Robust clustering algorithms

Yang, Xulei

2005

Yang, X. (2005). Robust clustering algorithms. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/3847>

<https://doi.org/10.32657/10356/3847>

Nanyang Technological University

Downloaded on 26 Apr 2023 21:40:41 SGT

Robust Clustering Algorithms

Yang Xulei

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University

in fulfillment of the requirements for the degree of

Doctor of Philosophy

2005

Statement of Originality

I hereby certify that the content of this thesis is the result of work done by me and has not been submitted for a higher degree to any other University or Institution.

.....

Date

.....

Yang Xulei

Acknowledgments

I wish to express my sincere gratitude to all the people who have assisted me during the years of my postgraduate study at Nanyang Technological University. I am most grateful to my supervisor, Associate Professor Song Qing, for his professional guidance, constant support and trust. His insightful comment has been an inspiration for my work. It has been a great pleasure to have Professor Song as my advisor.

I would like to thank all my colleagues at the School of Electrical and Electronic Engineering, for their support and help towards the completion of my study. Particularly, I want to say thanks to my friends at the Computer Control Laboratory and the Intelligent Machines Research Laboratory, for their valuable discussions on some work of this study.

I would like to acknowledge the School of Electrical and Electronic Engineering, Nanyang Technological University, for providing the financial support, research facilities and opportunity for this study.

Finally, my sincere thanks go to all the members of my family. Especially, I am very grateful to my wife, Xu Linfang, for her love, care, understanding and encouragement. I would not have accomplished this thesis if there were not their support.

This thesis is dedicated to my mother and my wife for their love and support.

Summary

Clustering is an important unsupervised classification technique which can be informally stated as grouping a given collection of unlabelled samples into meaningful clusters. There are a number of applications in a wide variety of areas such as pattern recognition, image processing, automatic control, communication, data mining, and bio-informatics. A very rich literature on clustering has been developed over the past three decades. Most traditional clustering algorithms, such as K-means and fuzzy c-means, however, suffer from several major drawbacks, e.g., the sensitivity to noise and outliers, the difficulty in determining the number of clusters, and the sensitivity to initialization. In this thesis, we focus on developing a few robust learning algorithms, which aim to overcome the major drawbacks of traditional clustering methods, for data clustering and its extensions.

The deterministic annealing (DA) algorithm, in which the annealing process with phase transitions leads to a natural hierarchical clustering, is independent of the choice of the initial data configuration. The DA approach to clustering has been demonstrated to possess substantial performance improvement over traditional unsupervised learning methods. However, the DA algorithm is sensitive to noise and outliers in addition to the pre-requirement of the optimal cluster number. A robust deterministic annealing (RDA) algorithm is therefore proposed in this thesis to improve the performance of standard DA algorithm for data clustering: the robustness against noise and outliers is obtained by using the basic idea of noise clustering (NC) algorithm; and the optimal cluster number is estimated by the Vapnik-Chervonenkis (VC) bound. The experimental results on artificial and real

data sets show that RDA clustering algorithm provides a performance that is superior to that of traditional NC and DA algorithms: it takes advantages of both NC and DA algorithms in addition to auto-selection of the optimal cluster number.

It is well-known that, by means of some specific shell-induced distance measures, the traditional clustering methods can be modified to partition shell-shaped data structures such as lines, circles and ellipses. However, most shells clustering methods, which are developed based on existing (traditional) clustering algorithms, inevitably suffer from the same drawbacks as the latter do. In this thesis, we make an effort to improve the robustness of existing spherical shells clustering methods. An information fuzzy c-spherical shells (IFCSS) algorithm is therefore developed for robust clustering of spherical shells, which possesses several improved features: first, it is robust against noise and outliers, which is achieved by the maximization of mutual information against the prior distribution; second, the optimal cluster number of spherical shells is auto-determined by the cluster validity, i.e., VC-bound, which is estimated based on the structural risk minimization principle. The effectiveness and superiority of the proposed IFCSS algorithm for clustering spherical shells are demonstrated by extensive experiments.

