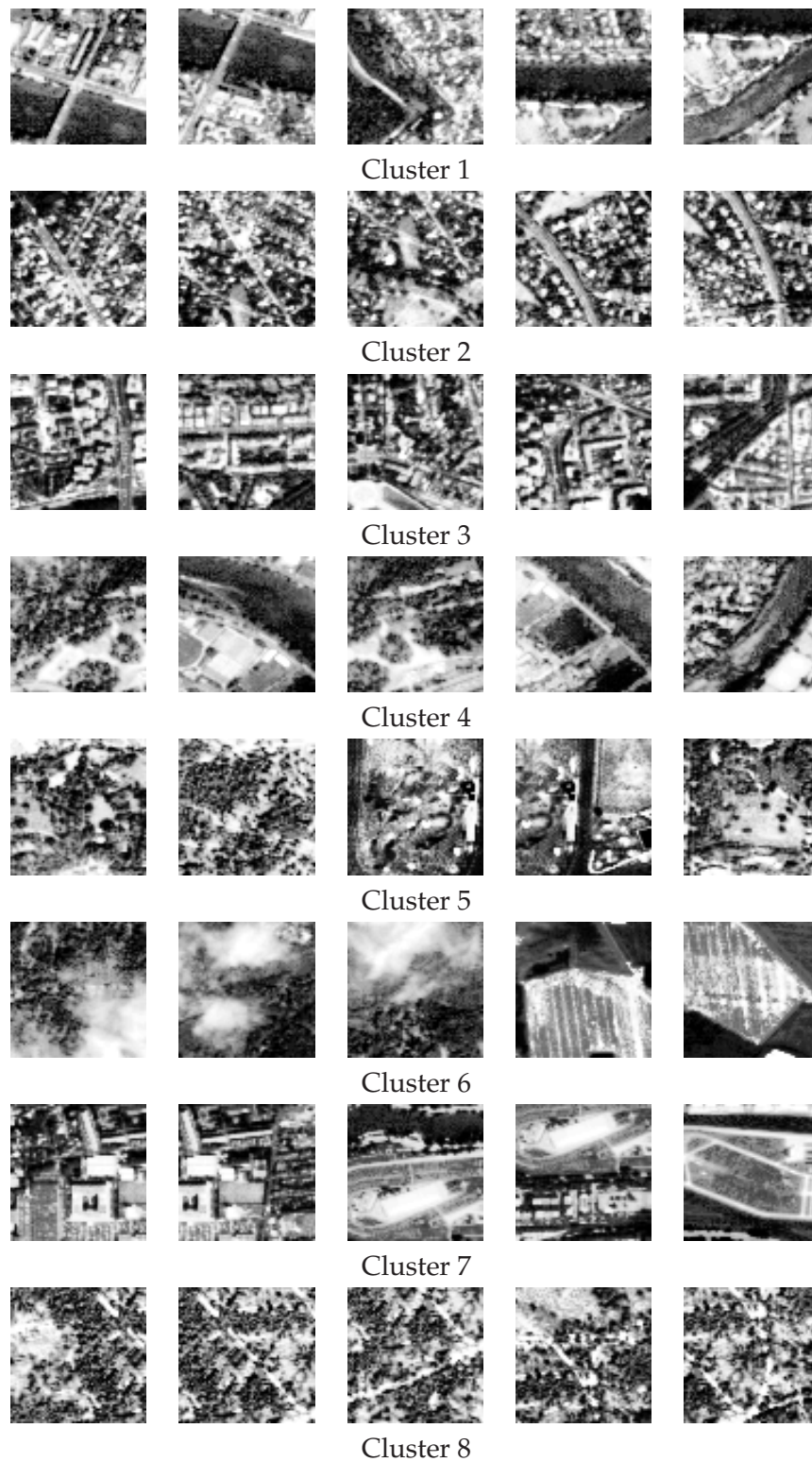


Figure 9.24: Clusters 1 – 8 detected by combination of clusterings of Paris.



Figure 9.25: Clusters 9 – 16 detected by combination of clusterings of Paris.

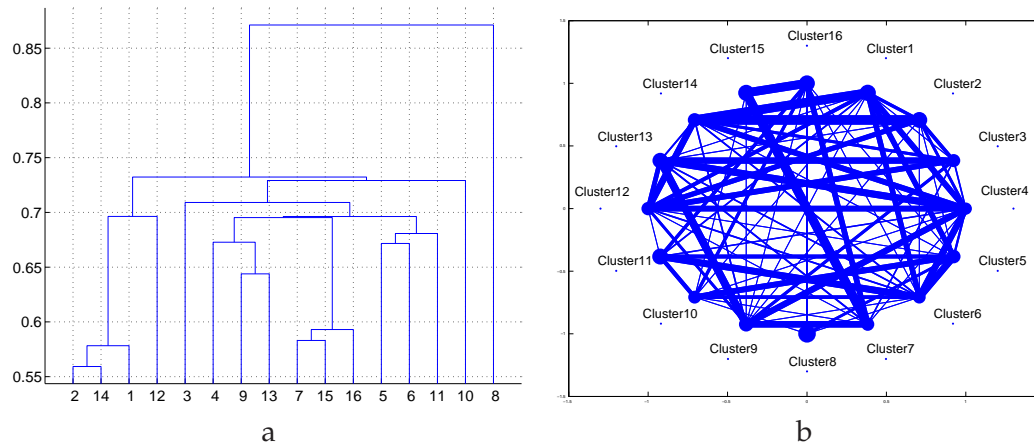


Figure 9.26: Semantic connections among concepts (clusters of combination presented in 9.23d): a - tree, b - graph.

9.5 Conclusions

In this Chapter the notion of image semantic, its principles, construction and analysis have been presented. Examples of semantic are shown on multimedia and satellite images. We have demonstrated how to derive image semantic in the unsupervised way and justified it with prior knowledge about images. For multimedia images both the visual perception and vocabulary descriptions have shown pertinent semantic. Satellite image semantic has been approved by visual interpretation.

Visualisation of combination results has been proposed via the tree and graph structures. A tree structure is able to generalise different clusterings using the combination result. It represents a top down hierarchy and helps to see how structures are near from others. The importance of connections is shown at each level of the hierarchy.

The graph structure shows connections among consensus clusters. The clusters are located on a circle with equal distance between them. Graph relations are shown as edges between consensus clusters. The importance of each relation is displayed by the thickness of the edge. These relations may help the user to find "semantic" links between consensus clusters.

In the first experiment we have constructed semantic of multimedia images. Independent classifications have been combined to obtain semantic representations of images. Classifications have been obtained from users and classes having been described by words. The combination of classifications as well as the combination of words give almost the same semantic. It shows that visual perception of images corresponds to textual description. We may conclude that the combination of different clusterings (classifications) of images derive image semantic which corresponds to information content of images.

In the second experiment, we have presented results of the combination of different clusterings of cities.

The third experiment has been carried out on satellite images in fully unsupervised way: feature extraction from images, unsupervised feature selection and unsupervised clustering. The number of clusters has been chosen automatically for each algorithm.

Combination of different clusterings has been used to construct satellite image semantic which has been represented by tree and graph structures. Visual perception of clustering corresponds to the semantic structure and justify pertinence of the proposed approach.

Chapter 10

Conclusions

In this thesis an unsupervised mining approach of optical satellite images of high resolution has been proposed. The general idea of mining includes information extraction from images, modelling by clustering algorithms, combining different clustering solutions and representing clusterings via a semantic structure. A software prototype of user interface for satellite image clustering has been proposed.

10.1 Summary

The unsupervised mining approach developed in this thesis has been evaluated on satellite images. However, the general idea of mining can be easily applied to other types of data. The accent in the thesis has been put on unsupervised methods because of the size of the data bases which require to be mined without human interaction to obtain an objective data modelling.

On the contrary of many similar works which argue to apply one single algorithm of data mining, we propose here different algorithms for data modelling and then combine their results. Some clustering algorithms are algebraic, others are probabilistic. For complex data, results obtained with different modelling often differ from one approach to another. We have proposed an unsupervised approach for combining clusterings issued from different algorithms.

