Robust clustering algorithms for image segmentation and curve analysis

Wang, Zhimin

2009

Wang, Z. (2009). Robust clustering algorithms for image segmentation and curve analysis.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/41537>

<https://doi.org/10.32657/10356/41537>

Robust Clustering Algorithms for Image Segmentation and Curve Analysis

Wang Zhimin

School of Electrical & Electronics Engineering

A thesis submitted to Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

August 2009

Acknowledgements

I would like to express my sincere thanks to my supervisors Professor Soh Yeng Chai and Dr. Song Qing for their endless encouragement, insightful guidance, and invaluable comments throughout my research study. I would not have been able to complete my Ph.D without the many years of support from them. I would also like to thank Dr. Sim Kang for his valued guidance with respect to the medical image applications.

I also wish to thank Nanyang Technological University, Singapore for awarding me the research scholarship and providing excellent facilities to pursue my graduate study. My special thanks goes to Mr. Sow Peck Heng and Mr. Yuen Sien Huan in Computer Vision Lab (CVL), for their kind support and assistance during my Ph.D study. The lab mates at CVL also deserve much thanks, especially Yeo Yong Kiang and Mahshid Farzinfar.

Personally, I would also like to express my deep appreciation to my parents for their endless love, support and encouragement through the whole course of my studies; and to my wife, Gu Yuan for her unconditional love, and understanding.

Summary

Data clustering has become one of the most important research areas of pattern recognition. The objective of data clustering is to use the cluster concept to simplify the representation of large amount of data objects and generate meaningful clusters for further analysis and interpretation. Such a technology is useful in many disciplines, such as computational biology, bioinformatics, medical image processing, digital image segmentation, affective computing, real-time market forecast and online document clustering search engine.

There are several crucial steps in a pattern analysis system based on data clustering methodologies. These include data collection, feature extraction/selection, clustering strategy, and clustering output interpretation. Among these issues, the clustering method is an especially important one. Robustness, efficiency, extendibility, and universality of a data clustering analysis system are usually determined by the data clustering method. However, there is no universal clustering technique that is always applicable for uncovering the variety of structures present in the data sets. This thesis focuses on the development of adaptive, robust, and generalized data clustering methods for real applications.

One of the key contributions of this thesis is the development of an adaptive spatial information-theoretic clustering' (ASIC) algorithm. The proposed algorithm can solve the lack of spatial information problem in data clustering based image segmentation methods, as well as the lack of ability to tolerate noise and outliers in data clustering algorithms. Furthermore, the adaptive similarity measure proposed in this algorithm will resolve the problem of non-adaptive and experimentally set spatial weighting factors used in the literature. With the proposed adaptive similarity measure, the spatial context incorporation process in the ASIC algorithm is fully adaptive to the local image content and does not require any experimentally adjusted parameter. This newly defined distance measure is also useful for enhanc-

ing the ability to tolerate noise and outliers. For high level outliers after spatial information clustering, the ASIC uses the capacity maximization concept in information theory to identify and reassign them based on a bigger adaptive window. In image segmentation, the strengths of the proposed ASIC algorithm include: 1) it can produce more homogenous segmentation results; 2) it preserves the fine details of the image while reducing the edge blurring artifact significantly in a relatively noisy environment; and 3) the spatial constraint is adaptive to the image, and does not require any user interference.

We also extend the concept of information-theoretic framework and adaptive similarity measure to the famous fuzzy C-means clustering algorithm for image segmentation. A robust fuzzy clustering algorithms named 'adaptive spatial information fuzzy clustering' (ASIFC) algorithm is proposed. Combining these two concepts and modifying the objective function of the FCM algorithm, we are able to solve the sensitivity of noisy data and the lack of spatial information problems and improve the image segmentation results. The experimental results with simulated MRI images and real MRI images demonstrate that the proposed algorithm can indeed produce more homogeneous segmentation results than the FCM-type algorithms.

Besides the efforts on the development of robust clustering algorithms based on the intensity-feature of the image, the robust analysis of curve based clustering methods are also addressed in this thesis. Rather than using intensity values as feature to analyze the image, it is more natural and convenient to use curves. For the conventional curve clustering approaches, the most popular Gaussian statistical model is not sufficient to handle the variability of the real data sets and outlier problems. This led to a need to develop a more robust statistical model. We proposed a more robust statistical model based on multivariate t -Distribution to tackle these problems. With this robust random mixture effect model, the performance in terms of estimating the model parameters has been improved and the negative impacts from the outlying observations have been well taken care of. Furthermore, the proposed model provides a flexible way to account for the varying real data distributions and to identify the outlying observations.

Table of Contents

Acknowledgements	ii
Summary	iv
List of Figures	xii
List of Acronyms	xiii
Author's Publications	xiv
1 Introduction	1
1.1 Motivations	2
1.2 Objectives	4
1.3 Main Contributions of the Thesis	5
1.4 Organization of the Thesis	7
2 Literature Review on Data Clustering and Image Segmentation	9
2.1 Introduction	9
2.2 Notation and Terminology	11
2.3 Data Representation	12

TABLE OF CONTENTS

v

2.4	Similarity Measure	13
2.5	Standard Data Clustering Techniques	16
2.5.1	K-means algorithm	17
2.5.2	Fuzzy C-means algorithm	18
2.5.3	Deterministic annealing	19
2.6	Outliers and Cluster Number	21
2.7	Image Segmentation by Clustering	24
2.8	Image Segmentation by Other Methods	28
2.9	Conclusions	29
3	Adaptive Spatial Information Clustering (ASIC) Algorithm	31
3.1	Preliminaries	33
3.1.1	Modified FCM algorithms and its variants	33
	BCFCM algorithm	34
	EnFCM algorithm	35
	FGFCM algorithm	36
3.1.2	Information clustering	36
	Mutual Information	37
	Minimization of mutual information and data clustering . . .	39
	Maximization of mutual information and data reliability . . .	44
3.2	Adaptive Spatial Information Clustering Algorithm	50
3.2.1	Improved similarity measure and MI minimization for basic clustering and low level outlier elimination	52

TABLE OF CONTENTS

vi

3.2.2	MI maximization and adaptive window for high level outlier elimination	57
3.2.3	Remarks on the ASIC algorithm	61
3.3	The ASIC Algorithm	62
3.4	Experimental Results	64
3.4.1	The need of incorporating the spatial information with data clustering approaches	64
3.4.2	Quantified and visualized comparisons between ASIC and DA, BCFCM, EnFCM, and FGFCM with synthetic images	69
	Segmentation results on a synthetic image corrupted by mixed noise	69
	Quantified comparison of SA results on synthetic images of different noise types and levels	71
	Edge blurring effect with low level outlier elimination	71
	Outliers identification by MI maximization with high level outlier elimination	72
3.4.3	Visual comparisons between ASIC and DA, BCFCM, EnFCM, and FGFCM with real images	76
	Real Image - Plane	76
	Real Image - Lena	76
	Real Image - Cameraman	79
3.4.4	3D Segmentation of Brain Magnetic Resonance Image (MRI) .	81
3.5	Conclusions	82

4 A Robust Information Fuzzy Clustering Algorithm for Image Seg-

TABLE OF CONTENTS

vii

mentation	84
4.1 An Adaptive Spatial Information Fuzzy Clustering (ASIFC) Algorithm	86
4.2 The Proposed ASIFC Algorithm	88
4.3 Experimental Results	90
4.3.1 Synthetic images	90
4.3.2 Phantom MRI brain image	93
4.3.3 MRI brain image	98
4.4 Conclusions	99
5 Data Clustering with Curve Data Representation	103
5.1 Introduction	103
5.2 Literature Review	104
5.3 B-spline Curve Representation	108
5.4 Multivariate Gaussian Random Effect Mixture Model	111
5.5 Expectation-Maximization (EM) Framework for the Multivariate Gaussian Random Effect Mixture Model	113
5.5.1 The complete data log-likelihood	114
5.5.2 E-step solution	116
5.5.3 M-step solution	118
5.6 Conclusions	120
6 Robust Curve Clustering based on Multivariate t-Distribution Model	121
6.1 Introduction	121
6.2 A Multivariate t -Distribution Model for Robust Curve Clustering . .	123

TABLE OF CONTENTS

viii

6.2.1	Robust statistical model using multivariate t -Distribution . . .	123
6.2.2	The complete-data log likelihood	124
6.2.3	ML solutions using the EM algorithm	126
	E step	127
	M step	129
6.2.4	Computational Complexity of the Proposed Model	131
6.2.5	Algorithm implementation	132
6.3	Experimental Results	134
6.3.1	Simulated curves	134
6.3.2	Silhouetted face profile contour	139
6.3.3	Vehicle curves	140
6.3.4	UCR times series database	144
6.4	Conclusions	148
7	Conclusions and Recommendations for Future Research	150
7.1	Conclusions	150
7.2	Recommendations for Future Research	152
	Bibliography	155

List of Figures

1.1	Stages for data clustering process.	4
2.1	An example of data clustering.	10
2.2	Intensity value extraction of a grey image.	13
2.3	Functional data representation by curve	14
2.4	The outliers are data points with equal or almost equal distances to their adjacent cluster centers.	22
2.5	The impact of outlier observation in data clustering	23
2.6	An example for image segmentation with thresholding	25
3.1	Effect of noise on segmentation result	35
3.2	Recursive algorithm to maximize the mutual information.	49
3.3	8-neighborhood system in a 3×3 window.	51
3.4	The map of weighting parameters	54
3.5	The adaptive spatial parameter $\{\lambda_{i,j}\}$ under homogeneous regions and inhomogeneous regions.	56
3.6	Example for high level outlier elimination	60
3.7	An image histogram with two Gaussian distribution like classes. . . .	66

LIST OF FIGURES

x

3.8	The histogram of class 1 in segmented image.	66
3.9	Comparison between ASIC algorithm and DA algorithm	67
3.10	Histograms of the segmentation results	68
3.11	Synthetic Image	70
3.12	Handling edge blurring effect results on synthetic images, corrupted with three different levels of artificial noise	73
3.13	Performance of several algorithms on edge blurring effect	74
3.14	High level outlier elimination	75
3.15	Real Image - Plane	78
3.16	Real Image - Lena	80
3.17	Real Image - Cameraman	81
3.18	3D MRI brain image segmentation with ASIC algorithm	82
4.1	Comparison of BCFCM, EnFCM, FGFCM, ASIC and ASIFC ap- proaches on a synthetic image	92
4.2	Example of simulated MRI brain image segmentation	94
4.3	Tissue Segmentation Accuracy (TSA) value of CSF, GM, and WM with different noise levels for 40 simulated MR brain images.	97
4.4	Example 1 - Segmentation on a sagittal view slice	100
4.5	Example 2 - Segmentation on a coronal view slice	101
4.6	Example 3 - Segmentation on a transverse view slice	102
5.1	Face contours extracted by landmark based method	105
5.2	Face contours represented by B-splines	105

5.3	Vehicle contours extracted using method in [113]	106
5.4	Anatomical contour representation via deformable model for MRI brain images	107
5.5	Construction of a cubic spline curve	109
5.6	An example of cubic B-spline curves with 6 control points.	110
5.7	B-spline curves with its control points and basis functions.	112
6.1	Plot of t-distribution densities with degree of freedom for 0.1, 3, 10 compared to Gaussian distribution density (Magenta).	122
6.2	Example 1. Simulated curve data from four different functions. (a) Y_1 . (b) Y_2 . (c) Y_3 . (d) Y_4	135
6.3	Example 2. Simulated curves with two outlier curves contaminated with anomalous noise.	137
6.4	Simulated face profile contour with 18 key points	140
6.5	Simulated face curve data for G1, G2, and G3 groups. (a) Simulated data for G1, 50 curves. (b) Simulated data for G2, 50 curves. (c) Simulated data for outliers cluster G3, 10 curves.	141
6.6	Clustering results true (solid line) and estimated (dashed line) mean curve profiles for the case with outlier face curves. (a) True and estimated mean curve profiles for G1 group based on Gaussian model. (b) True and estimated mean curve profiles for G2 group based on Gaussian model. (c) True and estimated mean curve profiles for G1 group based on t -Distribution model. (d) True and estimated mean curve profiles for G2 group based on t -Distribution model.	141
6.7	Vehicle curves classes. (a) Sedan; (b) MPV; (c) Sports car (outlier class).	143

6.8	UCR Times Series Database: synthetic control data set. (a) True mean curve for 6 classes for the training set. (b) Estimated mean curve with our proposed model. (c) Cluster 1. (d) Cluster 2. (e) Cluster 3. (f) Cluster 4. (g) Cluster 5. (h) Cluster 6.	145
6.9	UCR Times Series Database: ECG200 data set. (a) True mean curve for 2 classes for all the data samples. (b) Estimated mean curve with our proposed model.	147
6.10	True mean curves for Trace dataset from UCR times series database.	148

List of Acronyms

CPU	Central Processing Unit
ASIC	Adaptive Spatial Information-theoretic Clustering
EM	Expectation-Maximization
FCM	Fuzzy C-means
ML	Maximum Likelihood
MRI	Magnetic Resonance Imaging
DA	Deterministic Annealing
WM	White Matter
GM	Gray Matter
CSF	Cerebrospinal Fluid
BCFCM	Bias-Corrected Fuzzy C-means
AFCS	Adaptive Fuzzy Clustering/Segmentation
RFCM	Robust Fuzzy C-means
EnFCM	Enhance Fuzzy C-means
FGFCM	Fast Generalized C-means
RIC	Robust Information Clustering
MI	Mutual Information
ROI	Region of Interest
GMM	Gaussian Mixture Model
PCM	Possibilistic C-means
PFCM	Possibilistic Fuzzy C-means
ASIFC	Adaptive Spatial Information Fuzzy Clustering
MPRAGE	Magnetization Prepared Rapid Gradient Echo
DTI	Diffusion Tensor Imaging

Author's Publications

- P1. Zhi Min Wang, Yeng Chai Soh, Qing Song, and Kang Sim, "A Robust Information Fuzzy Clustering Algorithm for Medical Image Segmentation", accepted by GrC 2010.
- P2. Zhi Min Wang, Yeng Chai Soh, Qing Song, and Kang Sim, "Robust Information Fuzzy Clustering Algorithm for Image Segmentation", in preparation, 2009.
- P3. Qing Song, Xulei Yang, Yeng Chai Soh, Zhi Min Wang, "An information-theoretic fuzzy C-spherical shells clustering algorithm", in press, *Fuzzy Sets and Systems*, 2009.
- P4. Zhi Min Wang, Yeng Chai Soh, Qing Song, Kang Sim, "Adaptive Spatial Information-theoretic Clustering for Image Segmentation", *Pattern Recognition*, vol. 42, no. 9, pp. 2029-2044, 2009.
- P5. Zhi Min Wang, Qing Song, Yeng Chai Soh, and Kang Sim, "Robust Curve Clustering based on a Multivariate t-Distribution Model", accepted by *IEEE Transaction on Neural Network*, 2010.
- P6. Zhi Min Wang, Qing Song, Yeng Chai Soh, and Kang Sim, "Image clustering by incorporating adaptive spatial connectivity", 10th International Conference on Control, Automation, Robotics and Vision (ICARCV), pp. 657-661, 2008.
- P7. Zhi Min Wang, Qing Song, Yeng Chai Soh, and Kang Sim, "Improved Adaptive Spatial Information Clustering for Image Segmentation", 4th International Symposium on Visual Computing (ISVC), pp. 308-317, 2008.
- P8. XuLei Yang, Qing Song, Wenbo Zhang, and Zhi Min Wang, "Clustering Spherical Shells by a Mini-Max Information Algorithm", 7th Asian Conference on Computer Vision (ACCV), pp. 224-233, 2006.

- P9. Zhi Min Wang, Qing Song, Yeng Chai Soh, XuLei Yang, and Kang Sim, "Adaptive Spatial Information Clustering for Image Segmentation", International Joint Conference on Neural Network (IJCNN), pp. 4151-4158, 2006.
- P10. Zhi Min Wang, Qing Song, and Yeng Chai Soh, "MRI brain image segmentation by adaptive spatial deterministic annealing clustering", 3rd International Symposium on Biomedical Imaging (ISBI), pp. 299-302, 2006.

Chapter 1

Introduction

The rapid growth of the computational power has accelerated the mass deployment and use of computer technologies. Characterized by the Moore's law [1], the increase in storage capacities relieves the soaring needs of high capacity data storage from all industries. At the same time, the advances of central processor unit (CPU) and network technologies make faster processing and transmission of data and information possible. With these hardware improvements, advancing data processing technologies in the software aspect has also received special attentions from the engineering researchers. These advances have it possible to develop systems and programs to free human resources from some of the computationally intensive calculations and data-driven analysis tasks. This progress can be seen in the rapid development of pattern recognition methods, as well as in an increasing number of application software on the market. Within the many useful techniques in pattern recognition systems, data clustering is one of the key processing steps as it determines the categories of the data and also influences the subsequent decisions and actions in many applications. Data clustering, an unsupervised data mining technique, is therefore becoming an important tool for researchers of various disciplines, ranging from mathematics, statistics, bioscience, computer vision, image processing, financial analysis, traffic broadcasting and many others. Indeed, many theoretical research works have been done to achieve better and optimized data clustering methods in the past 40 years. One of the most significant practical benefits of using data clustering techniques is that it relieves human beings of the heavy computational work and tedious data

analysis activities. The saved human resource can be redirected and focused onto key application development and system design. In turn, modern data processing systems make faster and more accurate data manipulations possible, which further reduce labor and costs. Data clustering techniques are continuing to revolutionize the industries, with more and more complicated algorithms and advanced techniques being developed as the field grows.

Typical applications that have benefited from data clustering techniques include computational biology, bioinformatics, medical image processing, digital image segmentation, affective computing, real-time market forecast and online document clustering search engine. For example, in computational biology, data clustering techniques are used to determine the time-ordered gene expression data that have similar patterns of expression [2–5]. In the field of medical image processing and computer vision, data clustering techniques are widely used to segment an image or image sequences into several similar regions on which a clinical analysis or an object recognition can be performed [6–11]. Moreover, in some new academic disciplines such as affective computing and online document clustering search engine, we are also able to see the traces of data clustering techniques [12–16]. In fact, the advancement of data clustering techniques is still continuing and expanding. This thesis aims to contribute to this growing field by developing robust and efficient clustering methods for applications in image segmentation and functional data analysis.

1.1 Motivations

Essentially, data clustering techniques are geared towards finding internal structures in the data. The intrinsic structures of a dataset are helpful in simplifying the representation of the data, grouping the data objects sensibly, or labeling data points by a valid similarity measure. Once a dataset is separated into several clusters, further application-oriented assessments or visualization of these clusters can be applied, such as data mining, document retrieval, image segmentation, and pattern classification [17].

According to [18], the activities involved in a typical pattern analysis system based on clustering methodologies can be described by the flowchart in Fig. 1.1. Firstly, the raw data of interest are recorded/stored using a proper format in accordance with the applications. Then the raw data is converted to a form suitable for data clustering methods by means of feature extraction and selection. Once this representation is done, clustering methods are applied on these features and the output of the clustering process is ready for final assessment and interpretation. In a real application, e.g. medical image segmentation based on data clustering, this processing flow consists of the following steps:

1. *Data Collection*: 2-D slides or 3-D volume scans of the patient are generated with a specific imaging system and the digitalized image data is stored using hard disk or optical media.
2. *Feature Extraction/Selection*: The raw image data is described by some convenient data forms for suitable segmentation method, such as a vector formed by the intensity value of each pixel/voxel.
3. *Clustering Strategy*: A clustering method is chosen to cluster/group the obtained features and the output clusters are produced. Usually, the output of the clustering result is a simple and compact representation of the raw data.
4. *Output Interpretation*: The output of clustering results can be displayed with much less intensity levels which are easier for clinical analysis. In addition, the image segmentation results can be used together with other applications (for example, multi-modality image fusion) to provide more meaningful interpretation.

Among the many technical issues involved in the data clustering system, the clustering method is an especially important one. A good data clustering system should be able to reveal the inherent structure of the data and generate meaningful clusters for further analysis and interpretation. This is usually determined by the data clustering method. It also determines the performance and reliability of a data clustering

Figure 1.1: Stages for data clustering process.

system. Robustness, efficiency, extendibility, and universality [19] are the important factors when one evaluates a new data clustering method. However, there is no universal clustering technique that is always applicable for uncovering the variety of structures present in the data sets [17]. Therefore, the development of adaptive, robust, and generalized data clustering methods is the most important motivation of this thesis.

1.2 Objectives

The main objective of this research is to develop robust clustering methods for real applications. Two fields are of interest in this research. One is image segmentation, the other is functional data analysis. In image segmentation, we are aiming to develop an adaptive image segmentation method that is able to make use of spatial contextual information from the image so as to provide clearer and more robust segmentation result. The proposed method should allow the incorporation of the spatial information which is always lacking in conventional data clustering based image segmentation methods. Also, the model complexity of the proposed method should be simple, which means that it should use as little prior information or fewer experimental parameters as possible. While in the functional data analysis field, we shall investigate a robust statistical model for functional data clustering. This model should inherit the merits of existing statistical model, yet taking into consideration the existence of noisy data points or outlier samples. The key objectives, can therefore be summarized as:

- (i) To develop a novel similarity measure for image segmentation that is adaptive

to the image content and robust to noisy data points or outlier.

- (ii) To propose robust image segmentation methods based on data clustering algorithm such that they can handle the outlier problem when incorporating the spatial information.
- (iii) To develop a robust multivariate statistical model which is able to provide more accurate clustering results even in the presence of noisy data points or outlier sample.

1.3 Main Contributions of the Thesis

Three robust data clustering methods are to be proposed in this research work. The first one, named 'Adaptive Spatial Information-theoretic Clustering' (ASIC) algorithm, is developed for robust image segmentation. The second robust information fuzzy clustering, called 'Adaptive Spatial Information Fuzzy Clustering' (ASIFC) algorithm is also developed for robust image segmentation. The last robust clustering algorithm, which is based on multivariate t-Distribution model, is designed to provide robust clustering results for functional data represented by B-splines. Hence, the major contributions of this thesis can be summarized as follow.

1. ASIC - adaptive similarity measure

An adaptive similarity measure is proposed for incorporating the spatial information into the image segmentation method based on data clustering. This novel similarity measure employs a fully automated technique to choose the weighting factor for neighborhood effect during the clustering process. Totally adaptive to the local image context, the proposed adaptive similarity measure offers an effective way to eliminate the noise data or outliers through the given spatial constraints automatically and is controlled by the homogeneity degree of the neighborhood windows.

2. ASIC - two steps outlier elimination framework

A two-step information-theoretic clustering algorithm is proposed to efficiently eliminate outliers in a relatively noisy environment. In the first step, we incorporate the adaptive similarity measure into the objective function to perform the soft clustering by minimizing the mutual information between the clusters and input patterns. In the second step, we use the maximization mutual information scheme to evaluate the reliability of the input patterns at a higher level, i.e. to identify the equivalent outliers after incorporating the spatial information. These remaining outliers can be reassigned based on an adaptive window to further improve the segmentation results.

3. **A Robust Information Fuzzy Clustering Algorithm for Image Segmentation**

In order to improve the robustness of the conventional fuzzy C-means (FCM) clustering algorithms for image segmentation, a robust information fuzzy clustering algorithm is proposed. This is an extension of the information-theoretic framework into the FCM-type algorithms. It aims at solving two intertwined problems, i.e. the identification of noisy data and the lack of spatial information. The experimental results have shown that this robust clustering algorithm is useful for the MRI brain image segmentation and yield better segmentation results when compared to the conventional FCM approach.

4. **Robust Curve Clustering based on a Multivariate t-Distribution Model**

A curve clustering technique based on a new multivariate model is developed. Instead of the popular Gaussian random effect model, our new method uses the multivariate t-Distribution model which has better robustness to outliers and noise. In our method, we use B-spline curve to model curve data and apply the mixed effects model to capture the randomness and covariance of all curves within the same cluster. After fitting the B-spline based mixed-effects model to the newly proposed multivariate t-Distribution, we derive an Expectation-Maximization (EM) algorithm for estimating the parameters of the model. The proposed new model is able to overcome the lack of robustness of the

regular Gaussian distribution model. The simulation results of simulated data have demonstrated that this new robust mixed model has better performance and works well when there is outliers or noise data.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows:

Chapter 2 gives a detailed literature review of data clustering based image segmentation applications. Several popular clustering methods are included like the K-means and FCM clustering methods. We also introduce some modified variants of the FCM algorithm with the incorporation of the spatial information.

Chapter 3 formulates our proposed ASIC algorithm. The adaptive similarity measure will be presented in this chapter and the performance of its adaptiveness will also be demonstrated.

Chapter 4 presents a robust fuzzy clustering algorithm for image segmentation. We aim to extend the ideas in chapter 3 to address the noisy data and lack of spatial information in the FCM-type algorithms for image segmentation applications.

Chapter 5 reviews the data clustering techniques with the curve or contour representations. The popular B-spline curve representation method will be introduced, as well as the multivariate Gaussian random effect mixture model. Detail derivations of the maximum likelihood (ML) estimates via Expectation-Maximization algorithm will also be provided.

Chapter 6 proposes a robust curve clustering method based on the Multivariate t-Distribution Model. Mathematic formulations are presented in this chapter as well as the simulation results of its performance.

Chapter 7 concludes this thesis with a summary and some discussions on future work and research directions.

Chapter 2

Literature Review on Data Clustering and Image Segmentation

2.1 Introduction

Over the last twenty years, the research into the use of data clustering methods in pattern recognition has grown significantly. Data clustering is the unsupervised classification of a collection of patterns (observations, data items, or feature vectors) into groups (clusters) based on similarity. Intuitively, patterns within a valid cluster are more similar to each other than they are to a pattern belonging to a different cluster. The aim of the data clustering techniques is to identify and understand the structure from the data objects and provide better perception and representation of the data. A good example of clustering is illustrated in Fig. 2.1. The input patterns are shown in the left hand side of Fig. 2.1, and the outputs are shown in the right hand side of Fig. 2.1. Here, points belonging to the same cluster are given the same label.

Most of the clustering methods were developed in the past two decades. The major contribution to the development of clustering methods can be traced back to 1963. Robert Sokal [20] used a clustering method to group a set of organisms based the degree of similarity among these organisms. They used the clustering results to

Figure 2.1: An example of data clustering.

analyze the evolutionary process of biological species [19]. Since then, numerous books and influential review papers have been published on clustering. The book entitled *Pattern classification* [21], is a good starter to learn about all the basic concepts and elements on data clustering from the pattern recognition perspective. Another comprehensive book on data clustering algorithms was given by Jain *et al.* [18]. Their survey [17] on data clustering is also very famous in the field of data analysis. On top of these, you will find that the survey papers and books in [10, 11, 22–28] are useful to explore the problems, methods and applications on data clustering in computer science, biology, economics and psychology.

The rest of this chapter will present the literature review of data clustering technologies and its existing problems when applying them in image segmentation. In Section 2.2, the notation and terminology used throughout this thesis will be provided. Section 2.3 presents a few examples of data representation techniques. Several commonly used clustering algorithms will be reviewed in Section 2.5, and the problems of clustering techniques are presented in Section 2.6. Finally, the reviews of image segmentation by clustering are given in Section 2.7.

2.2 Notation and Terminology

In this section, the general notation and terminology that are used throughout this thesis will be briefly described. In later chapters, more specific notations will be provided when it is first introduced.

Features and Feature Vector

In general, a feature is an individual scalar component of a feature vector that is used by the data clustering algorithm. Feature vector can also be known as a pattern or an observation. Usually, the feature is represented as x_i and the feature vector is represented in boldface as $\mathbf{x} = (x_1, \dots, x_g)^T$, with g as the dimensionality of the feature vector. The input patterns can be presented by a capital boldface $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$, where N is the number of input patterns. By introducing the subscripts i, j , we can denote the input pattern set as an $i \times j$ matrix $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_N)$, where $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,j}, \dots, x_{i,g})^T$. In particular, if the feature dimension is 1, the representation will be simplified as $\mathbf{X} = (x_1, \dots, x_i, \dots, x_N)$.

Cluster Center and Cluster Number

The cluster center is usually represented in a similar way as $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_k, \dots, \mathbf{w}_K)$, where $\mathbf{w}_k = (w_{k,1}, \dots, w_{k,j}, \dots, w_{k,g})^T$. Here, K is the cluster number, which is pre-defined or can be estimated through the data clustering algorithm.

Class and Class label

The term *class*, in statistical analysis, is used to describe a set of patterns that generated by a specific probability density function [17]. In data clustering, we use this term to represent a set of patterns which are generated by some specific pattern generation process [29]. A class label is a class identifier to each pattern. Normally, the labels of all input patterns $\mathbf{X}$ can be represented as $\mathcal{L} = (l_1, \dots, l_i, \dots, l_N)$, where $l_i \in \{1, \dots, K\}$.

Hard Clustering and Soft Clustering

Hard clustering is a conventional data clustering technique which assigns the input

patterns to a particular class with either TURE (assign to it) or FAUSE (not assign to it) condition. The recent developed probabilistic framework for clustering offers better versatility to this association process [30]. Soft clustering allows the assigning of input patterns to each output cluster in probability. It is extremely useful in attacking the partial volume effect problem in medical image segmentation [31].

2.3 Data Representation

The proper representation of the application data is a very important prerequisite step for clustering methods. Simple, easy for visualization, and good representative data forms are always desired in various communities. Application-related data sets are all required to be converted by proper ways so that a specific data clustering method can be applied to reveal the intrinsic, topological and interpretable information from the original data. In many applications, the feature vectors are constructed from the discretized responses sampled from the data collection process, such as micro array gene data or magnetic resonance imaging (MRI) brain image data.

Although our research focus is not on data representation, we shall list a few representative examples of data representation in the real applications in order to provide a better understanding of data presentation in data clustering related applications.

1. Image Segmentation

In the digitalized image segmentation process, the intensity values of the gray image is usually extracted as the feature. Fig. 2.2 shows the whole process. The intensity values from the entire image grid are extracted to form a feature vector. The sequences of the extracted intensity values are not important to the data clustering process. However, it is of interest to know that the correspondences between the image location within the grid and the index of the feature is useful for segmentation visualization after the data clustering procedure.

Figure 2.2: Intensity value extraction of a grey image.

2. Functional Data Analysis

A convenient way of representing a set of functional data is to use curves. According to [32], this representation offers few advantages: 1) It greatly reduces the dimensionality of the functional data after fitting them with splines or smoothing functions. With this representation, the whole set of data points are substituted by a few functional basis parameters. 2) This representation will also help to smooth out the disturbance noise. 3) The curve representation is useful to preserve the intrinsic functional information and able to recover the missing data. Fig. 2.3 (a) shows a curve representation example for time course micro array gene data, and Fig. 2.3 (b) displays a process of using B-spline curve to model the face silhouette.

2.4 Similarity Measure

There are two types of clustering methods in unsupervised data clustering. One is hierarchical clustering, characterized by Single linkage clustering [35]; the other one is partitional clustering, represented by K-means algorithm [36]. As the name implied, the hierarchical clustering method is a sequence of partitional steps to group the input patterns into a nested groups. It is useful for applications that do not have specified cluster number and require supervision from a data analyst. On the other

Figure 2.3: Functional data representation by curve. (a). Gene expression data represented by curves (image is extracted from Wentzell *et al* [33]). (b). Face silhouette model by B-spline curve. (image is extracted from Davidenko [34]).

hand, partitional clustering offers one step solution for the clustering and widely used in engineering applications [18].

Both clustering methods need to use a similarity measure as criterion to cluster the input patterns. It is a significant step for any clustering methods. By modifying the similarity measure, one may be able to tune the clustering algorithm to fit a specific application requirement and improve the clustering results. We will discuss this in details later.

The similarity measure is usually done by calculating the total distance between the input patterns and cluster centers. Commonly used distance functions are: Euclidean distance, Mahalanobis distance [37], and maximum norm distance. The famous square-error distance in data clustering is Euclidean distance.

Mathematically, the similarity measure based on Euclidean distance can be written as

$$D = \sum_{i=1}^N \sum_{k=1}^K ||\mathbf{x}_i - \mathbf{w}_k||^2 \quad (2.4.1)$$

The clustering method using such a similarity measure favors a partition scheme that has a minimum value of D . It is equivalent to assigning the input pattern $\mathbf{x}_i$ to its nearest cluster center $\mathbf{w}_k$.

Rose [30] introduced association probability functions $\{p(\mathbf{w}_k|\mathbf{x}_i)\}$ in his soft clustering algorithm, named Deterministic Annealing (DA) clustering algorithm. With this probabilistic framework, the similarity measure for DA clustering algorithm can be rewritten as

$$D_{DA} = \sum_{i=1}^N \sum_{k=1}^K p(\mathbf{x}_i)p(\mathbf{w}_k|\mathbf{x}_i)||\mathbf{x}_i - \mathbf{w}_k||^2 \quad (2.4.2)$$

Similar to Eq. (2.4.2), the famous fuzzy c-means (FCM) algorithm defined its own similarity measure from the fuzzy set point of view. The standard FCM algorithm is based on minimizing the following similarity measure:

$$J_{FCM} = \sum_{i=1}^N \sum_{k=1}^K u_{ik}^q ||\mathbf{x}_i - \mathbf{w}_k||^2, 1 \leq q < \infty \quad (2.4.3)$$

where q is a real number greater than 1. It controls the fuzzyness of the clustering results. u_{ik} is the membership value of data point $\mathbf{x}_i$ inside the k th cluster.

Other forms of similarity measure using non-Euclidean distance also have their appropriate domains of applications. Babu *et al.* [38] used Manhattan distance in classification of compressed data encoded using run-length. A hierarchical clustering framework based on Mahalanobis distance was proposed by Szeto *et al.* [39] for gene express data clustering.

As we mentioned earlier, a shortcoming of these conventional similarity measure is that there is no spatial information in the distance functions. The subscript of $\mathbf{x}_i$ is the position of the element in the vector $\mathbf{X}$, and is determined by the way one forms this input vector. However, no matter how one defines the sequential position of all data points $\mathbf{x}_i$, the clustering results rely on values of $\mathbf{x}_i$ only. In other words, the clustering result is merely based on the similarity between feature values of input patterns and centers. This problem makes the standard data clustering methods very sensitive to noise and other imaging artifacts when applying to image segmentation.

2.5 Standard Data Clustering Techniques

There are many data clustering techniques available for data analysis. Since our research focus is mainly on the engineering side, we will only discuss the clustering methods for the branch of partitional clustering in this section. With the given similarity measure, a partitional clustering method produces a partition solution so that a given cost function can be mathematically minimized. This minimization is usually accompanied by some prescribed constraints as complete search of the optimal solution within the feature space is definitely computationally prohibitive. The solutions to these clustering algorithms are iterative, and sensitive to the initialization of the starting states.

For rest of this section, several popular standard clustering approaches that have

been widely used by many researchers will be briefly described. These methods are: (1) K-means algorithm, (2) fuzzy C-means algorithm, and (3) deterministic annealing algorithm.

2.5.1 K-means algorithm

The K-means clustering algorithm [36] is the most commonly used algorithm. K-means algorithm generates a specific number of disjoint, non-hierarchical clusters. The goal of K-means algorithm is to divide the input patterns into K clusters such that the similarity measure relative to the centers of the clusters are minimized. The K-Means method is fast, unsupervised, non-deterministic and iterative.