Data clustering in Mercer kernel-induced feature space is interesting: by non-linearly mapping the observed data from a low-dimensional input space into a high (possible infinite)-dimensional feature space by means of a given kernel function, the kernel-based clustering can discover some complicated data structures that may not be revealed by traditional clustering methods in the standard input space. However, some important issues related to kernel-based clustering, such as the adaptive selection of kernel function and its parameter(s), the robustness against noise and outliers, and the estimation of the optimal cluster number, remain unstudied in the literature. For this purpose, a kernel-based deterministic annealing (KDA) algorithm is constructed in this thesis for data clustering by using a Gaussian kernel function, whose width is adaptively selected by the scaled inverse of the input data covariance. Furthermore, other important issues men-

tioned above are also studied preliminarily. The experimental results on artificial and real data sets demonstrate the effectiveness of the proposed KDA algorithm for clustering a variety of data sets, which may possess different shape, distribution and size.

Finally, we develop robust clustering methods for image segmentation and pattern classification. First, we present an adaptive spatially-constrained deterministic annealing (ASDA) algorithm, which takes into account of the spatial continuity constraints by using a dissimilarity index that allows spatial interactions between image pixels, for image segmentation. The local spatial continuity constraint reduces the noise effect and the classification ambiguity. More importantly, the strength of spatial constraint for each given image pixel is auto-determined by the scaled variance of its neighbor pixels in a 3×3 window, which leads to the adaptiveness of the presented algorithm. The effectiveness and efficiency of the presented method for image segmentation are tested on synthetic and magnetic resonance images. Second, we present a weighted support vector machine (WSVM) to improve the outlier sensitivity problem of standard support vector machine (SVM) for two-class data classification. The basic idea is to assign different weights to different data points such that the WSVM algorithm learns the decision surface according to the relative importance of data points in the training data set. The weights used in WSVM are generated by the proposed robust clustering algorithm, kernel-based possibilistic c-means (KPCM), whose partition generates relative high values for important data points but low values for outliers. Experimental results on artificial and benchmark data indicate that the proposed method reduces the effect of outliers and yields higher classification rate than standard SVM does.

In summary, several novel and robust learning algorithms, with the aim to overcome the drawbacks of traditional clustering algorithms, are developed for data clustering and its applications. The effectiveness and superiority of the proposed methods are supported by experimental results.

Table of Contents

Acknowledgments	i
Summary	ii
List of Figures	xi
List of Tables	xvi
1 Introduction	1
1.1 Background, Motivation and Objective	1
1.2 Major Contributions of the Thesis	6
1.3 Organization of the Thesis	8
2 Literature Review of Data Clustering Algorithms	11
2.1 Introduction	11
2.2 Standard Clustering Methods	13
2.2.1 Fuzzy C-means Clustering	14
2.2.2 Deterministic Annealing Clustering	15
2.3 Robust Clustering Methods	18
2.3.1 Robust Fuzzy C-means Clustering	19

TABLE OF CONTENTS

vi

2.3.2	Popular Cluster Validity Indices	22
2.4	Spherical Shells Clustering Methods	24
2.4.1	Fuzzy C-shells Clustering Algorithm	25
2.4.2	Fuzzy C-Spherical Shells Clustering Algorithm	26
2.4.3	Fuzzy C-rings Clustering Algorithm	27
2.5	Kernel-based Clustering Methods	29
2.5.1	Nonlinear Mapping	29
2.5.2	Kernel Trick	31
2.5.3	Data Clustering in Feature Space	32
3	A Robust Deterministic Annealing Algorithm for Data Clustering	37
3.1	Introduction	37
3.2	The Proposed RDA Clustering Algorithm	38
3.2.1	Formulation and Derivation	38
3.2.2	Optimal Cluster Number Estimation	43
3.2.3	Pseudo-code of RDA algorithm	46
3.3	Experimental Results	47
3.3.1	Example 1: Optimal Cluster Number	49
3.3.2	Example 2: DA vs RDA	51
3.3.3	Example 3: NC vs RDA	52
3.3.4	Example 4: RDA on Real Data Sets	54
3.4	Conclusion	57
4	An Information Fuzzy C-Spherical Shells Clustering Algorithm	58
4.1	Introduction	58