We summarise now the new ideas demonstrated in this thesis:

- ★ Extraction of geometrical features from satellite images. These features are based on statistics of edges detected on gray scale images. In addition, a set of texture features is extracted from images: Haralick descriptors, Gabor coefficients and QMF features.
 - ★ The problem of curse of dimensionality obliges to select the most informative features. A new method of unsupervised feature selection which is based on feature clustering has been proposed. This approach is derived from the combination of different clusterings of the feature space.
 - ★ A minimum description length MDL criterion estimates the best clustering and the optimal number of clusters.
 - ★ New hierarchical algorithms have been derived from the simplified MDL criterion adapted for kernel K-means. The algorithms are based on the gradient descend
-

optimisation of the kernel MDL criterion.

- ★ A new method of unsupervised clustering combination is proven to achieve the exact global solution. It is based on the mean shift procedure to estimate the density of the clusterings.
- ★ All clusterings are presented by tree and graph structures. It helps the user to visualise clustering results and to learn data structures.

We present now in more details these propositions.

When going from high resolution images (i.e., SPOT) to very high resolution (i.e., IKONOS), the Earth surface goes from textural to structural representation. Therefore image descriptors also have to be adapted to the change in the resolution. For example, sea, forest and fields still look like texture surfaces, while urban and artificial areas (industrial zones, etc.) begin to have geometrical structures. Working with VHR images we should introduce both types of information: texture and geometrical. Texture features describe image regularities, while geometrical features capture information about lines, etc. In this thesis, geometrical features have been extracted from edges. An adaptive edge extraction has been proposed to reduce the effects of the change of contrast. Finally, various edge statistics have been calculated (length of line segments, edge density per surface, etc.).

The problem of high dimension of data has been solved by unsupervised feature selection. This approach has shown almost the same performances as the feature selection with supervised classification. The idea of this method is: (i) cluster features and (ii) combine clusterings by unsupervised approach. One single stable feature is selected from each cluster. Selected features are used further for data clustering. It has been assumed that features extracted from satellite images are not very noisy. MDL criterion has been efficient in selecting the optimal number of clusters and the optimal number of features. As a conclusion, two steps of feature selection can be considered. The first step is unsupervised feature selection by combination of clustered features. It selects features which are not correlated. The second step is removing of noisy features by MDL criterion.

Estimating the optimal number of clusters and the quality of clusterings has been carried out by MDL. The proposed simplification of MDL criterion makes it possible to be applied by different algorithms. A new hierarchical algorithm has been derived from simplified MDL criterion and kernel K-means algorithm. The hierarchy is constructed via gradient descend optimisation of the kernel MDL. Clustering results have demonstrated the efficiency of the algorithm.

Combination of clusterings obtained from different algorithms is performed by two methods. The first one is hierarchical and the second one is iterative (the mean shift). The hierarchical algorithm has no clear proof about the global optimum of the combination, while the exact solution can be obtained by the iterative algorithm.

Combination results have been presented by graph and tree structures. The tree structure expresses the hierarchical dependency among clusters, while the graph structure displays all possible connections among clusters. Experimentally, we have shown that trees and graphs reflect semantic meaning of data. Experiments have been carried out on different kinds of data issued from multimedia and satellite images. Obtained semantics have demonstrated interpretable links among concepts of images.

10.2 Perspectives

Several possible future research topics are emerging from the thesis.

One of the main drawbacks and consequently research directions, concerns feature extraction. As this subject is not the main issue of this thesis it has not been completely studied. Parameters for feature extraction algorithms have been set a priori from the knowledge of the properties of satellite images. It does not exactly reflect the richness of image information. A proposition would be to estimate optimal parameters for each feature extraction algorithm. This can be made via image modelling and parameter optimisation based on the quality of the feature model. The following parameters of geometrical features should be estimated: the optimal parameter of scale for Deriche edge detector, the error of edge approximation by line segments and the window size for edge statistics. Parameters of Haralick features as the size of the analysing window and the number of gray-scale levels should be optimal. Image frequencies for Gabor and QMF filtering should also be estimated optimally.

The drastically increasing size of databases (satellite images, multimedia images, etc.) poses the problem of the computation complexity of clustering algorithms. Many theoretical basis and developed algorithms exhibit a square complexity. Therefore they cannot be applied to large data sets in reasonable time. A new research direction consists in developing algorithms with linear memory complexity and time calculation. This rule should be kept imperatively for processing large databases, when algorithms with square complexities fail. One of the approaches satisfying these demands may be seen via mean shift like data clustering algorithms. This algorithm guarantees the global and exact clustering for a given model of data. One of the problems and consequently possible research directions is the estimation of parameters of functions used by the mean shift algorithm.

The third direction is the problem of feature selection. This procedure should be also integrated in the clustering algorithm. The selection should be considered via weighting each feature for each cluster separately.

Another kind of propositions consists in software development for large data bases. Data should be processed without doubling itself, e.g., without saving image patches, that is the case of many research works extracting features. It may significantly reduce processed memory.

Clustering is the first step of data mining which represents data in a compact form via clusters and relations among them. Analysis of characteristics of clusters and relations is a step towards high level data interpretation (under interpretation we mean human interaction for inference and generation of knowledge). An intermediate step between clustering results and human interpretation is very often considered. This step is called automatic construction of high level semantic. At this level clusters are considered as elementary items of information. For image processing tasks we have additional information about where clusters are located on the image. This spatial information may be used to find groups of clusters which have the same spatial organisation. We propose four levels of abstraction of data:

1. Zero level is data representation.
 2. First level is clusters and their relations discovered from data.
 3. Second level is spatial and textual links of clusters on the first level.
 4. Third level is human interpretation of data semantic.
-

The zero level represents data, i.e., satellite images. At the first level clusters are obtained from data. Relations among them are inferred either in the original feature space or in the feature space of different clusterings.

Information about spatial organisation of data is added at the second level. The elementary spatial organisation may be seen via blocks (regions) of an image. For example, an image can be divided into rectangular blocs. The goal of this step is to find clusters of blocks. Each of these clusters contains blocks which have similar spatial distribution of clusters obtained at the first level. Textual information may also be added at this level to enforce linkage among clusters. This level defines complete semantic of data.

Human interpretation of clusterings should involve the image semantic. At this level a system of mining proposes to a user all possible information about data: clusters of data, semantic clusters and possible relations among them. Here several scenario of data interpretation are possible: (i) hierarchical analysis of data, i.e., how particular data rely on context or how the context is built on data; (ii) selection of the particular cluster or group of clusters to classify data. This classification can be supervised or semi supervised. For semi-supervised classification a user gives an example to the system and iteratively selects appropriate responses.

In this thesis the first level of the semantic construction has been proposed. Several future research directions can be considered to enrich the image semantic. The first direction is determining the optimal block size for image clustering. These blocks can have different sizes. For example, blocks of urban zones should be smaller because urban zones contain quite different pieces of information. If this block is too large then important information may be mixed. On the contrary, a block of agricultural zones, e.g., fields, may be of larger size because it has homogeneous information. We conclude that the richer information the smaller the block size should be and vice-versa. In addition, any form of block should be taken into account, because the Earth surface reflected by satellite images has no rectangular frontiers. Various interpretations of the same zones are possible for different block sizes. Therefore, the second direction is estimating the optimal block size for semantic clustering. For example, in the case of urban zones small blocks describe images of residential buildings, warehouses, etc. When the block size is larger, then images of buildings constitute residential zones, images of warehouses correspond to industrial zones, etc. With larger block sizes it is possible to discriminate urban zones from rural and agricultural zones, etc. We conclude that an image can be divided into blocks of several optimal sizes depending on semantic meaning. Thus, each region of the image corresponds to a certain level of semantic hierarchy.

Satellite image semantic is a step towards a formal representation of concepts and relations to describe the Earth surface. This representation is needed to better explore large data bases of satellite images. Semantic can be formalised by ontology. The interest of mining images via ontology consists in linkage of image semantic and models of natural languages to better reflect scene understanding. Nowadays, it is a very promising research direction on data mining and knowledge reasoning. One of the projects on this subject is Differential and formal ontology editor, DAFOE ¹. Despite ontology for satellite images has not yet been constructed, many works on it have been done: cartography of the Earth surface, formal representation of concepts and relations among them, etc. Works on environment analysis using geographical formalism and satellite images is developing in project of Corine Land Cover [Bossard et al., 2000].

¹<http://dafoe4app.fr/>

In this thesis we have been interested in automatic modelling of images, extraction of concepts and relations. Formalization of these terms by ontology may be seen as new research direction.