The objective function and the details of K-means algorithm are provided below:

$$D = \sum_{i=1}^N \sum_{k=1}^K \|\mathbf{x}_i - \mathbf{w}_k\|^2 \quad (2.5.4)$$

Step 1 : Choose K initial cluster centers $\mathbf{w}_1; \dots, \mathbf{w}_K$ randomly from the N points $\mathbf{x}_1, \dots, \mathbf{x}_N$.

Step 2 Assign point $\mathbf{x}_i \in \mathbf{X}, i = 1, \dots, N$ to cluster $\mathbf{w}_k, k \in \{1, \dots, K\}$ iff

$$\|\mathbf{x}_i - \mathbf{w}_k\|^2 < \|\mathbf{x}_i - \mathbf{w}_p\|^2, p \in \{1, \dots, K\}, \text{ and } p \neq k \quad (2.5.5)$$

Ties are resolved arbitrarily.

Step 3 Compute new cluster centers $\mathbf{w}_1^*, \dots, \mathbf{w}_K^*$ as follows:

$$\mathbf{w}_k^* = \frac{1}{n_k} \sum_{\mathbf{x}_j \in \mathbf{w}_k} \mathbf{x}_j, k = 1, \dots, K \quad (2.5.6)$$

where n_k is the number of elements belonging to cluster $\mathbf{w}_k$.

Step 1 If a convergence criterion is met, then terminate. Otherwise continue from step 2

It is well known that K-means algorithm may converge to values that are not optimal [40] and is sensitive to the selection of the initial partition [18]. Several methods regarding the initialization issue of K-means algorithm have been reported in [41–44].

2.5.2 Fuzzy C-means algorithm

The K-means algorithm only clusters input patterns into crisp, or hard clusters and a feature vector can be a member of one cluster only, while fuzzy clustering allows feature vectors to have memberships of multiple clusters, rather than only one cluster in the case of hard clustering algorithm like K-means. The fuzzy set concept was introduced first by Zadeh [45] in 1965. It offers a new way to isolate and identify functional relationship of the data points.

One approach to fuzzy clustering, probably the most commonly used, is the FCM algorithm, which was proposed by Dunn [46] in 1973 and improved by Bezdek [47] in 1981. FCM is formulated as the minimization of the following objective function with respect to the membership function u and the centers $\mathbf{w}$:

$$J_{fcm} = \sum_{i=1}^N \sum_{k=1}^K u_{ik}^q \|\mathbf{x}_i - \mathbf{w}_k\|^2, \quad 1 \leq q < \infty \quad (2.5.7)$$

where q is any real number greater than 1, u_{ik} is the degree of membership of $\mathbf{x}_i$ in the k th cluster, and $\|\cdot\|$ is any norm expressing the similarity measure. By specifying the appropriate similarity measure, FCM algorithm can be applied to data that possesses ellipsoidal shaped clusters [48, 49], although typically the Euclidean distance measure is used.

Intuitively, the objective function (2.5.7) is minimized when the high membership values are assigned to input patterns which are close to their nearest cluster center and low membership values are assigned when they are far from the other cluster center.

With the hidden constraint $\sum_{k=1}^K u_{ik} = 1$, minimizing the objective function (2.5.7), we have

$$\frac{\partial J_{fcm}}{\partial u_{ik}} = 0 \quad , \quad \frac{\partial J_{fcm}}{\partial \mathbf{w}_k} = 0 \quad (2.5.8)$$

These lead to the iterative solutions

$$u_{ik} = \frac{||\mathbf{x}_i - \mathbf{w}_k||^{-2/(q-1)}}{\sum_{c=1}^K ||\mathbf{x}_i - \mathbf{w}_c||^{-2/(q-1)}} \quad (2.5.9)$$

and

$$\mathbf{w}_k = \frac{\sum_{i=1}^N u_{ik}^q \mathbf{x}_i}{\sum_{i=1}^N u_{ik}^q} \quad (2.5.10)$$

Iterating through these two necessary conditions leads to the minimum of J_{fcm} . This iteration will stop when $|u_{ik}^{(p+1)} - u_{ik}^{(p)}| < \varepsilon$, where ε is a termination condition between 0 and 1, p is the iteration steps. This procedure converges to a local minimum or a saddle point of J_{fcm} .

Even though the FCM algorithm is better than the 'hard' K-means algorithm by allowing the input patterns to be assigned to more than one clusters, it still has several problems. First, FCM may converge to local minima of the squared-errors criterion. The presence of noise will make this problem worse. It is very difficult to handle this problem in the standard FCM formulation [50]. This drawback has motivated researchers to find alternative ways to get a good initialization, for example, Pham *et al.* [51] used a kernel estimator to select the initial centers when segmenting the MRI brain images, and other researchers proposed robust version of FCM algorithms [50, 52, 53].

The other problem with this prototype-based clustering algorithm is the preset number of clusters. FCM algorithm requires the user to provide the number of clusters as an input. But the user in general has no idea about the number of clusters. This is problematic and the clustering result may not be good especially when the number of clusters is very large and not easy to guess [50].

2.5.3 Deterministic annealing

The deterministic annealing (DA) algorithm has been shown to be an effective optimization tool for clustering, compression, classification, regression and related optimization problems [30], [54]. As its name suggests, DA algorithm consists of two

aspects. On one hand, it is deterministic, which means the search strategy is not random. On the other hand, it is an annealing method and it aims at the global minimum, instead of being attracted to a local minimum like K-means, or FCM. This algorithm is formally based on the principles of probability theory and information theory [55], and it is aiming at minimizing the objective function with certain levels of randomness.

By using the concepts in information theory and introducing the probabilistic framework, the DA algorithm can be viewed as minimizing the average distortion function between the input pattern and its best reproduced cluster:

$$\begin{aligned}
 F &= D - TH \\
 &= \sum_{i=1}^N \sum_{k=1}^K p(\mathbf{x}_i) p(\mathbf{w}_k | \mathbf{x}_i) d(\mathbf{x}_i, \mathbf{w}_k) + \\
 &\quad T \sum_{i=1}^N \sum_{k=1}^K p(\mathbf{x}_i) p(\mathbf{w}_k | \mathbf{x}_i) \log(p(\mathbf{x}_i) p(\mathbf{w}_k | \mathbf{x}_i))
 \end{aligned} \tag{2.5.11}$$

where $p(\mathbf{x}_i)$ is the probability of choosing the i th input pattern $\mathbf{x}_i$, and $p(\mathbf{w}_k | \mathbf{x}_i)$ is the conditional probability relating input data $\mathbf{x}_i$ to center $\mathbf{w}_k$. The T is a Lagrangian multiplier. The first term D is the similarity measure, the second term H is randomness (Shannon's entropy [56]). Therefore, we are able to recast the DA algorithm as one of seeking the distribution which minimizes D subject to a specified level of randomness.

Minimizing the objective function F with respect to the free parameters $p(\mathbf{w}_k | \mathbf{x}_i)$ and $\mathbf{w}_k$ gives the solutions

$$p(\mathbf{w}_k | \mathbf{x}_i) = \frac{p(\mathbf{w}_k) \exp\left(-\frac{d(\mathbf{x}_i, \mathbf{w}_k)}{T}\right)}{Z_x} \tag{2.5.12}$$

and

$$\mathbf{w}_k = \frac{\sum_{i=1}^N \mathbf{x}_i p(\mathbf{x}_i) p(\mathbf{w}_k | \mathbf{x}_i)}{p(\mathbf{w}_k)} \tag{2.5.13}$$

where the normalization is

$$Z_x = \sum_{k=1}^K p(\mathbf{w}_k) \exp\left(-\frac{d(\mathbf{x}_i, \mathbf{w}_k)}{T}\right) \tag{2.5.14}$$

The central iteration of DA algorithm, therefore, consists of two main steps:

1. Fix the cluster centers and use (2.5.12) to compute the membership functions.
2. Fix the membership functions and optimize the centers according to (2.5.13).

Although DA algorithm suppresses the local minimum problem, meaning that it is not sensitive to the volume of clusters, it still suffers from the same problems as the FCM algorithm, i.e. it is not robust to the noise and outliers as pointed out in [50].

2.6 Outliers and Cluster Number

One of the basic challenges of data clustering techniques is the outlier problem. An outlier is a suspicious observation that is quite different from other observations in terms of measurement or position. The existence of outliers will alter the actual distribution of the underlying data set, and hinder the accuracy calculation of the true mass center during the data clustering process. This consecutive negative impacts will lead to clustering errors for the main part of the observations.

Most of the outlier observations are caused by errors during the data collection process. While it is necessary to improve the quality and accuracy of data acquisition, proper outlier detection and elimination methods are important to the development of new data clustering methods. Many methods have been developed for the detection of the outlier observations during the data clustering process [57–60]. More information about outlier detection can be found in [61, 62].

Besides those outliers which are far away from the main part of the observations, most of the outliers are the data points which have equal or almost equal similarity distances to their adjacent cluster centers (see Fig. 2.4). We refer to this kind of outlier observations as border outliers. The existence of border outliers may shift the true cluster center into wrong positions or may cause difficulty in assigning the data label using correct cluster numbers [60]. Fig. 2.5 shows a typical example, which has two well separated compact clusters with seven points in each cluster and two outliers sitting in the upper middle positions of the figure. Figs 2.5 (b)-(c) show that the standard mass constrained DA algorithm that separates the two clusters by

Figure 2.4: The outliers are data points with equal or almost equal distances to their adjacent cluster centers.

combining the two outliers into one cluster, and the two estimated cluster centers are shifted from the true positions as compared to Figs 2.5 (d)-(e).

Another important issue of clustering is the choice of number of clusters. While many applications may have a prior knowledge about the number of clusters, a proper scheme of choosing the cluster number is needed when the prior information is not available. Different cluster number will yield different quality of the clustering results. A large cluster number will emphasize the details of data set and suppress the purpose of data clustering. However it is able to reduce the clustering errors (In the extreme case, there is no clustering error if the cluster number is equal to the number of input data points!). On the other hand, if the cluster number is too small, the data cannot be well separated and the intrinsic features of the data cannot be revealed. But the model of the clustering system with a smaller cluster number is simple and easy to implement [P9].

One of the commonly used criteria for obtaining the optimal cluster number is to find the clustering result that has compact clusters and well separated cluster centers. The Davies-Boulding index [63] is such a criterion. The objective of this index is to minimize the intra-cluster scatter and maximize the inter-cluster separation. Hubert and Schultz [64] had also introduced a criterion named "C-index" to validate the cluster number. It uses the sum of distances of the specific pairs of patterns

Figure 2.5: The impact of outlier observation in data clustering. (a) The original data set. (b) The clustering result using standard DA algorithm ($K=2$). (c) The clustering result using standard DA algorithm ($K=3$). (d) The idea clustering result with outlier detection ($K=2$). (e) The idea clustering result with outlier detection ($K=3$). In the clustering results, "*" and "o" represent the patterns in two clusters, respectively, the bigger "*" denotes the cluster centers, and the black point is the identified outlier observation.

to determine the optimal cluster number. Usually, a small value of C indicates good clustering result. Other attempts can be found in [46, 65]. Moreover, many efforts have also been made in choosing an optimal cluster number to balance the needs of both model complexity and clustering errors [50, 60, 66, 67]. More generate information about this problem can be found in [29, 68].

2.7 Image Segmentation by Clustering

Image segmentation is one of the important steps in many computer vision applications. The goal of image segmentation is to partition a given image into regions or parts that have a strong correlation with patterns or areas corresponding to the objects in real world. In this sense, the image segmentation is closely related to data clustering and can be well interpreted through the clustering point of view. Using data clustering techniques in image segmentation is quite straightforward and only a few adaptations are needed during the implementation. In the rest of the section, we will focus on the image segmentation based on clustering techniques and the existing problems associated with it.

In order to perform the image segmentation using data clustering techniques, an actual or appropriate number of clusters should be provided in advance. This choice is application related. According to [9], $K = 2$ (binary segmentation) is adequate for displaying or printing when the storage and transmission are of great importance. Otherwise, $K = 4$ will be more appropriate for other general real images. While in MRI brain image segmentation applications, $K = 3$ is widely used as the goal of the segmentation is to divide the entire image into subregions such as the white matter (WM), gray matter (GM), and cerebrospinal fluid (CSF) spaces [69]. Our current research focus is not to develop clustering algorithms which are able to automatically identify an optimal cluster number for a specific image segmentation application. For the sake of simplicity, we assume that the cluster number are known in advance for all simulated or real images.

Once the cluster number of the target image is set, the next step is the feature vector

Figure 2.6: An example for image segmentation with thresholding. (a) Original image. (b) Good threshold segmentation. (c) Threshold too low. (d) Threshold too high.

construction. The dimensionality of the feature vector depends on the type of the image. For example, for gray-level image, the feature vector is a one-dimensional vector that consists of all the gray values and usually is displayed as histogram. For color image, the feature space is 3-Dimensional because the every pixel consists of three RGB (red, green and blue) values.

The simplest segmentation process is thresholding. The feature vector is clustered via high and low threshold values for each cluster. A simple example of using thresholding technique for image segmentation is shown in Fig. 2.6. It is obvious that a good threshold selection is crucial for successful segmentation, but it is also a drawback of this simple segmentation method.

Partitional clustering algorithms, however, offer a more effective way of finding the optimal threshold values for image segmentation. This process is automatic once the cluster number and clustering method are given. Compared to other segmentation

methods, image segmentation based on clustering method has high reproducibility because the induced segmentation results are mainly based on the original information of the image data, with little or no assumption on the model and data distribution. However, one major disadvantage of standard unsupervised clustering methods is that they rely only on the intensity distribution of the pixels, and disregard their geometric information, making them very sensitive to noise and other artifacts introduced during the imaging process. Recently, the issue of incorporating spatial smoothness into clustering techniques has attracted much interest in one of the most rapidly developing research areas, medical image analysis.

In medical image analysis, Wells *et al.* [7] presented an expectation-maximization (EM) algorithm as a tool for segmentation of MRI brain images with bias artifact. This method is simple, but assumes the intensity values of the brain tissues follow the Gaussian distribution. In addition, lacking the spatial information of each pixel, this method is sensitive to the intensity artifact introduced during the image data recording process. Later, Mohamed et al. [53] used a modified FCM algorithm for medical image segmentation. They introduced the spatial information into the similarity measure. The similarity measure is modified such that the pixel can be dragged closer to the cluster center if it is in homogenous regions. Although there is no more model assumption in the FCM algorithm, the drawbacks of the FCM algorithm [50] are its sensitivity to the non-descriptive initial centers and its massive computational load. Therefore, Ahmed et al. [70] used another similarity measure in a Bias-Corrected FCM (BCFCM) algorithm for bias field estimation and segmentation, which achieved a better performance when an additional term is added to the new objective function. The influence of the neighborhood effect can be adjusted with a constant α , without influencing the center pixel in the objective function. It is not effective if the center pixel is highly different from the neighbors, for example in the case of the salt and pepper noise. Recently, a possibility c-means based algorithm called the Adaptive Fuzzy Clustering/Segmentation (AFCS) [71] was proposed to impose spatial constraints by adding/subtracting a small positive constant ϵ to/from the membership of the central pixel in a 3×3 window. The

addition or subtraction depends on whether the pixel in the 8-neighborhood belongs to the same or different class as its central pixel. Inspired by [71], Liew et al. [67] proposed a spatial FCM clustering algorithm for clustering and segmenting by using both the feature space and spatial information. This was different from [70], as it is believed that the influence of the central pixel is varying in the similarity measure. The method uses a weighting function to weigh the influence of the central pixel and its neighbors. When the central pixel is in a homogeneous region, the similarity measure should be greatly dominated by its neighbors. The converse holds for edge regions. This algorithm was shown to have better performance than the others in terms of adaptive selection of the weighting factor. However, user intervention is still needed when choosing the parameter σ of the weighting function and a correct initialization of the FCM algorithm. Another variant of FCM algorithm called the Robust Fuzzy C-means algorithm (RFCM) was proposed by Dzung [72]. A penalty term is added into the objective function to constrain the membership values. It favors configurations where the central pixel has a high membership value in one class while adjacent pixels have low membership values in other classes. Similar to other spatial models, this RFCM algorithm still suffers from the over-smoothed edges problem. Moreover, the cross-validation scheme for choosing the smoothing parameter β will not always find the appropriate β . Recently, Szilágyi et al. [73] proposed an Enhanced Fuzzy C-Means (EnFCM) clustering algorithm to segment the image more effectively while using the spatial information. However, this method still depends on a fixed spatial parameter α which needs to be adjusted. The shortcoming of using fixed spatial parameter is evident. Another disadvantage emerges from this problem is that it will also blur the image feature like sharp edges while smoothing out noises. In order to overcome these problems, Cai et al. [74] proposed a Fast Generalized Fuzzy C-Means (FGFCM) algorithm by introducing a new localized similarity measure. This new similarity measure makes use of local and spatial intensity information, and so it performs better than the EnFCM algorithm. It has lower misclassification rate by reducing the blurring effect. However, the need for experimentally adjusted parameter and the blurring problems still exist in FGFCM.

We can also find the trace of this effort in computer vision area [8, 9, 75–79]. And the devoted effort is continuing, with more and more researchers developing more robust and effective data clustering methods for image segmentation.

2.8 Image Segmentation by Other Methods

Image segmentation is a process of pixel classification that aims to extract or segment objects or regions from the background. Pal and Pal [11] provided a good review of the various segmentation techniques. It should be noted that there is no single standard approach to segmentation. Many different types of scene parts can serve as the segments on which descriptions are based, and there are many different ways in which one can attempt to extract these parts from the image. The selection of an appropriate segmentation technique depends on the type of images and applications.

There are primarily four types of segmentation techniques: thresholding, boundary-based, region-based, and hybrid techniques. Thresholding is based on the assumption that clusters in the histogram correspond to either background or objects of interest that can be extracted by separating these histogram clusters. In addition to thresholding, many image segmentation algorithms are based on two basic properties of the pixel intensities in relation to their local neighborhood: discontinuity and similarity. Methods based on pixel discontinuity are called boundary-based or edge extraction methods, whereas methods based on pixel similarity are called region-based methods.

It is well known that such segmentation techniques - based on boundary or region information alone - often fail to produce accurate segmentation results. Hence, there has been a tendency towards hybrid segmentation algorithms which take advantage of the complementary nature of such information. Hybrid methods combine boundary detection and region growing to achieve a better segmentation [80].

The evaluation of segmentation algorithms has been mostly subjective; therefore people usually judge an algorithm's effectiveness based on intuition and the results

from several segmented images. Cardoso and Corte-Real [81] proposed a generic framework for segmentation evaluation using symmetric and asymmetric distance metric alternatives to meet the specificities of a wide class of applications. Unnikrishna *et al.* [82] demonstrated how a proposed measure of similarity, the normalized probabilistic rand (NPR) index, can be used to perform a quantitative comparison between image segmentation algorithms using a hand-labeled set of ground-truth segmentations.

Other variations of image segmentation algorithms can be referred to locating object contours by the snake model [83], linking edges by adaptive mathematical morphology [84], automatic seeded region growing [85], top-down region dividing [86], as well as some mathematic model based techniques, like hidden Markov Model [77], Bayesian methods [87], Genetic Algorithm [88] and Artificial Neural Network [89], and multicue partial differential equation [90].

2.9 Conclusions

Data clustering has recently become a generally useful tool for many kinds of applications, such as computer vision, economics, biology, medical image analysis and online document mining. From the literature review presented in this chapter, we can observe that developing appropriate, robust and effective data clustering method to meet the rapidly changing needs of existing and new branches of science is still a challenge to researchers.

Since many clustering applications are real-world applications, the data representation and the requirement of clustering process are not always the same. This requires the new data clustering techniques to be adaptable to the changes of application areas.

There are many inherited problems of data clustering techniques when applying them to a real application, such as choice of correct similarity measure, outlier detection or cluster number validation. In image segmentation, the lack of spatial

information in the similarity measure is one of the obstacles to successful segmentation. Although some work has been done to overcome this problem, this problem still requires further studies to make the data clustering techniques more robust to noise data and outliers.

Other subproblems like bad initialization, experimental-set parameters, model assumption of input patterns and pre-defined cluster number are still open problems in clustering analysis. Effective initialization techniques are needed to reduce the impact of initialization to the final clustering results. Experimentally-set parameters and assumptions that make current problems tractable are not always reliable. More complicated clustering techniques need to be developed to keep up with the fast pace of change in the research world.

Chapter 3

Adaptive Spatial Information Clustering (ASIC) Algorithm

A data clustering method must be robust if it is to be useful in practical applications. According to Huber [91], in engineering practice, a robust clustering should not be affected significantly by small deviations (caused by noise, outliers or other effects) from the assumed model. In image segmentation, robustness means the segmentation algorithm is tolerant to the presence of noise and variations in the intensity values of data. That is, a small drift in the intensity values in the magnitude image should not have a significantly impact (for example, misclassifying some points in gray matter into white matter) on the segmentation result.

Generally, when considering whether a data clustering algorithm is robust or not, there are three aspects: 1) robust to initialization of prototype and cluster number, 2) robust to cluster volumes, and 3) robust to noise and outliers [92]. The common strategy among the various robust clustering algorithms for image segmentation is to modify the objective functions by incorporating the spatial information so that a subset of the input patterns can be ignored or reconsidered by the influence from the neighborhood patterns, which can also be viewed as a robust weighting function to phase out outliers. However, there remain some problems in these robust clustering algorithms [53, 67, 70, 70, 73, 74].

One of the shortcomings is the use of a fixed amount of spatial weighting factors.

With a fixed amount of spatial influence from the neighboring pixels, the center pixel will be dragged to the cluster center where most of its neighboring pixels belong. It is equivalent to applying a mean filter. It may smooth out the important structures, such as regional borders or edges, in the image. Some of the existing methods proposed the use of adaptive spatial weighting factors [67, 74]. Another problem of using a fixed amount of spatial influence is that it is not adaptive to the image content. On one hand, the spatial parameter should be set as high as possible to smooth out the noise. On the other hand, it should be set small enough to preserve the fine details and sharpness of the image. Clearly, a fixed selection scheme is not adequate means to cope with this trade-off problem. A more robust, noise-insensitive, locally adaptive control parameter selection method should be used to consistently improve the clustering by incorporating the spatial information.

Some algorithms used localized and adaptive spatial weighting factor selection schemes [67, 74] to overcome the problem of using a fixed spatial factor and thereby improving the segmentation results. However, the need for experimentally adjusted parameters and the blurring problems still exist. In this chapter, we will introduce a novel *Adaptive Spatial Information-theoretic Clustering* (ASIC) algorithm for image segmentation based on the original *robust information clustering* (RIC) algorithm developed by Song [60]. The ASIC algorithm is a two-step image segmentation algorithm. In the mutual information (MI) minimization step, our adaptive spatial clustering algorithm is based on a 3×3 window and has a specific similarity measure that incorporates the spatial information to eliminate low level outliers. A new spatial parameter is introduced to overcome the problems in the bias corrected fuzzy c-means (BCFCM) algorithm, the fast generalized fuzzy c-means (FGFCM) algorithm and their variants. The new parameter has several advantages: 1) It can be computed before the clustering step without any prior knowledge and assumption of the noise in the image; 2) The information needed to compute the new parameter is totally from the local neighborhood window and the gray values of all the pixels within that window. It is totally adaptive to the local image context without any pre-defined parameters; 3) The new parameter influences not only the neighbor-

hood terms of the objective function but also the central pixel term of the objective function, in such a way that the center pixel will be greatly influenced by the neighboring pixels when it is in a homogeneous region, otherwise, the center pixel will have small neighboring constraint if it is in the edge region. This is very helpful for eliminating the edge blurring artifact. In the MI maximization step, the remaining outliers (high level outliers after the spatial information clustering), i.e. the remaining unreliable pixel assignments are identified from an information-theoretic point of view, and reassigned based on an adaptive window. This feature provides ASIC with the capabilities to eliminate outliers in a relatively stronger noisy environment as compared to other conventional algorithms with spatial variants. Comprehensive experiments to show the superiority of the proposed ASIC algorithm for image segmentation will be provided in Chapter 6.

3.1 Preliminaries

In this section, we will briefly introduce some robust data clustering algorithms for image segmentation. Most of these work in image segmentation is based on FCM algorithm and its variants. Since our ASIC algorithm is formulated from an information clustering point of view that unifies data clustering and outlier assessment in a single framework, the basic ideas of information clustering will be discussed as well. For the sake of simplicity, we will constrain our discussion to gray level images, i.e. one-dimensional gray-level feature vector. In the rest of this thesis, we will use scalar notations wherever it is appropriate.

3.1.1 Modified FCM algorithms and its variants

The standard objective function has been shown in Section 2.5. For convenience, we rewrite it again:

$$J_q = \sum_{i=1}^N \sum_{k=1}^K u_{ik}^q \|x_i - w_k\|^2, 1 \leq q < \infty \quad (3.1.1)$$

where q is a real number greater than 1 which controls the fuzzyness of the clustering results, and u_{ik} is the membership value of data point x_i inside the k th cluster.

A shortcoming of the above equation is that there is no spatial information in the objective function of standard FCM algorithm. Therefore, when we apply the FCM algorithm in image segmentation, the image matrix $\mathbb{I}_{M \times N}$ (supposing that we use a 2D image with resolution $M \times N$) will be represented into a one dimensional source vector $x_i \in \mathbf{X}, i = 1, \dots, M \times N$, and used as input for clustering. The subscript of x_i is the position of the element in the vector $\mathbf{X}$, and is determined by the way one forms this input vector. However, no matter how one defines the sequential position of all data points x_i , the clustering results rely on the values of x_i only. In other words, the clustering result is merely based on the similarity between pixel intensity values and centers, and the spatial locations of pixel points have no influence. This problem makes the standard FCM algorithm very sensitive to noise and other imaging artifacts.

Now, let's consider an example shown in Fig. 3.1. This is an artificial image with 2 clusters (white background and gray foreground). The gray square is corrupted by one noise pixel which has the same intensity value as the background. With a standard FCM algorithm, the segmentation result is still affected by the noise pixel. A more robust algorithm is needed to suppress the noise data to produce a better segmentation result as shown in Fig. 3.1 (c).

BCFCM algorithm

In order to remedy this deficiency, Ahmed et al. [93] modified the objective function so that the assignment of a pixel is influenced by its neighboring pixels. This method is known as Bias-Corrected FCM (BCFCM), and the modified objective function is

$$J_s = \sum_{i=1}^N \sum_{k=1}^K u_{ik}^q \|x_i - w_k\|^2 + \frac{\alpha}{N_R} \sum_{i=1}^N \sum_{k=1}^K u_{ik}^q \sum_{r \in N_i} \|x_r - w_k\|^2 \quad (3.1.2)$$

where J_s denotes the spatial FCM objective function in order to distinguish from J_q , N_i is the subset of neighborhood pixels of x_i and N_R is the size of neighborhood pixels. The parameter α reflects the preference of neighboring effect.

Figure 3.1: Effect of noise on segmentation result. (a) original image with noise, (b) FCM segmentation result, (c) expected segmentation result.

It can be seen from (3.1.2) that the new objective function has two components. One puts the constraints that the pixels are to be closed to the cluster centers. The other imposes spatial continuity, and is weighted by α . It favors a piecewise-homogenous clustering solution.

EnFCM algorithm

Another significant modification is proposed by Szilágyi *et al.* [73], referred to as the Enhanced FCM (EnFCM) algorithm. In order to reduce the computation load, this method generates a pre-filtered image by incorporating the spatial information, so that the clustering algorithm can be directly applied to gray levels instead of pixels.

The new intensity value in the pre-filter image can be obtained as follow:

$$\xi_i = \frac{1}{1 + \alpha} \left(x_i + \frac{\alpha}{N_R} \sum_{r \in N_i} x_r \right) \quad (3.1.3)$$

where ξ_i denotes the new intensity value of the i th pixel of the pre-filtered image. The other notations are the same as defined earlier, and α plays the same role as before.

Once the new image is created, it is clustered by an accelerated version of FCM clustering. The new objective function is defined as

$$J_{EnFCM} = \sum_{k=1}^K \sum_{l=1}^m \gamma_l u_{il}^q ||l - w_k||^2 \quad (3.1.4)$$

where γ_l denotes the number of pixels with gray value that is equal to the new intensity value l in the pre-filtered image, and m is the number of gray levels in the generated new image.

FGFCM algorithm

Cai et al. [74] modified the spatial factor α in (3.1.3) and replaced it by the follow:

$$S_{ij} = \begin{cases} S_{s_{ij}} \times S_{g_{ij}}, & j \neq i \\ 0, & j = i \end{cases} \quad (3.1.5)$$

with

$$S_{s_{ij}} = \exp\left(\frac{-\max(|p_j - p_i|, |q_j - q_i|)}{\lambda_s}\right), \quad (3.1.6)$$

and

$$S_{g_{ij}} = \exp\left(\frac{-||x_i - x_j||^2}{\lambda_g \times (\sum_{j \in N_i} ||x_j - x_i||^2 / N_R)}\right) \quad (3.1.7)$$

where $S_{s_{ij}}$ is the spatial information within the neighborhood window, (p_i, q_i) is the spatial coordinate of the i th pixel, and λ_s is a preset scalar for $S_{s_{ij}}$. The location gray level information is introduced by $S_{g_{ij}}$, with λ_g as an experimentally adjusted parameter to control the spread of $S_{g_{ij}}$. With this new spatial factors, the FGFCM algorithm can overcome the problem of using a fixed spatial factor and improve the segmentation results.

3.1.2 Information clustering

The general intuition about data clustering is that similar patterns should be grouped together to the same cluster while dissimilar ones should not. By doing so, we can

use the cluster identifier to represent each data point, which results in a simpler and more compact description of the input patterns. This motivated us to reinterpret the data clustering problem with information theory [55,60]. In [94], it is mentioned that a data compression is a process that intentionally discards the fine details of the data. Instead of transmitting each individual data point (input pattern), one can design a set of specific data compression codes (cluster centers) and represent the individual data point by the code words to which they belong, so that more information contents can be transmitted. During this compression, distortion (similarity measure or objective cost function) of the data set will occur. Finally, a good design of the data compression code (data clustering algorithm) is desired to reduce the distortion. These resemblances enable us to recast the data clustering problems in a more general probabilistic framework with solid mathematic foundation. Several successful attempts were reported in [30,60,95].

Furthermore, by using the capacity concept in information theory, we can assess data reliability and the preference of the input data points. These data points can be viewed as outliers in data clustering point of view [50]. With this advantage, we are able to propose a robust data clustering algorithm which can identify outliers and reduce their negative impacts.

Mutual Information

One of the most important concepts in information clustering is mutual information. As the name implied, it is a numerical scale to measure the amount of information between input and output [94]. It can tell us about how much information we can get of the input by observing occurrences of the output. A vivid example of this characteristic can be found in medical image registration. Mutual information is used as quantitative measure to align two images. If the target image is properly aligned to the reference image, the mutual information is maximized. Otherwise, the mutual information between these two images is small.

In information theory, if a source symbol x occurs with probability $p(x)$, so the

amount of information associated with it can be measured by

$$I(x) = -\log p(x) \quad (3.1.8)$$

The common bases of the logarithm are 2 or e .

The mutual information between input symbol x and output symbol w can be interpreted as the information provided about input symbol x by the occurrence of the output symbol w . In another word, the mutual information measures how much uncertainty (information) of input symbol x is left if the output symbol w is observed. Therefore we have

$$I(x, w) = -\log p(x) - (-\log p(x|w)) = \log \frac{p(x|w)}{p(x)} \quad (3.1.9)$$

This is the basic mutual information definition about two particular realizations of input x and output w .

The commonly used formulation of the mutual information between two ensembles $\mathbf{X}$ and $\mathbf{W}$ can be expressed as

$$I(\mathbf{X}; \mathbf{W}) = \sum_x \sum_w p(x, w) \log \frac{p(x|w)}{p(x)} \quad (3.1.10)$$

where x and w are random variables, and $p(x, w), p(x|w), p(x)$ are probabilities. The value of mutual information depends on particular outcomes of x and w . It itself is also a random variable (see [96]).

Using Bayes formula, the mutual information is symmetric in $\mathbf{X}$ and $\mathbf{W}$. That is

$$\begin{aligned} I(\mathbf{X}; \mathbf{W}) &= I(\mathbf{W}; \mathbf{X}) \\ &= \sum_x \sum_w p(x, w) \log \frac{p(w|x)}{p(w)} \\ &= \sum_x \sum_w p(x, w) \log \frac{p(x|w)}{p(x)} \end{aligned} \quad (3.1.11)$$

According to the fundamental definition of mutual information [96], the mutual

information can also be defined with Shannon entropy as

$$\begin{aligned}
 I(\mathbf{X}; \mathbf{W}) &= I(\mathbf{W}; \mathbf{X}) \\
 &= H(\mathbf{X}) - H(\mathbf{X}|\mathbf{W}) \\
 &= H(\mathbf{W}) - H(\mathbf{W}|\mathbf{X}) \\
 &= H(\mathbf{X}) + H(\mathbf{W}) - H(\mathbf{X}, \mathbf{W})
 \end{aligned} \tag{3.1.12}$$

where H is entropy given by $H(\mathbf{X}) = -\sum_x p(x) \log p(x)$. From (3.1.12), it is easy to see that the mutual information is the average information that output ensemble $\mathbf{W}$ has about input ensemble $\mathbf{X}$. For more information about the additional properties about mutual information, please refer to [94, 96].

Minimization of mutual information and data clustering

During communication, we favor the use of simple representations to reduce the entropy rate of the input source, which can be viewed as data compression in information theory [94]. Such a compression will purposely discard the details of the input source and incur a certain level of distortion. Let $w_1, \dots, w_k, \dots, w_K$ be a sequence of source output, $x_1, \dots, x_i, \dots, x_N$ be the input sequence, and $p(w_k|x_i)$ be the association probability measure of using w_k to represent x_i , the average distortion can be measured by $\sum_{i=1}^N \sum_{k=1}^K p(x_i)p(w_k|x_i)(x_i - w_k)^2$. The aim is to minimize the average distortion to be as small as possible. Zero distortion can be achieved if each input data is represented by itself.

Now the problem is about how one can define a proper function to minimize the average distortion over all assignments of the association probabilities $p(w_k|x_i)$. Remember that the association probabilities $p(w_k|x_i)$, together with the input source probabilities $p(x_i)$, determine the mutual information between input symbols and output symbols. Hereby, a straightforward solution is to minimize the average distortion under the constraint of mutual information.

The rate-distortion function, in information theory, is defined as

$$R(\mathcal{D}) = \min_{\{p(w_k|x_i)\} \in Q_{\mathcal{D}}} I(\mathbf{X}; \mathbf{W}) \tag{3.1.13}$$

where

$$\begin{aligned} Q_{\mathcal{D}} &= \{p(w_k|x_i) \in R^K \times R^N : \\ &\sum_{i=1}^N \sum_{k=1}^K p(x_i)p(w_k|x_i)(x_i - w_k)^2 \leq \mathcal{D}; \\ &\sum_k p(w_k|x_i) = 1, p(w_k|x_i) \geq 0\} \end{aligned} \quad (3.1.14)$$

and $I(\mathbf{X}; \mathbf{W})$ is the mutual information between $\{p(x_i)\}$ and $\{p(w_k|x_i)\}$, which is given by

$$I(\mathbf{X}; \mathbf{W}) = \sum_{i=1}^N \sum_{k=1}^K p(x_i)p(w_k|x_i) \log \frac{p(w_k|x_i)}{p(w_k)} \quad (3.1.15)$$

Sometimes it is also convenient to use the following inverse function

$$D(\mathcal{R}) = \min_{\{p(w_k|x_i)\} \in Q_{\mathcal{R}}} \sum_{i=1}^N \sum_{k=1}^K p(x_i)p(w_k|x_i)(x_i - w_k)^2 \quad (3.1.16)$$

where

$$\begin{aligned} Q_{\mathcal{R}} &= \{p(w_k|x_i) \in R^K \times R^N : I(\mathbf{X}; \mathbf{W}) \leq \mathcal{R}; \\ &\sum_k p(w_k|x_i) = 1, p(w_k|x_i) \geq 0\} \end{aligned} \quad (3.1.17)$$

The average distortion is minimized under a specified randomness constraint by the mutual information.