TABLE OF CONTENTS

vii

4.2	The Proposed Clustering Method	59
4.2.1	FCSS Algorithm for Spherical Shells	60
4.2.2	Robustness through Maximization of Mutual Information . .	62
4.2.3	Optimal Cluster Number by VC-Bound	66
4.2.4	Pseudo-Code of the Proposed Method	69
4.3	Experimental Results	70
4.3.1	Example 1: Robustness Against Noisy Points	70
4.3.2	Example 2: The Optimal Cluster Number	71
4.3.3	Example 3: Several Additional Specific Cases	73
4.4	Conclusion and Discussion	75
5	Data Clustering in Mercer Kernel-Induced Feature Space	77
5.1	Introduction	77
5.2	Derivation and Implementation	78
5.2.1	Kernel-induced Distance Measure	78
5.2.2	The KDA Clustering Algorithm	80
5.2.3	Adaptive Selection of Gaussian Kernel Parameter	84
5.2.4	Pseudo-code of KDA Implementation	86
5.3	Experimental Results	89
5.3.1	Artificial Data	90
5.3.2	Real-life Data	96
5.4	Several Further Investigations	97
5.4.1	Optimal Cluster Number Estimation	98
5.4.2	Robust Pruning of Noisy Points	99

TABLE OF CONTENTS

viii

5.4.3	Image Illustration in Feature Space	100
5.5	Conclusion	102
6	Further Investigation on VC-bound as Cluster Validity	103
6.1	Introduction	103
6.2	The Proposed New Cluster Validity	106
6.2.1	SRM Principle and VC-bound	107
6.2.2	Cluster Number Selection by VB index	108
6.2.3	Pseudo-Code of the Proposed Method	111
6.3	Experimental Results	112
6.3.1	Numerical Examples	113
6.3.2	Two Real Data Sets	117
6.4	Conclusion and Discussion	119
7	Image Segmentation By Deterministic Annealing Algorithm With Adaptive Spatial Constraints	122
7.1	Introduction	122
7.2	Review of Spatially-Constrained Fuzzy C-means Algorithm	125
7.3	The proposed ASDA Clustering Algorithm	127
7.3.1	Formulation and Derivation	127
7.3.2	A Complexity Reduced Version of ASDA	129
7.3.3	Adaptive Selection of Spatial Penalty α_j	129
7.3.4	Pseudo-code of ASDA for Image Segmentation	131
7.4	Experimental Results	131
7.4.1	Synthetic Image	133

TABLE OF CONTENTS

ix

7.4.2	Simulated MR Data	134
7.5	Conclusion and Discussion	135
8	A Weighted Support Vector Machine for Data Classification	137
8.1	Introduction	137
8.2	Standard SVM	139
8.2.1	Formulation of SVM	139
8.2.2	Outlier Sensitivity Problem of SVM	141
8.3	The Proposed Weighted SVM	142
8.3.1	Formulation of WSVM	142
8.3.2	Weights Generating Algorithm	144
8.4	Experimental Results	147
8.4.1	Artificial Data Set	147
8.4.2	Benchmark Data Set	148
8.5	Remarks and Discussions	150
8.6	Conclusion	153
9	Conclusions and Future Work	154
9.1	Conclusions	154
9.2	Future Work	158
	Appendix A: Abbreviations' List	161
	Appendix B: The Proof of Fano's Inequality	163
	Appendix C: AIC and BIC	167