Appendix A

Haralick features

In this Appendix we list Haralick features [Haralick et al., 1977] mentioned in Section 3.2. The features are computed on a co-occurrence matrix. The matrix is the second-order histogram of the joint probability distribution $P(i, j, \rho, \theta)$ Eq. (3.6) of a pair of pixels which are separated by ρ pixels and have an angle θ with respect to the horizontal axis. Let p_{ij} be an element of normalised CM P_{ij} Eq. (3.6) for some ρ and θ : $p_{ij} = P_{ij} / \sum_{ij} P_{ij}$. Then Haralick features are:

1. Angular second moment

$$X_6 = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p_{ij}^2 \quad (\text{A.1})$$

2. Contrast

$$X_7 = \sum_{n=0}^{L-1} n^2 \left\{ \sum_{\substack{i=0, \\ |i-j|=n}}^{L-1} \sum_{j=0}^{L-1} p_{ij} \right\} \quad (\text{A.2})$$

3. Correlation

$$X_8 = \frac{\left\{ \sum_i \sum_j (ij) p_{ij} \right\} - \mu_x \mu_y}{\sigma_x \sigma_y}, \quad (\text{A.3})$$

where μ_x and σ_x are mean and standard deviation of $\sum_j p_{ij}$, analogously for μ_y, σ_y and $\sum_i p_{ij}$.

4. Variance

$$X_9 = \sum_i \sum_j (i - \mu)^2 p_{ij} \quad (\text{A.4})$$

5. Inverse difference moment

$$X_{10} = \sum_i \sum_j \frac{p_{ij}}{1 + (i - j)^2} \quad (\text{A.5})$$

6. Sum average

$$X_{11} = \sum_{i=2}^{2L} i p_{x+y}(i) \quad (\text{A.6})$$

7. Sum variance

$$X_{12} = \sum_{i=2}^{2L} (i - X_1)^2 p_{x+y}(i) \quad (\text{A.7})$$

8. Sum entropy

$$X_{13} = - \sum_{i=2}^{2L} p_{x+y}(i) \log p_{x+y}(i) \quad (\text{A.8})$$

9. Entropy

$$X_{14} = - \sum_i \sum_j p_{ij} \log(p_{ij}) \quad (\text{A.9})$$

10. Difference variance

$$X_{15} = \sum_{i=2}^{2L} (i - X_1)^2 p_{x-y}(i) \quad (\text{A.10})$$

11. Difference entropy

$$X_{16} = \sum_{i=0}^{L-1} p_{x-y}(i) \log\{p_{x-y}(i)\} \quad (\text{A.11})$$

12. Information measure 1

$$X_{17} = \frac{H_{xy} - H_{xy}^1}{\max\{H_x, H_y\}} \quad (\text{A.12})$$

13. Information measure 2

$$X_{18} = \sqrt{(1 - \exp(-2(H_{xy}^2 - H_{xy})))} \quad (\text{A.13})$$

where H_{xy} , H_x and H_y are

$$H_{xy}^1 = - \sum_i \sum_j p_{ij} \log(p_x(i)p_y(j)) \quad (\text{A.14})$$

$$H_{xy}^2 = - \sum_i \sum_j p_x(i)p_y(j) \log(p_x(i)p_y(j)) \quad (\text{A.15})$$

Appendix B

Features of line segments and edges

In this Appendix we propose features of linear segments and edges discussed in Section 3.3. Let L_i be a length of the i^{th} linear segment and α_i is its angle of rotation, where $i = 1, \dots, N$ and N is the number of linear segments.

We propose to use the following features:

1. $X_1 = N$ - the number of linear segments;
2. $X_2 = N_T$ - the number of linear segments for $L_i > T$, where $T = \text{constant}$;
3. The mean length of linear segments:

$$X_3 = \frac{1}{N} \sum_{i=1}^N L_i; \quad (\text{B.1})$$

4. The mean length of linear segments for $L_i > T$:

$$X_4 = \frac{1}{N_T} \sum_{i=1}^{N_T} \{L_i | L_i > T\}; \quad (\text{B.2})$$

5. The weighted length of linear segments:

$$X_5 = \frac{\sum_{i=1}^N L_i^2}{\sum_{i=1}^N L_i}; \quad (\text{B.3})$$

6. The weighted length of linear segments for $L_i > T$:

$$X_6 = \frac{\sum_{i=1}^N \{L_i^2 | L_i > T\}}{\sum_{i=1}^N \{L_i | L_i > T\}}; \quad (\text{B.4})$$

7. X_7 - the length of curves which corresponds to the number of pixels;

8. Features of pixels distribution with sliding window. Let $N'_{i,j}$ is a number of pixels in the window of size $n_w \times m_w$ pixels where $n_w = 10, m_w = 10$ and $1 \leq i \leq S, 1 \leq j \leq S, S = 64 - (m_w - 1)$:

Mean

$$X_8 = \frac{1}{S^2} \sum_{i,j} N'_{i,j} \quad (\text{B.5})$$

9. Standard deviation

$$X_9 = \sqrt{\frac{1}{S^2} \sum_{i,j} (N'_{i,j} - X_8)^2} \quad (\text{B.6})$$

10. Skew

$$X_{10} = \frac{1}{S^2} \sum_{i,j} \left(\frac{N'_{i,j} - X_8}{X_9} \right)^3 \quad (\text{B.7})$$

11. Kurtosis

$$X_{11} = \frac{1}{S^2} \sum_{i,j} \left(\frac{N'_{i,j} - X_8}{X_9} \right)^4 - 3 \quad (\text{B.8})$$

12. The number of directions of linear segments X_{12} ;

An angle of linear segment is $\alpha_i = \arctan(a(i, j))$ Eq.3.16. We use eight angles: $[0, 22.5), [22.5, 45), [45, 67.5), [67.5, 90), [90, 112.5), [112.5, 135), [135, 157.5), [157.5, 180)$. These angles are used to compute a histogram H , where $\sum_{i=1}^8 H_i = 1$. The number of directions are the number of histogram values which exceed a threshold 0.4. Thus, we may have 0, 1, or 2 directions.

13. A co-occurrence matrix (CM) Eq. (3.6) introduced in Section 3.2 is used to extract features from a gray-tone image [Shanmugan et al., 1973; Haralick et al., 1977]. We propose to calculate statistics on CM from an image of edges. We remind that CM is a matrix of frequencies $P_{i,j}$ of two pixels separated by distance d where one pixel has gray-level i and the other has gray-level j . Let $N_x \times N_y$ be a size of the image with the horizontal spatial domain $L_x = \{1, 2, \dots, N_x\}$, vertical spatial domain $L_y = \{1, 2, \dots, N_y\}$ and $I(k, l) \in \{0, 1, \dots, N_g\}$, N_g is the number of gray-levels.

A binary image has two levels 0 and 1, thus $N_g = 2$. In this case the CM is a frequency of occurring of two pixels with gray-level 1. Thus, we can compute frequencies for four different angles: $P(d, 0^\circ), P(d, 45^\circ), P(d, 90^\circ), P(d, 135^\circ)$. where $0 \leq d \leq D$, D - maximal distance. D is chosen 20. Frequencies of the binary image which do not depend on angles can be defined as:

$$P'(d) = \frac{1}{4} (P(d, 0^\circ) + P(d, 45^\circ) + P(d, 90^\circ) + P(d, 135^\circ))$$

We use three statistical features of $P'(d)$.

Mean of the frequencies:

$$X_{13} = \frac{1}{D} \sum_d P'(d) \quad (\text{B.9})$$

14. Standard deviation of the frequencies:

$$X_{14} = \sqrt{\frac{1}{D} \sum_d (P'(d) - X_{13})^2} \quad (\text{B.10})$$

15. Entropy of the frequencies:

$$X_{15} = - \sum_d P'(d) \log P'(d) \quad (\text{B.11})$$

Appendix C

MDL for the Complete Log-likelihood of GMM

In this Appendix we propose to simplify the complete log-likelihood $\log(P(X, z|\Theta))$ Eq. (6.29) which has been presented in Chapter 6 Section 6.4.