From the rate-distortion function (3.1.13) and its constraint (3.1.14), it is easy to see that $R(\mathcal{D})$ is a function of $\mathcal{D}$. We will show later that $R(\mathcal{D})$ is convex and nonincreasing in $\mathcal{D}$, which is proved in the following theorem [94]. Other useful references can be found in [96, 97].

Lemma 3.1 *The mutual information $I(\mathbf{X}; \mathbf{W})$ is a concave function of $p(x)$ and a convex function of $p(w|x)$.*

Proof. For the concave property, let $p_1(x)$ and $p_2(x)$ be two arbitrary sets of input probabilities, and $p(x)$ is a combination probabilities $p(x) = \lambda p_1(x) + \bar{\lambda} p_2(x)$ for

any λ , $0 < \lambda < 1$, $\bar{\lambda} = (1 - \lambda)$. Let $I_1(X; W)$, $I_2(X; W)$, $I(X; W)$ be the mutual information for these probabilities, we will have

$$\begin{aligned}
& I(X; W) - \lambda I_1(X; W) - \bar{\lambda} I_2(X; W) \\
&= \sum_x \sum_w p(x)p(w|x) \log \frac{p(w|x)}{\sum_x p(w|x)p(x)} - \lambda \sum_x \sum_w p_1(x)p(w|x) \log \frac{p(w|x)}{\sum_x p(w|x)p_1(x)} \\
&\quad - \bar{\lambda} \sum_x \sum_w p_2(x)p(w|x) \log \frac{p(w|x)}{\sum_x p(w|x)p_2(x)} \\
&= \lambda \sum_x \sum_w p_1(x)p(w|x) \log \frac{\sum_x p_1(x)p(w|x)}{\sum_x p(x)p(w|x)} + \bar{\lambda} \sum_x \sum_w p_2(x)p(w|x) \log \frac{\sum_x p_2(x)p(w|x)}{\sum_x p(x)p(w|x)} \\
&\geq \lambda \sum_x \sum_w p_1(x)p(w|x) \left(1 - \frac{\sum_x p(x)p(w|x)}{\sum_x p_1(x)p(w|x)}\right) + \bar{\lambda} \sum_x \sum_w p_2(x)p(w|x) \left(1 - \frac{\sum_x p(x)p(w|x)}{\sum_x p_2(x)p(w|x)}\right) \\
&= 0
\end{aligned} \tag{3.1.18}$$

In the above, we use the inequality $\log x \geq 1 - \frac{1}{x}$. This concludes the proof for concave property. The convexity property can be proved by a similar way. $\square$

Theorem 3.1 $R(\mathcal{D})$ is a convex function and nonincreasing in $\mathcal{D}$.

Proof. To proof that $R(\mathcal{D})$ is a convex function, let $p_1(w|x)$ and $p_2(w|x)$ achieve the minimum in (3.1.13) for $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively. Set $p(w|x) = \lambda p_1(w|x) + \bar{\lambda} p_2(w|x)$ for any λ , $0 < \lambda < 1$, $\bar{\lambda} = (1 - \lambda)$, and the average distortion $\mathcal{D}$ for $p(w|x)$ is

$$R(\mathcal{D}) \leq I(\mathbf{X}; \mathbf{W}) \tag{3.1.19}$$

Since the mutual information is convex in $p(w|x)$, we can further upper bound $R(\mathcal{D})$ as

$$\begin{aligned}
R(\mathcal{D}) &\leq \lambda I_1(\mathbf{X}; \mathbf{W}) + \bar{\lambda} I_2(\mathbf{X}; \mathbf{W}) \\
&= \lambda R(\mathcal{D}_1) + \bar{\lambda} R(\mathcal{D}_2)
\end{aligned} \tag{3.1.20}$$

Finally, the average distortion associated with $p(w|x)$ is

$$\begin{aligned}
\mathcal{D} &= \sum_x \sum_w p(x)p(w|x)(x - w)^2 \\
&= \sum_x \sum_w p(x)(\lambda p_1(w|x) + \bar{\lambda} p_2(w|x))(x - w)^2 \\
&= \lambda \mathcal{D}_1 + \bar{\lambda} \mathcal{D}_2
\end{aligned} \tag{3.1.21}$$

Thus, we have

$$R(\lambda \mathcal{D}_1 + \bar{\lambda} \mathcal{D}_2) \leq \lambda R(\mathcal{D}_1) + \bar{\lambda} R(\mathcal{D}_2), \quad (3.1.22)$$

and the convexity is proved. Note that the minimization in 3.1.13 is over a constraint set, and this set will enlarge as the $\mathcal{D}$ is increased. Therefore, the minimum $R(\mathcal{D})$ is nonincreasing in $\mathcal{D}$. This completes the proof. $\square$

Since the rate-distortion function is nonincreasing and convex in $\mathcal{D}$, it must be strictly decreasing for any distortion smaller than $\mathcal{D}$. Therefore, the inequality constraint on $\mathcal{D}$ must be satisfied with equality constraint. Please refer to [94] for details. With this property, we can rewrite the rate-distortion function as

$$\begin{aligned} R(\mathcal{D}) = \min_{p(w|x)} & \left[\sum_{i=1}^N \sum_{k=1}^K p(x_i) p(w_k|x_i) \log \frac{p(w_k|x_i)}{\sum_{i=1}^N p(x_i) p(w_k|x_i)} \right. \\ & \left. - s \left(\sum_{i=1}^N \sum_{k=1}^K p(x_i) p(w_k|x_i) (x_i - w_k)^2 - \mathcal{D} \right) \right] \end{aligned} \quad (3.1.23)$$

where s is Lagrange multiplier. Now the association probabilities $\{p(w_k|x_i)\}$ are unconstrained and the minimum of rate-distortion can be achieved by a set of probabilities $\{\bar{p}(w_k|x_i)\}$. The value of distortion $\bar{\mathcal{D}}$ for the minimum solution can be calculated by

$$\bar{\mathcal{D}} = \sum_{i=1}^N \sum_{k=1}^K p(x_i) \bar{p}(w_k|x_i) (x_i - w_k)^2 \quad (3.1.24)$$

Note that the values for the pair $\mathcal{D}, R(\mathcal{D})$ are parameterized by the Lagrange multiplier s .

To this end, we have shown that the rate-distortion function is adequate to be used as objective function to minimize the distortion measure in a probabilistic way. By minimizing the mutual information across all possible values of association probabilities $\{p(w_k|x_i)\}$, we can find the minimum value of distortion measure $\bar{\mathcal{D}}$ which is controlled by the Lagrange multiplier s . With proper tuning mechanism (e.g. annealing process) of parameter s , we could find the optimal minimum distortion of the given transmission channels (the counterpart in data clustering is model complexity, i.e. cluster number).

Minimizing $R(\mathcal{D})$ with respect to the association probabilities $\{p(w_k|x_i)\}$, we have

$$\begin{aligned} \frac{\partial R(\mathcal{D})}{\partial p(w_k|x_i)} &= \frac{\partial}{\partial p(w_k|x_i)} \left[\sum_{i=1}^N \sum_{k=1}^K p(x_i)p(w_k|x_i) \log \frac{p(w_k|x_i)}{\sum_{i=1}^N p(x_i)p(w_k|x_i)} \right. \\ &\quad \left. - s \left(\sum_{i=1}^N \sum_{k=1}^K p(x_i)p(w_k|x_i)(x_i - w_k)^2 - \mathcal{D} \right) \right] \\ &= \log \frac{p(w_k|x_i)}{p(w_k)} + p(w_k) - s(x_i - w_k)^2 = 0 \end{aligned} \quad (3.1.25)$$

Then

$$p(w_k|x_i) = \frac{p(w_k)e^{s(x_i-w_k)^2}}{e^{p(w_k)}} \quad (3.1.26)$$

Using the fact that $\sum_{k=1}^K p(w_k|x_i) = 1$, again we have

$$p(w_k|x_i) = \frac{p(w_k)e^{s(x_i-w_k)^2}}{\sum_{k=1}^K p(w_k)e^{s(x_i-w_k)^2}} \quad (3.1.27)$$

To minimize the $R(\mathcal{D})$ with respect to $\{w_k\}$, we set the derivative to zero to yield the following condition

$$\begin{aligned} \frac{\partial R(\mathcal{D})}{\partial w_k} &= 2s \sum_{i=1}^N p(x_i)p(w_k|x_i)(x_i - w_k) \\ &= 0 \end{aligned} \quad (3.1.28)$$

Hence

$$w_k = \frac{\sum_{i=1}^N x_i p(x_i)p(w_k|x_i)}{p(w_k)} \quad (3.1.29)$$

Therefore, minimization of $R(\mathcal{D})$ in (3.1.13) with respect to $\{p(w_k|x_i), w_k\}$ would lead to an iterative scheme

$$p(w_k|x_i) = \frac{p(w_k)e^{(x_i-w_k)^2/T}}{\sum_{k=1}^K p(w_k)e^{(x_i-w_k)^2/T}} \quad (3.1.30)$$

$$w_k = \frac{\sum_{i=1}^N x_i p(x_i)p(w_k|x_i)}{p(w_k)} \quad (3.1.31)$$

which turns out to be the iterative solutions (2.5.12) and (2.5.13) in the Deterministic Annealing (DA) clustering algorithm. Here, the Lagrange multiplier $T = 1/s$.

The advantages of using rate-distortion function as data clustering objective function are notable. It is insensitive to the volume of input patterns as it tries to split the whole input patterns into several effective clusters during the annealing process [30]. It is a soft clustering process which can avoid the local minima of the hard clustering algorithm like k -means algorithm. It can also explore the details of the input pattern structure without any limit as long as a pre-specified cluster number is given [60].

Maximization of mutual information and data reliability

In the last section, we can obtain an optimal set of association probabilities during the minimization process of mutual information. Note that the prior probabilities $\{p(x_i)\}$ of choosing input patterns are usually fixed during the constrained minimization of mutual information, which implies that initially we do not have any preferences about the input patterns. A natural question arising from this limitation is that *'are we be able to access the input pattern reliability with these optimal association probabilities?'* or alternatively, *'what kinds of input patterns are more preferred than others with the given model and data compact representation?'*

Generally, one could be able to reaccess the data reliability by using the Bayes formula to obtain the *posteriori* probabilities $\{p(x_i|w_k)\}$, which is given by

$$p(x_i|w_k) = \frac{p(x_i)\bar{p}(w_k|x_i)}{\sum_{i=1}^K p(x_i)\bar{p}(w_k|x_i)} \quad (3.1.32)$$

However, directly using this *posteriori* probability to access the data reliability will give rise to an ill-posed optimization problem and it is not a good idea for outlier rejection [98]. Nevertheless, the capacity maximization in information theory offers us an alternative way to solve this problem.

In information theory, the capacity function is defined by the maximization form of mutual information, which is presented as

$$C = \max_{p(\mathbf{X})} I(\mathbf{X}; \mathbf{W}) = I(p(\mathbf{X}); \bar{p}(\mathbf{W}|\mathbf{X})) \quad (3.1.33)$$

where the maximization is over all probability distributions of the input patterns. We use an alternative representation of mutual information [94] to emphasize that

this maximization is under a fixed set of association probabilities $\{p(w_k|x_i)\}$. The capacity concept is an important concept in information theory, especially in the channel reliability assessment. It offers strong mathematical foundation of how to find the appropriate input probability assignments such that information can be transmitted at capacity. Actually, the capacity and rate-distortion function are inter-related in information theory. One wants to minimize the information of input data due to the limit of channel capacity. On the other hand, one also wants to fully utilize the capacity provided by the channels to maximize the information transmission.

Later, we will show that any data transmission with input information rate larger than capacity cannot be reliably transmitted through the channel. This is the well-known *converse theorem* in information theory [94]. It provides a new way to assess the data compression reliability (identifying outliers) from an information-theoretic point of view.

Before we begin the detail proof of the *converse theorem*, we introduce some notations which are to be used in the proof. Data clustering (compression) is treated as a coding scheme in the information theory [94] as the set of channel output $\mathbf{W}$ is estimated from the set of channel input $\mathbf{X}$. Furthermore, the coding error probability p_e of the input data can be defined asⁱ

$$p_e = \sum_{i=1}^{L_e} \sum_{k=1}^K p(x_i, w_k) = \sum_{i=1}^{L_e} \sum_{k=1}^K p(x_i) \bar{p}(w_k|x_i) \quad (3.1.34)$$

Similarly, the probability of coding the correct data can be defined as

$$p_c = 1 - p_e = \sum_{i=1}^{L_c} \sum_{k=1}^K p(x_i, w_k) \quad (3.1.35)$$

where L_e and L_c denote respectively the number of errors and correct data points that are compressed (clustered) into the K clusters with $L = L_e + L_c$.

Theorem 3.2 Suppose that $C + 1$ is larger than the input uncertainty $H(p(\mathbf{X}))$. If we intend to identify the most unreliable data from the given coding (clustering)

ⁱSimilar to [94], we temporary ignore $p_l = 0$, which turns out to represent the noisy data point in the discrete-time communication channel. Without loss of generality, we also assume that $L_e > 0$, otherwise there is an error free case.

result, i.e. $p_e \geq p_c$ with $L_e \geq 1$ and $L_c \geq 1$, then the coding error probability p_e is bounded away from zero as presented by the following extended Fano's inequality, which is a general case of the original one in the standard information theoryⁱⁱ

$$p_e \geq \frac{1}{\log(L_c L_e)} (H(p(\mathbf{X})) - C - 1) \quad (3.1.36)$$

Proof. Uncertainty in the estimate of channel input data can be broken into two parts: the uncertainty in the channel whether or not an error probability p_e was made, and given that an error is made, the uncertainty in the true value, i.e. $p_e \log(L - L_e) = p_e \log(L_c)$ [94]. The error occurs with probability p_e such that the first uncertainty is the error entropy

$$H(p_e) = -(1 - p_e) \log(1 - p_e) - p_e \log p_e \quad (3.1.37)$$

The second uncertainty occurred with probability p_e can be no larger than $p_e \log(L_c) \leq p_e \log(L_c L_e)$. Therefore, if the total uncertainty can be interpreted as the information lost, we should have the inequality

$$H(p(\mathbf{X}), p(\mathbf{X}|\mathbf{W})) \leq H(p_e) + p_e \log(L_c L_e) \quad (3.1.38)$$

To show (3.1.38) is true, using the definition of the entropy we have

$$\begin{aligned} H(p(\mathbf{X}), p(\mathbf{X}|\mathbf{W})) &= - \sum_{i=1}^{L_e} \sum_{k=1}^K p(x_i, w_k) \log p(x_i|w_k) \\ &\quad - \sum_{l=1}^{L_c} \sum_{k=1}^K p(x_l, w_k) \log p(x_l|w_k). \end{aligned} \quad (3.1.39)$$

ⁱⁱIn the standard discrete-time communication channel problem [94], a single pattern mismatch between the channel input and output is discussed, i.e. $L_e = 1$ and $L_c = L - L_e = L - 1$, such that the Fano's inequality is $p_e \geq \frac{1}{\log(L-1)} (H(p(\mathbf{X})) - C - 1)$.

Subtracting $H(p_e) + p_e \log(L_e L_c)$ from the left side of (3.1.39), we get

$$\begin{aligned}
& H(p(\mathbf{X}), p(\mathbf{X}|\mathbf{W})) - (H(p_e) + p_e \log(L_e L_c)) \\
&= H(p(\mathbf{X}), p(\mathbf{X}|\mathbf{W})) + p_c \log(p_c) + p_e \log p_e - p_e \log(L_e) - p_e \log(L_c) \\
&\leq H(p(\mathbf{X}), p(\mathbf{X}|\mathbf{W})) + p_c \log(p_c) + p_e \log p_e - p_e \log(L_e) - p_c \log(L_c) \\
&= \sum_{i=1}^{L_e} \sum_{k=1}^K p(x_i, w_k) \log \frac{p_e}{(L_e)p(x_i|w_k)} + \sum_{i=1}^{L_c} \sum_{k=1}^K p(x_i, w_k) \log \frac{p_c}{(L_c)p(x_i, w_k)} \\
&\leq \sum_{i=1}^{L_e} \sum_{k=1}^K p(x_i, w_k) \left[\frac{p_e}{(L_e)p(x_i|w_k)} - 1 \right] + \sum_{i=1}^{L_c} \sum_{k=1}^K p(x_i, w_k) \left[\frac{p_c}{(L_c)p(x_i|w_k)} - 1 \right] \\
&= \sum_{i=1}^{L_e} \sum_{k=1}^K p(w_k) \left[\frac{p_e}{L_e} \right] + \sum_{i=1}^{L_c} \sum_{k=1}^K p(w_k) \left[\frac{p_c}{L_c} \right] - 1 \\
&= p_e + p_c - 1 = 0
\end{aligned} \tag{3.1.40}$$

where the first inequality above is based on the assumption of $p_e \geq p_c$, i.e. the most unreliable (noisy) data points are identified by the decoding scheme, while the second inequality is based on $\log(x) \leq (x - 1)$, and $p(w_k) = p(x_i, w_k)/p(x_i|w_k)$ is the marginal distribution.

Considering that the maximum of MI which is a concave function of the probability set $p(\mathbf{X})$ [94], we have

$$H(p(\mathbf{X}), p(\mathbf{X}|\mathbf{W})) = H(p(\mathbf{X})) - I(p(\mathbf{X}), p(\mathbf{X}|\mathbf{W})) \geq H(p(\mathbf{X})) - C \tag{3.1.41}$$

Then according to (3.1.41) and (3.1.38), the extended Fano's inequality for robust data clustering [94] can be represented as

$$H(p(\mathbf{X})) - C \leq H(p_e) + p_e \log(L_e L_c) \leq 1 + p_e \log(L_e L_c) \tag{3.1.42}$$

This completes the proof of the theorem. $\square$

This inverse theorem gives an important indication that any input information transmitted beyond the capacity C will generate unreliable data transmission. With an efficient numerical algorithm, we are able to find those input data probabilities that achieve the capacity and identify those unreliable input data at the same time.

This can be done by maximization of the mutual information between input data probability distributions $\{p(x_i)\}$ and the estimated optimal association probabilities $\{\bar{p}(w_k|x_i)\}$. Similar to the minimization of the rate distortion function in Eq. (3.1.23), constrained capacity maximization can be expressed as:

$$C(\mathcal{D}) = \max_{p(x)} \left[\sum_{i=1}^N \sum_{k=1}^K p(x_i) \bar{p}(w_k|x_i) \log \frac{\bar{p}(x_i|w_k)}{p(x_i)} - s \left(\sum_{i=1}^N \sum_{k=1}^K p(x_i) \bar{p}(w_k|x_i) (x_i - w_k)^2 - \mathcal{D} \right) \right] \quad (3.1.43)$$

where $\bar{p}(x_i|w_k) = \frac{p(x_i) \bar{p}(w_k|x_i)}{\sum_{i=1}^N p(x_i) \bar{p}(w_k|x_i)}$.

To obtain the maximization solutions, we use a new Lagrange multiplier s_1 to include the constraint $\sum_{i=1}^N p(x_i) = 1$, then

$$\begin{aligned} \frac{\partial}{\partial p(x_i)} \left\{ \sum_{i=1}^N \sum_{k=1}^K p(x_i) \bar{p}(w_k|x_i) \log \frac{\bar{p}(x_i|w_k)}{p(x_i)} - s \left(\sum_{i=1}^N \sum_{k=1}^K p(x_i) \bar{p}(w_k|x_i) (x_i - w_k)^2 - \mathcal{D} \right) + s_1 \left(\sum_{i=1}^N p(x_i) - 1 \right) \right\} &= 0 \\ -\log p(x_i) - 1 + \sum_{k=1}^K \bar{p}(w_k|x_i) \left(\log \bar{p}(x_i|w_k) - s \bar{p}(w_k|x_i) (x_i - w_k)^2 \right) + s_1 &= 0 \\ \log p(x_i) = s_1 - 1 + \sum_{k=1}^K \bar{p}(w_k|x_i) \left(\log \bar{p}(x_i|w_k) - s \bar{p}(w_k|x_i) (x_i - w_k)^2 \right) \\ p(x_i) = \frac{\exp \sum_{k=1}^K \bar{p}(w_k|x_i) \left(\log \bar{p}(x_i|w_k) - s \bar{p}(w_k|x_i) (x_i - w_k)^2 \right)}{\exp(1 - s_1)} \end{aligned} \quad (3.1.44)$$

With $\sum_{i=1}^N p(x_i) = 1$, we further have

$$\exp(1 - s_1) = \sum_{i=1}^N \exp \sum_{k=1}^K \bar{p}(w_k|x_i) \left(\log \bar{p}(x_i|w_k) - s \bar{p}(w_k|x_i) (x_i - w_k)^2 \right) \quad (3.1.45)$$

Hence

$$p(x_i) = \frac{\exp \sum_{k=1}^K \bar{p}(w_k|x_i) \left(\log \bar{p}(x_i|w_k) - s \bar{p}(w_k|x_i) (x_i - w_k)^2 \right)}{\sum_{i=1}^N \exp \sum_{k=1}^K \bar{p}(w_k|x_i) \left(\log \bar{p}(x_i|w_k) - s \bar{p}(w_k|x_i) (x_i - w_k)^2 \right)} \quad (3.1.46)$$

Note that the inequality constraint $p(x_i) \geq 0$ will be satisfied as we can see from the solution for $p(x_i)$ is always positive.

Figure 3.2: Recursive algorithm to maximize the mutual information.

Replacing $\bar{p}(x_i|w_k)$ with $\frac{p(x_i)\bar{p}(w_k|x_i)}{\sum_{i=1}^N p(x_i)\bar{p}(w_k|x_i)}$ in the maximization solution for $p(x_i)$ yields

$$p(x_i) = \frac{p(x_i) \exp \sum_{k=1}^K \bar{p}(w_k|x_i) \left[\log \frac{\bar{p}(w_k|x_i)}{\sum_{i=1}^N p(x_i)\bar{p}(w_k|x_i)} - s \bar{p}(w_k|x_i) (x_i - w_k)^2 \right]}{\sum_{i=1}^N p(x_i) \exp \sum_{k=1}^K \bar{p}(w_k|x_i) \left[\frac{\log \bar{p}(w_k|x_i)}{\sum_{i=1}^N p(x_i)\bar{p}(w_k|x_i)} - s \bar{p}(w_k|x_i) (x_i - w_k)^2 \right]} \quad (3.1.47)$$

In [96], the non-negativity of the average mutual information had been given. However, the individual items in the sum of the maximization of mutual information can be negative (i.e. for some i and k , $\log \frac{p(w_k|x_i)}{\sum_{i=1}^N p(x_i)p(w_k|x_i)} < 1$). This implies that the probability of the k th output (cluster center) is decreased by the observation of i th input data $p(x_i)$. This particular input data may be viewed as an unreliable

input data with less preference (outlier), and the negative effect incurred by it must be offsetted by others since the mutual information is not negative. Intuitively, if $p(x_i|w_k) = 0$, then $p(x_i)$ should be set equal to zero in order to obtain the maximum of the capacity, so that these unreliable input data points can be identified and deleted/dropped from the whole input ensemble. Therefore, the maximization of the capacity function provides a robust density estimation mechanism to identify the unreliable input data (outliers) in the sense that the average information is maximized over all output cluster centers and input data.

We use similar recursive algorithm to calculate the $p(x_i)$ as in [55]. The recursive algorithm is shown in Fig. 3.2 and the detailed proof can be found in [94].

3.2 Adaptive Spatial Information Clustering Algorithm

With these necessary preliminaries introduced in the last section, we are now in the position to introduce our proposed robust data clustering algorithm for image segmentation.

Usually, there are two types of misclassified pixels, called outliers, in image segmentation: (1) noise pixels caused by intensity shift, for example, salt and pepper noise. (2) border pixels, in other words, the data points with equal similarity distances to their adjacent cluster centers. The existence of outliers may shift the true cluster center into wrong positions or may cause difficulty when assigning the data label using correct cluster numbers [60]. We propose the ASIC algorithm with a two-step solution, i.e. the MI minimization and MI maximization to eliminate the two kinds of outliers, respectively.

To deal with the first type of outliers and to cope with the problems of the conventional spatial algorithms as presented in the last section, a new objective function with the introduction of spatial information into the data clustering based image segmentation with full adaptive capability is proposed in this section. This new

Figure 3.3: 8-neighborhood system in a 3×3 window.

objective function contains a weighting parameter that can adaptively control the neighboring effect in terms of the homogeneity of the 3×3 neighbor window (Fig. 3.3), while preserving the fine image structures such as borders or edges for the low level outlier elimination with MI minimization for this specific objective function.

For the second type of outliers, conventional methods classify these data points according to the specified similarity distance (including the spatial information). In this case, a small numerical error would cause misclassification of these data points. For the worst case scenario in which a data point has equal similarity distances to its nearest cluster centers, it is very difficult for the conventional algorithm, even with spatial information, to decide which cluster this data point should belong to (as shown in Fig. 2.4). The ASIC algorithm has the capability to further eliminate this kind of outliers and provide improved image segmentation performance as compared to the conventional algorithms in the second step. This is called the high level outlier elimination based on MI maximization in which an adaptive window is proposed.

3.2.1 Improved similarity measure and MI minimization for basic clustering and low level outlier elimination

In order to introduce the spatial information into the data clustering algorithm in a more robust and adaptive way, we propose a new similarity measure d_s and express the distortion measure $\mathcal{D}_s$ as follow:

$$\mathcal{D}_s = \sum_{i=1}^M \sum_{j=1}^N \sum_{k=1}^K p(x_{i,j}) p(w_k | x_{i,j}) d_s(x_{i,j}, w_k) \quad (3.2.48)$$

where

$$d_s(x_{i,j}, w_k) = \lambda_{i,j} \|x_{i,j} - w_k\|^2 + \frac{(1 - \lambda_{i,j})}{N_R} \sum_{x_{r,c} \in N_{i,j}} \|x_{r,c} - w_k\|^2 \quad (3.2.49)$$

with the image patterns $x_i \in \mathbf{X}$ converted into $M \times N$ matrix, i.e. $x_{i,j} \in \mathbb{I}_{M \times N}$. The location of each pixel is presented by (i, j) , where $i \in [1, M]$ and $j \in [1, N]$. The parameter $\lambda_{i,j}$ is a spatial factor which can be computed locally in advance. The $N_{i,j}$ is the subset of neighborhood pixels of (i, j) in a 3×3 window. N_R is the window size, and $p(x_{i,j})$ is a fixed arbitrary value in the MI minimization step. It is usually set to represent uniform distribution, i.e. $p(x_{i,j}) = \frac{1}{M \times N}$, as we have no information about the prior distribution of the input data points. $p(w_k | x_{i,j})$ is the conditional probability of assigning observation $x_{i,j}$ to center w_k . $\|x_{i,j} - w_k\|^2$ is usually represented by Euclidean distance. The subscripts have been extended to the 2D case, because we investigate the performance of our algorithm on 2D grey scale images. For 3D image volume or RGB color images, our proposed method can be applied in a straightforward manner.

As can be seen from (3.2.49), the proposed similarity measure has two components. One puts the constraint that the pixels are to be close to the cluster centers. The other imposes spatial continuity. These two terms are weighted by $\lambda_{i,j}$. With this new objective function, we can choose $\lambda_{i,j}$ to be small so that the pixel $x_{i,j}$ is greatly influenced by its neighbors if it is in a homogeneous region. On the other hand, a big $\lambda_{i,j}$ will be used to reduce the neighboring effect, if $x_{i,j}$ is along the borders

or edges. Thus, the over-smoothed edges problem can be solved by this adaptive weighting scheme.

In order to make the smoothing parameter $\{\lambda_{i,j}\}$ adaptive to the image content, the value of $\{\lambda_{i,j}\}$ should be adjusted according to the homogeneity of the neighboring window. Consider a 3×3 window, for the pixel in location (i, j) , we can calculate the dispersion of intensity differences between its intensity value and its neighbors':

$$d(x_{i,j}, x_{i+r,j+c}) = ||(x_{i,j} - x_{i+r,j+c})||^2 \quad (3.2.50)$$

with $r = \{-1, 0, 1\}$, and $c = \{-1, 0, 1\}$, but $(r, c) \neq (0, 0)$. Thus, the homogeneousness of this 3×3 neighboring window can be measured by the standard derivation of the intensity difference between the center pixel and its neighboring pixels, leading to

$$\begin{aligned} \eta_{i,j} &= \left(\frac{1}{N_R} \sum_{r=-1}^1 \sum_{c=-1}^1 (d(x_{i,j}, x_{i+r,j+c}) - \mu)^2 \right)^{\frac{1}{2}} \\ \mu &= \frac{1}{N_R} \left(\sum_{r=-1}^1 \sum_{c=-1}^1 d(x_{i,j}, x_{i+r,j+c}) \right) \end{aligned} \quad (3.2.51)$$

It is apparent from (3.2.51) that the standard derivation of the intensity difference $\{d(x_{i,j}, x_{i+r,j+c})\}$ does not take into account the magnitude of intensity transition in the edge region, and hence will over smooth obscure edge regions. Therefore, a further modification is required to eliminate the unbalanced effect on the weighting parameters between the obvious and obscure edge regions.

Since (3.2.51) measures the local variance of the intensity difference between the center pixel and its neighbors, a straightforward solution is to divide $\eta_{i,j}$ by the local variance of intensity of all the pixels over the 3×3 window, i.e.

$$\hat{\lambda}_{i,j} = \eta_{i,j} / \zeta_{i,j} \quad (3.2.52)$$

where

$$\begin{aligned} \zeta_{i,j} &= \left(\frac{1}{9} \sum_{r=-1}^1 \sum_{c=-1}^1 (x_{i+r,j+c} - \tau)^2 \right)^{\frac{1}{2}} \\ \tau &= \frac{1}{9} \left(\sum_{r=-1}^1 \sum_{c=-1}^1 x_{i+r,j+c} \right) \end{aligned}$$

Figure 3.4: The map of weighting parameters. (a) Original simulated image; (b) without the consideration of pixels intensity variation; (c) with the consideration of pixels intensity variation

Finally, we normalize $\hat{\lambda}_{i,j}$ and obtain $\lambda_{i,j}$

$$\lambda_{i,j} = \hat{\lambda}_{i,j} / \max(\hat{\lambda}_{i,j}) \quad (3.2.53)$$

As can be seen from Figs. 3.4 (b)-(c), we have effectively eliminated the dominance of obvious edges by the division of the local variation of all pixel intensity in the 3×3 window.

Obviously, this approach is totally adaptive to the local image content, and can be preprocessed before the clustering operation. There is no experimentally adjusted parameters in the whole process. As discussed before, there are two main

characteristics in our method:

- (a) When the center pixel is in a homogeneous region, the spatial parameter $\lambda_{i,j}$ should be chosen as small as possible so that the clustering result of center pixel is greatly influenced by its neighbors. It is especially useful when the center pixel is contaminated by the salt and pepper noise. On the other hand, if the intensity value of center pixel is similar to its neighbors', this method still yields the correct result because there is no much difference between the center pixel and its neighbors in terms of similarity measurement. Yet another advantage of this method is that it favors a smoother segmentation result against the homogeneous regions as compared to other counterparts.
- (b) When the center pixel is in an inhomogeneous region, like edge or border, the spatial parameter $\lambda_{i,j}$ should be chosen as high as possible so that the clustering result depends mainly on its own intensity information. By offsetting the spatial information within the neighborhood window, the center pixel is hardly influenced by its neighbors and the fine details of the image could be preserved. The level of smoothness is automatically adjusted according to the inhomogeneity within the local window. This adaptive method is suitable for those images corrupted by Gaussian noise, which tends to smooth the image details.

Fig. 3.5 shows four examples of factor $\{\lambda_{i,j}\}$ under different window features. Example (i) shows the case when the center pixel is located along the image edge, and example (ii) is for a less inhomogeneous region. As can be seen from Fig. 3.5, both $\lambda_{i,j}$ s are big in order to preserve the fine details of the image. On the other hand, examples (iii) and (iv) show the cases that the center pixel is in the homogeneous region. It is clearly shown in Fig. 3.5 that $\lambda_{i,j}$ s are very small for such situations.

According to the information-theoretic clustering framework discussed in Section 3.1.2, any data compression scheme must preserve the average amount of information of the input data such that we can reproduce the input data from the compressed data with an average distortion that is less than or equal to some specified distortion

$\lambda_{i,j} = 0.9583$	$\lambda_{i,j} = 0.3258$	$\lambda_{i,j} = 0.029$	$\lambda_{i,j} = 0.0246$
234	243	235	125
239	119	255	116
239	125	238	122
33	230	229	120
255	117	243	121
243	116	235	119
31	242	235	115
31	28	251	113
244	31	245	113
(i)	(ii)	(iii)	(iv)

Figure 3.5: The adaptive spatial parameter $\{\lambda_{i,j}\}$ under homogeneous regions and inhomogeneous regions.

$\mathcal{D}_s$. With this in mind, the objective function of the MI minimization of the ASIC algorithm can be defined as

$$R(\mathcal{D}_s) = \min_{\{p(w_k|x_{i,j})\} \in Q_{\mathcal{D}_s}} I(\mathbf{X}; \mathbf{W}) \quad (3.2.54)$$

where

$$\begin{aligned} Q_{\mathcal{D}_s} = \{ & p(w_k|x_{i,j}) \in R^K \times R^{M \times N} : \\ & \sum_{i=1}^M \sum_{j=1}^N \sum_{k=1}^K p(x_{i,j}) p(w_k|x_{i,j}) d_s(x_{i,j}, w_k) \leq \mathcal{D}_s; \\ & \sum_k p(w_k|x_{i,j}) = 1, p(w_k|x_{i,j}) \geq 0 \} \end{aligned} \quad (3.2.55)$$

$I(\mathbf{X}; \mathbf{W})$ is the mutual information (MI) between $\{p(x_{i,j})\}$ and $\{p(w_k|x_{i,j})\}$, which can be expressed as

$$I(\mathbf{X}; \mathbf{W}) = \sum_{i=1}^M \sum_{j=1}^N \sum_{k=1}^K p(x_{i,j}) p(w_k|x_{i,j}) \log \frac{p(w_k|x_{i,j})}{p(w_k)} \quad (3.2.56)$$

Similar to the formulation in Section 3.1.2, it is easy to see that the optimization problem can be reformulated as

$$R = \min_{\{p(w|x)\}} I(\mathbf{X}; \mathbf{W}) + s \left(\mathcal{D}_s - \sum_{i=1}^M \sum_{j=1}^N \sum_{k=1}^K p(x_{i,j}) p(w_k|x_{i,j}) d_s(x_{i,j}, w_k) \right) \quad (3.2.57)$$

with

$$\mathcal{D}_s = \sum_{i=1}^M \sum_{j=1}^N \sum_{k=1}^K p(x_{i,j}) \bar{p}(w_k|x_{i,j}) d_s(x_{i,j}, w_k) \quad (3.2.58)$$

where $s = 1/T$ is the Lagrange multiplier, $\bar{p}(w_k|x_{i,j}) \in \bar{P}(\mathbf{W}|\mathbf{X})$ achieves a minimum point of $R(\mathcal{D}_s)$ at a specific T [94].