Let z be a hidden variable which attributes any sample i to classes: $z = \{z_1, \dots, z_i, \dots, z_I\}$. Then complete likelihood function $\log(P(X, z|\Theta))$ Eq. (6.29) of the finite mixture Eq. (5.27) is [Figueiredo, 2002; Govaert, 2003]:

$$\begin{aligned} \log(P(X, z | \Theta)) &= \log \left(\prod_{i=1}^I \sum_{k=1}^K z_{ik} \alpha_k P_k(X_i | \Theta_k) \right) = \\ &= \sum_{i=1}^I z_{ik} \log(\alpha_k P_k(X_i | \Theta_k)) . \end{aligned} \quad (\text{C.1})$$

By substituting the multivariate Gaussian distribution $P_k(X_i | \Theta_k)$ (5.29) in the complete log-likelihood (C.1), we obtain:

$$\begin{aligned} \sum_{i=1}^I z_{ik} \log(\alpha_k \mathcal{N}(X_i | \mu_k, \Sigma_k)) &= \sum_{i=1}^I z_{ik} \log \left(\alpha_k \frac{e^{-\frac{1}{2}((X_i - \mu_k) \Sigma_k^{-1} (X_i - \mu_k))^T}}{(2\pi)^{D/2} |\Sigma_k|^{1/2}} \right) = \\ &= \sum_{i=1}^I z_{ik} \left(\log \left(\frac{\alpha_k}{|\Sigma_k|^{1/2}} \right) - \frac{D}{2} \log(2\pi) - \frac{1}{2} ((X_i - \mu_k) \Sigma_k^{-1} (X_i - \mu_k)^T) \right) = \\ &= \frac{1}{2} \sum_{i=1}^I z_{ik} \log \left(\frac{\alpha_k^2}{|\Sigma_k|} \right) - \frac{1}{2} \sum_{i=1}^I z_{ik} J \log(2\pi) \\ &\quad - \frac{1}{2} \sum_{i=1}^I z_{ik} ((X_i - \mu_k) \Sigma_k^{-1} (X_i - \mu_k)^T) . \end{aligned} \quad (\text{C.2})$$

In this equation, some terms are constant:

$$- \frac{1}{2} \sum_{i=1}^I z_{ik} J \log(2\pi) = - \frac{1}{2} \sum_{k=1}^K n_k J \log(2\pi) = - \frac{1}{2} I J \log(2\pi) = \text{const}_1 . \quad (\text{C.3})$$

Moreover, to calculate the matrix Σ_k (5.31) the only samples from the cluster k are needed, therefore:

$$-\frac{1}{2} \sum_{i=1}^I z_{ik} ((X_i - \mu_k) \Sigma_k^{-1} (X_i - \mu_k)^T) = -\frac{1}{2} \sum_{k=1}^K n_k J = -\frac{JI}{2} = \text{const}_2. \quad (\text{C.4})$$

Then, the complete log-likelihood $\log(P(X, z|\Theta))$ (C.1) may be written as:

$$\frac{1}{2} \sum_{i=1}^I z_{ik} \log \left(\frac{\alpha_k^2}{|\Sigma_k|} \right) + \text{const} = \frac{1}{2} \sum_{k=1}^K n_k \log \left(\frac{\alpha_k^2}{|\Sigma_k|} \right) + \text{const}. \quad (\text{C.5})$$

In the right part of the MDL definition (6.28), \mathbb{k} is the model free parameters number. In case of Gaussian mixture model free parameters are:

- ★ $K - 1$ parameters for K weights α_k (since $\sum \alpha_k = 1$);
- ★ J parameters for each mean μ_k ;
- ★ $J(J + 1)/2$ parameters for each covariance matrix Σ_k .

Therefore, the number of free parameters is:

$$\mathbb{k} = K - 1 + K(J + J(J + 1)/2) = K(J^2 + 3J + 2)/2 - 1. \quad (\text{C.6})$$

Using the complete log-likelihood (C.5) and the free parameter number of (C.6), the description length (6.28) of Gaussian mixture model with K clusters is:

$$-\frac{1}{2} \sum_{k=1}^K n_k \log \left(\frac{\alpha_k^2}{|\Sigma_k|} \right) + (K(J^2 + 3J + 2)/2 - 1) \log(I)/2 + \text{const}. \quad (\text{C.7})$$

The *const* term having no influence on MDL for different cluster numbers and as $\alpha_k = n_k/I$, we may rewrite Eq. (C.7) as:

$$\Lambda = - \sum_{k=1}^K n_k \log \left(\frac{n_k^2}{|\Sigma_k|} \right) + K(J^2 + 3J + 2) \log(I)/2. \quad (\text{C.8})$$

Appendix D

Proof of Theorem 7.5.1

In this Appendix we prove proposed Theorem 7.5.1 which states the mean shift algorithm finds the global minimum of error E Eq. (7.64).

Firstly, we show the maximisation of the mean shift vector norm. **Proposition 7.5.1** is a particular case of **Theorem 1** proposed in [Comaniciu, 2003] or **Theorem 3** derived in [Fashing & Tomasi, 2005] that establish that the optimum solution is found when the mean shift procedure maximises the norm of the mean shift vector.

Secondly, we prove that during optimisation the number of points n_j falling into cluster j is a strictly monotonic increasing sequence. Let y_k be a point where density is estimated within the d -dimensional window $W(y_k)$. Let the density estimation \hat{f} Eq. (7.65) with Epanechnikov kernel Eq. (7.66) for k and $k + 1$ consecutive steps be \hat{f}_k and \hat{f}_{k+1} respectively:

$$\begin{aligned}\hat{f}_k &= \frac{1}{(Ih^d)} \sum_{b_u \in W(y_k)} K\left(\frac{b - b_u}{h}\right) = \frac{(d+2)}{2Ic_d} \sum_{b_u \in W(y_k)} (1 - \|y_k - b_u\|^2) = \\ &= \frac{(d+2)}{2Ic_d} \frac{1}{n_k} \sum_{b_u, b_v \in W(y_k)} b_v b'_u.\end{aligned}\tag{D.1}$$

and

$$\hat{f}_{k+1} = \frac{(d+2)}{2Ic_d} \sum_{b_u \in W(y_{k+1})} (1 - \|y_{k+1} - b_u\|^2) = \frac{(d+2)}{2Ic_d} \frac{1}{n_{k+1}} \sum_{b_u, b_v \in W(y_{k+1})} b_v b'_u.\tag{D.2}$$

It was proved in [Comaniciu & Meer, 1999] **Theorem 1** that the positive sequence $\{\hat{f}_k\}$ of density estimation by mean-shift algorithm and Epanechnikov kernel is converging and

$$\hat{f}_{k+1} - \hat{f}_k \geq \frac{d+2}{2Ic_d} n_k \|y_{k+1}\|^2,\tag{D.3}$$

consequently the condition $\hat{f}_{k+1} > \hat{f}_k$ holds. Using this condition we may prove that $n_{k+1} > n_k$. Let us rewrite inequality (D.3) by substituting equations Eq. (7.62), Eq. (D.1)

and Eq. (D.2):

$$\begin{aligned} \frac{(d+2)}{2Ic_d} \left(\frac{1}{n_{k+1}} \sum_{b_u, b_v \in W(y_{k+1})} b_v b_u - \frac{1}{n_k} \sum_{b_u, b_v \in W(y_k)} b_v b_u \right) \geq \\ \frac{(d+2)}{2Ic_d} \frac{n_k}{n_{k+1}^2} \sum_{b_u, b_v \in W(y_{k+1})} b_v b_u. \end{aligned} \quad (D.4)$$

Dividing inequality (D.4) by \hat{f}_{k+1} Eq. (D.2) and using conditions $0 < \hat{f}_k / \hat{f}_{k+1} < 1$ the inequality (D.4) becomes:

$$1 - \frac{\hat{f}_k}{\hat{f}_{k+1}} \geq \frac{n_k}{n_{k+1}} > 0 \Rightarrow 0 < 1 - \frac{n_k}{n_{k+1}} < 1 \Rightarrow 0 < n_k < n_{k+1}. \quad (D.5)$$

When the optimal value is achieved, then $\hat{f}_{k+1} \equiv \hat{f}_k$ and $n_j = n_{k+1} \equiv n_k$. We proved here that the number of samples n_j^2 is strictly increasing (D.5). The condition $\|\mu_j\|^2 > 0.5$ Eq. (7.61) provides strictly negative values during minimising error E Eq. (7.64) by the mean-shift algorithm with Epanechnikov kernel Eq. (7.66).