With the new similarity measure, minimization of R in (3.2.57) with respect to $\{p(w_k|x_{i,j}), w_k\}$ would lead to a new iterative scheme:

$$p(w_k|x_{i,j}) = \frac{p(w_k)e^{d_s(x_{i,j},w_k)/T}}{\sum_{k=1}^K p(w_k)e^{d_s(x_{i,j},w_k)/T}} \quad (3.2.59)$$

$$w_k = \frac{1}{p(w_k)} \sum_{i=1}^M \sum_{j=1}^N p(x_{i,j})p(w_k|x_{i,j}) \left(\lambda_{i,j}x_{i,j} + \frac{(1-\lambda_{i,j})}{N_R} \sum_{x_{r,c} \in N_{i,j}} x_{r,c} \right) \quad (3.2.60)$$

By iteratively minimizing (3.2.57) via (3.2.59) and (3.2.60), the minimization step of the ASIC algorithm can explore the details of the input data and group them effectively into several clusters subject to a specified level of randomness, which can be measured by the minimization of the mutual information [50].

3.2.2 MI maximization and adaptive window for high level outlier elimination

In the last subsection, by minimizing the MI, we have obtained the sub-optimal membership values without considering the high level outlier, i.e. $\{\bar{p}(w_k|x_{i,j})\}$, when the probability distribution of input data $\{p(x_{i,j})\}$ is fixed. We now want to reassess the reliability of the input data under the obtained sub-optimal membership values $\{\bar{p}(w_k|x_{i,j})\}$. Some of the input data are more desired than the others when we know the sub-optimal membership values. For example, if the input data is closer to its adjacent cluster center than the rest, this data point is more likely to belong to that cluster, i.e. high reliability.

Remember that the MI maximization method in information theory provides us a way to evaluate the reliability of input data. It can be formulated by maximizing

MI against the input probability distribution $p(x_{i,j})$:

$$\begin{aligned} C &= \max_{\{p(x_{i,j})\}} I(\mathbf{X}; \mathbf{W}) \\ &= \max_{\{p(x_{i,j})\}} \sum_{i=1}^M \sum_{j=1}^N \sum_{k=1}^K p(x_{i,j}) \bar{p}(w_k|x_{i,j}) \log \frac{p(x_{i,j}|w_k)}{p(x_{i,j})} \end{aligned} \quad (3.2.61)$$

where C represents the channel (cluster) capacity.

According to the information theory [94], [60], for fixed $\bar{p}(w_k|x_{i,j})$, the MI in (3.2.61) is maximized by

$$p(x_{i,j}) = \frac{\exp \left(\sum_{k=1}^K \bar{p}(w_k|x_{i,j}) \log p(x_{i,j}|w_k) \right)}{\sum_{i=1}^M \sum_{j=1}^N \exp \left(\sum_{k=1}^K \bar{p}(w_k|x_{i,j}) \log p(x_{i,j}|w_k) \right)}$$

where

$$p(x_{i,j}|w_k) = \frac{p(x_{i,j}) \bar{p}(w_k|x_{i,j})}{\sum_{i=1}^M \sum_{j=1}^N p(x_{i,j}) \bar{p}(w_k|x_{i,j})} \quad (3.2.62)$$

For any value of (i, j) , if $p(x_{i,j})$ in (3.2.62) has a zero value, such that a corresponding input data point $x_{i,j}$ will have no contribution to the MI maximization, and should be removed from the sum in the MI maximization procedure and treated as outlier (unreliable data point). Song [60] gave an intuitive discussion on the reason why such a data point with $p(x_{i,j}) = 0$ should be considered as an outlier without a detailed theoretical proof. Here, we further develop this into a complete theorem, called the information-theoretic approach for high level outlier identification as presented in the appendix, to show that the MI maximization is indeed a powerful tool to identify outliers as the highly unreliable data point (pixel). This forms the basis of our ASIC algorithm for high level outlier elimination. Note also that these unreliable data points are not to be deleted, they are labeled as outliers and are going to be clustered again with a bigger neighboring window (please see the implementation section 3.3 for details).

As discussed earlier, the second type of outliers are those pixels with equal or almost equal similarity distances to their adjacent cluster centers (refer to Fig. 2.4).

Although we are able to improve the segmentation result by the low level outlier elimination with a 3×3 window, the second type of outlier problem may still exists in the new equivalent dissimilarity distance measure, as shown in Fig. 3.6. In order to further improve the segmentation result, we need to identify the second type of outliers under the new similarity distance measurement and use an adaptive window to label them more effectively, This procedure can be considered as high level outlier elimination.

With the low level outlier elimination, most of the misclassified errors have been corrected by incorporating the spatial information. The outliers identified in the MI maximization step form a small portion of pixels whose clustering results are highly uncertain. Since the MI maximization is based on the inherent spatial information of a 3×3 neighborhood window from $\bar{p}(w_k|x_{i,j})$, an adaptive neighborhood window may be needed for resegmentation of high level noisy pixel with more spatial information.

In this research work, we introduce a simple and fast reassignment method to label the ambiguous pixel for high level outlier elimination. This method is based on the concept that the center pixel should be assigned to the cluster that most of its neighbors belong. This has been widely used in a number of existing spatial clustering algorithms [31, 73, 79, 93]. The only difference is that we do it within an adaptive window for the reassignment of outlier based on MI maximization. The implementation of high level outlier elimination and reassignment is given as follows:

1. Consider an initial 3×3 window centered at the outlier, obtain the clustering labels for all neighboring pixels within this neighborhood window from the minimization step.
2. Obtain all label counts for each cluster, and store the count values into $\xi_k, k = 1, 2, \dots, K$.
3. Find the maximum value of ξ_k and its corresponding cluster index m , i.e. $\xi_m = \max(\xi_k)$.
4. If $\xi_1 = \xi_2 = \dots = \xi_K$, increase the current window edge by 2 more pixels per

Figure 3.6: Example for high level outlier elimination. (a) Outlier existence in equivalent similarity measurement. (b) Relabeling the outlier pixels with an adaptive window.

iteration, (for example, from 3×3 to 5×5 , or even larger if there are still equal values), go to 1); otherwise, update the outlier label with m .

5. Iterate for all outlier pixels until there is no further change, stop.

3.2.3 Remarks on the ASIC algorithm

- The ASIC algorithm is a two-step image segmentation algorithm. In the MI minimization step, the input image intensity data points with the specific spatial measure are to be partitioned into groups in order to reduce the data distortion. For the input data points, there are no prior preferences, i.e. $p(x_{i,j}) = 1/l$ is fixed. After we have obtained the optimal association probabilities $\{\bar{p}(w_k|x_{i,j})\}$, it is good to reevaluate the preferences of input data points under the current clustering results. The second step can be achieved by MI maximization. The new $\{p(x_{i,j})\}$ are used to identify the outliers.
- The adaptive spatial similarity measure is rather general. It can be easily applied to other clustering algorithms to derive other algorithms. With the new definition of similarity measure, the adaptive spatial factors $\{\lambda_{i,j}\}$ can play a key role in controlling the trade-off between noise removal and detail preservation.
- The selection framework for $\{\lambda_{i,j}\}$ is adaptive and localized. The value of $\lambda_{i,j}$ is not fixed for all neighborhood windows over the image. There is no prior knowledge of the noise and all the information to compute $\{\lambda_{i,j}\}$ are from the local gray level and spatial information. It can reflect the homogeneity degree of the neighborhood windows, and automatically control the spatial constraints. Another advantage of using $\{\lambda_{i,j}\}$ is that it can be computed before the segmentation process, and so the computational complexity is reduced.
- In the MI maximization step, the remaining outliers, i.e. the unreliable pixel assignments are identified from an information-theoretic view point and reas-

signed based on an adaptive window. This feature provides ASIC with the capabilities to eliminate outliers in a relatively noisy environment.

3.3 The ASIC Algorithm

The complete ASIC algorithm is described below. Supposed that we are given the feature vectors of input image $X = \{x_{1,1}, x_{1,2}, \dots, x_{M,N}\}$ with the resolution of $L = M \times N$.

MI Minimization for basic clustering and low level outlier elimination:

- 1). Set the maximum number of clusters K_{max} , and minimum temperature T_{min} . Then $p(x_{i,j}) = 1/(M \times N)$ for $i = 1$ to M , $j = 1$ to N .
- 2). Initialization: $T > 2\lambda_{max}(Cov_x)$ where $\lambda_{max}(Cov_x)$ is the largest eigenvalue of the covariance matrix Cov_x of the input feature vectors X [30], initialize cluster number $K = 1$, and $p(w_k) = 1$. Calculate the weighting function $\lambda_{i,j}$ which occurs in formulation (3.2.59).
- 3). Perform fixed point iteration according to (3.2.59) and (3.2.60).
- 4). Convergence Test: If not satisfied, then go to 3).
- 5). If $T < T_{min}$, perform the last iteration and stop.
- 6). Cooling step: $T \leftarrow \alpha T, (\alpha < 1)$.
- 7). If $K < K_{max}$, check condition for phase transition. If a critical temperature $T = 2\lambda_{max}(Cov_x)$ is reached for the clustering, add a new center $w_{K+1} = w_K + \delta$, with $p(w_{K+1}) = p(w_K)/2$, $p(w_K) \leftarrow p(w_K)/2$, and update $K+1 \leftarrow K$. Repeat 3)-6) with new cluster number. If $K = K_{max}$, go to step 8).

MI Maximization for high level outlier elimination: 8). Select $p(x_{i,j}) = 1/L$, for $i = 1, \dots, M, j = 1, \dots, N$, and $\varepsilon > 0$, then start the fixed point iteration of the robust density estimation according to steps 9) to 10).

9). Iteratively update the prior distribution by using

$$c_{i,j} = \exp\left[\sum_{k=1}^K \bar{p}(w_k|x_{i,j}) \ln \frac{\bar{p}(w_k|x_{i,j})}{\sum_{i=1}^M \sum_{j=1}^N p(x_{i,j}) \bar{p}(w_k|x_{i,j})}\right]$$

and

$$p(x_{i,j}) = p(x_{i,j}) \frac{c_{i,j}}{\sum_{i=1}^M \sum_{j=1}^N c_{i,j}}$$

10). If

$$\ln \sum_{i=1}^M \sum_{j=1}^N p(x_{i,j}) c_{i,j} - \ln \max_{\{i,j\}} c_{i,j} < \varepsilon$$

then go to 9) where $\varepsilon > 0$.

11). Label unreliable data points, set $T \rightarrow 0$ for the titled distribution to obtain the membership of all data points for a hard clustering solution.

12). Create a 3×3 window centered at the outlier, obtain the clustering labels for all neighboring pixels within this neighborhood window from the minimization step.

13). Store all label counts for each cluster into $\xi_k, k = 1, 2, \dots, K$.

14). Find the maximum value of ξ_k and its corresponding cluster index, i.e. $\xi_m = \max(\xi_k)$.

15). If $\xi_1 = \xi_2 = \dots = \xi_K$, increase the current window edge by 2 more pixels, (for example, from 3×3 to 5×5), go to 12); Otherwise, update the outlier label with m .

16). Iterate for all outlier pixels until there is no further change, then stop.

The weighting factors $\{\lambda_{i,j}\}$ do not change in each fixed point iteration, so we can pre-compute them at step 2). After this, the algorithm begins the fixed point iteration. We need to do the convergence test before cooling the temperature $T \leftarrow \alpha T$. The fixed point iteration terminates on the following condition:

$$\Delta_{change}^{n+1} = \sum_{k=1}^K (w_k^{n+1} - \mathbf{w}_k^n)^T (w_k^{n+1} - w_k^n) < \epsilon$$

When the change in the $(n + 1)th$ iteration converges to a user predefined threshold value, let's say $\epsilon = 0.01$, the algorithm enters into the cooling step.

3.4 Experimental Results

In this section, we will examine the performance of our algorithm through a variety of simulated images and real images. We will start by illustrating the need of incorporating the spatial information with data clustering based image segmentation approaches. After this, we will use quantified and visualized comparisons between our proposed method and the other four algorithms of deterministic annealing (DA), bias-corrected fuzzy c-means (BCFCM), enhanced fuzzy c-means (EnFCM), and Fast Generalized fuzzy c-means (FGFCM) with the simulated image series. Furthermore, we will also demonstrate how the proposed adaptive similarity measure is able to outperform other spatial data clustering variants with a two-step outlier elimination process and reduce the edge blurring effects. Finally, the effectiveness of the proposed algorithm is demonstrated through experiments on real image examples.

In all examples, the cooling factor $\alpha = 0.95$ was used unless otherwise stated. The fuzziness control parameter q has been set to 2, and $N_R = 8$ (i.e. a 3×3 neighbor window is centered around each pixel). The spatial parameter α in BCFCM algorithm is obtained by searching the interval $[0.5, 0.9]$ with a step size of 0.05 in terms of the optimal SA value. Similarly, the α in EnFCM is obtained by searching the interval $[0.2, 8]$ as suggested in [74]. The λ_g in FGFCM is selected from 0.5 to 6 with an increment of 0.5.

3.4.1 The need of incorporating the spatial information with data clustering approaches

As mentioned earlier, data clustering based image segmentation methods resemble the thresholding approaches but they offer a more effective way of finding the optimal threshold values for image segmentation. However, similar to the thresh-

olding approaches, these standard image segmentation methods discard the image geometric information and rely only on the intensity distribution. Therefore, the segmentation results are always contaminated by noise and other image artifacts. For example, in medical image segmentation, due to the noise and intensity inhomogeneities introduced in the imaging process, different tissues at different locations may have similar intensity appearance, while the same tissue at different locations may have a different intensity appearance. Thus, the segmentation results would be totally wrong without the incorporation of spatial information.

Suppose we now have an ideal image with histogram as shown in Fig. 3.7 (a). This image contains two Gaussian distribution like classes, with the same variance but different mean. We set the mean and variance of each class so that they overlap each other. The overlap regions are labeled as $e1$ and $e2$. Clearly, if we segment the image with conventional data clustering algorithms and the optimal threshold is displayed by the gray line, class 1 in the segmented image will have a histogram as show in Fig. 3.8. The pixels in the overlapped region $e1$ which belonged to class 2 would have been clustered as class 1 as well, because they are closer to the class center of class 1. This process will probably be the cause of some unavoidable misclustered pixels in the final image segmentation results.

A straightforward solution to this problem is to modify the conventional similarity measure so that the data clustering process will not only rely on the individual pixel intensity itself, but also on the neighboring pixels' intensity values. Recall our proposed similarity measure in Eq. (3.2.49)

$$d_s(x_{i,j}, w_k) = \lambda_{i,j} \|x_{i,j} - w_k\|^2 + \frac{(1 - \lambda_{i,j})}{N_R} \sum_{x_{r,c} \in N_{i,j}} \|x_{r,c} - w_k\|^2$$

we can see that, by incorporating the spatial information into the similarity measure function, the new distance measure of pixel in position (i, j) is now influenced by the intensity values of its neighboring pixels. When the center pixel (i, j) is in a homogeneous region, the new similarity measure should be greatly dominated by its neighbors. The contrary holds for nonhomogeneous regions. With this modification,

Figure 3.7: An image histogram with two Gaussian distribution like classes.

Figure 3.8: The histogram of class 1 in segmented image.

Figure 3.9: Comparison between ASIC algorithm and DA algorithm. (a) Original image. (b) adaptive weighting functions map. (c) ASIC algorithm. (d) DA algorithm

those contaminated pixels mainly caused by intensity shift will be dragged back to their 'corrected' cluster centers because the criterion of clustering a data point will now depend on the homogeneity of the local neighboring window it belongs to. The image segmentation can now be casted in a more reasonable framework.

Let us now see how our proposed ASIC algorithm solves the problem mentioned above. Suppose we have a synthetic test image as shown in Fig. 3.9 (a). This image contains four-class patterns corrupted by Gaussian noise and has been created with means of (30, 120, 180, 240) (grey level) and variance of 15 (grey level). Compared to the segmentation result by the DA algorithm in Fig. 3.9 (d), the reduction of misclusterd pixels with our ASIC algorithm is apparent.

Fig. 3.10 shows the histograms of the segmentation results with DA and ASIC

Figure 3.10: Histograms of the segmentation results of Fig. 3.9. (a) Using DA and ASIC algorithm. (a)-(d) DA algorithm. (e)-(h) ASIC algorithm.

algorithms for Fig. 3.9 (a). As pointed out earlier, the DA algorithm performs like thresholding as can be seen from the histograms in Figs. 3.10 (a)-(d). The histograms look like being cut into 4 parts. Clearly, the number of misclassified pixels is high for the DA algorithm. With our ASIC algorithm, the histograms have longer 'tails'. Intuitively, it seems that the ASIC algorithm has dragged some pixels from other clusters as compared to the DA algorithm, to make the shapes of histogram more similar to the Gaussian distribution. Note that we generate this image with white Gaussian noise, hence the clustering results nicely represented the 'true' distribution. In image segmentation, wider tails in the histogram tend to yield greater details in the segmentation results. In other words, with the information of the pixel location, the information lost through intensity shift has been recovered, and the accuracy of the clustering has been increased as well.

3.4.2 Quantified and visualized comparisons between ASIC and DA, BCFCM, EnFCM, and FGFCM with synthetic images

We use another synthetic test image as shown in Fig. 3.4(a), and the resolution is 256×256 pixels and the grey-scale range is 0 to 255.

To validate the segmentation results, we use the segmentation accuracy (SA) which is computed by

$$SA = \frac{\text{number of misclassified pixels}}{\text{total number of pixels}} \quad (3.4.63)$$

Segmentation results on a synthetic image corrupted by mixed noise

Fig. 3.11 (e) shows the segmentation result by using our algorithm. Compared to the clustering results of the DA algorithm in Fig. 3.11 (b), the BCFCM algorithm [93] in Fig. 3.11 (c), the superiority of our ASIC algorithm is apparent. More detailed comparisons for other types of noise and edge blurring effect will be given in the next subsections.

Figure 3.11: Synthetic Image. (a) Original image. (b) DA algorithm. (c) BCFCM algorithm. (d) EnFCM algorithm. (e) FGFCM algorithm. (f) ASIC algorithm.

Table 3.1: SA values on synthetic images for DA, BCFCM, EnFCM, FGFCM and ASIC methods.

Noise levels	DA	BCFCM	EnFCM	FGFCM	ASIC
salt&pepper 3%	2.26%	2.38%	2.37%	0.59%	0.34%
salt&pepper 5%	3.80%	3.93%	3.93%	0.98%	0.50%
salt&pepper 7%	5.36%	5.49%	5.49%	1.37%	0.69%
salt&pepper 15%	11.20%	11.45%	11.45%	4.40%	1.99%
Gaussian 4%	0.01%	0.16%	0.16%	0.02%	0.13%
Gaussian 8%	2.96%	0.69%	0.67%	0.05%	0.17%
Gaussian 12%	8.31%	2.08%	1.98%	0.53%	0.24%
Gaussian 16%	12.97%	4.43%	3.02%	1.76%	0.40%
salt&pepper 5% + Gaussian 4%	3.82%	4.01%	3.99%	0.72%	0.52%
salt&pepper 5% + Gaussian 12%	11.64%	6.27%	6.20%	1.17%	0.77%

Quantified comparison of SA results on synthetic images of different noise types and levels

Table 3.1 shows the SA values obtained when applying five different algorithms to Fig. 3.11 for different levels and types of noise. Three types of noise are used in this experiment: salt and pepper noise, Gaussian noise, and mixed noise. From Table 3.1, we can see that our proposed algorithm has the lowest misclassification rates in most cases. When the noise level is low, for example 4% and 8% Gaussian noise, the FGFCM algorithm shows slightly better performance than our algorithm. However, under mixed noise and median-to-high noise levels, our ASIC algorithm is much more robust to the presence of noise as compared to other algorithms, and it is easier to implement in real applications because there is no experimentally adjusted parameter.

Edge blurring effect with low level outlier elimination

In order to further illustrate the effectiveness of the adaptive weighting functions, we shall test the four algorithms on their performance of handling edge blurring effect of the four algorithms with spatial context. In this experiment, we will select the segmentation result from one particular cluster and use the black and white

color to display it. This particular cluster is chosen with the criterion that it has the largest number of misclassified pixels caused by the edge blurring for most of the algorithms.

Three synthetic images with different noise levels are tested in this experiment. From Fig. 3.12, we can see that the BCFCM, EnFCM and FGFCM algorithms have the edge blurring artifacts in the segmentation results, while the artifact has been nicely overcome by our ASIC algorithm. Because the constant spatial constraints in BCFCM and EnFCM algorithms fail to cope with the discontinuity of edges pixels, the membership values of other classes have been increased by the spatial penalty imposed on the actual class. Similarly, the FGFCM algorithm also cannot nicely handle the edge blurring effect because the pre-filtered image is just a less blurred averaged image, although it uses a localized filter window. In ASIC, the adaptive weighting functions can make better trade-off between edge blurring and spatial penalty. The robust estimation step can further reduce misclassified pixels which are caused by outliers, thus improving the performance of segmentation.

Another way to investigate this problem is to look at the distribution of the membership function. We create another synthetic image that is used in [72] for this experiment. The test image contains four classes with intensity values of 0, 25, 50, and 75. The resolution of the image is 96×96 . No noise is added as our purpose is to test the effectiveness of reducing the edge blurring effect. We generate the membership function which corresponds to the class with intensity value of 50. The membership functions on Figs. 3.13(d)-(e) show that the BCFCM, EnFCM and FGFCM algorithms lack sufficient robustness to the edge blurring effect as discussed before, while our ASIC algorithm has much better robustness to this artifact.

Outliers identification by MI maximization with high level outlier elimination

It is interesting to illustrate how the robust estimation step plays a key role in our ASIC algorithm.

Figure 3.12: Handling edge blurring effect results on synthetic images, corrupted with three different levels of artificial noise. Each row contains the segmentation results by using four spatial clustering algorithms for one particular synthetic image. From top to bottom, the row (a)-(d) is corrupted with 8% Gaussian noise, the second row (e)-(h) is corrupted with 5% salt & pepper noise, and the last row (i)-(l) is corrupted with mixed noise of 5% salt & pepper noise + 12% Gaussian noise. From left to right, the column images are the segmentation results of BCFCM, EnFCM, FGFCM, and ASIC, respectively.

Figure 3.13: Performance of several algorithms on edge blurring effect. (a) Original image; (b) Adaptive weighting functions of ASIC algorithm; (c) ASIC membership function; (d) BCFCM membership function; (e) EnFCM membership function; (f) FGFCM membership function.

A synthetic image with mixed noise (5% salt & pepper + 4% Gaussian) is shown in Fig. 3.14 (a). With the ASIC algorithm, the outlier pixels are obtained and shown in Fig. 3.14 (b). Besides the pixels along the borders, some other pixels located exactly or around the contaminated positions have been detected as outliers. By relabeling these outliers with bigger neighborhood window, the segmentation result in Fig. 3.14 (e) visually shows a better performance as compared to the one shown in Fig. 3.14 (d). In order to appropriately illustrate the effectiveness of our high level outlier elimination method, we plot those corrected outlier pixels in Fig. 3.14 (c). From the figure, we found that not only the outlier pixels in homogeneous regions have been corrected, but also the ones along the borders and edges. It experimentally proved that this simple but fast high level outlier elimination method is efficient for our purpose.

Table 3.2 gives the quantified comparisons of SA values for different noise levels at different steps of the ASIC algorithm. From these results, it is clear that ASIC

Figure 3.14: High level outlier elimination. (a) Original image; (b) Outlier identified by MI maximization; (c) Outlier pixels which have been corrected during the high level outlier elimination process; (d) Segmentation result by step1 of ASIC algorithm; (e) Segmentation result by ASIC algorithm.

algorithm is able to reduce the misclassified error by a combination of the low and high level outlier eliminations, thus producing a more consistent and robust image segmentation result. Moreover, it verifies that the adaptive window method produces better results than the fixed 3×3 neighborhood window in the high level outlier elimination step.

3.4.3 Visual comparisons between ASIC and DA, BCFCM, EnFCM, and FGFCM with real images

After the studies of simulated images, we shall now examine the performance of our algorithm with some real images. In order to investigate the performance (in terms of noise removal performance and edge blurring effect handling) of the four segmentation algorithms, a mixed noise (3% salt & pepper + 4% Gaussian) will be artificially added into these real images unless specified otherwise. As for the cluster number of each real image, we manually select c with values between 2 and 4 such that the segmentation results preserve the characteristics of the main objects while keeping a simple representation of the image [9].

Real Image - Plane

The original image is shown in Fig. 3.15 (a). The resolution is 512×512 and the gray scale range is 0 to 255. The contaminated image is shown in Fig. 3.15 (b). For the class number, we choose $c = 3$ because it produces a better representation of the image. From Figs. 3.15 (c)-(f), we can see that our adaptive algorithm gives a much better segmentation, even though a small portion of the wing of the small plane is not well recovered. The mixed noise effect has been greatly reduced by our ASIC algorithm.

Real Image - Lena

The original image (512×512 pixels) is shown in Fig. 3.16 (a), and Fig. 3.16 (b) is the same image contaminated by mixed noise. Figs. 3.16 (c)-(e) show the results

Table 3.2: SA values comparison of ASIC algorithm on synthetic images in different steps.

Noise levels	ASIC		
	low level outlier elimination (step1 only)	high level outlier elimination (combined steps 1 and 2)	
		fixed 3×3 window	adaptive window
salt&pepper 3%	1.03%	0.36%	0.34%
salt&pepper 5%	1.80%	0.52 %	0.50%
salt&pepper 7%	2.73%	0.70%	0.69%
salt&pepper 15%	8.00%	2.05%	1.99%
Gaussian 4%	0.41%	0.14%	0.13%
Gaussian 8%	0.71%	0.18%	0.17%
Gaussian 12%	1.09%	0.31%	0.24%
Gaussian 16%	2.20%	0.60%	0.40%
salt&pepper 5% + Gaussian 4%	2.53%	0.57%	0.52%
salt&pepper 5% + Gaussian 12%	4.45%	0.90%	0.77%

Figure 3.15: Real Image - Plane. (a) Original image. (b) Noisy image corrupted by 3% salt & pepper + 4% Gaussian mixed noise. (c) Segmentation results using BCFCM. (d) Segmentation result using EnFCM. (e) Segmentation result using FGFCM. (f) Segmentation result using ASIC.

of the BCFCM, EnFCM, and FGFCM algorithms. Fig. 3.16 (f) shows the result of our ASIC algorithm. In this example, we choose $c = 4$. From Figs. 3.16 (c)-(d), we could say that the BCFCM and EnFCM cannot produce good segmentation results due to the influence of the mixed noise. Many small patches are evident in the segmentation results. FGFCM and our algorithm can better eliminate the impact of mixed noise, even though there are still some discontinuous regions. Compared to FGFCM, our ASIC can provide a more homogenous segmentation result while suppressing the effect of mixed noise. This is improved by the influence of stronger neighborhood effect imposed by the adaptive selection of the weighting functions.

Real Image - Cameraman

Another example with a real image is shown in Fig. 3.17 (a). The resolution is 256×256 , and $c = 2$ is used in this example. A 4% Gaussian noise has been added to this image. Fig. 3.17 (b) shows the adaptive weighting functions. Suppose that our goal of applying clustering algorithm is to extract the camera man with his equipment from the background. This means that we want the segmented foreground image to be with less isolated background regions, while preserving as much foreground details as possible for better display. The segmentation result of the ASIC algorithm is shown in Fig 3.17 (c). We can see that the camera man with his tripod has been well separated from the background, and only several small isolated background regions are present at the lower half of the image. Figs. 3.17 (d)-(e) show the results of the BCFCM algorithm with $\alpha = 0.5$ and $\alpha = 0.7$. When the neighboring effect is small (α is small), there are many perceptible isolated speckles at the bottom-right corner of the image. Increasing the spatial penalty term α will result in a better result. With high spatial penalty settings as shown in Figs. 3.17 (g) and (i), EnFCM and FGFCM can provide comparable results with our ASIC algorithm, however, it will oversmooth the details of the foreground which is obvious at the tripod support area. At low spatial penalty settings (Figs. 3.17 (f) and (h)), the details of the tripod can be preserved at the cost of poor segmentation results. To this end, we can see that the ASIC method is an effective method in smoothing out the image

Figure 3.16: Real Image - Lena. (a) Original image. (b) Noisy image corrupted by 3% salt & pepper + 4% Gaussian mixed noise. (c) Segmentation results using BCFCM. (d) Segmentation result using EnFCM. (e) Segmentation result using FGFCM. (f) Segmentation result using ASIC.

Figure 3.17: Real Image - Cameraman. (a) Original image of a camera man with 4% Gaussian noise; (b) adaptive weighting functions for ASIC algorithm; (c) ASIC algorithm; (d) BCFCM algorithm ($\alpha = 0.5$); (e) BCFCM algorithm ($\alpha = 0.7$); (f) EnFCM algorithm ($\alpha = 0.5$); (g) EnFCM algorithm ($\alpha = 3.5$). (h) FGFCM ($\lambda_g = 0.5, \lambda_s = 3$). (i) FGFCM ($\lambda_g = 2.5, \lambda_s = 3$).

noise while preserving the important details of the image.

3.4.4 3D Segmentation of Brain Magnetic Resonance Image (MRI)

To further evaluate the performance of our clustering algorithm on multidimensional images, we also apply the ASIC algorithm on real 3D brain MRI data. The MRI data is provided by Institute of Mental Health, Singapore. The tested brain image is T1 modality, with a resolution of $256 \times 256 \times 180$ voxels and $1mm$ slice thickness.

Figure 3.18: 3D MRI brain image segmentation with ASIC algorithm. (a) DA algorithm. (b) ASIC algorithm.

Figs. 3.18 show the 3D segmentation results obtained by the DA and ASIC algorithms. We removed some of the segmentation volume to show the performance. As can be seen from Fig. 3.18 (b), our ASIC algorithm generates more homogenous segmentation results and also preserves the details between the different brain tissues (white matter, gray matter and cerebrospinal fluid), as compared to the simple DA algorithm.

3.5 Conclusions

We have presented a robust image segmentation technique based on information clustering in this chapter. The proposed method is aimed to solve the problems of the lack of spatial information and adaptiveness of the existing methods in the literature. Within our information-theoretic clustering framework, an adaptive similarity measure is developed for incorporating the spatial information into the image segmentation method based on data clustering. This similarity measure is fully adaptive to the image content and is able to eliminate misclustered data in the low level. Further, the remaining unreliable data points are reassess via a capacity max-

imization process and are to be reassigned based on an adaptive window to further purify the segmentation results.

The ASIC algorithm is effective in that it is adaptive to the image content to favor the solution of piecewise-homogeneous labeling without affecting the edges, thus preserving most of the details of the image. The amount of spatial constraint is controlled by the image itself without the need of user intervention. Furthermore, the two-step approach of ASIC provides the capability to eliminate outliers in a relatively noisy environment. A comparison of our ASIC algorithm with other variants of spatial clustering algorithms, like the BCFCM, EnFCM and FGFCM algorithms, illustrates that while these algorithms are effective in eliminating the noise or small isolated regions, our algorithm has a better performance in terms of preserving the details and reducing the edge blurring artifact in a relatively noisy environment. The proposed method can be further improved to fit many other images (e.g. 3D MRI brain image), and can be integrated into other image segmentation systems to achieve a fully automatic image segmentation system.

Chapter 4

A Robust Information Fuzzy Clustering Algorithm for Image Segmentation

Fuzzy C-means (FCM) is one of the most popular fuzzy clustering algorithms. It has been shown to be more advantageous than crisp clustering in that the crisp assignment of a data point to cluster center is not reasonable in real applications. Further, when compared to the crisp approaches, the FCM method is more tolerant to variations and noise in the input data [50]. Therefore, this method has been widely used in different engineering and scientific fields such as pattern recognition, data mining, and image processing [17].

In order to make the standard FCM algorithm [47] more robust to noise and outliers, Davé *et al.* [99] proposed a robust version of the FCM algorithm. It is probably one of the most known robust fuzzy clustering algorithm in the literature. In this approach, a separate noise cluster is introduced into the clustering process and is represented by a prototype that has a predefined constant distance δ . With this *noise cluster*, the clustering results are less sensitive to the noise and outliers. However, the value of δ is critical to the robustness of this approach and the determination of a suitable value of δ is not easy in real clustering applications. Another attempt to overcome the relative membership problem can be found in the possibilistic c-means (PCM) algorithm of [100]. The primary objective of the PCM method is to achieve

membership values that are possibilistic. With this modification and relaxation of the constraint on the membership values, the membership values for representative data points can be set as high as possible while unrepresentative data points have low membership values in all clusters [100]. This idea is further expanded in the possibilistic fuzzy C-means (PFCM) [101] approach which combines the objective functions of the FCM-type and PCM-type algorithms with extra hyper-parameters to include both the fuzzy membership and possibility values for further robustness improvements.

For image segmentation, the robust analysis in clustering approaches mainly focuses on the lack of spatial information in the objective function. Many modified fuzzy clustering approaches had been reported in [53, 67, 70–74]. These methods are better than the conventional fuzzy clustering algorithms because the clustering decision of a single pixel are now reconsidered by the influence from the neighborhood patterns. This improvement tends to smooth out the isolated noise or image artifacts and produces more homogeneous segmentation results.

In this chapter, we propose a new robust fuzzy clustering algorithm, which are called adaptive spatial information fuzzy clustering (ASIFC) algorithm, to address the noise problem and the lack of spatial information for the conventional FCM algorithm in image segmentation aspect. We aim to extend the idea in [60] to specifically enhance the robustness of FCM algorithms. The lack of spatial information problem is taken into consideration by using the adaptive similarity measure proposed in Chapter 3, while the noise and outliers are to be identified through the mutual information (MI) maximization. This new information framework allows us to add robust capability to the basic FCM algorithm and inherits its advantages such as fast processing time and less sensitive to the density of clusters.

4.1 An Adaptive Spatial Information Fuzzy Clustering (ASIFC) Algorithm

The objective function of the famous fuzzy C-means clustering is given by

$$J_q = \sum_{i=1}^N \sum_{k=1}^K u_{ik}^q \|x_i - w_k\|^2, 1 \leq q < \infty \quad (4.1.1)$$

where q is a real number greater than 1 that controls the fuzzyness of the clustering results, and u_{ik} is the membership value of data point x_i inside the k th cluster.

In order to overcome the noise problem of FCM algorithm, Krishnapuram *et al.* [100] proposed the possibilistic clustering algorithm to achieve membership values that are possibilistic. Here, one input data point has little effect on good clusters if its possibility as noise is high, so that the clustering results are less distorted as compared to the FCM method. The modified objective function of PCM algorithm is

$$J_{PCM} = \sum_{i=1}^N \sum_{k=1}^K t_{ik}^q \|x_i - w_k\|^2 + \sum_{k=1}^K \eta_k \sum_{i=1}^N (1 - t_{ik})^q \quad (4.1.2)$$

where η_k is the estimated noisy cluster hyper-parameter, and there are no constraints on the possibility membership values t_{ik} other than the requirement that they should be in $[0, 1]$.