Appendix E

Dictionary of image classes

In this appendix we represent words which have been extracted from descriptions of the 50 image classifications in Section 9.3.

Each user characterises its image classes by words. Descriptions of image classifications have been done mainly with English words. Mainly nouns have been selected while articles and endings have been removed manually to avoid mistakes, presence or absence of comas, etc. In the total 157 words (or group of words) have been obtained to describe image classes.

The dictionary is :

activity, air, alive, animal, architecture, art, artificial, art_photograph, art_pictural, artistique, automobile, balloon, beach, bizarre, bird, boat, bridge, building, car, castle, celebrity, child, city, city_landscape, coast, construction, costume, countryside, diversite, disguise, dance, dense, dragon, details, driver, detail, earth, electricity, eau, fly, forest, famous, folklore, forest, famous_sight, fair, figure, go, green, grass, group, great_horizon, house, holidays, history, human, historic_building, historic_places_of_interest, human_being, historic_places_of_interest, hard_to_classify, interieur, interesting, installation, image_non_naturelle, images_of_home, live, little_building, landscape, landscape_with_human_building, landscape_without_visible_human_presence, mammal, men, mosaic, modern_building, monument, means_of_transport, marine, motor, man, mountain, nature, natural_zone, nationality, natural_landscape, not_natural, not_a_gategory, neon, old, original, object, other, outlier, panoramic_view, postcard, people, performance, place, painting, people, paysage, paint, plane, plage, pelleteuse, paysage_de_cartes_postales, panoramic_view, photos_of_sport, people_doing_something_together_or_alone, photos_of_folklore_around_the_world, photos_of_nature, religious_building, recognizable, roue, sea, sky, swim, someone_with_claws_and_fur, scenery, seat, stand, smart, show, scenery, street, sculpture, stained_glass_window, statue, stained_glasses, ship, should_be_renamed_vehicle, sport, social_activities, theatre, transport, tower, totem, tradition, tree, unique, urban_landscape, useless, vitrail, vehicule, vehicules_de_toute_sorte, verdure, vacances, voyage, voiture, women, walk, wise, water, widescreen, wheel, woman, zoom_on_a_detail_of_an_object

The following Table E.1 represents the extracted word description of each cluster obtained in Section 9.3. Words of each cluster in Table E.1 are very similar by their sense and corresponds to semantic concepts.

Cluster 1:	any image with an animal as a main character, an animal is the main subject, birds, animals, nature, sea, sky, animals, natural zone, live, swim, go, earth, birds, someone with claws and fur, animaux, betes, bestioles, animal, nature, ANIMALS, All kinds of animals, birds, mammals, alive, animal, green, animal placed in a tree, panoramic views, animals, oiseau, mammifere, unique, Living animals,
Cluster 2:	any image containing people, people is the main subject, people, costume, activity, men, women, group, seat, walk, stand, one, or, few, persons, human being, art, folklore, child, live, smart, wise, personnages humains, nationalities, traditions, societies humaines; diversite; peoples, ce qui fait penser aux vacances, aux lieux touristiques, endroits a visiter... Photos of folklore around the world, show; celebrity; costume; disguise, hommes, traditional, costume, performance, people doing something together or alone, wear for special cultural events, social activities
Cluster 3:	nice scenery that could be taken on holidays, well-known monuments, building, street, electricity, monument, castle, recognizable, landscape with human buildings, towers, monument, batiment, pont, constructions humaines, paysages non naturels, famous sights, monuments, bridges, ce qui fait penser aux vacances, aux lieux touristiques, endroits a visiter, monuments, architecture, ARCHITECTURE, MONUMENTS Famous sites around the world, buildings; beaches; holidays; scenery; castles, castles, famous, old, city landscape, interest object mostly surrounded by sky, Historic buildings, religious buildings
Cluster 4:	nice scenery that could be taken on holidays, landscape without particular focus, grass, sky, nature, nature, postcard, nature, landscape without visible human presence, landscape, countryside, sea, beach, mountain, paysage, nature, water, tree, sky, earth, coasts, forests, castels, houses, paysage de cartes postales, plage, mer, cite, rivage, SEA, SHORE, SEASHORE, BEACH, buildings; beaches; holidays; scenery; castles nature, widescreen, water and land separated by a long line, natural landscapes, urban landscapes, panoramic views, animals, eau, montagne, verdure, plage
Cluster 5:	nice scenery that could be taken on holidays, landscape without particular focus, water, sea, ships, boat, nature, postcard, landscape with human buildings, cars, boats, planes, nature, ensemble de batiments, ship, water, vehicules de toute sorte, cars, ships, details ce qui fait penser aux vacances, aux lieux touristiques, endroits a visiter, plage, mer, cite, rivage, SEA, SHORE, SEASHORE, BEACH Photos of sports, buildings; beaches; holidays; scenery; castles vehicule, marine, boats, means of transport, panoramic views, animals, machines or instruments, eau, montagne, verdure, plage, everyday life objects

Table E.1: Discovered clusters described by the text.

Appendix F

Human-computer interface for unsupervised image clustering

In this appendix we demonstrate a developed human - computer interface for unsupervised satellite image clustering. Human - computer interface is a mean of dialog between a user and a computer system. This program is applied by a user to load satellite image, to visualise the image, set some parameters (if needed) and cluster the image. The system returns as a result labels of the clustered image and display them by superimposing it with the original image. In addition, examples of each cluster can be displayed in the form of small patches. A screen shot of the interface is presented in Figure F.1.

The user's manual for this application consists in the following steps:

1. The default image is loaded for clustering. The user can select another satellite image by clicking on the button "Load image".
2. The user can scroll image either to zoom in or to zoom out it. The same action can be done by sliding a bar at the left side of the program.
3. The user can select desired number of clusters, (the number of clusters is 2, by default).
4. Clicking on the button of "Cluster image" the user runs K-means clustering algorithm to cluster image.
5. Clusters are displayed in the original image by different colours.
6. The user can visualise samples of the obtained clusters by clicking on the "Textures" button. These patches correspond to the nearest samples of each cluster (nearest in the sense of Euclidean distance to the mean vector of corresponding cluster).

Specification The "Unsupervised image clustering" (UIC) program has been written in C++ using QT widgets and can be compiled either for Windows OS or Linux OS. The demo version of the program is able to load two satellite images of size 3000×3000 pixels. Extensions of the program to a higher number of images may be done easily. Images are either pre-cut or online extracted from larger images. The algorithm of K-means is build in the program.

The result of clustering is displayed by different colours on original image. RGB colour map depends on the number of clusters. Finally, the gray level of the image is scaled and added to each colour.

The application shown in Figure F.1 represents UIC program with a loaded and clustered image of Madrid. Clusters are displayed by different colours. Examples of textures of each cluster are given in the right part of the image. This program can be improved by implementing additional functions of data mining: as unsupervised features selection, clustering algorithms, criteria to determine the number of clusters, visualisation of clustering results, etc.