The PCM algorithm utilizes different hyper-parameters η_k to form K different noisy clusters and has the capability to identify noisy data through the possibility membership values, i.e. the sum of possibilistic value t_{ik} is not equal to one in a particular cluster as compared to the FCM algorithm. However, PCM is much more sensitive to the initialization and η_k 's selection as compared to the FCM-type algorithms, and this has limited its applications [50].

To circumvent these shortcomings in the basic fuzzy clustering approaches and to take into consideration the incorporation of spatial information, we introduce input data distribution and an improved similarity measure (refer to (3.2.49)) into the objective function of FCM algorithm. This can be done by modifying the objective

function (4.1.1) as follows:

$$J_{ASIFC} = \sum_{i=1}^M \sum_{j=1}^N \sum_{k=1}^K e_{i,j} u_{k|i,j}^q d_s(x_{i,j}, w_k), 1 \leq q < \infty \quad (4.1.3)$$

where

$$d_s(x_{i,j}, w_k) = \lambda_{i,j} \|x_{i,j} - w_k\|^2 + \frac{(1 - \lambda_{i,j})}{N_R} \sum_{x_{r,c} \in N_{i,j}} \|x_{r,c} - w_k\|^2 \quad (4.1.4)$$

and $e_{i,j}$ is the estimated input data distribution (with constraint $\sum_{i=1}^M \sum_{j=1}^N e_{i,j} = 1$), which is set to $1/M \times N$ in the initial stage of the ASIFC algorithm. $u_{k|i,j}$ is actually another form of notation for the membership value as we try to make the notations consistent with the information theory. The choice of the spatial weighting factor $\lambda_{i,j}$ has been discussed in Section 3.2.1.

Note that the estimated input data distribution plays an important role in identifying the noise data and outliers. During the initial stage, since we have no preference for any input data points, the initial values for $e_{i,j}$ are usually set to $1/M \times N$ unless specified user preferences are available. In the pruning phase, the data reliability/preference is assessed by MI maximization. The identified unreliable data points can be deleted (for common data clustering applications) or relabeled (for image segmentation applications) to achieve better clustering results. Embedded in the new objective function (4.1.3), the estimated input data distribution $e_{i,j}$ also plays a similar role of the possibility value as in the PCM algorithm to eliminate the outliers. Furthermore, the improved similarity measure $d_s(x_{i,j}, w_k)$ offers an optimal way to tackle the lack of contextual information from the image coordinates. Except for the fuzzifier q , there is no pre-fixed parameter in the specific objective function (4.1.3) for the clustering phase, which makes our proposed ASIFC algorithm relatively simpler to use when compared to other robust versions of fuzzy clustering algorithms like the PCM-type algorithms.

By introducing the hidden constraint $\sum_{k=1}^K u_{k|i,j} = 1$, and minimizing the objective function (4.1.3), we have

$$\frac{\partial J_{ASIFC}}{\partial u_{ik}} = 0 \quad , \quad \frac{\partial J_{ASIFC}}{\partial w_k} = 0 \quad (4.1.5)$$

These lead to the iterative solutions

$$u_{k|i,j} = \frac{d_s(x_{i,j}, w_k)^{-1/(q-1)}}{\sum_{c=1}^K d_s(x_{i,j}, w_c)^{-1/(q-1)}} \quad (4.1.6)$$

and

$$w_k = \frac{\sum_{i=1}^M \sum_{j=1}^N e_{i,j} u_{k|i,j}^q (\lambda_{i,j} x_{i,j} + \frac{(1-\lambda_{i,j})}{N_R} \sum_{x_{r,c} \in N_{i,j}} x_{r,c})}{\sum_{i=1}^M \sum_{j=1}^N e_{i,j} u_{k|i,j}^q} \quad (4.1.7)$$

To derive the robust estimation of input data distribution $e_{i,j}$, we first define the mutual information between the set of optimal membership distributions of input data points obtained via (4.1.6) and (4.1.7) and the input data distribution as follows:

$$I(\mathbf{X}, \mathbf{W}) = \sum_{i=1}^M \sum_{j=1}^N \sum_{k=1}^K e_{i,j} u_{k|i,j} \log \frac{u_{k|i,j}}{e_{i,j} \sum_{i=1}^M \sum_{j=1}^N (e_{i,j} u_{k|i,j})} \quad (4.1.8)$$

Using similar procedure as in Section 3.1.2, we can obtain the robust estimate for $e_{i,j}$ by the following function

$$e_{i,j} = \frac{e_{i,j} \exp \left(\sum_{k=1}^K u_{k|i,j} \log \frac{u_{k|i,j}}{\sum_{i=1}^M \sum_{j=1}^N (u_{k|i,j} e_{i,j})} \right)}{\sum_{i=1}^M \sum_{j=1}^N \exp \left(\sum_{k=1}^K u_{k|i,j} \log \frac{u_{k|i,j}}{\sum_{i=1}^M \sum_{j=1}^N (u_{k|i,j} e_{i,j})} \right)} \quad (4.1.9)$$

The ASIFC is basically a two-step clustering algorithm for image segmentation. First, it uses a conventional FCM algorithm together with the improved similarity measure as the starting point to explore the details of the input features from the image and group them effectively into several clusters subject to the adaptive spatial weighting factors $\lambda_{i,j}$. Given the estimated cluster centers w_k and the optimal membership value $u_{k|i,j}$ from ASIFC, the reliabilities of the input data points are assessed by the MI maximization. Similar to our adaptive spatial information clustering (ASIC) algorithm, these identified unreliable data points are going to be clustered again with a bigger neighboring window.

4.2 The Proposed ASIFC Algorithm

The implementation of the ASIFC algorithm is described below. Suppose that we are given the feature vectors of input image $X = \{x_{1,1}, x_{1,2}, \dots, x_{M,N}\}$ with the resolution of $L = M \times N$.

Fuzzy clustering with adaptive spatial weighting factors:

- 1). Set the pre-specified number of clusters K , convergence parameter $\epsilon = 0.001$, fuzzifier $q > 0$, and the input data distribution $e_{i,j} = 1/(M \times N)$ for $i = 1$ to M , $j = 1$ to N .
- 2). Initialization: Calculate the weighting function $\lambda_{i,j}$ which occurs in formulation (3.2.59).
- 3). Perform fixed point iteration according to (4.1.6) and (4.1.7).
- 4). Convergence Test: If not satisfied, then go to 3).

MI Maximization for identifying the unreliable data points:

- 5). Select $e_{i,j} = 1/(M \times N)$, for $i = 1, \dots, M, j = 1, \dots, N$, and $\varepsilon > 0$, then start the fixed point iteration of the robust density estimation according to steps 6) to 7).
- 6). Iteratively update the input data distribution by using

$$c_{i,j} = \exp \left(\sum_{k=1}^K u_{k|i,j} \log \frac{u_{k|i,j}}{\sum_{i=1}^M \sum_{j=1}^N (u_{k|i,j} e_{i,j})} \right)$$

and

$$e_{i,j} = e_{i,j} \frac{c_{i,j}}{\sum_{i=1}^M \sum_{j=1}^N c_{i,j}}$$

- 7). If

$$\ln \sum_{i=1}^M \sum_{j=1}^N e_{i,j} c_{i,j} - \ln \max_{\{i,j\}} c_{i,j} < \varepsilon$$

then go to 9) where $\varepsilon > 0$.

- 8). Label unreliable data points, and obtain the membership of all data points for a hard clustering solution.
- 9). Create a 3×3 window centered at the outlier, obtain the clustering labels for all neighboring pixels within this neighborhood window from the fuzzy clustering step.
- 10). Store all label counts for each cluster into $\xi_k, k = 1, 2, \dots, K$.

11). Find the maximum value of ξ_k and its corresponding cluster index, i.e. $\xi_m = \max(\xi_k)$.

12). If $\xi_1 = \xi_2 = \dots = \xi_K$, increase the current window edge by 2 more pixels, (for example, from 3×3 to 5×5), go to 12); Otherwise, update the outlier label with m .

13). Iterate for all outlier pixels until there is no further change, then stop.

The weighting factors $\{\lambda_{i,j}\}$ do not change in each fixed point iteration, so we can pre-compute them at step 2). After this, the algorithm begins the fixed point iteration.

4.3 Experimental Results

In this section, the results of applying ASIFC algorithm to synthetic and real images are present. In the comparative study, we also included other competing techniques from the literature like bias-corrected fuzzy c-means (BCFCM) [70], enhanced fuzzy c-means (EnFCM) [73], Fast Generalized fuzzy c-means (FGFCM) [74], and our own Adaptive Spatial Information-Theoretic Clustering (ASIC) [P5].

In all examples, the cooling factor $\alpha = 0.95$ was used unless otherwise stated. The default values for fuzzifier $q = 2$, convergence value $\epsilon = 0.001$ and maximum iteration number of 100, are used unless otherwise stated. The neighborhood window size is 3×3 . The spatial parameter α in BCFCM algorithm is obtained by searching the interval $[0.5, 0.9]$ with a step size of 0.05 in terms of the optimal SA value. Similarly, the α in EnFCM is obtained by searching the interval $[0.2, 8]$ as suggested in [74]. The λ_g in FGFCM is selected from 0.5 to 6 with an increment of 0.5.

4.3.1 Synthetic images

In order to compare the ASIFC algorithm with the other four methods, we create a synthetic test image as shown in Fig. 4.1 (a), and the resolution of this image is

Table 4.1: SA values of BCFCM, EnFCM, FGFCM, ASIC and ASIFC methods for the synthetic image.

Noise levels	BCFCM	EnFCM	FGFCM	ASIC	ASIFC
Gaussian 3%	0.03%	1.25%	0.01%	0%	0%
Gaussian 5%	0.04%	1.39%	0.02%	0.02%	0.02%
Gaussian 10%	0.27%	1.53%	0.01%	0.22%	0.22%
Gaussian 20%	6.14%	2.05%	0.95%	0.73%	0.64%

128×128 pixels and the grey-scale range is 0 to 255. Gaussian noise is used in this experiment, with noise levels of 3, 5, 10, and 20.

To validate the segmentation results, we use the segmentation accuracy (SA) which is computed by

$$SA = \frac{\text{number of misclassified pixels}}{\text{total number of pixels}} \quad (4.3.10)$$

Fig. 4.1 (a) displays the synthetic image with Gaussian noise level of 20 added. We applied the five techniques and obtained the segmentation results as shown in Figs. 4.1 (c)-(g). Overall, the ASIC and ASIFC yield better results than the rest. Visually, we can observe that the ASIFC performs slightly better than the ASIC algorithm. This can be further illustrated by the SA values listed in Table 4.1. Table 4.2 shows the average execution time for each algorithm implemented with MATLAB. It is obtained by averaging 10 optimal trials. From the comparison of the computational time among these five techniques, we can see that the ASIC algorithm has the largest computation loan among them due to its annealing-like optimization process. The robust estimation step also contributes some computational burden to this. On the other hand, EnFCM and FGFCM perform much faster than ASIC as they are derived from fuzzy C-means algorithm. They also have faster convergence rates than the BCFCM's as they have better spatial constraints. For the proposed ASIFC algorithm, although it performs slower than the other FCM variants, it did improve significantly on the computational complexity by using the fuzzy clustering framework. This gap can be further narrowed by using more efficient method to compute the weighting image and estimate the outliers.

Figure 4.1: Comparison of BCFCM, EnFCM, FGFCM, ASIC and ASIFC approaches on a synthetic image: (a) original synthetic image, (b) weighting image for ASIFC algorithm, (c) BCFCM segmentation result, (d) EnFCM segmentation result, (e) FGFCM segmentation result, (f) ASIC segmentation result, and (g) ASIFC segmentation result.

Table 4.2: Average computational time (in seconds) with MATLAB on synthetic image for BCFCM, EnFCM, FGFCM, ASIC and ASIFC methods.

Noise levels	BCFCM	EnFCM	FGFCM	ASIC	ASIFC
Gaussian 3%	0.4141s	0.2360s	0.2594s	2.2766s	1.2078s
Gaussian 5%	0.4891s	0.2344s	0.3047s	3.4641s	1.3563s
Gaussian 10%	0.5891s	0.2500s	0.3065s	3.6172s	1.6484s
Gaussian 20%	0.8391s	0.3547s	0.3766s	3.3172s	2.4781s

4.3.2 Phantom MRI brain image

To test the performance of our method on medical images, we will test our algorithm with the Brainweb MRI Brain Database [102] and compare the results with other competing methods. In this example, we will focus on the segmentation of white matter (WM), grey matter (GM), and cerebrospinal fluid (CSF), i.e. the three main tissues of the human brain. Before we apply the clustering methods, we will perform the extracranial tissues removal by using the discrete model provided by [103]. This will help to remove the negative effect from the extracranial tissues during segmentation, and it will also make the validation easier to manipulate. The images we chose in this example are simulated T1-weighted MR brain images with 1-mm spacing, with 1%, 3%, 5% noise levels, and 0%, 20%, 40% inhomogeneities.

To validate the segmentation results, we use the same evaluation measure as in [77], namely, tissue segmentation accuracy (TSA):

$$TSA = \frac{2N_{CTk}}{N_{CITk} + N_{GTk}} \quad (4.3.11)$$

where N_{CTk} denotes the number of pixels that were correctly (inside the mask of ground truth) assigned to tissue k by a given method, N_{CITk} is the total (inside and outside the mask of ground truth) number of pixels assigned to tissue k , and N_{GTk} is the number of pixels belonging to tissue k in the discrete anatomical model (the ground truth mask). The ground truth mask can be obtained by using the discrete model provided by [103].

Fig. 4.2 shows the sample slides of simulated MRI brain image, the true partial

Figure 4.2: Example of simulated MRI brain image segmentation: (a) Sample slides of a simulated MR brain image with 5% noise, and no intensity inhomogeneity. (b) Discrete anatomical model for WM, GM, and CSF.

volume model of that was used to generate the simulated image. To perform a more comprehensive comparison, we selected 40 MR brain images (from index 61 to index 100) from the BrainWeb database, which consist a variety of shapes and sizes of the three main brain tissues. Fig. 4.3 shows the TSA values of three main brain tissues obtained when BCFCM, EnFCM, FGFCM, ASIC and the proposed ASIFC algorithms are applied to the selected images. The 3, 5, and 7% noise cases are tested. Figs. 4.3 (a), (d) and (g) show that the FGFCM and BCFCM algorithms perform quite well for all the noise levels on CSF tissue segmentation. EnFCM, ASIC and our proposed ASIFC algorithm seem to yield bigger error rates. From the sample MRI image in Fig. 4.2 (a), we can see that CSF tissue consists of many small regions with very low intensity values. Some even smaller than a 3×3 window. Therefore, for the ASIC and our ASIFC algorithm, the dissimilarity measure tends to trust more on the center pixel itself and imposes little spatial constraint to it. This is the main reason of the relatively lower TSA values. This problem can also be found in the TSA values for the GM tissue, although some of the misclustering pixels are contributed by the CSF tissue. For the segmentation of WM tissue, it is apparent that the ASIFC algorithm performs better. This is improved by the influence of stronger neighborhood effect imposed by the adaptive selection of the

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Figure 4.3: Tissue Segmentation Accuracy (TSA) value of CSF, GM, and WM with different noise levels for 40 simulated MR brain images.

weighting functions. This is more obvious by looking at the TSA values obtained by ASIC algorithm. Although the ASIC algorithm outperforms the ASIFC algorithm for the WM tissue, the proposed ASIFC has better segmentation for the GM and CSF tissues when compared to ASIC algorithm as the fuzzy c-means algorithm is more robust to the deterministic annealing algorithm [50].

From this comprehensive comparison we can conclude that, the FGFCM and BCFCM algorithms are useful in the segmentation of CSF and GM tissues, while the ASIC and our proposed algorithm would provide better segmentation results for the WM tissue, and probably the GM tissue as well.

4.3.3 MRI brain image

In this experiment, we shall evaluate the performance of our proposed clustering algorithm on real MRI brain image data. The real MRI brain images are obtained using a Magnetization Prepared Rapid Gradient Echo (MPRAGE) imaging sequence. The size of the image volume is $256 \times 256 \times 180$ voxels. The voxel size is $0.9 \times 0.9 \times 0.9mm$.

We choose 3 sample slices under different anatomical views (sagittal, coronal and transverse) as our test images. Visually, all the competing algorithms are able to produce pretty good segmentation results for these three MRI brain images. When we try to examine the results closely, we can see that there are some noticeable differences among them. For example, the segmentation results produced by the BCFCM is a bit 'noisier' when compared to other methods. You can see many isolated small regions in Fig. 4.5 (d). The ASIC method produce a more consistent and smoother segmentation results. However, some details are lost in the cerebellum region as compared to the EnFCM, FGFCM and the proposed ASIFC algorithms, which can be seen in the cerebellum region in Fig. 4.4 (g) and Fig. 4.5 (g). Among the EnFCM, FGFCM, and ASIFC algorithms, our algorithm yields an optimal trade-off between smoothing out noise and preserving the fine details for the whole image. These improvements can be visualized in Fig. 4.5 (c) and

Fig. 4.6 (c). With the adaptive similarity measure and information-theoretic framework, the newly proposed fuzzy clustering algorithm has strengths to overcome the sensitivity of noise or outlier data points and the lack of spatial information. The power of the ASIFC algorithm does not only lie in providing more homogeneous and smoother segmentation results, but also in preserving the image details and boundary information.

4.4 Conclusions

In this chapter, we have presented a novel robust information fuzzy clustering method that uses the information-theoretic clustering concept and adaptive spatial weighting factors to improve the image segmentation results. The proposed ASIFC algorithm addresses the two intertwined problems, of the identification of noisy data and the lack of spatial contextual information in data clustering based image segmentation. The algorithm is fully automated, except for the initialization of fuzzifier, cluster number and stop condition parameters. The experimental results based on synthetic images show that the proposed algorithm can effectively improve the segmentation accuracy. The experiment on real MRI image also demonstrates that the ASIFC method can deliver useful segmentation results for brain image analysis. Further adaptations and validation studies are required if the method is to be used for clinical studies.

These two robust techniques, i.e. ASIC and ASIFC are mainly based on the intensity-feature of the image. However, image understanding can be advanced if the image objects are represented in a more natural way, for example, by its contour, shape or spatial relationships to other objects in the scene [104]. In the next chapter, we will briefly review some data clustering techniques for image understanding based on the curve, contour representation and find solution that can be useful for developing robust data clustering framework.

Figure 4.4: Example 1: Segmentation on a sagittal view slice. (a) Sample slides from real 3D MRI brain image volume. (b) adaptive weighting parameters of our ASIFC algorithm. (c) ASIFC segmentation. (d) BCFCM segmentation. (e) EnFCM segmentation. (f) FGFCM segmentation. (g) ASIFC segmentation.

Figure 4.5: Example 2: Segmentation on a coronal view slice. (a) Sample slides from real 3D MRI brain image volume. (b) adaptive weighting parameters of our ASIFC algorithm. (c) ASIFC segmentation. (d) BCFCM segmentation. (e) EnFCM segmentation. (f) FGFCM segmentation. (g) ASIFC segmentation.

Figure 4.6: Example 3: Segmentation on a transverse view slice. (a) Sample slides from real 3D MRI brain image volume. (b) adaptive weighting parameters of our ASIFC algorithm. (c) ASIFC segmentation. (d) BCFCM segmentation. (e) EnFCM segmentation. (f) FGFCM segmentation. (g) ASIFC segmentation.

Chapter 5

Data Clustering with Curve Data Representation

5.1 Introduction

In addition to the traditional techniques that used intensity feature vectors (intensity-based approaches) to understand the image objects, a fair bit of results have also been reported in recent years on using curves and/or contours (feature-based approaches) to analyze the image objects [34, 104–111]. The representation of image objects using curves or contours is the most natural way of describing and recognizing shapes by human beings. The curve or contour descriptors can provide smooth and continuous representation of particular region-of-interest (ROI) in image. In addition, it can avoid the negative impact caused by noise and other image artifacts and it resolves the boundary discontinuity problem that always exists in the pixel-based image analysis methods.

In this chapter, we shall first review some recent research findings in data clustering analysis based on the curve representation. We then examine the derivation of expectation maximization for a widely used random effect model, i.e. the Gaussian random effect mixture model. This chapter serves as a supplementary literature review to Chapter 2 and it also provides useful preliminaries for Chapter 6.

5.2 Literature Review

The curve or contour descriptors are useful in many applications. For example, in surveillance, we can use them to differentiate vehicle types, classify civil airplanes and fighter planes, or even retrieve different tank views [112–116]. In medical image analysis, contour-based descriptors can be used to model the anatomical structures so that clinical analysis can be performed, such as segmentation, registration, control/patient group analysis, or building atlases analysis. In psychology, curves can be used to extract the silhouetted face profiles used in psychological experiments [34]. It is also useful to depict the characters in handwriting recognition application. Moreover, we also can find the trace of contour-based methods in other data clustering related applications. In bioinformatics, curve representation are used model the large amount of gene expression data in order to circumvent the huge dimensionality issue and missing data problem in gene data analysis [2]. In stock market and traffic control, curve clustering analysis are also favored [32].

Of course, the first and most important step in using contour-based data clustering analysis methods is the extraction of contours representation itself. Although this problem is application oriented, finding a proper contour representation will definitely improve the performance of the analysis. In early years, landmark-based analysis had been successfully used in many applications [104, 118–123]. It uses discrete, coarse sampled points from the object contour as features, and then cluster the objects based on the similarity of these landmarks. However, these methods require some user interferences to choose the landmarks and lack physical interpretation [108]. The landmark features sometimes can be further approximated by polygonal function [105], Fourier transform [124] or wavelet function [117, 125]. Using parametrical descriptor for contours provide simple, smooth and efficient way to model the contour variability. Moreover, it can greatly reduce the computational cost as the contours are usually described by a few basis coefficients. Deformable model based methods like active contour or snakes [111, 126–128] have achieved many successes in medical image analysis, especially in the MRI brain image registration area. They can automatically detect the object contour via some energy optimiza-

Figure 5.1: Face contours extracted by landmark based method. (image is extracted from Wang *et al.* [117]).

Figure 5.2: Face contours represented by B-splines. (Image is extracted from Davidenko [34]).

Figure 5.3: Vehicle contours extracted using method in [113]. (image is extracted from Thakoor *et al.* [110]).

tion schemes and are less sensitive to the initial conditions [129]. Other representation methods like level set [109, 130] also can be found in the literature. The contour representation by level set can be easily integrated into Expectation-Maximization (EM) based clustering algorithms to obtain the "average" shape contour and its variations within each contour class.

Several real examples in terms of feature-based contour representation techniques are presented in Figs. 5.1-5.4. Among these representation techniques, parametrical contour descriptors (especially B-splines) are considerably more popular in data clustering based methods. As the representative model, B-spline mode has the following advantages: 1) The complex object contour can be fitted by B-spline curve which is determined by using a small number of parameters; 2) The fitting is linear and smooth, and can be adjusted locally by the inherent well-defined properties of the B-spline model; 3) This parametric model is rather generative and can be advanced to other polynomial variants, Fourier or wavelet transform without much difficulty; and 4) With its intrinsic parametric formulation, integration with popular data clustering methods is straightforward. B-spline model also have been successful in functional data analysis applications [2, 4, 5, 131–133]. It is not only capable of capturing and preserving the functional structure of the data, but is also effective to reduce the large data dimensionality and high computational cost as compared to traditional data-driven data clustering methods.

In many B-spline based curve clusterization applications, the problem formulation is

Figure 5.4: Anatomical contour representation via deformable model for MRI brain images. (image is extracted from Yang *et al.* [134]).

usually cast by a random-effect model [2–4, 109]. Suppose $\mathbf{Y}_i = \{y_{ij}\}_{i=1, \dots, n}^{j=1, \dots, g}$ denote the g -dimensional vector of the observed data for i th individual curve, and $\boldsymbol{\mu}_c$ denote the mean or average curve profile of the c th cluster. The random effect model can be generalized as:

$$\mathbf{Y}_i = \boldsymbol{\mu}_c + \mathbf{b}_i + \boldsymbol{\epsilon}_i \quad (5.2.1)$$

where the second term $\mathbf{b}_i$ models the random effect for the i th individual curve, and the last term $\boldsymbol{\epsilon}_i$ is used to model the uncorrelated measurement error. The main advantage of this model is that it can capture the intra-cluster derivation from the mean curve profile which is not due to the measurement error. If we are able to calculate the 'average' curve within each contour class, as well as the within cluster 'shift' for each individual one, we can explore the whole structure of the whole curve objects by just the means and variances. Note that the representations of the whole data set at this stage is quite simple, just the mean profiles and specified individual deviations. For this random effect model, the Expectation-Maximization (EM) [135] framework is obviously the most appropriate method to solve this problem. For

example, Jank [133] applied an ascent EM algorithm to cluster the basis coefficients obtained from the bidder curves in a large online auction databases. Tsai *et al.* [109] introduced an level set based EM approach to classify a contour database and estimate the best exemplified mean contour for each contour class. The contour is fitted by the zero level set and the iterative EM algorithm is used to estimate and classify the level set function of the specified curves. Similarly, Luan *et al.* [2] proposed a Gaussian mixture model with B-splines to cluster time-course gene expression data. This approach utilized a multivariate Gaussian mixture model to represent the gene expression data and use the EM algorithm to estimate the associated parameters by treating the cluster indicator and random effect as missing data. We shall see that the EM framework is well defined and generalized. Therefore, any appropriate contour representation (eg. B-spline or level set function) can be integrated into this framework.

To summarize, the major work in the literature on clustering contours involves: 1) the use of parametric curve representations (B-spline model); 2) the use of random effect model to formulate the clustering problem; 3) applying EM algorithm in the optimization process. In the rest of this chapter, we will explore these three concepts.

5.3 B-spline Curve Representation

The simplest way to represent a curve is to use a polynomial or a combinations of a set of polynomials. The spline is one special function defined by polynomials over several small piecewise intervals. The most obvious difference between normal polynomials and spline is that the defined range is local in spline so that it can overcome the undesirable oscillations and noise problems when the interval range is larger [136]. Further, the smoothness condition is always imposed between neighboring polynomials for spline such that the whole curve is smooth and continuously differentiable.

A B-spline, i.e. basic spline, is actually the smallest element (also a spline function) with regard to the given degree, smoothness and domain partitions to represent a

Figure 5.5: Construction of a cubic spline curve. (image is extracted from wiki online website).

spline function. Any spline function with given degree, smoothness and domain partitions can be viewed as a linear combination of B-splines under the same conditions [137].

In general, B-spline curves, are defined by three elements: 1) control points, 2) a set of knots, and 3) a set of basis functions for each control point. Suppose one is given n valued t_i with $0 \leq t_0 \leq t_1 \leq \dots \leq t_n \leq 1$, the domain of the spline is control by t under the interval $[0, 1]$. The knots t_i will divide the closed interval so that the spline has local support in the half open interval $[t_i, t_{i+1})$. The control point is used to constraint the local shape of the spline function [137]. Fig. 5.5 shows how the control point works during the construction of a cubic spline. In Fig. 5.5, P_0, P_1, P_2, P_3 are control points and Q_0, Q_1, Q_2, R_0, R_1 are intermediate points that parameterize the path of B .

If we know the degree p of the basis function, we can define the i th B-spline basis function by the Cox-de Boor recursive algorithm [137] as

$$\begin{aligned} B_{i,0}(t) &= \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ B_{i,p}(t) &= \frac{t - t_i}{t_{i+p} - t_i} B_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} B_{i+1,p-1}(t) \end{aligned} \quad (5.3.2)$$

The most commonly used degree is 3 (i.e. cubic B-spline basis) as it provide the best balance between smoothness and degree complexity of polynomials.

Figure 5.6: An example of cubic B-spline curves with 6 control points.

To this end, a B-spline function can be expressed as a linear combination of m basis functions of degree p :

$$p(t) = \sum_{i=1}^m \beta_i B_{i,p}(t), t \in [t_{p-1}, t_m] \quad (5.3.3)$$

where $\{B_{i,p}\}$ are basis functions with degree p and $\{\beta_i\}$ are the given control points. t are knots that are used to parameterize the spline with $0 \leq t_0 \leq t_1 \leq \dots \leq t_n \leq 1$, and n is chosen by the user. If the knots are equally spaced, the resulting B-spline is uniform B-spline; otherwise, it is non-uniform B-spline [137].

A cubic B-spline curve constructed by MATLAB is shown in Fig. 5.6. Since the basis function is determined once the degree of the basis function is fixed, it is obvious by Eq. (5.3.2) that the B-spline representation is only determined by the small number of control points and the pre-specified knots. Note that this representation is continuous and any missing data point or detailed object curvature can be pre-

dicted by the evaluation of the specified location along the curve. This benefit of the B-spline curves had been used by Abraham *et al.* in [132] to analyze the temperature and evolution of pH value in a food industry process. We will conclude the introduce of B-spline curve representation at this stage. More information about other important properties and computation issues for the B-spline functions can be found in [138, 139].

5.4 Multivariate Gaussian Random Effect Mixture Model

The random effect model has been widely studied in the literature [2–4, 131]. It is useful when the applications have patterns that depend on some intrinsic model, but not all of them can be known or measured [140]. The random effect model can model this kind of uncertainty and dependency. Besides this, the random effect model has computation convenience and can be easily fitted into the EM framework [141].

One of the most popular random effect models is multivariate Gaussian random effect mixture model. If we model the $\mathbf{Y}_i$ in Eq. (5.2.1) by a mixture Gaussian distribution and introduce mixing proportions π_c (where $\sum_{c=1}^C \pi_c = 1$), then $\mathbf{Y}_i$ follows the mixture model which can be expressed as

$$\mathbf{Y}_i \sim \sum_{c=1}^C \pi_c N(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \quad (5.4.4)$$

where $\boldsymbol{\mu}_c$ is the mean and $\boldsymbol{\Sigma}_c$ the covariance matrix for the c th Gaussian component.

Furthermore, if the mean profile $\boldsymbol{\mu}_c$ and random effect $\mathbf{b}_i$ are modeled using cubic B-spline functions, and suppose $\bar{\mathbf{B}}_l$ are the basis functions for mean curve profile, $l = 1, \dots, p$, and $\mathbf{B}_k$ are the basis functions for random effect, $k = 1, \dots, q$, the random effect model in (5.2.1) can be rewritten as

$$\mathbf{Y}_i = \sum_{l=1}^p \beta_{cl} \bar{\mathbf{B}}_l + \sum_{k=1}^q \varphi_{ik} \mathbf{B}_k + \boldsymbol{\epsilon}_i, \quad (\boldsymbol{\mu}_c = \sum_{l=1}^p \beta_{cl} \bar{\mathbf{B}}_l, \mathbf{b}_i = \sum_{k=1}^q \varphi_{ik} \mathbf{B}_k) \quad (5.4.5)$$

where $\boldsymbol{\beta}_c = (\beta_{c1}, \dots, \beta_{cp})'$ and $\boldsymbol{\varphi}_i = (\varphi_{i1}, \dots, \varphi_{iq})'$ are the coefficients vectors corresponding to the mean of c th cluster and random effect of i th individual curve,

Figure 5.7: B-spline curves with its control points and basis functions.

respectively. Fig. 5.7 shows the notation relationship for the random effect model of a *cth* mean curve profile.

Therefore, the generalized multivariate Gaussian random effect mixture model based on B-spline curves can be expressed hierarchically as

$$\begin{aligned}
 \mathbf{Y}_i &= \sum_{l=1}^p \beta_{cl} \bar{\mathbf{B}}_l + \sum_{k=1}^q \varphi_{ik} \mathbf{B}_k + \boldsymbol{\epsilon}_i \\
 \mathbf{Y}_i &\sim \sum_{c=1}^C \pi_c N(\bar{\mathbf{B}} \boldsymbol{\beta}_c, \mathbf{B} \boldsymbol{\Lambda} \mathbf{B}' + \sigma^2 \mathbf{I}) \\
 \text{cov}(\boldsymbol{\varphi}_i) &= \boldsymbol{\Lambda}
 \end{aligned} \tag{5.4.6}$$

Usually, the random coefficients vector $\boldsymbol{\varphi}_i$ is assumed to be independent of the cluster indicators in order to reduce the model complexity. $\boldsymbol{\epsilon}_i$ is the uncorrelated measurement errors, and has $E(\boldsymbol{\epsilon}_i) = 0$, $\text{Var}(\boldsymbol{\epsilon}_i) = \sigma^2$.

5.5 Expectation-Maximization (EM) Framework for the Multivariate Gaussian Random Effect Mixture Model

The EM algorithm is a general technique of iteratively finding the maximum-likelihood (ML) estimates of a given probability distribution model. In pattern recognition community, it is an elaborate approach for the incomplete-data problem. It provides an alternative numerical solution when the optimization problem is analytically intractable [142]. It usually involves two steps, called the Expectation step (i.e. E-step) and the Maximization step (i.e. M-step). This is also the reason why this algorithm is called EM algorithm. It uses an iterative scheme during the optimization process. Intuitively, in the E-step, the missing data values are filled with their expectation values based on the initial values of the unknown parameters and the observed data. In the M-step, the unknown parameters are updated by their estimated values by using the expectation values of missing data. This process is performed recursively until convergence.

In the rest of this section, we will give detail derivations of the EM algorithm based on the multivariate Gaussian random effect mixture model discussed in Section 5.4. The derivations are useful for a better understandings of how EM algorithm is solving the incomplete data problem and also provide mathematical preliminaries for the next chapter. Readers are encouraged to read the book [135] to get a full understanding of the EM framework.

5.5.1 The complete data log-likelihood

The generalized multivariate Gaussian random effect mixture model based on B-spline curves is restated here as

$$\begin{aligned} \mathbf{Y}_i &= \sum_{l=1}^p \beta_{cl} \bar{\mathbf{B}}_l + \sum_{k=1}^q \varphi_{ik} \mathbf{B}_k + \boldsymbol{\epsilon}_i \\ \mathbf{Y}_i &\sim \sum_{c=1}^C \pi_c N(\bar{\mathbf{B}}\boldsymbol{\beta}_c, B\boldsymbol{\Lambda}B' + \sigma^2\mathbf{I}) \\ \text{cov}(\boldsymbol{\varphi}_i) &= \boldsymbol{\Lambda} \end{aligned} \quad (5.5.7)$$

The complete data log-likelihood l for this model is

$$\begin{aligned} l &= \sum_{i=1}^n \sum_{c=1}^C I(Z_i = c) \left(\log \pi_c - \frac{1}{2} \log |\boldsymbol{\Lambda}| - \frac{1}{2} \boldsymbol{\varphi}_i' \boldsymbol{\Lambda}^{-1} \boldsymbol{\varphi}_i - \frac{g}{2} \log \sigma^2 \right. \\ &\quad \left. - \frac{1}{2\sigma^2} \|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 \right) \\ &= \sum_{i=1}^n \sum_{c=1}^C I(Z_i = c) \log \pi_c - \frac{n}{2} \log |\boldsymbol{\Lambda}| - \frac{1}{2} \sum_{i=1}^n \sum_{c=1}^C I(Z_i = c) \boldsymbol{\varphi}_i' \boldsymbol{\Lambda}^{-1} \boldsymbol{\varphi}_i \\ &\quad - \frac{ng}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{c=1}^C I(Z_i = c) \|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 \end{aligned} \quad (5.5.8)$$

where $I(Z_i = c)$ is cluster indicator functions (it takes the value of 1 if $\mathbf{Y}_i$ is generated by the c th component), and g is the dimensionality of $\mathbf{Y}_i$.