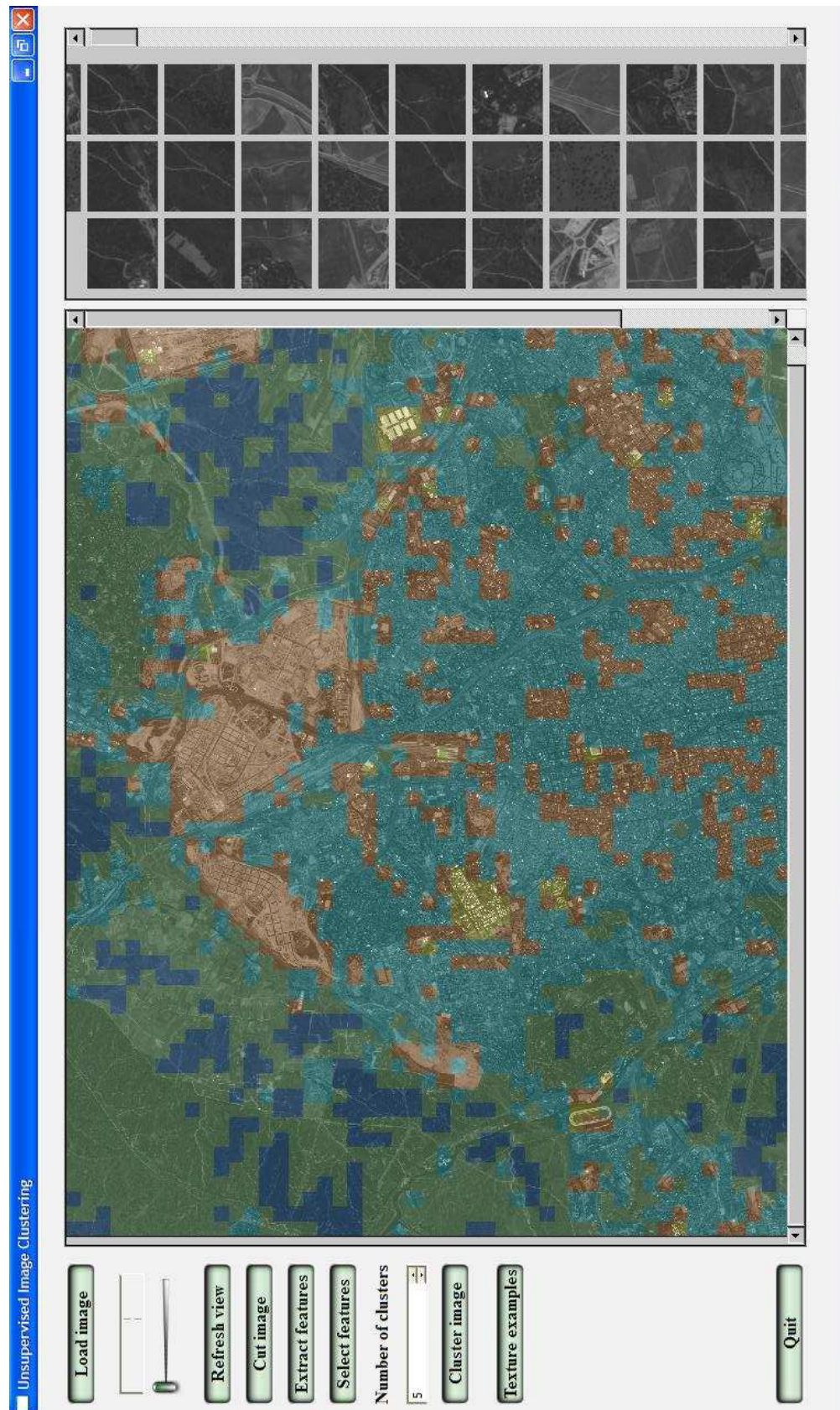


Figure F.1: Screenshot of program for unsupervised satellite image clustering.

Bibliography

- Al-Ani, A. & Deriche, M. (2002). A new technique for combining multiple classifiers using the Dempster-Shafer theory of evidence, *Journal of Artificial Intelligence Research* **17**: 333–361.
- Atkinson, P. M. & Lewis, P. (2000). Geostatistical classification for remote sensing: an introduction, *Comput. Geosci.* **26**(4): 361–371.
- Ayad, H. & Kamel, M. S. (2005). Cluster-based cumulative ensembles., *Multiple Classifier Systems*, pp. 236–245.
- Barnes, C. (2007). Image-driven data mining for image content segmentation, classification, and attribution, *Geoscience and Remote Sensing, IEEE Transactions on*.
- Barron, A., Rissanen, J. & Yu, B. (1998). The minimum description length principle in coding and modeling, *IEEE Trans. Inform. Theory* **44**(6): 2743–2760.
- Benhadia, H. & Marcotorchino, F. (1998). Introduction à la similarité régularisée en analyse relationnelle., *Math. Sci. Hum.* **46**: 45–69.
- Bhattacharya, A., Roux, M., Maitre, H., Jermyn, I. H., Descombes, X. & Zerubia, J. (2007). Indexing satellite images with features computed from man-made structures on the earth's surface, *Proc. International Workshop on Content-Based Multimedia Indexing*, Bordeaux, France.
- Biernacki, C., Celeux, G. & Govaert, G. (1999). An improvement of the NEC criterion for assessing the number of clusters in a mixture model., *Pattern Recognition Letters* **20**(3): 267–272.
- Biernacki, C., Celeux, G. & Govaert, G. (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate gaussian mixture models., *Computational Statistics & Data Analysis* **41**(3-4): 561–575.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer.
- Bordes, J.-B. & Maître, H. (2007). Semantic annotation of satellite images, *Machine Learning and Data Mining in Pattern Recognition, 5th International Conference, MLDM 2007, Leipzig, Germany, July 18-20, 2007, Poster Proceedings*, IBAI publishing, pp. 120–133.
- Bossard, M., Feranec, J. & Otahel, J. (2000). Corine land cover technical guide - addendum 2000, European environment agency, *Technical report*.
-

- Boulis, C. & Ostendorf, M. (2004). Combining multiple clustering systems, *8th European conference on Principles and Practice of Knowledge Discovery in Databases(PKDD)*, LNAI 3202, pp. 63–74.
- Campedel, M., Kyrgyzov, I. & Maître, H. (2007). Sélection non supervisé d'attributs - application à l'indexation d'images satellitaires, *XIVe Rencontre de la Société francophone de classification, SFC*.
- Campedel, M., Luo, B., Maître, H., Moulines, E., Roux, M. & Kyrgyzov, I. (2004). Indexation des images satellitaires. détection et évaluation des caractéristiques de classification, *Technical report*, École Nationale Supérieure des Télécommunications, Département Traitement du Signal et des Images. Available from: http://www.tsi.enst.fr/~campedel/Contribution/TSIreport_2004D008.pdf.
- Campedel, M., Moulines, E., H., M. & Datcu, M. (2005). Feature selection for satellite image indexing, *ESA-EUSC: Image Information Mining*.
- Canny, J. (1983). Finding edges and lines in images, *Technical report*.
- Canny, J. (1986). A computational approach to edge detection, *IEEE Transactions on pattern Analysis and Machine Intelligence* 8(6): 679–698.
- Carneiro, G., Chan, A. B., Moreno, P. J. & Vasconcelos, N. (March 2007). Supervised learning of semantic classes for image annotation and retrieval, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29(3): 394–410.
- Chan, P. K., Schlag, M. D. F. & Zien, J. Y. (1994). Spectral k-way ratio-cut partitioning and clustering, *IEEE Trans. on CAD of Integrated Circuits and Systems* 13: 1088–1096.
- Chapelle, O., Vapnik, V., Bousquet, O. & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines, *Machine Learning* 46: 131–159.
- Cheeseman, P. & Stutz, J. (1996). Bayesian classification (AUTOCLASS): Theory and results, in U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth & R. Uthurusamy (eds), *Advances in Knowledge Discovery and Data Mining*, AAAI Press/MIT Press, pp. 153–180.
- Coleman, G. & Andrews, H. (1979). Image segmentation by clustering., *Proceedings of the IEEE* 67: 773–785.
- Comaniciu, D. (2003). An algorithm for data-driven bandwidth selection, *PAMI* 25(2): 281–288.
- Comaniciu, D. & Meer, P. (1999). Mean shift analysis and applications, *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, IEEE Computer Society, p. 1197.
- Comaniciu, D. & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis, *PAMI* 24(5): 603–619.
- Comaniciu, D., Ramesh, V. & Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift, *CVPR00*, pp. II: 142–149.
-