Detailed Derivations. In this multivariate model, the observed data is $\mathbf{Y}_i$. The cluster indicator vector Z and random coefficients $\boldsymbol{\varphi}_i$ (take note that there are n realizations from this g -dimensional multivariate normal distribution) are missing data. Let $\Theta = (\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_c, \dots, \boldsymbol{\beta}_C)$ be the parameters to be estimated, we have the

complete data likelihood

$$\begin{aligned}
 p(\mathbf{Y}, \boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_n, Z_1, \dots, Z_n | \Theta) &= \prod_{i=1}^n p(\mathbf{Y}_i, \boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_n, Z_1, \dots, Z_n | \Theta) \\
 &= \prod_{i=1}^n p(\mathbf{Y}_i | \boldsymbol{\varphi}_i, Z_i, \Theta) p(\boldsymbol{\varphi}_i, Z_i | \Theta) \\
 &= \prod_{i=1}^n p(\mathbf{Y}_i | \boldsymbol{\varphi}_i, Z_i, \Theta) p(\boldsymbol{\varphi}_i | \Theta) p(Z_i | \Theta) \\
 &= \prod_{i=1}^n \sum_{c=1}^C p(\mathbf{Y}_i | \boldsymbol{\varphi}_i, Z_i = c, \Theta) p(\boldsymbol{\varphi}_i | \Theta) p(Z_i = c | \Theta)
 \end{aligned} \tag{5.5.9}$$

Given the all the unknown parameters and data, for the i th observed data, the incomplete data likelihood is

$$\begin{aligned}
 p(\mathbf{Y}_i | \boldsymbol{\varphi}_i, Z_i = c, \Theta) &= \prod_{j=1}^g \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{Y}_i - \sum_{l=1}^p \beta_{cl} \bar{\mathbf{B}}_l - \sum_{l=1}^q \boldsymbol{\varphi}_{ij} \mathbf{B}_l\|^2\right) \right) \\
 &= \left(\frac{1}{2\pi\sigma^2} \right)^{\frac{g}{2}} \exp\left(\sum_{j=1}^g \left(-\frac{1}{2\sigma^2} \|\mathbf{Y}_i - \sum_{l=1}^p \beta_{cl} \bar{\mathbf{B}}_l - \sum_{l=1}^q \boldsymbol{\varphi}_{ij} \mathbf{B}_l\|^2 \right) \right) \\
 &= \left(\frac{1}{2\pi\sigma^2} \right)^{\frac{g}{2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{j=1}^g \|\mathbf{Y}_i - \sum_{l=1}^p \beta_{cl} \bar{\mathbf{B}}_l - \sum_{l=1}^q \boldsymbol{\varphi}_{ij} \mathbf{B}_l\|^2 \right) \\
 &= \left(\frac{1}{2\pi\sigma^2} \right)^{\frac{g}{2}} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 \right)
 \end{aligned} \tag{5.5.10}$$

where $\bar{\mathbf{B}}$ and $\mathbf{B}$ are $T \times q$ matrices, while $\boldsymbol{\beta}_c$ and $\boldsymbol{\varphi}_i$ are $q \times 1$ vectors.

In order to get the complete data likelihood, we still need to derive $p(\boldsymbol{\varphi}_i | \Theta) p(Z_i | \Theta)$, which is given as the following:

$$\begin{aligned}
 p(\boldsymbol{\varphi}_i | \Theta) &= p(\boldsymbol{\varphi}_i) \\
 &= \frac{1}{(2\pi)^{n/2} |\boldsymbol{\Lambda}|^{1/2}} \exp\left(-\frac{1}{2} \boldsymbol{\varphi}_i' \boldsymbol{\Lambda}^{-1} \boldsymbol{\varphi}_i\right)
 \end{aligned} \tag{5.5.11}$$

and

$$p(Z_i = c | \Theta) = p(Z_i = c) = \pi_c \tag{5.5.12}$$

Therefore, we have the complete data log-likelihood

$$\begin{aligned}
l &= \log(p(\mathbf{Y}, \boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_n, Z_1, \dots, Z_n | \Theta)) \\
&= \sum_{i=1}^n \sum_{c=1}^C \log(p(\mathbf{Y}_i | \boldsymbol{\varphi}_i, Z_i = c, \Theta) p(\boldsymbol{\varphi}_i | \Theta) p(Z_i = c | \Theta)) \\
&= \sum_{i=1}^n \sum_{c=1}^C I(Z_i = c) \log \pi_c - \frac{n}{2} \log |\boldsymbol{\Lambda}| - \frac{1}{2} \sum_{i=1}^n \sum_{c=1}^C I(Z_i = c) \boldsymbol{\varphi}_i' \boldsymbol{\Lambda}^{-1} \boldsymbol{\varphi}_i \\
&\quad - \frac{ng}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{c=1}^C I(Z_i = c) \|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 \quad (5.5.13)
\end{aligned}$$

5.5.2 E-step solution

From the last subsection, we note that the values of $I(Z_i = c)$, and $\boldsymbol{\varphi}_i$ are unobservable. We need to use expectation value to substitute them.

Therefore, for the ML estimation, in E-Step, we need to calculate

$$E[I(Z_i = c) | \mathbf{Y}] = Pr(Z_i = c | \mathbf{Y}) \quad (5.5.14)$$

$$E[I(Z_i = c) \boldsymbol{\varphi}_i' \boldsymbol{\Lambda}^{-1} \boldsymbol{\varphi}_i | \mathbf{Y}] = \pi_{c|i} E(\boldsymbol{\varphi}_i' \boldsymbol{\Lambda}^{-1} \boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c) \quad (5.5.15)$$

$$\begin{aligned}
E[I(Z_i = c) \|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 | \mathbf{Y}] &= \|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c\|^2 - 2(\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c)' \mathbf{B} E(\boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c) \\
&\quad + E(\boldsymbol{\varphi}_i' \mathbf{B}' \mathbf{B} \boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c) \quad (5.5.16)
\end{aligned}$$

with

$$\begin{aligned}
E(\boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c) &= \boldsymbol{\Lambda} \mathbf{B}' (\mathbf{B} \boldsymbol{\Lambda} \mathbf{B}' + \sigma^2 I)^{-1} (\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c) \\
E(\boldsymbol{\varphi}_i \boldsymbol{\varphi}_i' | \mathbf{Y}, Z_i = c) &= Var(\boldsymbol{\varphi}_i \boldsymbol{\varphi}_i' | \mathbf{Y}, Z_i = c) + [E(\boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c)][E(\boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c)]'
\end{aligned}$$

The justification for the above calculation are given below. For the first conditional expectation, if X is the indicator function of an event S , then $E(X|A)$ is just the conditional probability $P(S|A)$. Therefore we have

$$E[I(Z_i = c) | \mathbf{Y}] = Pr(Z_i = c | \mathbf{Y}) = \pi_{c|i} \quad (5.5.17)$$

This probability is the *posterior* clustering probability given the data and known parameters.

For the second conditional expectation, using marginalization theory, we have

$$\sum_{c=1}^C \pi_{c|i} E[I(Z_i = k) \boldsymbol{\varphi}_i' \boldsymbol{\Lambda}^{-1} \boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c] = E[I(Z_i = k) \boldsymbol{\varphi}_i' \boldsymbol{\Lambda}^{-1} \boldsymbol{\varphi}_i | \mathbf{Y}] \quad (5.5.18)$$

Therefore

$$\begin{aligned} E[I(Z_i = k) \boldsymbol{\varphi}_i' \boldsymbol{\Lambda}^{-1} \boldsymbol{\varphi}_i | \mathbf{Y}] &= \sum_{c=1}^C \pi_{c|i} E[I(Z_i = k) \boldsymbol{\varphi}_i' \boldsymbol{\Lambda}^{-1} \boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c] \\ &= \pi_{k|i} E[\boldsymbol{\varphi}_i' \boldsymbol{\Lambda}^{-1} \boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = k] \\ &= \pi_{c|i} E[\boldsymbol{\varphi}_i' \boldsymbol{\Lambda}^{-1} \boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c] \end{aligned} \quad (5.5.19)$$

Similarly, the third condition expectation is

$$\begin{aligned} &E[I(Z_i = c) \|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 | \mathbf{Y}] \\ &= \pi_{c|i} E(\|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 | \mathbf{Y}, Z_i = c) \\ &= \pi_{c|i} E[(\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i)'(\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i) | \mathbf{Y}, Z_i = c] \\ &= \pi_{c|i} E[\|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c\|^2 - (\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c)' \mathbf{B}\boldsymbol{\varphi}_i - \boldsymbol{\varphi}_i' \mathbf{B}'(\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c) \\ &\quad + \boldsymbol{\varphi}_i' \mathbf{B}' \mathbf{B} \boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c] \\ &\quad \left\{ \because E[X'] = E[X]' \right\} \\ &= \pi_{c|i} (\|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c\|^2 - 2(\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c)' \mathbf{B} E(\boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c) \\ &\quad + E(\boldsymbol{\varphi}_i' \mathbf{B}' \mathbf{B} \boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c)) \end{aligned} \quad (5.5.20)$$

Note that for multivariate Gaussian distribution, if $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are partitioned as follow

$$\begin{aligned} \boldsymbol{\mu} &= \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix} \text{ with sizes } \begin{bmatrix} q \times 1 \\ (N - q) \times 1 \end{bmatrix} \\ \boldsymbol{\Sigma} &= \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \text{ with sizes } \begin{bmatrix} q \times q & q \times (N - q) \\ (N - q) \times q & (N - q) \times (N - q) \end{bmatrix} \end{aligned} \quad (5.5.21)$$

then the distribution of $\mathbf{x}_1$ conditional on $\mathbf{x}_2 = a$ is multivariate normal $(\mathbf{X}_1 | \mathbf{X}_2 = a) \sim N(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}})$ where

$$\bar{\boldsymbol{\mu}} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_{22}^{-1} (a - \boldsymbol{\mu}_2) \quad (5.5.22)$$

and covariance matrix

$$\bar{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_{22}^{-1} \boldsymbol{\Sigma}_{21} \quad (5.5.23)$$

Further note that

$$\begin{bmatrix} \boldsymbol{\varphi}_i \\ \mathbf{Y}_i \end{bmatrix} | \{Z_i = c\} \sim N\left(\begin{bmatrix} 0 \\ \bar{\mathbf{B}}\boldsymbol{\beta}_c \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Lambda} & \boldsymbol{\Lambda}\mathbf{B}' \\ \mathbf{B}\boldsymbol{\Lambda} & \mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2 I \end{bmatrix}\right). \quad (5.5.24)$$

Therefore,

$$\boldsymbol{\varphi}_i | \{\mathbf{Y}, Z_i = c\} \sim N(\boldsymbol{\Lambda}\mathbf{B}'(\mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2 I)^{-1}(\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c), \boldsymbol{\Lambda} - \boldsymbol{\Lambda}\mathbf{B}'(\mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2 I)^{-1}\mathbf{B}\boldsymbol{\Lambda}). \quad (5.5.25)$$

So,

$$\begin{aligned} E(\boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c) &= \boldsymbol{\Lambda}\mathbf{B}'(\mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2 I)^{-1}(\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c) \\ \text{Var}(\boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c) &= \boldsymbol{\Lambda} - \boldsymbol{\Lambda}\mathbf{B}'(\mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2 I)^{-1}\mathbf{B}\boldsymbol{\Lambda} \end{aligned} \quad (5.5.26)$$

and

$$E(\boldsymbol{\varphi}_i \boldsymbol{\varphi}_i' | \mathbf{Y}, Z_i = c) = \text{Var}(\boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c) + [E(\boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c)][E(\boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c)]' \quad (5.5.27)$$

According to [2], $\boldsymbol{\varphi}_i' \mathbf{B}' \mathbf{B} \boldsymbol{\varphi}_i$ is a standard quadratic form, $E(\boldsymbol{\varphi}_i' \mathbf{B}' \mathbf{B} \boldsymbol{\varphi}_i | \mathbf{Y}, Z_i = c)$ can be computed easily via $\mathbf{B}' \mathbf{B}$ and $E(\boldsymbol{\varphi}_i \boldsymbol{\varphi}_i' | \mathbf{Y}, Z_i = c)$. Thus, the proof is completed.

5.5.3 M-step solution

Remember that the M-step is used to estimate all unknown parameters, i.e. $\pi_{c|i}$, $\bar{\boldsymbol{\Lambda}}$, $\bar{\boldsymbol{\beta}}_c$ and σ^2 . We shall first look into the $\pi_{c|i}$.

By definition, we have

$$\begin{aligned} \pi_{c|i} &= \frac{\Pr(Z_i = c) \Pr(\mathbf{Y}_i | Z_i = c)}{\Pr(\mathbf{Y}_i)} \\ &= \frac{\bar{\pi}_c \exp\left[-\frac{1}{2}(\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c)'(\mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2 I)^{-1}(\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c)\right]}{\sum_{c=1}^C \bar{\pi}_c \exp\left[-\frac{1}{2}(\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c)'(\mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2 I)^{-1}(\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c)\right]} \end{aligned} \quad (5.5.28)$$

where

$$\bar{\pi}_c = \frac{1}{n} \sum_{i=1}^n \pi_{c|i} \quad (5.5.29)$$

Second, the terms which involve the Λ parameters are

$$-\frac{n}{2} \log |\Lambda| - \frac{1}{2} \sum_{i=1}^n \sum_{c=1}^C \pi_{c|i} E(\varphi_i' \Lambda^{-1} \varphi_i | \mathbf{Y}_i, Z_i = c) = -\frac{1}{2} E \left[\sum_{i=1}^n (\log |\Lambda| + \varphi_i' \Lambda^{-1} \varphi_i) | \mathbf{Y}_i \right]. \quad (5.5.30)$$

According to matrix calculus

$$\begin{aligned} \nabla_{\mathbf{X}} a^T \mathbf{X}^{-1} b &= -\mathbf{X}^{-T} a b^T \mathbf{X}^{-T} \\ \nabla_{\mathbf{X}} \log |\mathbf{X}| &= \mathbf{X}^{-T} \end{aligned} \quad (5.5.31)$$

and Λ is symmetric, so the Λ which maximizes this term is given by

$$\bar{\Lambda} = E \left(\frac{1}{n} \sum_{i=1}^n \varphi_i \varphi_i' | \mathbf{Y}_i \right) = \frac{1}{n} \sum_{i=1}^n \sum_{c=1}^C \pi_{c|i} E(\varphi_i \varphi_i' | \mathbf{Y}_i, Z_i = c) \quad (5.5.32)$$

Third, there is only one term that is related to parameter β_c

$$-\frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{c=1}^C \pi_{c|i} E(\|\mathbf{Y}_i - \bar{\mathbf{B}}\beta_c - \mathbf{B}\varphi_i\|^2 | \mathbf{Y}_i, Z_i = c). \quad (5.5.33)$$

From previous sections, we have

$$\begin{aligned} E(\|\mathbf{Y}_i - \bar{\mathbf{B}}\beta_c - \mathbf{B}\varphi_i\|^2 | \mathbf{Y}_i, Z_i = c) &= \|\mathbf{Y}_i - \bar{\mathbf{B}}\beta_c\|^2 - 2(\mathbf{Y}_i - \bar{\mathbf{B}}\beta_c)' \mathbf{B} E(\varphi_i | \mathbf{Y}_i, Z_i = c) \\ &\quad + E(\varphi_i' \mathbf{B}' \mathbf{B} \varphi_i | \mathbf{Y}_i, Z_i = c) \end{aligned} \quad (5.5.34)$$

Taking derivative with respect to β_c , we have

$$\begin{aligned} &\frac{\partial}{\partial \beta_c} \left(-\frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{c=1}^C \pi_{c|i} E(\|\mathbf{Y}_i - \bar{\mathbf{B}}\beta_c - \mathbf{B}\varphi_i\|^2 | \mathbf{Y}_i, Z_i = c) \right) \\ &= -\frac{1}{2\sigma^2} \sum_{i=1}^n \frac{\partial}{\partial \beta_c} \left(\pi_{c|i} E(\|\mathbf{Y}_i - \bar{\mathbf{B}}\beta_c - \mathbf{B}\varphi_i\|^2 | \mathbf{Y}_i, Z_i = c) \right) \\ &= -\frac{1}{2\sigma^2} \sum_{i=1}^n \left(2\bar{\mathbf{B}}' \bar{\mathbf{B}}\beta_c - 2\bar{\mathbf{B}}' \mathbf{Y}_i + 2\bar{\mathbf{B}}' \mathbf{B} E(\varphi_i | \mathbf{Y}_i, Z_i = c) \right) \\ &= 0 \end{aligned} \quad (5.5.35)$$

So the update for β_c is

$$\bar{\beta}_c = \left(\sum_{i=1}^n \pi_{c|i} \bar{\mathbf{B}}' \bar{\mathbf{B}} \right)^{-1} \left(\sum_{i=1}^n \pi_{c|i} \bar{\mathbf{B}}' (\mathbf{Y}_i - \mathbf{B} E(\varphi_i | \mathbf{Y}_i, Z_i = c)) \right) \quad (5.5.36)$$

The last term is the one that involves σ^2 , and is given by

$$-\frac{ng}{2} \log \sigma^2 - \frac{1}{\sigma^2} \sum_{i=1}^n \sum_{c=1}^C \pi_{c|i} E(\|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 | \mathbf{Y}_i, Z_i = c)$$

Taking the derivative with respect to σ^2 , then the equation for updating σ^2 is given by

$$\bar{\sigma}^2 = \frac{1}{ng} \sum_{i=1}^n \sum_{c=1}^C \pi_{c|i} E(\|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 | \mathbf{Y}_i, Z_i = c) \quad (5.5.37)$$

5.6 Conclusions

In this chapter, we have briefly reviewed some feature-based representation techniques for image understanding. Within the data clustering based applications, we found that B-spline curve representation is one of the most popular methods widely used in surveillance, medical image diagnosis and functional data analysis. The B-spline based data clustering analysis is usually modeled by random effect model and optimized by the elaborate EM algorithm.

Through the derivation of the iterative EM solutions for the multivariate Gaussian random effect mixture model, we notice that the Gaussian mixture model lacks robustness with respect to outliers as there is no way for Gaussian distribution to ignore the negative impact from irregular data points [143]. In the next chapter, we will propose a robust random effect mixture model, called a robust multivariate t -Distribution Model, to overcome this problem.

Chapter 6

Robust Curve Clustering based on Multivariate t -Distribution Model

6.1 Introduction

In the previous chapter, we have introduced a commonly used data clustering approach for image analysis with curve representations, i.e. the multivariate Gaussian random effect mixture model. This model is also ubiquitous in many other disciplines like data mining, machine learning and others. The power of this model is obvious. However, its lack of robustness with respect to outliers problem is well known as well, because the maximum likelihood (ML) solutions for this model via Expectation Maximization (EM) algorithm is equivalent to finding the least-square solutions [144].

Since the models based on Gaussian distribution are not robust to real data sets which usually have heavy-tailed distributions [143], a straightforward solution is to find an alternative wider-tailed distribution to replace the Gaussian distribution. Many research work have been done of adopting the Student's t -distribution [145] to provide robustness in their models [136, 141, 143, 144, 146], because 1) the t -distribution has longer tail than the Gaussian distribution (as shown in Fig. 6.1), hence it can address some observations that are lying relatively far away from the mean; 2) the additional parameter ν (called degree of freedom) in the t -distribution model provides a flexible robust accommodation for outliers as it can down-weighted

Figure 6.1: Plot of t -distribution densities with degree of freedom for 0.1, 3, 10 compared to Gaussian distribution density (Magenta).

the atypical observations in the calculation of its parameters and yield better estimates [147]; and 3) it contains a wider class of elliptically symmetric distributions including Gaussian distribution (ν tends to infinity) [141], therefore the variability of the real data sets can be well taken care of.

In this chapter, we will introduce a robust random effect mixture model for robust curve clustering based on multivariate t -distribution. Compared to the general multivariate Gaussian model, the proposed model has advantages in robust estimation of means, regression coefficients, and variance-covariance matrices in multivariate linear models, even in cases with missing data [146].

6.2 A Multivariate t -Distribution Model for Robust Curve Clustering

6.2.1 Robust statistical model using multivariate t -Distribution

We rewrite the the generalized multivariate Gaussian random effect mixture model based on B-spline curves used in the previous chapter as

$$\begin{aligned} \mathbf{Y}_i &= \sum_{l=1}^p \beta_{cl} \bar{\mathbf{B}}_l + \sum_{k=1}^q \varphi_{ik} \mathbf{B}_k + \boldsymbol{\epsilon}_i \\ \mathbf{Y}_i &\sim \sum_{c=1}^C \pi_c N(\bar{\mathbf{B}} \boldsymbol{\beta}_c, \mathbf{B} \boldsymbol{\Lambda} \mathbf{B}' + \sigma^2 \mathbf{I}) \\ \text{cov}(\boldsymbol{\varphi}_i) &= \boldsymbol{\Lambda} \end{aligned} \quad (6.2.1)$$

In order to add robustness to the above random effect mixture model, we assume $\mathbf{Y}_i$ now follows a wider tail probability density function (p.d.f.), the t -Distribution, instead of the Gaussian distribution. This can also be viewed as introducing a new random variable $\mathbf{U}(u_1, \dots, u_n)$ with Gamma distribution to scale the normal distribution as mentioned in Peel and McLachlan's paper [141], i.e.

$$\mathbf{U} \sim \text{Gamma}\left(\frac{1}{2}\nu, \frac{1}{2}\nu\right) \quad (6.2.2)$$

where ν is the degree of freedom.

With new assumption, the new C-component multivariate mixture model for g -dimensional random sample $\mathbf{Y}_i$ can be expressed using the following model

$$\mathbf{Y}_i \sim \sum_{c=1}^C \pi_c t_g(\mathbf{Y}_i; \bar{\mathbf{B}} \boldsymbol{\beta}_c, \mathbf{B} \boldsymbol{\Lambda} \mathbf{B}' + \sigma^2 \mathbf{I}, \nu) \quad (6.2.3)$$

where $t_g(\mathbf{Y}_i; \bar{\mathbf{B}} \boldsymbol{\beta}_c, \mathbf{B} \boldsymbol{\Lambda} \mathbf{B}' + \sigma^2 \mathbf{I}, \nu)$ is the multivariate t -Distribution density function.

The g -dimensional multivariate t -Distribution density function is given by

$$t_g(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \frac{\Gamma(\frac{\nu+g}{2}) |\boldsymbol{\Sigma}|^{-1/2}}{(\pi\nu)^{\frac{1}{2}} \Gamma(\frac{\nu}{2}) \{1 + \delta(\mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\Sigma})/\nu\}^{\frac{1}{2}(\nu+g)}} \quad (6.2.4)$$

where $\delta(\mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (\mathbf{y} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{y} - \boldsymbol{\mu})$ is the Mahalanobis distance between $\mathbf{y}$ and the center $\boldsymbol{\mu}$.

By introducing the unobserved cluster indicator functions $z_{ci} = I(Z_i = c)$ (it takes value 1 if $\mathbf{Y}_i$ is generated by the c th component $t_g(\mathbf{Y}_i; \bar{\mathbf{B}}\boldsymbol{\beta}_c, \mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2\mathbf{I}, \nu)$), we can express our proposed robust statistical model in the following hierarchical models:

$$\begin{aligned} \mathbf{Y}_i &= \sum_{l=1}^p \beta_{cl} \bar{\mathbf{B}}_l + \sum_{k=1}^q \varphi_{ik} \mathbf{B}_k + \boldsymbol{\epsilon}_i \\ \mathbf{Y}_i &\sim \sum_{c=1}^C \pi_c t_g(\mathbf{Y}_i; \bar{\mathbf{B}}\boldsymbol{\beta}_c, \mathbf{B}\boldsymbol{\Lambda}\mathbf{B}', \nu) \\ \begin{bmatrix} \mathbf{Y}_i \\ \boldsymbol{\varphi}_i \end{bmatrix} | z_{ci} &\sim t_g \left(\begin{bmatrix} \bar{\mathbf{B}}\boldsymbol{\beta}_c \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2\mathbf{I} & \mathbf{B}\boldsymbol{\Lambda} \\ \mathbf{B}'\boldsymbol{\Lambda} & \boldsymbol{\Lambda} \end{bmatrix}, \nu \right) \\ u_i | z_{ci} &\sim \text{Gamma}(\frac{\nu}{2}, \frac{\nu}{2}) \end{aligned} \quad (6.2.5)$$

The u_i is the hidden robust estimator influenced by the Mahalanobis distance $\delta(\mathbf{Y}_i, \boldsymbol{\mu}_c, \boldsymbol{\Sigma})$ and the degree of freedom ν . As we know, for the t -Distribution, if $\nu > 1$, $\boldsymbol{\mu}_c$ is the mean of the data samples, and if $\nu > 2$, $\nu/(\nu - 2)\boldsymbol{\Sigma}$ is the covariance matrix. For the case $\nu \rightarrow \infty$, the model will reduce to a multivariate Gaussian distribution with $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}$. Therefore, by tuning the degree of freedom ν , we can tune the model's robustness to a particular data set or a particular application [143].

6.2.2 The complete-data log likelihood

Before we consider the ML solutions for this proposed model, it is always favorable to factorize the complete-data log likelihood into several terms. Suppose that $\mathbf{Y}_i$ denote an observed data samples from the c th t -Distribution $t_g(\bar{\mathbf{B}}\boldsymbol{\beta}_c, \boldsymbol{\Sigma}, \nu)$. Then we can denote

$$\mathbf{Y} = (\mathbf{Y}'_1, \dots, \mathbf{Y}'_i, \dots, \mathbf{Y}'_n)', (\mathbf{Y}_i = \sum_{l=1}^p \beta_{cl} \bar{\mathbf{B}}_l + \sum_{k=1}^q \varphi_{ik} \mathbf{B}_k + \boldsymbol{\epsilon}_i) \quad (6.2.6)$$

as the observed random vector and use

$$f(\mathbf{Y}; \boldsymbol{\Theta}) = \sum_{c=1}^C \sum_{i=1}^n \pi_c t_g(\mathbf{Y}_i; \bar{\mathbf{B}}\boldsymbol{\beta}_c, \boldsymbol{\Sigma}, \nu) \quad (6.2.7)$$

to denote the p.d.f of this random vector $\mathbf{Y}$, where $\pi_c, \dots, \pi_C$ are the mixing proportions and $\boldsymbol{\Theta}$ is the vector containing the unknown parameters, i.e. $\boldsymbol{\Theta} = (\pi_1, \dots, \pi_{C-1}, \boldsymbol{\theta}'_1, \dots, \boldsymbol{\theta}'_C, \sigma^2)$, with $\boldsymbol{\theta}_c$ containing the elements of $\bar{\mathbf{B}}\boldsymbol{\beta}_c$ and the elements of $\boldsymbol{\Sigma}$, ($c = 1, \dots, C$).

Under this mixed model, let $\mathbf{x}_{obs} = \mathbf{Y}$ denote the observed data and $\mathbf{x}_{mis} = (\mathbf{U}', \mathbf{Z}', \boldsymbol{\varphi}')'$ denote the missing data. The complete data is $\mathbf{x}_{com} = (\mathbf{x}_{obs}', \mathbf{x}_{mis}')$, and where $\mathbf{U} = (u_1, \dots, u_n)'$ is the unobserved robust scale factor of each observed sample, $\mathbf{Z} = (Z_1, \dots, Z_n)'$ is the *prior* component label vector (where $z_{ci} = 1$ if *i*th sample is generated by the *c*th component), and $\boldsymbol{\varphi} = (\boldsymbol{\varphi}_1', \dots, \boldsymbol{\varphi}_n')'$ is the unknown random effect coefficients. With these assumptions, the complete-data log likelihood $L_c(\mathbf{x}_{com}|\boldsymbol{\Theta})$ can be factorized into the product of marginal densities of the $\mathbf{Z}$, the conditional densities of $\mathbf{U}$ given the $\mathbf{Z}$, the conditional densities of the $\boldsymbol{\varphi}$ given the $\mathbf{U}$ and the $\mathbf{Z}$, and the conditional densities of the $\mathbf{Y}$ given the $\boldsymbol{\varphi}$, the $\mathbf{U}$ and the $\mathbf{Z}$. Therefore, we can express the complete-data log likelihood as the following equation:

$$L_c(\boldsymbol{\Theta}|\mathbf{Y}, \boldsymbol{\varphi}, \mathbf{U}, \mathbf{Z}) = L_1(\mathbf{Z}) + L_2(\mathbf{U}|\mathbf{Z}) + L_3(\mathbf{U}, \mathbf{Z}|\boldsymbol{\varphi}) + L_4(\mathbf{Y}|\boldsymbol{\varphi}, \mathbf{U}, \mathbf{Z}) \quad (6.2.8)$$

The marginal densities for the $\mathbf{Z}$ is straightforward and it is

$$L_1(\mathbf{Z}) = \sum_{c=1}^C \sum_{i=1}^n z_{ci} \log \pi_c. \quad (6.2.9)$$

For the conditional densities for the $\mathbf{U}$ given the $\mathbf{Z}$, $u_1, \dots, u_i$ are independently distributed according to

$$u_i|Z_i = c \sim \Gamma\left(\frac{1}{2}\nu, \frac{1}{2}\nu\right). \quad (6.2.10)$$

Hence, we have

$$L_2(\mathbf{U}|\mathbf{Z}) = \sum_{c=1}^C \sum_{i=1}^n z_{ci} \left\{ -\log \Gamma\left(\frac{1}{2}\nu\right) + \frac{1}{2}\nu \log\left(\frac{1}{2}\nu\right) + \frac{1}{2}\nu(\log u_i - u_i) - \log u_i \right\}. \quad (6.2.11)$$

Given u_i and $Z_i = c$, $\boldsymbol{\varphi}_i$ follows a normal distribution with zero mean and covariance matrix $\boldsymbol{\Lambda}/u_i$, i.e.

$$\boldsymbol{\varphi}_i|u_i, Z_i = c \sim N(0, \boldsymbol{\Lambda}/u_i). \quad (6.2.12)$$

Hence,

$$L_3(\mathbf{U}, \mathbf{Z}|\boldsymbol{\varphi}) = \sum_{c=1}^C \sum_{i=1}^n z_{ci} \left(-\frac{1}{2} \log\left(\frac{|\boldsymbol{\Lambda}|}{u_i}\right) - \frac{1}{2} \text{trace}(\boldsymbol{\Lambda}^{-1} u_i \boldsymbol{\varphi}_i \boldsymbol{\varphi}_i') \right). \quad (6.2.13)$$

Finally, given $\boldsymbol{\beta}_c$, $\boldsymbol{\varphi}_i$, u_i and the $Z_i = c$, the conditional density of the Y_i is also a normal distribution

$$Y_i | \boldsymbol{\beta}_c, \boldsymbol{\varphi}_i, u_i, Z_i = c \sim N(0, \sigma^2 / u_i), \quad (6.2.14)$$

so we have

$$\begin{aligned} L_4(\mathbf{Y}|\boldsymbol{\varphi}, \mathbf{U}, \mathbf{Z}) &= \sum_{i=1}^n \sum_{c=1}^C z_{ci} \left(-\frac{g}{2} \log(\sigma^2 / u_i) - \frac{u_i}{2\sigma^2} \|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 \right) \\ &= -\frac{ng}{2} \log \sigma^2 + \sum_{i=1}^n \sum_{c=1}^C z_{ci} \left(\frac{g}{2} \log u_i - \frac{u_i}{2\sigma^2} \|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 \right). \end{aligned} \quad (6.2.15)$$

6.2.3 ML solutions using the EM algorithm

In this section, we consider the ML estimation of the parameter in the above multivariate t -Distribution based random effect model (6.2.6). The p.d.f. for model (6.2.6) with unknown mixing proportions $\pi_1, \dots, \pi_C$ is

$$f(\mathbf{Y}_i; \boldsymbol{\Theta}) = \sum_{c=1}^C \pi_c t_g(\mathbf{Y}_i; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}, \nu) \quad (6.2.16)$$

where

$$\boldsymbol{\Theta} = (\pi_1, \dots, \pi_{C-1}, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_C, \boldsymbol{\Sigma})$$

$$\boldsymbol{\mu}_c = \bar{\mathbf{B}}\boldsymbol{\beta}_c$$

$$\boldsymbol{\Sigma} = \mathbf{B}\mathbf{A}\mathbf{B}' + \sigma^2 \mathbf{I}$$

In the EM framework, if we consider $\boldsymbol{\varphi}_i, \mathbf{U} = (u_1, \dots, u_n)'$, $\mathbf{Z} = (z_{c1}, \dots, z_{cn})'$ as missing data, the complete-data vector $\boldsymbol{\chi}$ can be expressed as

$$\begin{aligned} \boldsymbol{\chi} &= (\boldsymbol{\chi}'_{obs}, \boldsymbol{\chi}'_{mis})' \\ \boldsymbol{\chi}_{obs} &= (\mathbf{Y}_{1j}^T, \dots, \mathbf{Y}_{nj}^T)' \\ \boldsymbol{\chi}_{mis} &= (\mathbf{Z}', \mathbf{U}', \boldsymbol{\varphi}'_1, \dots, \boldsymbol{\varphi}'_n)' \end{aligned} \quad (6.2.17)$$

Hence, the complete data log-likelihood, which is derived in the previous sub-section, can be factorized into the following equation

$$L_c(\Theta|\mathbf{Y}, \boldsymbol{\varphi}, \mathbf{U}, \mathbf{Z}) = L_1(\mathbf{Z}) + L_2(\mathbf{U}|\mathbf{Z}) + L_3(\mathbf{U}, \mathbf{Z}|\boldsymbol{\varphi}) + L_4(\mathbf{Y}|\boldsymbol{\varphi}, \mathbf{U}, \mathbf{Z}) \quad (6.2.18)$$

where

$$\begin{aligned} L_1(\mathbf{Z}) &= \sum_{c=1}^C \sum_{i=1}^n z_{ci} \log \pi_c \\ L_2(\mathbf{U}|\mathbf{Z}) &= \sum_{c=1}^C \sum_{i=1}^n z_{ci} \left\{ -\log \Gamma\left(\frac{1}{2}\nu\right) + \frac{1}{2}\nu \log\left(\frac{1}{2}\nu\right) + \frac{1}{2}\nu(\log u_i - u_i) - \log u_i \right\} \\ L_3(\mathbf{U}, \mathbf{Z}|\boldsymbol{\varphi}) &= \sum_{c=1}^C \sum_{i=1}^n z_{ci} \left(-\frac{1}{2} \log\left(\frac{|\boldsymbol{\Lambda}|}{u_i}\right) - \frac{1}{2} \text{trace}(\boldsymbol{\Lambda}^{-1} u_i \boldsymbol{\varphi}_i \boldsymbol{\varphi}_i') \right) \\ L_4(\mathbf{Y}|\boldsymbol{\varphi}, \mathbf{U}, \mathbf{Z}) &= -\frac{ng}{2} \log \sigma^2 + \sum_{c=1}^C \sum_{i=1}^n z_{ci} \left(\frac{g}{2} \log u_i - \frac{u_i}{2\sigma^2} \|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 \right) \end{aligned} \quad (6.2.19)$$

E step

From the complete-data log likelihood equations, we note that the values of z_{ci}, u_i and $\boldsymbol{\varphi}_i$ are unobservable. Therefore, we have to use the conditional expectation values instead. For the ML estimation, in the E step we need to calculate

$$\begin{aligned} E(z_{ci}|\mathbf{Y}_i) \\ E(u_i|\mathbf{Y}_i, z_{ci}) \\ E(\log u_i|\mathbf{Y}_i, z_{ci}) \\ E(\boldsymbol{\varphi}_i \boldsymbol{\varphi}_i'|\mathbf{Y}_i, z_{ci}, u_i) \\ E(\|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2|\mathbf{Y}_i, z_{ci}, u_i) \end{aligned} \quad (6.2.20)$$

For the first conditional expectation, we use the property that if X is the indicator function of an event S , then $E(X|A)$ is just the conditional probability $P(S|A)$. Therefore we have

$$\hat{z}_{ci} = E(z_{ci}|\mathbf{Y}_i) = Pr(Z_i = c|\mathbf{Y}_i) = \frac{\pi_c t_g(\mathbf{Y}_i; \bar{\mathbf{B}}\boldsymbol{\beta}_c, \boldsymbol{\Sigma}, \nu)}{\sum_{c=1}^C \pi_c t_g(\mathbf{Y}_i; \bar{\mathbf{B}}\boldsymbol{\beta}_c, \boldsymbol{\Sigma}, \nu)}. \quad (6.2.21)$$

Given the observed data $\mathbf{Y}_i$ and class indicator value z_{ci} , u_i is distributed as gamma distribution. According to Bayes theorem and conditional p.d.f. definition, i.e.

$$\begin{aligned} p(u_i|\mathbf{Y}_i, Z_i = c) &= \frac{p(\mathbf{Y}_i|u_i, z_{ci})p(u_i|Z_i = c)}{p(\mathbf{Y}_i|z_{ci})} \\ \mathbf{Y}_i|u_i, Z_i = c &\sim N(\bar{\mathbf{B}}\boldsymbol{\beta}_c, \frac{\boldsymbol{\Sigma}}{u_i}) \\ u_i|Z_i = c &\sim \Gamma(\frac{1}{2}\nu, \frac{1}{2}\nu) \\ \mathbf{Y}_i|Z_i = c &\sim t(\mathbf{Y}_i; \bar{\mathbf{B}}\boldsymbol{\beta}_c, \boldsymbol{\Sigma}, \nu) \end{aligned} \quad (6.2.22)$$

where $\boldsymbol{\Sigma} = \mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2\mathbf{I}$.

Thus,

$$p(u_i|\mathbf{Y}_i, z_{ci}) = \Gamma(\frac{\nu + g}{2}, \frac{\nu + \delta(\mathbf{Y}_i, \bar{\mathbf{B}}\boldsymbol{\beta}_c; \boldsymbol{\Sigma})}{2}) \quad (6.2.23)$$

Hence,

$$\hat{u}_i = E(u_i|\mathbf{Y}_i, z_{ci}) = \frac{\nu + g}{\nu + \delta(\mathbf{Y}_i, \bar{\mathbf{B}}\boldsymbol{\beta}_c; \mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2\mathbf{I})} \quad (6.2.24)$$

To calculate the conditional $\log u_i$ expectation, we need to use the property that if a random variable R has a Gamma (α, β) distribution, then

$$E(\log R) = \Psi(\alpha) - \log \beta \quad (6.2.25)$$

where

$$\Psi(s) = \frac{\Gamma'(s)}{\Gamma(s)} \quad (6.2.26)$$

is the Digamma function [141]. Applying this formulation, it follows that

$$\begin{aligned} E(\log u_i|\mathbf{Y}_i, z_{ci}) &= \Psi(\frac{\nu_c + g}{2}) - \log \left[\frac{1}{2} \{ \nu + \delta(\mathbf{Y}_i, \bar{\mathbf{B}}\boldsymbol{\beta}_c; \boldsymbol{\Sigma}) \} \right] \\ &= \log E(u_i|\mathbf{Y}_i, z_{ci}) + \left\{ \Psi(\frac{\nu + g}{2}) - \log(\frac{\nu + g}{2}) \right\} \end{aligned} \quad (6.2.27)$$

In order to calculate the last two expectations, from (6.2.6) we have

$$\begin{bmatrix} \mathbf{Y}_i \\ \boldsymbol{\varphi}_{ik} \end{bmatrix} | \{z_{ci}\} \sim t \left(\begin{bmatrix} \bar{\mathbf{B}}\boldsymbol{\beta}_c \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2\mathbf{I} & \mathbf{B}\boldsymbol{\Lambda} \\ \mathbf{B}'\boldsymbol{\Lambda} & \boldsymbol{\Lambda} \end{bmatrix}, \nu \right) \quad (6.2.28)$$

Therefore,

$$\begin{bmatrix} \mathbf{Y}_i \\ \boldsymbol{\varphi}_{ik} \end{bmatrix} | \{z_{ci}, u_i\} \sim N \left(\begin{bmatrix} \bar{\mathbf{B}}\boldsymbol{\beta}_c \\ 0 \end{bmatrix}, \frac{1}{u_i} \begin{bmatrix} \mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2\mathbf{I} & \mathbf{B}\boldsymbol{\Lambda} \\ \mathbf{B}'\boldsymbol{\Lambda} & \boldsymbol{\Lambda} \end{bmatrix} \right) \quad (6.2.29)$$

For multivariate Gaussian distribution, we have

$$\boldsymbol{\varphi}_i | \{\mathbf{Y}_i, z_{ci}, u_i\} \sim N \left(\boldsymbol{\Lambda}\mathbf{B}'(\mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2\mathbf{I})^{-1}(\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c, \frac{1}{u_i}(\boldsymbol{\Lambda} - \boldsymbol{\Lambda}\mathbf{B}'(\mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2\mathbf{I})^{-1}\mathbf{B}\boldsymbol{\Lambda})) \right). \quad (6.2.30)$$

So,

$$\begin{aligned} E(\boldsymbol{\varphi}_i | \mathbf{Y}_i, z_{ci}, u_i) &= \boldsymbol{\Lambda}\mathbf{B}'(\mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2\mathbf{I})^{-1}(\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c) \\ \text{Var}(\boldsymbol{\varphi}_i | \mathbf{Y}_i, z_{ci}, u_i) &= \frac{1}{u_i}(\boldsymbol{\Lambda} - \boldsymbol{\Lambda}\mathbf{B}'(\mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2\mathbf{I})^{-1}\mathbf{B}\boldsymbol{\Lambda}) \end{aligned} \quad (6.2.31)$$

and

$$E(\boldsymbol{\varphi}_i \boldsymbol{\varphi}_i' | \mathbf{Y}_i, z_{ci}, u_i) = \text{Var}(\boldsymbol{\varphi}_i | \mathbf{Y}_i, z_{ci}, u_i) + [E(\boldsymbol{\varphi}_i | \mathbf{Y}_i, z_{ci}, u_i)] [E(\boldsymbol{\varphi}_i | \mathbf{Y}_i, z_{ci}, u_i)]' \quad (6.2.32)$$

Finally, the last expectation is

$$\begin{aligned} &E(\|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 | \mathbf{Y}_i, z_{ci}, u_i) \\ &= \|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c\|^2 - 2(\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c)' \mathbf{B} E(\boldsymbol{\varphi}_i | \mathbf{Y}_i, z_{ci}, u_i) + E(\boldsymbol{\varphi}_i' \mathbf{B}' \mathbf{B} \boldsymbol{\varphi}_i | \mathbf{Y}_i, z_{ci}, u_i) \end{aligned} \quad (6.2.33)$$

Since $\boldsymbol{\varphi}_i' \mathbf{B}' \mathbf{B} \boldsymbol{\varphi}_i$ is a standard quadratic form, the expectation of $E(\boldsymbol{\varphi}_i' \mathbf{B}' \mathbf{B} \boldsymbol{\varphi}_i | \mathbf{Y}_i, z_{ci}, u_i)$ can be calculated by the summation of all entries of the point product matrix of $\mathbf{B}' \mathbf{B}$ and $E(\boldsymbol{\varphi}_i \boldsymbol{\varphi}_i' | \mathbf{Y}_i, z_{ci}, u_i)$ [2].

M step

In the M step, the maximum likelihood solutions for parameter set $\boldsymbol{\Theta}$ is to be computed. First, the mixing proportion factor is

$$\hat{\pi}_c = \frac{1}{n} \sum_{i=1}^n Pr(Z_i = c | \mathbf{Y}_i) \quad (6.2.34)$$

Second, the covariance matrix $\mathbf{\Lambda}$ for random effect can be obtained by maximizing the following term

$$\sum_{c=1}^C \sum_{i=1}^n z_{ci} \left(-\frac{1}{2} \log\left(\frac{|\mathbf{\Lambda}|}{\hat{u}_i}\right) - \frac{1}{2} \text{trace}(\mathbf{\Lambda}^{-1} u_i \boldsymbol{\varphi}_i \boldsymbol{\varphi}_i') \right) \quad (6.2.35)$$

So, the $\mathbf{\Lambda}$ that maximizes this term is given by

$$\hat{\mathbf{\Lambda}} = \frac{1}{n} \sum_{i=1}^n \sum_{c=1}^C \hat{z}_{ci} \hat{u}_i E(\boldsymbol{\varphi}_i \boldsymbol{\varphi}_i' | \mathbf{Y}_i, z_{ci}, u_i) \quad (6.2.36)$$

Third, the term that involves the mean profile coefficients $\boldsymbol{\beta}_c$ in $L_4(\boldsymbol{\varphi}, \mathbf{U}, \mathbf{Z} | \mathbf{Y})$ is

$$-\frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{c=1}^C z_{ci} u_i E(\|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 | \mathbf{Y}_i, z_{ci}, u_i) \quad (6.2.37)$$

By maximizing this term, we can get the iteration function for $\boldsymbol{\beta}_c$ as

$$\hat{\boldsymbol{\beta}}_c = \left(\sum_{i=1}^n \hat{z}_{ci} \hat{u}_i \bar{\mathbf{B}}' \bar{\mathbf{B}} \right)^{-1} \left(\sum_{i=1}^n \bar{\mathbf{B}}' (\mathbf{Y}_i - \mathbf{B} E(\boldsymbol{\varphi}_i | \mathbf{Y}_i, z_{ci}, u_i)) \right) \quad (6.2.38)$$

Fourth, the term involving σ^2 is

$$-\frac{ng}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{c=1}^C z_{ci} u_i E(\|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 | \mathbf{Y}_i, z_{ci}, u_i) \quad (6.2.39)$$

Therefore, the ML solution for σ^2 is

$$\hat{\sigma}^2 = \frac{1}{ng} \sum_{i=1}^n \sum_{c=1}^C \hat{z}_{ci} \hat{u}_i E(\|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 | \mathbf{Y}_i, z_{ci}, u_i) \quad (6.2.40)$$

Therefore, the ML solution for σ^2 is

$$\hat{\sigma}^2 = \frac{1}{ng} \sum_{i=1}^n \sum_{c=1}^C \hat{z}_{ci} \hat{u}_{ic} E(\|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 | \mathbf{Y}_i, z_{ci}, u_i) \quad (6.2.41)$$

The last term in this M-step is the term which involves the degree of freedom parameter ν , which is given by

$$\sum_{c=1}^C \sum_{i=1}^n z_{ci} \left\{ -\log \Gamma\left(\frac{1}{2}\nu\right) + \frac{1}{2}\nu \log\left(\frac{1}{2}\nu\right) + \frac{1}{2}\nu (\log u_i - u_i) - \log u_i \right\} \quad (6.2.42)$$

As the above equation involving the estimate of ν is nonlinear, and it is computational intensive by a search over ν , Shoham gives a accurate approximation to update the ν in [143]. We will use this approximate solution in our work to estimate ν as well, and it is given by

$$\begin{aligned}\hat{\nu} &= \frac{2}{y + \log y - 1} + 0.0416(1 + \operatorname{erf}(0.6594 \log(\frac{2.1971}{y + \log y - 1}))), \quad (6.2.43) \\ y &\equiv -\sum_{i=1}^n \sum_{c=1}^C \hat{z}_{ci} \left(\log(\hat{u}_{ic}) - \hat{u}_{ic} + \Psi(\frac{\nu + g}{2}) - \log(\frac{\nu + g}{2}) \right) / n\end{aligned}$$

6.2.4 Computational Complexity of the Proposed Model

In order to compare the computational complexity of the proposed algorithm with the Gaussian model, we reproduce the EM solutions of multivariate Gaussian model based on random mixed-effects model as follow:

- E-step

$$\begin{aligned}E(z_{ci}|\mathbf{Y}_i) &= \Pr(Z_i = c|\mathbf{Y}_i) = \frac{\pi_c N_g(\mathbf{Y}_i; \bar{\mathbf{B}}\boldsymbol{\beta}_c, \boldsymbol{\Sigma})}{\sum_{c=1}^C \pi_c N_g(\mathbf{Y}_i; \bar{\mathbf{B}}\boldsymbol{\beta}_c, \boldsymbol{\Sigma})} \\ E(\boldsymbol{\varphi}_i|\mathbf{Y}_i, z_{ci}) &= \boldsymbol{\Lambda}\mathbf{B}'(\mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2\mathbf{I})^{-1}(\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c) \\ E(\|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2|\mathbf{Y}_i, z_{ci}) &= \|\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c\|^2 - 2(\mathbf{Y}_i - \bar{\mathbf{B}}\boldsymbol{\beta}_c)' \mathbf{B} E(\boldsymbol{\varphi}_i|\mathbf{Y}_i, z_{ci}) \\ &\quad + E(\boldsymbol{\varphi}_i' \mathbf{B}' \mathbf{B} \boldsymbol{\varphi}_i|\mathbf{Y}_i, z_{ci})\end{aligned} \quad (6.2.44)$$

where

$$\begin{aligned}Var(\boldsymbol{\varphi}_i|\mathbf{Y}_i, z_{ci}) &= \boldsymbol{\Lambda} - \boldsymbol{\Lambda}\mathbf{B}'(\mathbf{B}\boldsymbol{\Lambda}\mathbf{B}' + \sigma^2\mathbf{I})^{-1}\mathbf{B}\boldsymbol{\Lambda} \\ E(\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i'|\mathbf{Y}_i, z_{ci}) &= Var(\boldsymbol{\varphi}_i|\mathbf{Y}_i, z_{ci}) + [E(\boldsymbol{\varphi}_i|\mathbf{Y}_i, z_{ci})][E(\boldsymbol{\varphi}_i|\mathbf{Y}_i, z_{ci})]'\end{aligned}$$

- M-step

$$\begin{aligned}
\hat{\pi}_c &= \frac{1}{n} \sum_{i=1}^n Pr(Z_i = c | \mathbf{Y}_i) \\
\hat{\beta}_c &= \left(\sum_{i=1}^n \hat{z}_{ci} \bar{\mathbf{B}}' \bar{\mathbf{B}} \right)^{-1} \left(\sum_{i=1}^n \bar{\mathbf{B}}' (\mathbf{Y}_i - \mathbf{B} E(\boldsymbol{\varphi}_i | \mathbf{Y}_i, z_{ci})) \right) \\
\hat{\mathbf{\Lambda}} &= \frac{1}{n} \sum_{i=1}^n \sum_{c=1}^C \hat{z}_{ci} E(\boldsymbol{\varphi}_i \boldsymbol{\varphi}_i' | \mathbf{Y}_i, z_{ci}) \\
\hat{\sigma}^2 &= \frac{1}{ng} \sum_{i=1}^n \sum_{c=1}^C \hat{z}_{ci} E(\|\mathbf{Y}_i - \bar{\mathbf{B}}\beta_c - \mathbf{B}\boldsymbol{\varphi}_i\|^2 | \mathbf{Y}_i, z_{ci})
\end{aligned} \tag{6.2.45}$$

The above EM solutions reveal that the computational complexity of the Gaussian model is mainly determined by the equation of $\hat{\sigma}^2$, which is also influenced by the equation of $\hat{\beta}_c$. The computational complexity for the equation of $\hat{\beta}_c$ is $\mathcal{O}(ngp^2 + ngq^2 + ng^2p + npq + ng^3 + p^3)$, where p and q are the numbers of basis functions for mean and random effects. Hence, the computational complexity of the EM algorithm based on Gaussian model is $\mathcal{O}(n^2gp^2C + n^2gq^2C + n^2g^2pC + n^2pgC + n^2g^3C + np^3C)$.

Compared to the M-step solutions for the proposed t -Distribution model, the difference in the equation for $\hat{\sigma}^2$ is the robust estimate parameter u_i . The robust estimate parameter $\hat{u}_{ic}$ can be obtained via $\hat{u}_{ic} = E(u_i | \mathbf{Y}_i, z_{ci}) = \frac{\nu+g}{\nu+\delta(\mathbf{Y}_i, \bar{\mathbf{B}}\beta_c; \mathbf{B}\mathbf{\Lambda}\mathbf{B}' + \sigma^2\mathbf{I})}$, in which the computational complexity is given by $\mathcal{O}(g^3)$. In another word, compared to the Gaussian model, in order to compute $\hat{\sigma}^2$, it needs additional computational cost of $\mathcal{O}(nCg^3)$. Since this additional computational cost is still smaller than $\mathcal{O}(n^2g^3C)$ as in the computational complexity of Gaussian model, we can conclude that the total computational complexity of t -Distribution model is still the same as that of the Gaussian model.

6.2.5 Algorithm implementation

Summarizing the above discussions, the detail implementation steps of the proposed algorithm are listed as follows:

1. **Initialization:** Initialize the cluster center C , the degree-of-freedom parameter ν , and the basis function matrices $\bar{\mathbf{B}}$, $\mathbf{B}$. Apply fuzzy C-means (FCM) clustering algorithm to determine the parameters β_c for each cluster and the data covariance $\mathbf{\Omega}$. Set σ^2 to a small number and get the initial value of $\mathbf{\Lambda}$ according to $\mathbf{\Omega} = \mathbf{B}\mathbf{\Lambda}\mathbf{B}' + \sigma^2\mathbf{I}$. Set the maximum iteration number $iter_max$ (e.g. 500) and iteration counter $k = 1$, threshold for parameter change $\varepsilon = 10^{-5}$.

2. **E step:** For the k th iteration of the sequence of EM algorithm

Calculate the expectation $E(z_{ci}|\mathbf{Y}_i)$ of class indicator value using (6.2.21).

Calculate the expectations involving the robust estimation parameter u_i using (6.2.24) and (6.2.27).

Calculate the expectations involving the random effect φ_i using (6.2.32) and (6.2.33).

3. **M step:** For each cluster's parameters, $\hat{\pi}_c$, $\hat{\beta}_c$, $\hat{\mathbf{\Lambda}}$, $\hat{\sigma}^2$ and $\hat{\nu}$ are updated according to:

$$\begin{aligned}\hat{\pi}_c &= \frac{1}{n} \sum_{i=1}^n Pr(Z_i = c|\mathbf{Y}_i) \\ \hat{\beta}_c &= \left(\sum_{i=1}^n \hat{z}_{ci} \hat{u}_{ic} \bar{\mathbf{B}}' \bar{\mathbf{B}} \right)^{-1} \left(\sum_{i=1}^n \bar{\mathbf{B}}' (\mathbf{Y}_i - \mathbf{B} E(\varphi_i|\mathbf{Y}_i, z_{ci}, u_i)) \right) \\ \hat{\mathbf{\Lambda}} &= \frac{1}{n} \sum_{i=1}^n \sum_{c=1}^C \hat{z}_{ci} \hat{u}_{ic} E(\varphi_i \varphi_i' | \mathbf{Y}_i, z_{ci}, u_i) \\ \hat{\sigma}^2 &= \frac{1}{ng} \sum_{i=1}^n \sum_{c=1}^C \hat{z}_{ci} \hat{u}_{ic} E(\|\mathbf{Y}_i - \bar{\mathbf{B}}\beta_c - \mathbf{B}\varphi_i\|^2 | \mathbf{Y}_i, z_{ci}, u_i) \\ \hat{\nu} &= \frac{2}{y + \log y - 1} + 0.0416 \left(1 + erf \left(0.6594 \log \left(\frac{2.1971}{y + \log y - 1} \right) \right) \right), \\ &\quad \left(y \equiv - \sum_{i=1}^n \sum_{c=1}^C \hat{z}_{ci} \left(\log(\hat{u}_{ic}) - \hat{u}_{ic} + \Psi\left(\frac{\nu + g}{2}\right) - \log\left(\frac{\nu + g}{2}\right) \right) / n \right)\end{aligned}$$

4. **Convergence test:** Iterate between steps 2 and 3 until the parameter change is less than ε or the maximum number of iteration is reached.
5. Stop and plot the result.

For the initialization of the degrees of freedom, one can always use the value one (Cauchy distribution) to maximize the robustness at the beginning. Smaller degrees of freedom ν_i can appropriately assign small weights to the outliers via the mean and covariance of the data, while relatively larger values of the degrees of freedom tends to fit a GMM to the data [148]. That would be a trade-off between robustness and efficiency. In our experiments, as we know the levels of the outliers in the data, we choose the initial values from 10, 20, 50, and 100 according to the contamination of the data. This works well and provides possible gain in efficiency. Nevertheless, for the cases without any prior knowledge, it is advisable to use small initialization values of degrees of freedom which corresponding to fat-tailed distributions.

6.3 Experimental Results

6.3.1 Simulated curves

In this example, we will demonstrate the performance of our proposed model in comparison to the multivariate Gaussian model. First, we generate 100 curves for each cluster from the following four functions [4]:

$$\begin{aligned} Y_1(t) &= [-\exp(t)/1000] \\ Y_2(t) &= \tan(t/6.6) \\ Y_3(t) &= [5(t-4)^2/\max(t-4)^2] \\ Y_4(t) &= \cos(t) \end{aligned}$$

with the time series length of 30 and the number of basis functions for mean and random effect as $p = q = 10$. We use the same variance of 0.15 for each time knot, and covariance of 0.2 between time knots. The generated data for these four functions are shown in Fig. 6.2.

From Fig. 6.2, we observe that cluster Y_1 strongly overlaps with cluster Y_2 , while cluster Y_3 is well separated from the other three clusters. Hence, we choose three combinations (i.e. $Y_1 - Y_2$, $Y_1 - Y_3$, and $Y_1 - Y_4$) to study the performance and

Figure 6.2: Example 1. Simulated curve data from four different functions. (a) Y_1 . (b) Y_2 . (c) Y_3 . (d) Y_4 .

robustness of the proposed t -Distribution model. The clustering accuracy (CA) is determined by (the number of correctly clustered curves)/(total number of curves).

Comparing the clustering accuracy for combinations $Y_1 - Y_2$ and $Y_1 - Y_4$ in Table 6.1, we see that the proposed t -Distribution model yields better clustering results than the Gaussian model. For the combination $Y_1 - Y_3$, as the curves are well separated, both models are able to correctly cluster them. However, it is of interest to note that the estimated degree of freedom ν for the combination $Y_1 - Y_3$ indicates that a Gaussian model is actually adequate for the clustering purpose. When ν is relatively large, it approximates the Gaussian model. This proves that the proposed t -Distribution model is more flexible, and robust, than the Gaussian model.

To better understand the robustness of the multivariate mixed-effects t -Distribution model, let us consider another example with manually selected ν . Now, 100 curves are created according to 'sine' and 'cosine' functions with 50 curves for each cluster. We artificially contaminated two curves at a single time knot t with anomalous noise, which are randomly selected from each curve cluster (See Fig. 6.3).

Table 6.1: The clustering accuracy for three different combinations of simulated curves.

Combination	CA values for 3 different combinations		
	Gaussian model	t -distribution model	estimated degree of freedom $\hat{\nu}$
$Y_1 - Y_2$	57%	65%	19.31
$Y_1 - Y_3$	100%	100%	303.53
$Y_1 - Y_4$	88%	92%	77.4

As pointed out earlier, the inherent robust estimation parameter u_i and the degree of freedom ν play important roles in the t -Distribution model. In Table 6.3, we listed out the corresponding expectation values of $\hat{u}_{ic}$ for the two contaminated curves, as well as the expectation value of $\hat{u}_{ic}$ for a 'normal' curve observation from the 'sine' curve cluster.

Comparing these robust estimation values $\hat{u}_{ic}$, we see that both contaminated curves are given almost equal weighting, with slight preference for their original cluster. For the normal curve, this weighting preference is essentially bigger. On the other hand, when the t -Distribution is approaching the Gaussian model ($\nu = 300$), this preference becomes smaller, which implies that the Gaussian model may not be able to identify the outliers and can give wrong clustering result. By tuning the degree of freedom parameter ν , we can control the robustness of the t -Distribution model and offset the negative effect from the potential outlier observations. Moreover, this robustness parameter $\hat{u}_{ic}$ is useful if one wants to identify the outliers and further trim them to get better clustering results.

In addition, we also perform some analysis on the estimation of the uncorrelated measurement errors σ^2 for both Gaussian model and our proposed model. We still use the 100 curves in the previous example and artificially add in the uncorrelated measurement errors. We use the same variance of 0.02 for each time knot, and covariance of 0.08 between time knots. We test several different values of σ^2 , i.e.

Table 6.2: Analysis of the uncorrelated measurement errors σ^2 .

		estimated σ^2		sum of square errors of the control points	
		Gaussian	t-Distribution	Gaussian	t-Distribution
$\sigma^2 =$	0.01	0.0202	0.0181	0.1960	0.1171
	0.09	0.0942	0.1042	0.4286	0.0662
	0.16	0.1635	0.1753	0.7432	0.2443
	0.25	0.2527	0.2659	0.7224	0.3241

Figure 6.3: Example 2. Simulated curves with two outlier curves contaminated with anomalous noise.

0.01, 0.09, 0.16, and 0.25. The estimated values and the sum of square errors between the estimated control points and the ground truth ones has been listed in Table 6.2.

From Table 6.2 we can see that both model can correctly estimate the uncorrelated measurement errors σ^2 . However, the parameter σ^2 has different interpretations under the Gaussian model and the multivariate t-distribution model [147]. Hence we cannot compare them directly. But it is clear from the sum of square errors of the control points that the t-distribution model is more robust and can provide more accuracy mean profiles estimation.

Table 6.3: The expectation of parameter $\hat{u}_{ic}$ for outlier curve and normal curve with manually selected ν corresponding to Fig. 6.3.

	DOF	contaminated curve 1 (red)	contaminated curve 2 (blue)	normal curve (sine function)
$\hat{u}_{ic} = E(u_i \mathbf{Y}_i, z_{ci})$	$\nu = 7$	c = 1 3.6231	3.4403	3.6952
		c = 2 4.1153	3.1424	4.5577
	$\nu = 100$	c = 1 0.3264	0.5376	0.5105
		c = 2 0.6044	0.3619	1.073
	$\nu = 300$	c = 1 0.6114	0.8154	0.7304
		c = 2 0.8049	0.7134	1.0183

6.3.2 Silhouetted face profile contour

In this example, we generate a set of face contour curves manually using Matlab. Similar to [34], 18 key points are used to model the significant landmarks for the face contour (See Fig. 6.4). The resulting face contour is created by fitting a cubic B-spline model to these key points. For random effects, we generate the random contour from these 18 key points. The correlations between these key points are experimentally set in order to balance the the overall shape of the face contour (the shape of the forehead and nose) and the inter-cluster face variations.

Three groups of face contours with a population of 50 contours per group are created and then sample contours are shown in Figs. 6.5. Group 1 is created purposely with bigger nose and forehead in contrast to Group 2, while Group 3 is an outlier group which is specifically generated by averaging Group 1 and Group 2.

To compare the performance between Gaussian model and our proposed t -Distribution model, three combinations (i.e. $G1 - G2$, $G2 - G3$, and $G1 - G3$) among these three groups are carried out. In these experiments, the test curves are created from different combinations with an even size of 50 contours for each group. Our attention is focused on whether the test models are able to cluster the contours of well-separated groups and groups with strong overlap.

As a final experiment, we consider the overall performance based on the data of all groups with 110 contours generated by combining three groups together, with a configuration of 50 contours from Group 1, 50 contours from Group 2, and 10 contours from Group 3. The purpose of this experiment is to investigate the robustness of both models when the observed curves are contaminated with an additional set of outlier curves.

The clustering results for the above four experiments are shown in Table 6.4. It can be seen that the clustering accuracies for the first three combinations reflect well what we have expected. The Gaussian model cannot cluster correctly when two cluster curves have strong overlap. Our proposed t -Distribution model is able to recover most of the mis-clustered curves, which are affected by the strong contamina-

Figure 6.4: Simulated face profile contour with 18 key points

tion. When the noise level becomes reasonable, the advantage of the t -Distribution becomes significant. It is worth noting that the clustering results can be further improved with an optimal value of parameter ν of the degree of freedom for the proposed algorithm. The estimated values of the degree of freedom $\hat{\nu}$ in this example are listed in Table 6.4.

We present the clustering results of the final experiment in Fig. 6.6. Clearly, the estimated mean curve profiles obtained from the Gaussian model is strongly affected by the outlier observations as can be seen from Figs. 6.6 (a)-(b). In contrast, our proposed algorithm works well in this experiment, and it provides smoother and more accurate results in the presence of noise and outliers as compared to the Gaussian model. Moreover, from Fig. 6.6 (c) we also observe that the robust parameter $\hat{u}_{ic}$ plays an important role in keeping the estimated mean curves closer to the true mean when the covariance of outliers is large.

6.3.3 Vehicle curves

In this example, we consider a set of vehicle curves from two main classes: sedan or MPV, and outlier class: sports car. The simulated vehicle curves are generated by

Figure 6.5: Simulated face curve data for G1, G2, and G3 groups. (a) Simulated data for G1, 50 curves. (b) Simulated data for G2, 50 curves. (c) Simulated data for outliers cluster G3, 10 curves.

Figure 6.6: Clustering results true (solid line) and estimated (dashed line) mean curve profiles for the case with outlier face curves. (a) True and estimated mean curve profiles for G1 group based on Gaussian model. (b) True and estimated mean curve profiles for G2 group based on Gaussian model. (c) True and estimated mean curve profiles for G1 group based on t -Distribution model. (d) True and estimated mean curve profiles for G2 group based on t -Distribution model.

Table 6.4: The clustering accuracy for different combinations of 3 groups of silhouetted face curves.

Model	Different combinations of 3 groups			
	G1-G2 (2 clusters)	G2-G3 (2 clusters)	G1-G3 (2 clusters)	all groups (2 clusters with one outlier cluster)
Gaussian	100%	53.9%	55.3%	72.3%
t-Distribution	100% ($\hat{\nu} = 167.32$)	75.6% ($\hat{\nu} = 60.75$)	77.4% ($\hat{\nu} = 73.62$)	96.36 % ($\hat{\nu} = 25.34$)

Figure 6.7: Vehicle curves classes. (a) Sedan; (b) MPV; (c) Sports car (outlier class).

using the same method as in the previous examples. We choose 21 key control points to capture the characteristics of different cars. Some sample curves are represented in Figs. 6.7 (a)-(c). Similar to the first example, variations are added to the key control points to generate the random effect within the same vehicle class. The number of observation for each vehicle class is still 50. However, the outlier class in this experiment is not the average between sedan and MPV. It is created based on a different vehicle class: the sport car. We choose this class because its front contour is similar to the one of MPV, and the contour of its rear part is similar to that of the sedan car.

We use 100 vehicle curves (50 each for sedan and MPV) to test the performance of both the t -Distribution model and the Gaussian counterpart. The first row in Table 6.5 shows the clustering accuracy for the sedan and MPV clusters. For the t -Distribution model, the estimated degrees of freedom parameter implies that the

Table 6.5: The clustering accuracy for vehicle curve.

Classes	Models	
	Gaussian	t -Distribution
sedan and MPV	100%	100% ($\hat{\nu} = 178.98$)
sedan, MPV, and sports car	52.7%	94.4% ($\hat{\nu} = 52.95$)

Gaussian model is adequate for the well separated case since $\hat{\nu}$ has relatively large value. When there are outliers, the t -Distribution model preforms better than the Gaussian model as shown in the second row of Table 6.5.

6.3.4 UCR times series database

To further investigate the performance of our proposed method, we use the proposed algorithm on a benchmark time series database, the "UCR Time Series Classification/Clustering Page" [149].

We first take the 'Synthetic Control' data set, which has 6 classes, and 300 samples each, in the training and test data sets respectively. We choose the training data set for clustering purpose and use a uniform cubic B-spline with 25 control points to model the data. Fig. 6.8 (a) shows the true mean curves for the 6 classes in the training set. We test both the Gaussian model and our t -Distribution model on this data set. According to our experiment, the Gaussian model fails and causes numerical error during the simulation. In contrast, the t -Distribution model works well. The estimated mean curves from our proposed algorithm with the estimated degree of freedom of $\hat{\nu} = 14.35$ using (6.2.43) in the Appendix is given in Fig. 6.8 (b). The result is very promising in that the mean curves for 5 classes have been correctly estimated and the shapes of the true mean curves are well preserved by our proposed model. Fig. 6.8 (c)-(h) show the difference between true (red color) and estimated (blue color) mean curves for each cluster.

We perform another experiment on the ECG200 data set in [149], which has 2 classes with a total of 200 samples. In this example, we use all the data samples from both

Figure 6.8: UCR Times Series Database: synthetic control data set. (a) True mean curve for 6 classes for the training set. (b) Estimated mean curve with our proposed model. (c) Cluster 1. (d) Cluster 2. (e) Cluster 3. (f) Cluster 4. (g) Cluster 5. (h) Cluster 6.

training set and test set. The time series length of this data set is 96, and the average mean curve for each class is given in Fig. 6.9 (a). With our proposed model, we obtain a 152/200 clustering accuracy with 8 control points. The estimated degree of freedom is $\hat{\nu} = 13.01$ and the final estimated mean curve for each cluster is plotted in Fig. 6.9 (b). For the Gaussian model, we are unable to separate the data samples when we use the same control points as that of the t -Distribution model. However, we are able to get a 158/200 clustering accuracy when the number control point is increased to 27. Although the clustering result is slightly better as compared to the t -Distribution model, it is at the cost of increasing the model complexity and computational cost.

From these two experiments, we can see that the B-spline based random mixed-effects model is effective to capture the variation and mean shape of time series data. Together with the t -Distribution model, the proposed algorithm offers an appropriate and flexible way to model the data distribution for clustering analysis in real applications.

In addition, we test three additional datasets from the UCR times series database, namely "Trace", "FaceFour", and "CBF" with the Gaussian model and our proposed model, and summarize the clustering accuracy in Table 6.6. "-" means that we are unable to make the specific model work on this dataset. From the clustering results, we can see that even the proposed model can work with some of these datasets, the accuracy is not high. One of the reasons can be that the curve model may not work with these datasets, as the curve representation may destroy some useful information and reduce the discriminating power. The other reason of this low accuracy can be caused by using the uniform B-spline curves. During the experiments, we found some clusters are very similar to each other and they only has small variations within a relative short time range (see Fig. 6.10). Using a uniform B-spline curve may not be able to capture these distinguish details between the clusters. This could be improved by using non-uniform B-spline curve and increasing the control points at the specific location to enhance the distinguish power of the proposed method, although it is quite challenge and very application oriented.

(a)

(b)

Figure 6.9: UCR Times Series Database: ECG200 data set. (a) True mean curve for 2 classes for all the data samples. (b) Estimated mean curve with our proposed model.

Table 6.6: The clustering accuracy for five datasets from UCR database.

	Synthetic control	ECG200	Trace	FaceFour	CBF
Gaussian model	-	158/200	-	-	-
t-distribution model	178/300	152/200	52/100	-	514/900

Figure 6.10: True mean curves for Trace dataset from UCR times series database.

Unlike the simulated examples we used in the previous section, for real applications, it is necessary to select a proper number of control points and their locations in order to maximize the advantages of our proposed method. Usually, we first visually inspect the mean curve of each cluster and estimate how many control points are needed to capture the overall trend of this cluster. Then we select the largest one as the number of control points for all clusters. For some complicated cases, we may have to resort to the trial and error method to find the proper number of control point.