- Costache, M. & Datcu, M. (2007). Learning - unlearning for mining high resolution eo images, *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*.
- Datcu, M. & Seidel, K. (2000). Image information mining: exploration of image content in large archives, *Aerospace Conference Proceedings, IEEE* 3: 253 – 264.
- Datcu, M. & Seidel, K. (2005). Human-centered concepts for exploration and understanding of earth observation images, *Geoscience and Remote Sensing, IEEE Transactions on*.
- Datcu, M., Daschiel, H., Pelizzari, A., Quartulli, M., Galoppo, A., Colapicchioni, A., Pastori, M., Seidel, K., Marchetti, P. & D'Elia, S. (2003). Information mining in remote sensing image archives: system concepts, *Geoscience and Remote Sensing, IEEE Transactions on*.
- Daugman, J. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by 2d visual cortical filters, *JOSA-A* 2(7): 1160–1169.
- Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statistical Society. Series B (Methodological)* 39: 1–38.
- Deriche, R. (1987a). Optimal edge detector using recursive filtering, *Proceeding of the First International Conf. on Computer Vision* pp. 501–505.
- Deriche, R. (1987b). Using Canny's criteria to derive a recursively implemented optimal edge detector, *International Journal of Comp.Vision* 1(2): 167–187.
- Dhillon, I. S., Guan, Y. & Kulis, B. (2004). Kernel k-means: spectral clustering and normalized cuts, *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM Press, pp. 551–556.
- Diday, E. (1979). *Optimisation en classification automatique. Tome 1, 2. (French) [Optimization in automatic classification. Vol. 1, 2]*, Institut National de Recherche en Informatique et en Automatique (INRIA), Rocquencourt.
- Duda, R. O., Hart, P. E. & Stork, D. G. (2000). *Pattern Classification (2nd Edition)*, Wiley-Interscience.
- Dunn, D. & Higgins, W. (1995). Optimal gabor filters for texture segmentation, *IP* 4(7): 947–964.
- Dunn, D., Higgins, W. E. & Wakeley, J. (1994). Texture segmentation using 2-d gabor elementary functions, *IEEE Trans. Pattern Anal. Mach. Intell.* 16(2): 130–149.
- Fashing, M. & Tomasi, C. (2005). Mean shift is a bound optimization, *PAMI* 27(3): 471–474.
- Feder, M. & Merhav, N. (1996). Hierarchical universal coding, *IEEE Transactions on Information Theory* 42: 1354 – 1364.
- Ferecatu, M. & Boujemaa, N. (2007). Interactive remote-sensing image retrieval using active relevance feedback, *GeoRS* 45(4): 818–826.
-

- Figueiredo, M.A.F. Jain, A. (2002). Unsupervised learning of finite mixture models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(3): 381–396.
- Floudas, C. A. & Visweswaran, V. (1994). Quadratic optimization, *Handbook of global optimization*, Kluwer Academic Publishers, pp. 217–270.
- Forsyth, D. A. & Ponce, J. (2002). *Computer Vision: A Modern Approach*, Prentice Hall Professional Technical Reference.
- Fred, A. & Jain, A. (2005). Combining multiple clusterings using evidence accumulation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(6): 835–850.
- Friedman, J., Trevor, H. & Robert, T. (2001). *The Elements of Statistical Learning, Data Mining, Inference and Prediction*, Statistics, Springer.
- Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*, Academic Press.
- Gionis, A., Mannila, H. & Tsaparas, P. (2005). Clustering aggregation, *icde* **0**: 341–352.
- Giros, A. (July 31 2006-Aug. 4 2006). Comparison of partitions of two images for satellite image time series segmentation, *Geoscience and Remote Sensing Symposium, 2006. IGARSS 2006. IEEE International Conference on* pp. 2592–2595.
- Gleyzes, J.-P., Meygret, A., Fratter, C., Panem, C., Baillarin, S. & Valorge, C. (2003). SPOT5: system overview and image ground segment, *IEEE International Geoscience and Remote Sensing Symposium. IGARSS. Proceedings.* **1**: 300–302.
- Gorte, B. & Stein, A. (1998). Bayesian classification and class area estimation of satellite images using stratification, *GeoRS* **36**(3): 803–812.
- Gotlieb, C. C. & Kumar, S. (1968). Semantic clustering of index terms, *J. ACM* **15**(4): 493–513.
- Govaert, G. (2003). *Analyse des données*, Hermes Science Publications.
- Govaert, G. & Nadif, M. (2003). Clustering with block mixture models., *Pattern Recognition* **36**(2): 463–473.
- Govaert, G. & Nadif, M. (2005). An em algorithm for the block mixture model., *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(4): 643–647.
- Govaert, G. & Nadif, M. (2007). Clustering of contingency table and mixture model, *European Journal of Operational Research* **127**(3): 1055–1066.
- Gueguen, L. & Datcu, M. (2007). Image time-series data mining based on the information-bottleneck principle, *IEEE Transactions on Geoscience and Remote Sensing* **45**: 827 – 838.
- Guyon, I. (2002). Gene selection for cancer classification using support vector machines, *Mach. Learn.* **46**: 389–422.
- Haralick, R., Shanmugam, K. & Dinstein, I. (1977). Textural features for image classification, *CMetImAly77*, pp. 141–152.
- Hardle, W., Hdrdle, W. & Simar, L. (2003). *Applied Multivariate Statistical Analysis*, Springer.
-

- Heas, P. & Datcu, M. (2005). Modelling trajectory of dynamic clusters in image time-series for spatio-temporal reasoning, *IEEE Transactions on Geoscience and Remote Sensing* **43**(7): 1635–1647.
- Hsu, C.-W. & Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines, *IEEE Transactions on Neural Networks* **13**: 415–425.
- Huang, C., Davis, L. S. & Townshend, J. R. G. (2002). An assessment of support vector machines for land cover classification, *International Journal of Remote Sensing* **23**: 725–749.
- Huang, Y. S. & Suen, C. Y. (1995). A method of combining multiple experts for the recognition of unconstrained handwritten numerals., *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**: 90–94.
- Imai, H. & Iri, M. (1988). Polygonal approximations of a curve (formulations and algorithms), *Computational Morphology* pp. 71–86.
- Jain, A. & Dubes, R. C. (1988). *Algorithms for Clustering Data*, Prentice-Hall, Englewood Cliffs, NJ.
- Jain, A. K. & Farrokhnia, F. (1991). Unsupervised texture segmentation using gabor filters, *Pattern Recogn.* **24**(12): 1167–1186.
- Kang, H.-J. & Kim, J. (1997). A probabilistic framework for combining multiple classifiers at abstract level, *ICDAR97* pp. We-1B.
- Kannan, R., Vempala, S. & Veta, A. (2000). On clusterings-good, bad and spectral, *FOCS '00: Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, IEEE Computer Society, Washington, DC, USA, pp. 367–377.
- Kaufman, L. & Rousseeuw, P. (1990). *Finding Groups in Data: an introduction to cluster analysis*, Wiley.
- Kuhn, A., Ducasse, S. & Gîrba, T. (2007). Semantic clustering: Identifying topics in source code., *Information & Software Technology* **49**(3): 230–243.
- Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience.
- Kurozumi, Y. & Davis, W. (1982). Polygonal approximation by the minimax method, *Computer Vision, Graphics and Image Processing* **19**: 248–264.
- Kyrgyzov, I., Maître, H. & Campedel, M. (2005). Combining clustering results for the analysis of textures of spot5 images, *ESA-EUSC: Image Information Mining*.
- Kyrgyzov, I., Maître, H. & Campedel, M. (2007a). A method of clustering combination applied to satellite image analysis, *IEEE - International Conference on Image Analysis and Processing ICIAP 2007* pp. 81–86.
- Kyrgyzov, I., Maître, H. & Campedel, M. (2008). A method of clustering combination, *The Journal of the Pattern Recognition Society* (submitted).
-

- Kyrgyzov, I. O., Kyrgyzov, O. O., Maître, H. & Campedel, M. (2007b). Kernel MDL to determine the number of clusters, *In International Conference on Machine Learning and Data Mining MLDM* **4571/2007**: 203–217.
- Lance, G. N. & Williams, W. T. (1967). A general theory of classificatory sorting strategies: Ii clustering systems, *Computer Journal* **10**: 271–277.
- Lange, T. & Buhmann, J. M. (2005). Combining partitions by probabilistic label aggregation, *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, ACM, pp. 147–156.
- Larose, D. T. (2006). *Data Mining Methods and Models*, Wiley-IEEE Press.
- Lau, K. & Wade, G. (Aug 1991). Spatial-spectral clustering using recursive spanning trees, *Communications, Speech and Vision, IEE Proceedings I* **138**(4): 232–238.
- Le Hegarat-Masclé, S., Bloch, I. & Vidal-Madjar, D. (1997). Application of dempster-shafer evidence theory to unsupervised classification in multisource remote sensing, *Geoscience and Remote Sensing, IEEE Transactions on* **35**(4): 1018–1031.
- Li, T., Ogihara, M. & Ma, S. (2004). On combining multiple clusterings, *CIKM '04: Proceedings of the thirteenth ACM conference on Information and knowledge management*, ACM Press, New York, NY, USA, pp. 294–303.
- Liu, Y., Zhang, D., Lu, G. & Ma, W. (2007). A survey of content-based image retrieval with high-level semantics, *PR* **40**(1): 262–282.
- Mackay, D. J. C. (2002). *Information Theory, Inference & Learning Algorithms*, Cambridge University Press.
- Manthalkar, R., Biswas, P. & Chatterji, B. (2003). Rotation invariant texture classification using even symmetric gabor filters, *Pattern Recognition Letters* **24**(12): 2061–2068.
- Marcotorchino, F. & El ayoubi, N. (1991). Logical paradigm of relational writings of some fundamental measures of association. (Paradigme logique des écritures relationnelles de quelques critères fondamentaux d'association.), *Rev. Stat. Appl.* **39**(2): 25–46.
- Marcotorchino, F. & Michaud, P. (1982). Agrégation de similarités en classification automatique., *Rev. Stat. Appl.* **30**(2): 21–44.
- Marine Campedel, Marie Lienou, I. K. H. M. (2008). Vers la construction d'une ontologie appliquée à l'imagerie satellitaire, *Extraction de COonnaissance et Images, ECOI*.
- Marques de Sá, J. P. (2001). *Pattern Recognition: Concepts, Methods and Applications*, Springer, Berlin.
- Mclachlan, G. & Peel, D. (2000). *Finite Mixture Models*, Wiley-Interscience.
- Michaud, P. & Marcotorchino, F. (1979). Modeles d'optimisation en analyse des données relationnelles., *Math. Sci. Hum.* **67**: 7–38.
- Muchoney, D.M., B. J. & Strahler, A. (1996). Global landcover classification validation issues and requirements, **1**: 233 – 235.