6.4 Conclusions

In this chapter, we have investigated the lack of robustness of the Gaussian distribution which has been widely used in curve clustering methods. We then present a robust curve clustering approach based on the multivariate t -Distribution random

effect mixture model to capture the within-cluster random effect of the curve data and combined features of both B-spline curve and robust multivariate t -Distribution model. Compared to the the regular Gaussian distribution model, the newly proposed model will perform better in the presence of outlier observations. The analysis of examples and experimental results demonstrate that our proposed robust statistical model indeed outperforms the Gaussian model when outliers are present in the data. The parameter estimation efficiency under outlier contamination has been observed in the simulated curve examples. Furthermore, from the face and car curve clustering examples, we are able to see that the proposed multivariate t -Distribution model provides a robust clustering framework for the curve model based image analysis and it can be a potential diagnostics tool for identifying outlying curves.

Chapter 7

Conclusions and Recommendations for Future Research

7.1 Conclusions

Data clustering based image analysis has become an integral component of image processing. Many technical issues are involved when using data clustering techniques for image understanding, these include data collection, feature extraction/representation, clustering algorithm, and output interpretation. Among these issues, the development of robust, efficient, extendible, and general clustering strategies is especially important because the performance and reliability of a data clustering system is usually determined by the clustering method. The existing problems of data clustering based image analysis approaches should be carefully considered with the fast changing requirements in this rapidly developing image processing field.

In this thesis, robust data clustering approaches have been proposed for intensity based and model based image analysis applications. The good performance of these approaches have been demonstrated by experimental results, both quantitatively and visually, on simulated and real image examples.

The first key contribution of this thesis is the introduction of an adaptive similarity measure. The advantages of the proposed adaptive similarity measure are that it

is adaptive to the local image content and does not require any experimentally adjusted parameter. Furthermore, it can readily incorporate the spatial context into the image segmentation process, while preserving the details of the image and reducing the edge blurring artifact in a relatively noisy environment. In fact, the proposed similarity measure is rather general and can be easily adopted into any clustering objective function based on Euclidian distance.

The second key contribution of this thesis is the robust segmentation method of intensity based feature representation. The existing data clustering based image segmentation techniques either lack the spatial information in their objective functions, or they require experimentally adjusted parameter which is not adaptive to the varieties of real image data. To circumvent these problems, an adaptive similarity measure is proposed and added into the information-theoretic clustering framework to form a robust image segmentation approach, which we named 'adaptive spatial information-theoretic clustering' (ASIC) algorithm. With this adaptive similarity measure, the amount of spatial constraint is controlled by the image itself without the need of user intervention. Furthermore, the two-step approach of ASIC is a first step towards a low and high level outlier elimination in a relatively noisy environment. The main characteristic of the ASIC algorithm is that it uses the capacity maximization concept in information theory to assess the reliability of the input data, thus enabling it to identify the outliers or unreliable data and further improves the clustering results.

The third contribution of this research work is concerned with the development of a robust fuzzy clustering algorithm for image segmentation. The conventional fuzzy C-means (FCM) approach has been widely used in many image segmentation applications and demonstrated better segmentation results when compared to the hard or crisp segmentation methods. However, the problem of noise sensitivity and the lack of spatial information problems in FCM-type algorithms have received great attentions in recent years. In order to address these two problems, we proposed a robust fuzzy clustering algorithm based on the information-theoretic framework and our novel adaptive similarity measure. The adaptive similarity measure is introduced

to add contextual information into the clustering process for image segmentation. The information-theoretic framework is incorporated into FCM algorithm by using the concept of input data distribution. Maximizing the mutual information against the input data distribution, we are able to identify those unreliable data points and relabel them with bigger neighborhood window to improve the segmentation. The simulation results based on simulated and real MRI brain images have shown that the proposed method can produce more homogeneous segmentation results as compared to the conventional FCM approach. Furthermore, our proposed robust clustering method does not use any experimentally set parameters except for the pre-specify fuzzifier value and cluster number.

The fourth key contribution of this research work is the development of a robust curve clustering method for model based image analysis. The result is especially useful for application where it is more convenient and concise to describe the image objects using curves or contours, as it is the most natural way of representing and recognizing shapes by human beings. Although the existing multivariate Gaussian mixture random effect model is effective and flexible to provide statistical modeling of this curvature representation, the lack of robustness of the conventional Gaussian mixture model is a big obstacle to provide robust clustering analysis in this field. An alternative approach to solve this problem is to use a wider-tailed statistical distribution model to build a new random effect model, so that the negative effect from atypical observations in the data set can be considered. The proposed multivariate t -Distribution model uses a robust parameter called degree of freedom to reduce the weights of atypical observations in the calculation of model parameters. Also, this robust statistical model provides a flexible way to account for the varying real data distributions and identify the outlying observations.

7.2 Recommendations for Future Research

The research results and methodologies on data clustering based image analysis as reported in this thesis offer many opportunities for further investigations. Here,

we recommend several future research directions which can further enhance the performance of the proposed result:

Automatic Cluster Number Selection. In this thesis, we mainly focus on the cluster analysis with a fixed cluster number. Future research can be carried out on the analysis of the relationship between outliers and model complexity of the clustering algorithm. It would be useful to propose an automatic cluster number selection framework to minimize the outlier number while preserving the simplicity of the model (small cluster number). A possible solution to this cluster validity problem is to use the structural risk minimization principle. It could be able to give a good guess about the optimal cluster number based on the optimization of Vapnik-Chervonenkis bound.

Integration of Prior Knowledge to Guide the Cluster Analysis. Knowledge-driven image analysis has received a great deal of attention in recent years, especially in medical image analysis [150–152]. The image information alone sometimes leads to implausible segmentation result. Additional knowledge or information is needed to direct the image analysis to statistically plausible directions. These prior information can usually be obtained via statistical average atlas from a training population or different imaging techniques for the same object. In addition to the spatial information, the prior knowledge of shapes or spatial locations of image object is useful to constraint the image segmentation to more reasonable directions. This process usually involves proper alignments either based on rigid body registration or elastic registration.

Degree of Freedom Parameter Adaptation. For the sake of simplicity, we have used fixed degree of freedom when performing the robustness analysis of the curve data in our research work. It is always desirable to use an adaptive degree of freedom parameter in real applications. The degree of freedom parameter should be inferred from or adaptive to the real data. We can expect better robust performance with an adaptive selection framework for the degree of freedom parameter.

Fusion of Structural MRI and DTI Data Although our adaptive similarity measure and robust information-theoretic clustering framework have achieved great success in compensating the noise artifact and improving the structural MRI brain image segmentation results, there are still some limitations due to low intensity contrast in structural-MR images between WM and GM in certain regions [153]. In recent years, there is a trend of using diffusion tensor images (DTI) to provide complementary information to the structural MRI image data [153–157], because the DTI image data can provide better tissue contrasts for WM and non-WM tissues, CSF and non-CSF tissues [158]. With this complimentary information provided by the DTI image data, we believe that it is possible to adopt our methods to produce more robust and effective segmentation approaches for clinical use.

Bibliography

- [1] G. E. Moore, "Progress in digital integrated electronics," in *Proceedings of the 1975 IEDM*, 1975, pp. 11–13.
- [2] Y. H. Luan and H. Z. Li, "Clustering of time-course gene expression data using a mixed-effects model with B-splines," *Bioinformatics*, vol. 19, no. 4, pp. 474–482, 2003.
- [3] C. Zhou, "A bayesian mixture model for partitioning gene expression data," *Biometrics*, vol. 62, pp. 515–525, 2006.
- [4] P. Ma, C. I. Castillo-Davis, W. X. Zhong, and J. S. Liu, "A data-driven clustering method for time course gene expression data," *Nucleic Acids Research*, vol. 34, no. 4, pp. 1261–1269, 2006.
- [5] H. Li, Y. Sun, and M. Zhan, "Analysis of gene coexpression by B-spline based cod estimation," *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2007, no. 1, pp. 1–10, 2007.
- [6] L. P. Clarke, R. P. Velthuisen, M. A. Camacho, J. J. Heine, M. Vaidyanathan, L. O. Hall, R. W. Thatcher, and M. L. Silbiger, "MRI segmentation: Methods and applications," *Magn. Res. Imag.*, vol. 13, no. 3, pp. 343–368, 1995.
- [7] W. M. Wells, W. E. L. Grimson, R. Kikinis, and F. A. Jolesz, "Adaptive segmentation of MRI data," *IEEE Trans. Med. Imag.*, vol. 15, pp. 429–442, Apr. 1996.
- [8] S. Z. Li, "On discontinuity-adaptive smoothness priors in computer vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 6, pp. 576–586, 1995.

-
- [9] T. Pappas, "An adaptive clustering algorithm for image segmentation," *IEEE Trans. Signal Processing*, vol. 40, no. 4, pp. 901–914, Apr. 1992.
 - [10] J. C. Bezdek, L. O. Hall, and L. P. Clarke, "Review of MR image segmentation techniques using pattern recognition," *Med. Phys.*, vol. 20, pp. 1033–1048, 1993.
 - [11] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Patt. Rec.*, vol. 40, pp. 1277–1294, 1993.
 - [12] X. W. Chen and T. Huang, "Facial expression recognition: A clustering-based approach," *Pattern Recognition Letters*, vol. 24, pp. 1295–1302, 2003.
 - [13] M. Murugappan, M. Rizon, R. Nagarajan, S. Yaacob, D. Hazry, and I. Zunaiddi, "Time-frequency analysis of eeg signals for human emotion detection," in *4th Kuala Lumpur International Conference on Biomedical Engineering 2008*. Kuala Lumpur, Malaysia: Springer Berlin Heidelberg, 2008, pp. 262–265.
 - [14] S. Chakrabarti, *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan-Kaufmann, 2002.
 - [15] P. Baldi, P. Frasconi, and P. Smyth, *Modeling the Internet and the Web: Probabilistic Methods and Algorithms*. Wiley, 2003.
 - [16] A. Vakali and T. Dalamagas, "An overview of Web data clustering practices," in *Proceedings of the 9th International Conference on Extending Database Technology - EDBT' 04*. Springer-Verlag, 2004, pp. 597–606.
 - [17] A. K. Jain, M. N. Murthy, and P. J. Flynn, "Data clustering: A review," Department of Computer Science, Michigan State University, East Lansing, Michigan, Tech. Rep., August 2000.
 - [18] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Upper Saddle River, NJ: Prentice Hall, 1988.

-
- [19] M. S. Aldenderfer and R. K. Blashfield, *Cluster Analysis*. Sage Publications, Inc., 1984.
 - [20] R. R. Sokal, *Principles of numerical taxonomy*. W. H. Freeman (San Francisco), 1963.
 - [21] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley-Interscience, 2000.
 - [22] K. W. Bowyer and N. Ahuja, *Advances in Image Understanding : A Festschrift for Azriel Rosenfeld*. Wiley, 1996.
 - [23] D. L. Pham, C. Xu, and J. L. Prince, “A survey of current methods in medical image segmentation,” in *Annual Review of Biomedical Engineering*, M. L. Yarmush, K. R. Diller, and M. Toner, Eds. Annual Reviews, 2000, vol. 2, pp. 315–337.
 - [24] J. Ramsay and B. Silverman, *Functional Data Analysis*. Springer, New York, 1997.
 - [25] T. Tarpey and K. K. J. Kinatader, “Clustering functional data,” *Journal of Classification*, vol. 20, pp. 93–114, 2003.
 - [26] M. W. Berry, *Survey of Text Mining I: Clustering, Classification, and Retrieval*. Springer, 2003.
 - [27] A. K. Jain, R. P. W. Duin, and J. Mao, “Statistical pattern recognition: A review,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 1, pp. 4–37, 2000.
 - [28] K. J. dna A. Sokolowski and H.-H. Bock, *Classification, Clustering, and Data Analysis:Recent advances and applications*. Springer-Verlag Berlin, 2002.
 - [29] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. John Wiley & Sons Inc., 1973.

-
- [30] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," in *Proc. of the IEEE*, vol. 86, no. 11, 1998, pp. 2210–2239.
 - [31] A. W. C. Liew and Y. Hong, "An adaptive spatial fuzzy clustering algorithm for 3D MR image segmentation," *IEEE Trans. Med. Imag.*, vol. 22, pp. 1063–1075, Sep. 2003.
 - [32] L. A. García-Escudero and A. Gordaliza, "A proposal for robust curve clustering," *Journal of Classification*, vol. 22, pp. 185–201, 2005.
 - [33] P. D. Wentzell, T. K. Karakach, S. Roy, M. M. J., C. P. Allen, and M. Werner-Washburne, "Multivariate curve resolution of time course microarray data," *BMC Bioinformatics*, no. 7, 2006. [Online]. Available: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1539028>
 - [34] N. Davidenko, "Silhouetted face profiles: A new methodology for face perception research," *Journal of Vision*, vol. 7, no. 4, pp. 1–17, 2007.
 - [35] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*. Springer, 2003.
 - [36] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, CA, 1985, pp. 281–297.
 - [37] P. Mahalanobis, "On the generalized distance in statistics," in *Proceedings of the National Institute of Science of India* 12, 1936.
 - [38] T. R. Babua, M. N. Murtya, and V. Agrawalb, "Classification of run-length encoded binary data," *Pattern Recognition*, vol. 1, pp. 321–323, 2006.
 - [39] L. K. Szeto, A. W.-C. Liew, H. Y., and S.-S. Tang, "Gene expression data clustering and visualization based on a binary hierarchical clustering framework," *Journal of Visual Languages & Computing*, vol. 14, no. 4, pp. 341–362, 2003.

-
- [40] S. Z. Selim and M. A. Ismail, "K-means type algorithms: a generalized convergence theorem and characterization of local optimality," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, pp. 81–87, 1984.
 - [41] P. S. Bradley and U. M. Fayyad, "Refining initial points for K-Means clustering," in *Proc. 15th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 1998, pp. 91–99.
 - [42] M. Meila and D. Heckerman, "An experimental comparison of several clustering and initialization methods," in *Machine Learning*, 1998, pp. 386–395.
 - [43] J. L. Tian, L. Zhu, S. Q. Zhang, and L. Liu, "Improvement and parallelism of k-means clustering algorithm," *Tsinghua Science & Technology by Elsevier B.V.*, vol. 10, no. 3, pp. 277–281, 2006.
 - [44] S. S. Khan and A. Ahmad, "Cluster center initialization algorithm for k-means clustering," *Pattern Recognition Letters*, vol. 25, no. 11, pp. 1293–1302, 2004.
 - [45] L. A. Zadeh, "Fuzzy sets," *Information Control*, vol. 8, pp. 338–353, 1965.
 - [46] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybernetics*, vol. 3, pp. 32–57, 1973.
 - [47] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York, NY: Plenum Press, 1981.
 - [48] R. N. Dave, "Fuzzy-shell clustering and applications to circle detection in digital images," *Int. J. General Syst.*, vol. 16, pp. 343–355, 1990.
 - [49] R. N. Dave and T. Fu, "Adaptive fuzzy C-shells clustering and detection of ellipses," *IEEE Trans. Neural Networks*, vol. 3, pp. 643–662, Sep. 1992.
 - [50] R. N. Dave and R. Krishnapuram, "Robust clustering methods: A unified view," *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 2, pp. 270–293, 1997.
 - [51] D. L. Pham and J. L. Prince, "Adaptive fuzzy segmentation of magnetic resonance images," *IEEE Trans. Med. Imag.*, vol. 18, pp. 737–752, Sep. 1999.
-

-
- [52] R. Krishnapuram and J. M. Keller, "The possibilistic c-means algorithm: insights and recommendation," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 3, pp. 385–393, 1996.
 - [53] N. A. Mohamed, M. N. Ahmed, and A. A. Farag, "Modified fuzzy c-mean in medical image segmentation," in *Proc. of IEEE-EMBS*, vol. 20, 1998, pp. 1377–1380.
 - [54] K. Rose, E. Gurewitz, and G. C. Fox, "A deterministic annealing approach to clustering," *Pattern Recogn. Lett.*, vol. 11, pp. 589–594, Sep. 1990.
 - [55] R. E. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. Inform. Theory*, vol. 18, no. 4, pp. 460–473, 1972.
 - [56] C. E. Shannon, "A mathematical theory of communication," *Bell Systems Technical Journal*, vol. 27, pp. 623–656, 1948.
 - [57] M. R. Brito, E. L. Chávez, A. J. Quiroz, and J. E. Yukich, "Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection," *Statistics & Probability Letters*, vol. 35, no. 1, p. 33427, 1997.
 - [58] M. I. Petrovskiy, "Outlier detection algorithms in data mining systems," *Program. Comput. Softw.*, vol. 29, no. 4, pp. 228–237, 2003.
 - [59] T. Hu and S. Y. Sung, "Detecting pattern-based outliers," *Pattern Recognition Letters*, vol. 24, no. 16, pp. 3059–3068, 2003.
 - [60] Q. Song, "A robust information clustering algorithm," *Neural Computation*, vol. 17, no. 12, 2005.
 - [61] D. M. Hawkins, *Identification of Outliers*. Chapman anHall, London, 1980.
 - [62] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. Wiley, 1987.
 - [63] D. L. Davies and D. w. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 1, pp. 224–227, 1979.
-

-
- [64] L. Hubert and J. Schultz, "Quadratic assignment as a general data-analysis strategy," *British Journal of Mathematical and Statistical Psychology*, vol. 29, pp. 190–241, 1976.
 - [65] J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, no. 3, pp. 301–315, 1998.
 - [66] R. N. Dave, "Validating fuzzy partitions obtained through C-shells clustering," *Pattern Recognition Letters*, vol. 17, no. 6, pp. 613–623, 1996.
 - [67] A. W. C. Liew, S. H. Leung, and W. H. Lau, "Fuzzy image clustering incorporating spatial continuity," in *IEE Proc. Vis. Image Signal Process*, vol. 147, no. 2, 2000, pp. 185–192.
 - [68] R. Dubes and A. K. Jain, "Validity studies in clustering methodologies," *Pattern Recognition*, vol. 11, no. 4, pp. 235–254, 1979.
 - [69] K. Lim and A. Pfefferbaum, "Segmentation of mr brain images into cerebrospinal fluid spaces, white and gray matter," *J. Comput. Assist. Tomogr.*, vol. 13, pp. 588–593, 1989.
 - [70] M. N. Ahmed, S. M. Yamany, A. A. Farag, and T. Moriarty, "Bias field estimation and adaptive segmentation of MRI data using a modified fuzzy c-means algorithm," in *Proc. of the 13th International Conf. on Computer Assisted Radiology and Surgery, CARS'99*, Paris, 1999, p. 1004.
 - [71] Y. A. Tolias and S. M. Panas, "Image segmentation by a fuzzy clustering algorithm using adaptive spatially constrained membership functions," *IEEE Trans. Syst., Man, Cybern. A*, vol. 28, no. 3, pp. 359–369, 1998.
 - [72] D. L. Pham, "Spatial models for fuzzy clustering," *Computer Vision and Image Understanding*, no. 84, pp. 285–297, 2001.
 - [73] L. Szilágyi, Z. Benyó, S. M. Sizlányi, and H. S. Adam, "MR brain image segmentation using an enhanced fuzzy C-means algorithm," in *Proc. of the*

-
- 25th Annual International Conference of the IEEE EMBS*, Cancun, Mexico, September 17-21 2003, pp. 724–726.
- [74] W. Cai, S. Chen, and D. Zhang, “Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation,” *Pattern Recognition*, vol. 40, pp. 835–838, 2007.
- [75] S. Z. Li, *Markov Random Field Modeling in Computer Vision*. Springer, 1995.
- [76] C. Raphael, “Automatic segmentation of acoustic musical signals using hidden markov models,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 360–370, Apr. 1999.
- [77] J. L. Marroquin, E. A. Santana, and S. Botello, “Hidden markov measure field models for image segmentation,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 11, pp. 1380–1387, 2003.
- [78] K. Blekas, A. Likas, N. P. Galatsanos, and I. E. Lagaris, “A spatially constrained mixture model for image segmentation,” *IEEE Trans. Neural Networks*, vol. 16, no. 2, pp. 494–498, Mar. 2005.
- [79] S. Makrogiannis, G. Economou, and S. Fotopoulos, “A region dissimilarity relation that combines feature-space and spatial information for color image segmentation,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 35, no. 1, Feb. 2005.
- [80] T. Pavlidis and Y. T. Liow, “Integrating region growing and edge detection,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 3, pp. 225–233, Mar. 1990.
- [81] J. S. Cardoso and L. Corte-Real, “Toward a generic evaluation of image segmentation,” *IEEE Trans. Image Processing*, vol. 14, no. 11, pp. 1773–1782, 2005.

-
- [82] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 6, pp. 929–944, June 2007.
 - [83] F. Y. Shih and K. Zhang, "Efficient contour detection based on improved snake model," *Pattern Recognition and Artificial Intelligence*, vol. 18, no. 2, pp. 197–209, Mar. 2004.
 - [84] F. Y. Shih and S. Cheng, "Adaptive mathematical morphology for edge linking," *Information Sciences*, vol. 167, no. 1-4, pp. 9–21, Dec. 2004.
 - [85] —, "Automatic seeded region growing for color image segmentation," *Image and Vision Computing*, vol. 23, no. 10, pp. 877–886, Sept. 2005.
 - [86] Y. Wu, F. Y. Shih, J. Shi, and Y. Wu, "A top-down region dividing approach for image segmentation," *Pattern Recognition*, vol. 41, no. 6, pp. 1948–1960, June 2008.
 - [87] P. Orbanz and J. M. Buhmann, "Nonparametric bayesian image segmentation," *International Journal of Computer Vision*, vol. 77, pp. 25–45, 2008.
 - [88] U. Maulik, "Medical image segmentation using genetic algorithms," *IEEE Trans. Inform. Technol. Biomed.*, vol. 13, no. 2, pp. 166–173, 2009.
 - [89] M. Egmont, D. de Ridderb, and H. Handels, "Image processing with neural networks a review," *Pattern Recognition*, vol. 35, no. 10, pp. 2279–2301, 2002.
 - [90] A. Sofou and P. Maragos, "Generalized flooding and multicue pde-based image segmentation," *IEEE Trans. Image Processing*, vol. 17, no. 3, pp. 364–376, 2008.
 - [91] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.
 - [92] M. S. Yang and K. L. Wu, "A similarity-based robust clustering method," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 4, pp. 434–448, Apr. 2004.

-
- [93] M. N. Ahmed, S. M. Yamany, N. Mohamed, A. A. Farag, and T. Moriarty, "A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data," *IEEE Trans. Med. Imag.*, vol. 21, pp. 193–199, Mar. 2002.
 - [94] R. E. Blahut, *Principle and Practice of Information Theory*. Addison-Wesley, 1988.
 - [95] I. S. Dhillon, S. Mallela, R. K., I. G., and A. Elisseeff, "A divisive information-theoretic feature clustering algorithm for text classification," *Journal of Machine Learning Research*, vol. 3, pp. 1265–1287, 2003.
 - [96] R. G. Gallager, *Information Theory and reliable communication*. New York, NY: John Wiley & Sons, Inc., 1968.
 - [97] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY: Wiley InterScience, 2006.
 - [98] D. C. Mackay, "Comparision of approximate methods for handling hyperparameters," *Neural Computation*, vol. 11, pp. 1035–1068, 1999.
 - [99] R. N. Davé, "Characterization and detection of noise in clustering," *Pattern Recognition Letter*, vol. 12, no. 11, pp. 657–664, 1991.
 - [100] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 98–110, 1993.
 - [101] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A possibilistic fuzzy c-means clustering algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 4, pp. 363–399, 2005.
 - [102] (1997) BrainWeb: Simulated brain database. [Online]. Available: <http://www.bic.mni.mcgill.ca/brainweb>
 - [103] D. L. Collins, A. P. Zijdenbos, V. Kollokian, J. G. Sled, N. J. Kabani, C. J. Holmes, and A. C. Evans, "Design and construction of a realistic digital brain phantom," *IEEE Trans. Med. Imag.*, vol. 17, no. 3, pp. 463–468, 1998.

-
- [104] N. Duta, A. K. Jain, and M. P. D. Jolly, "Automatic construction of 2D shape models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 5, pp. 433–446, 2001.
 - [105] F. S. Cohen and J. Y. Wang, "Part I: Modeling image curves using invariant 3-D object curve models - a path to 3-D recognition and shape estimation from image contours," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 1, pp. 1–12, 1994.
 - [106] F. S. Cohen, Z. H. Huang, and Z. W. Yang, "Invariant matching and identification of curves using B-splines curve representation," *IEEE Trans. Image Processing*, vol. 4, no. 1, pp. 1–10, 1995.
 - [107] E. Klassen, A. Srivastava, W. Mio, and S. H. Joshi, "Analysis of planar shapes using geodesic paths on shape spaces," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 3, pp. 372–383, 2004.
 - [108] A. Srivastava, S. H. Joshi, W. Mio, and X. W. Liu, "Statistical shape analysis: Clustering, learning, and testing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 4, pp. 590–602, 2005.
 - [109] A. Tsai, W. M. Wells, S. K. Warfield, and A. S. Willsky, "An EM algorithm for shape classification based on level sets," *Medical Image Analysis*, vol. 9, pp. 491–502, 2005.
 - [110] N. Thakoor, J. Gao, and S. Y. Jung, "Hidden markov model-based weighted likelihood discriminant for 2-D shape classification," *IEEE Trans. Image Processing*, vol. 16, no. 11, pp. 2707–2718, 2007.
 - [111] Y. Wang and E. K. Teoh, "2D affine-invariant contour matching using B-spline model," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 10, pp. 1853–1858, 2007.
 - [112] Y. K. Jung and Y. S. Ho, "A feature-based vehicle tracking system in congested traffic video sequences," in *PCM '01: Proceedings of the Second IEEE Pacific Rim Conference on Multimedia*. Springer-Verlag, 2001, pp. 190–197.
-

-
- [113] N. Thakoor and J. Gao, "Automatic video object shape extraction and its classification with camera in motion," in *ICIP '05: IEEE International Conference on Image Processing, 2005*, vol. 3, Sept. 2005, pp. 437–440.
 - [114] H. B. Wang, Q. L. Lu, L. L. Zhang, L. Z. Xu, and X. F. Li, "A feature-based vehicle tracking system in congested traffic video sequences," in *ICCEE '08: 2008 International Conference on Computer and Electrical Engineering*, 2008, pp. 553–557.
 - [115] S. C. Tien, T. L. Chia, and Y. B. Lu, "Using cross-ratios to model curve data for aircraft recognition," *Pattern Recognition Letters*, vol. 24, no. 12, pp. 2047–2060, Aug. 2003.
 - [116] S. H. Joshi and A. Srivastava, "Shape estimation and object classification in images using geometric priors," in *ACSSC '06: 2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, 2006, pp. 1575–1579.
 - [117] Y. P. Wang, S. L. Lee, and K. Toraichi, "Multiscale curvature-based shape representation using B-spline wavelets," *IEEE Trans. Image Processing*, vol. 8, no. 11, pp. 1586–1592, 1999.
 - [118] C. G. Small, *The statistical theory of shape*. Springer, 1996.
 - [119] I. L. Dryden and K. Mardia, *Statistical shape analysis*. John Wiley & Sons Inc., 1998.
 - [120] S. L. Free, P. O'Higgins, D. D. Maudgil, I. L. Dryden, L. Lemieux, D. R. Fish, and S. D. Shorvon, "Landmark-based morphometrics of the normal adult brain using MRI," *Neuroimage*, vol. 13, no. 5, pp. 801–813, 2001.
 - [121] J. Shi, A. Samal, and D. Marx, "How effective are landmarks and their geometry for face recognition?" *Computer Vision and Image Understanding*, vol. 102, no. 2, pp. 117–133, 2006.

-
- [122] A. Millonig and K. Schechtner, "Developing landmark-based pedestrian-navigation systems," *IEEE Trans. Intell. Transport. Syst.*, vol. 8, no. 1, pp. 43–49, 2007.
- [123] W. Zhang, X. Li, J. Zhang, A. Luft, D. F. Hanley, P. van Zijl, M. I. Miller, L. Younes, and S. Mori, "Landmark-referenced voxel-based analysis of diffusion tensor images of the brainstem white matter tracts: application in patients with middle cerebral artery stroke," *Neuroimage*, vol. 44, no. 3, pp. 906–913, 2009.
- [124] L. H. Staib and J. S. Duncan, "Boundary finding with parametrically deformable models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 11, pp. 1061–1075, 1992.
- [125] A. Neumann and C. Lorenz, "Statistical shape model based segmentation of medical images," *Computerized Medical Imaging and Graphics*, vol. 22, pp. 133–143, 1998.
- [126] A. Delbimbo and P. Pala, "Visual image retrieval by elastic matching of user sketches," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 2, pp. 121–132, 1997.
- [127] D. Christos, X. D. Tao, and D. G. Shen, "Hierarchical active shape models, using the wavelet transform," *IEEE Trans. Med. Imag.*, vol. 22, no. 3, pp. 414–423, 2003.
- [128] H. Li, A. Yezzi, and L. D. Cohen, "3D brain segmentation using dual-front active contours with optional user interaction," *International Journal of Biomedical Imaging*, pp. 1–17, 2006.
- [129] S. Scalroff and L. F. Liu, "Deformable shape detection and description via model-based region grouping," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 5, pp. 475–489, 2001.
- [130] S. Bourouis, K. Hamrouni, and N. Betrouni, "Automatic MRI brain segmentation with combined atlas-based classification and level-set approach," in

-
- ICIAR '08: Proceedings of the 5th international conference on Image Analysis and Recognition*. Springer-Verlag, 2008, pp. 770–778.
- [131] G. M. James and C. A. Sugar, “Clustering for sparsely sampled functional data,” *Journal of the American Statistical Association*, vol. 98, no. 462, pp. 397–408, 2003.
- [132] C. Abraham, P. A. Cornillon, E. Matzner-Løber, and N. Molinari, “Unsupervised curve clustering using B-splines,” *Scandinavian Journal of Statistics*, vol. 30, no. 3, pp. 581–595, 2003.
- [133] W. Jank, “Ascent em for efficient curve-clustering in large on-line auction databases,” *SSRN eLibrary*, 2004. [Online]. Available: <http://ssrn.com/paper=902908>
- [134] J. Yang, L. H. Staib, and J. S. Duncan, “Neighbor-constrained segmentation with level set based 3-D deformable models,” *IEEE Trans. Med. Imag.*, vol. 23, no. 8, pp. 940–948, 2004.
- [135] G. J. McLACHLAN, *The EM Algorithm and Extensions*. New York, NY: John Wiley & Sons, Inc., 1997.
- [136] H. X. Wang, Q. B. Zhang, B. Luo, and S. Wei, “Robust mixture modelling using multivariate t-distribution with missing information,” *Pattern Recognition Letters*, vol. 24, pp. 701–710, 2004.
- [137] C. de Boor, *A Practical Guide to Splines*. Springer-Verlag, 1978.
- [138] D. F. Rogers and J. A. Adams, *Mathematical Elements for Computer Graphics, second edition*. McGraw-Hill, 1990.
- [139] G. Farin, *Curves and Surfaces for CAGD: A Practical Guide, fourth edition*. Academic Press, 1997.
- [140] Y. Lee, J. A. Nelder, and Y. Pawitan, *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman & Hall/CRC, 2006.

-
- [141] D. Peel and G. J. McLachlan, “Robust mixture modelling using the t -distribution,” *Statistics and Computing*, vol. 10, pp. 339–348, 2000.
 - [142] J. A. Bilmes, “A gentle tutorial of the EM algorithm and its applications to parameter estimation for gaussian mixture and hidden markov models,” Department of Electrical Engineering and Computer Science, U. C. Berkeley, Berkeley, CA, Tech. Rep., April 1998.
 - [143] S. Shoham, “Robust clustering by deterministic agglomeration em of mixtures of multivariate t -distributions,” *Pattern Recognition*, vol. 35, pp. 1127–1142, 2002.
 - [144] M. Svensén and C. M. Bishop, “Robust bayesian mixture modelling,” *Neurocomputing*, vol. 64, pp. 235–252, 2005, trends in Neurocomputing: 12th European Symposium on Artificial Neural Networks 2004.
 - [145] STUDENT, “The probable error of a mean,” *Biometrika*, vol. 6, no. 1, pp. 1–25, 1908.
 - [146] C. Liu and D. B. Rubin, “ML estimate of the t distribution using EM and its extensions, ECM and ECME,” *Statistica Sinica*, vol. 5, pp. 19–39, 1995.
 - [147] J. C. Pinheiro, C. H. Liu, and Y. N. Wu, “Efficient algorithms for robust estimation in linear mixed-effects models using the multivariate t distribution,” *Journal of Computational and Graphical Statistics*, vol. 10, no. 2, pp. 249–276, 2001.
 - [148] D. Gerogiannis, C. Nikou, and A. Likas, “The mixtures of Student’s t -distribution as a robust framework for rigid registration,” *Image and Vision Computing*, vol. 27, pp. 1285–1294, 2008.
 - [149] E. Keogh, X. Xi, L. Wei, and C. A. Ratanamahatana. (2006) The UCR time series classification/clustering homepage. [Online]. Available: http://www.cs.ucr.edu/~eamonn/time_series_data/

-
- [150] P. P. Wyatt and J. A. Nobel, "MAP MRF joint segmentation and registration of medical images," *Med. Image Anal.*, vol. 7, no. 4, pp. 539–552, 2003.
 - [151] J. Yang and J. S. Duncan, "3d image segmentation of deformable objects with joint shape-intensity prior models using level sets," *Med. Image Anal.*, vol. 8, no. 3, pp. 285–294, 2004.
 - [152] S. Dambreville, Y. Rathi, and A. Tannenbaum, "A framework for image segmentation using shape models and kernel space shape priors," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 30, no. 8, pp. 1385–1399, 2008.
 - [153] S. P. Awate, H. Zhang, T. J. Simon, and J. C. Gee, "Multivariate segmentation of brain tissues by fusion of mri and dti data," in *ISBI '08: 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2008.*, 2008, pp. 213–216.
 - [154] A. Hadjiprocopis, W. Rashid, and P. S. Tofts, "Unbiased segmentation of diffusion-weighted magnetic resonance images of the brain using iterative clustering," *Magnetic Resonance Imaging*, vol. 23, pp. 877–885, 2005.
 - [155] T. M. Liu, G. Young, L. Huang, N. K. Chen, and T. C. Wong, "76-space analysis of grey matter diffusivity: Methods and applications," *NeuroImage*, vol. 31, pp. 51–65, 2006.
 - [156] S. P. Awate, H. Zhang, and J. C. Gee, "A fuzzy, nonparametric segmentation framework for DTI and MRI analysis: With applications to DTI-tract extraction," *NeuroImage*, vol. 31, pp. 51–65, 2006.
 - [157] C. F. Lu, P. S. Wang, Y. C. Chou, H. C. Li, B. W. Soong, and Y. T. Wu, "Segmentation of diffusion-weighted brain images using expectation maximization algorithm initialized by hierarchical clustering," in *30th Annual International IEEE EMBS Conference*, Aug. 2008, pp. 5502–5505.
 - [158] T. M. Liu, H. Li, K. Wong, A. Tarokh, L. Guo, and T. C. Wong, "Brain tissue segmentation based on DTI data," *NeuroImage*, vol. 38, no. 1, pp. 114–123, 2007.
-