- Ng, A. Y., Jordan, M. I. & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm, *Advances in Neural Information Processing Systems 14*, MIT Press, Cambridge, MA, pp. 849–856.
- Papakonstantinou, G. (1985). Optimal polygonal approximation of digital curves, *Signal Processing* **8**: 131–135.
- Parulekar, A., Datta, R., Li, J. & Wang, J. Z. (2005). Large-scale satellite image browsing using automatic semantic categorization, *iccvw* **0**: 1873.
- Pratt, W. K. (2001). *Digital Image Processing*, John Wiley & Sons, Inc.
- R. Hanson, J. S. & Cheeseman, P. (May, 1991). Bayesian classification theory, *Technical report*, FIA-90-12-7-01, NASA Ames Research Center, Artificial Intelligence Branch.
- Rencher, A. C. (2002). *Methods of multivariate analysis.*, Vol. 2nd, Wiley-Interscience.
- Rissanen, J. (1978). Modeling by shortest data description, *Automatica* **14**: 465–471.
- Rissanen, J. (1984). Universal coding, information, prediction, and estimation, *IEEE Trans. Inform. Theory* **30**(4): 629–636.
- Rissanen, J. (1995). Stochastic complexity and its applications, *Technical report*.
- Rowe, D. B. (2002). *Multivariate Bayesian Statistics: Models for Source Separation and Signal Unmixing*, Chapman and Hall/CRC.
- Scholkopf, B. & Smola, A. J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA, USA.
- Schwarz, G. (1978). Estimating the dimension of a model, *The Annals of Statistics* **6**(2): 461–464.
- Schölkopf, B., Smola, A. & Müller, K.-R. (1996). Nonlinear component analysis as a kernel eigenvalue problem, *Technical Report 44*, Max Planck Institute for Biological Cybernetics, Tübingen, Germany.
- Shanmugan, K., Haralick, R. M. & Dinstein, I. (1973). Textural features for image classification, *IEEE Transactions on Systems, Man and Cybernetics* **3**(6): 610–621.
- Shawe-Taylor, J. & Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*, Cambridge University Press.
- Stein, A., Meer, F. v. d. & Gorte, B. (2002). *Spatial Statistics for Remote Sensing*, Springer.
- Strehl, A. & Ghosh, J. (2002). Cluster ensembles - a knowledge reuse framework for combining multiple partitions, *Journal on Machine Learning Research (JMLR)* **3**: 583–617.
- Suykens, J. & Horvath, G. (2002). *Advances in Learning Theory: Methods, Models, and Applications*, I O S Press, Incorporated.
- Theodoridis, S. & Koutroumbas, K. (2003). *Pattern Recognition, Second Edition*, Academic Press.
-

- Topchy, A., Jain, A. & Punch, W. (2004a). A mixture model for clustering ensembles, in *Proc. SIAM Conf. on Data Mining*, pp. 379–390.
- Topchy, A., Minaei-Bidgoli, B., Jain, A. & Punch, W. (2004b). Adaptive clustering ensembles, In *Proc. Intl. Conf. on Pattern Recognition, ICPR'04*, pp. 272–275.
- Vapnik, V. N. (1998). *Statistical Learning Theory*, Wiley-Interscience.
- Vapnik, V. N. (1999). *The Nature of Statistical Learning Theory (Information Science and Statistics)*, Springer.
- Vetterli, M. (1986). Filter banks allowing perfect reconstruction, *Signal Process.* **10**(3): 219–244.
- Ward, J. H. (1963). Hierarchical grouping to optimize an objective function, *Journal of the American Statistical Association* **58**: 236–244.
- Webb, A. R. (2002). *Statistical Pattern Recognition, 2nd Edition*, John Wiley & Sons.
- Weldon, T., Higgins, W. & Dunn, D. (1996). Efficient gabor filter design for texture segmentation, *PR* **29**(12): 2005–2015.
- Witten, I. H. & Frank, E. (1999). *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann.
- Xu, L., Krzyzak, A. & Suen, C. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition, *SMC* **22**(3): 418–435.
- Y. Qian, C. S. (2000). Clustering combination method., *ICPR* **2**: 732–735.
- Yu, S. & Shi, J. (2003). Multiclass spectral clustering, *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*.
- Zammit, O., Descombes, X. & Zerubia, J. (2007). Assessment of different classification algorithms for burnt land discrimination, *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International* pp. 3000–3003.
- Zhao, Y. & Karypis, G. (2004). Empirical and theoretical comparisons of selected criterion functions for document clustering., *Machine Learning* **55**(3): 311–331.
-

Index

-
- Akaike information criterion, 93
 - Bayesian decision theory, 81
 - Bayesian information criterion, 92
 - Bernoulli distribution, 125
 - Bernoulli mixture model, BMM, 125
 - Between-, within- cluster criteria, 90
 - Binomial distribution, 124
 - Cholesky decomposition, 134
 - Clustering, 71
 - Clustering stability, stable patterns, 155
 - Co-association matrix, 130
 - Co-occurrence matrix, CM, 47
 - Combination of clusterings, 117
 - Combinatorial clusterings, 123
 - Curse of dimensionality, 61
 - Data mining, 33, 39, 183
 - Earth observation, 31
 - Edge detection, 49
 - Eigen vector decomposition, 111, 132
 - EM-algorithm for BMM, 126
 - EM-algorithm for MMM, 129
 - Epanechnikov kernel, 148
 - Estimation of clustering, 89, 160
 - Expectation-Maximisation (EM) algorithm for GMM, 84
 - Feature extraction, 45
 - Gabor features, 47
 - Gaussian Mixture Model, GMM, 83
 - Geometrical features, 49
 - Haralick features, 46
 - Hierarchical agglomerative clustering, 73, 76
 - Hierarchical average-link clustering, 75
 - Hierarchical Bi-section clustering, 77
 - Hierarchical centroid-link clustering, 75
 - Hierarchical clustering based on KMDL, 106
 - Hierarchical complete-link clustering, 75
 - Hierarchical divisive clustering, 77
 - Hierarchical K-section clustering, 78
 - Hierarchical median-link clustering, 75
 - Hierarchical single-link clustering, 74
 - Hierarchical Ward's clustering, 76
 - Image content, 32
 - Image semantic, 181
 - Information measure, 92
 - K-means, 78
 - Kernel K-means, 79
 - Kernel MDL, KMDL, 98
 - Maximum Likelihood Classification, 82
 - Mean shift combination, 147
 - Minimum description length, MDL, 93
 - Model selection, 87
 - Multinomial mixture model, 128
 - Nominal data clustering, 121
 - Partitional clustering, 78
 - QMF features, 48
 - Quadratic programming, 135
 - Spectral K-means, 81
 - SPOT5, 31, 41
 - Stochastic complexity, 93
 - Supervised classification, 59
 - Support vector machines, SVM, 59
 - Unsupervised classification, 71
 - Validity criteria for hierarchical clustering, 90
 - Validity criteria for partitional clustering, 90
 - Visualisation of clustering results, 184
-