TABLE OF CONTENTS x

Author's Publications 169

Bibliography 171

List of Figures

2.1	The annealing process with phase transitions of DA clustering algorithm on a simple data. The original cluster centers are denoted by “o”, the partitioned cluster centers are denoted by “*”.	17
2.2	Schematic drawing for calculation of the cluster prototypes. v_k is the k th cluster center, r_k is the radius of the k th cluster, x_j is the j th data point. Figure taken from [68].	25
2.3	Toy example of a binary classification problem mapped into feature space. In two dimensions a rather complicated nonlinear decision surface is necessary to separate the classes (a), whereas in a feature space the decision surface becomes a hyperplane (b). Figure taken from [93].	30
2.4	Comparisons of K-means and kernel-based K-means on a linearly nonseparable data set.	33
3.1	Clustering results of DA and RDA with an added outlier coordinate (50,0), a simple demonstration of the robustness of RDA against noisy points. Note the different locations of the cluster centers (denoted by “*”) obtained by DA and RDA.	41
3.2	The plot of VC-bound with different cluster number for data sets in example 1. V_b reaches the minimum at $c^* = 3$ for both data sets. . .	47

3.3	The partitioned results and values of VC-bound of RDA with different cluster number (from 2 to 6) for the first data set in example 1. V_b reaches the minimum at $c^* = 3$ corresponding to the most reasonable partitioned result.	48
3.4	The partitioned results and values of VC-bound of RDA with different cluster number (from 2 to 6) for the second data set in example 1. V_b reaches the minimum at $c^* = 3$ corresponding to the most reasonable partitioned result.	49
3.5	The plot of VC-bound with different cluster number for data sets in example 2. V_b reaches the minimum at $c^* = 2$ for noisy data set 1 and $c^* = 4$ for noisy data set 2, respectively.	50
3.6	Clustering noisy data sets, comparison between DA and RDA. The values of original and partitioned cluster centers are shown in Table 3.1.	51
3.7	The plot of VC-bound with different cluster number for data sets in example 3. V_b reaches the minimum at $c^* = 3$ for both data sets. . .	52
3.8	Clustering balanced and unbalanced data sets, comparison between NC and RDA. Where $x/20$ means the according result occurs x times out of 20 runs.	53
3.9	RDA on Iris and Wine data sets. V_b reaches the minimum at $c^* = 3$ for both data. The number of misclassified points are 11 for Iris data and 5 for Wine data.	54
4.1	Clustering results of standard FCSS and the proposed IFCSS with different values of λ for concentric shell-shaped clusters with some outliers. The larger value λ is, the more outliers are eliminated. . .	70

4.2	VC-bound vs cluster number with different values of λ for the data set in example 1. V_b reaches the minimum at $c^* = 2$ with independence of λ	71
4.3	Partitioned results and VC-bound of IFCSS with different cluster number c for data set 1 in example 2. V_b reaches the minimum at $c^* = 3$, which is corresponding to the most reasonable result.	72
4.4	Partitioned results and VC-bound of IFCSS with different cluster number c for data set 2 in example 2. V_b reaches the minimum at $c^* = 3$, which is corresponding to the most reasonable result.	72
4.5	VC-bound vs cluster number for the data sets in example 2. V_b reaches the minimum at $c^* = 3$ for both data sets.	73
4.6	First row: four original data sets used in example 3; Second row: according VC-bound revealing the optimal cluster for each data set; Third row: according partitioned results of IFCSS with optimal cluster number for each data set.	74
5.1	Plot of the r.h.s of (5.20) as a function of distance $\ x_i - x_j\ $ in data space for four different values of ρ . At distances $\ x_i - x_j\ \gg \rho$ and $\ x_i - x_j\ \ll \rho$ the assignment costs are constant.	83
5.2	Three artificial data sets tested in the experiments.	89
5.3	The partition results by applying KDA with different values of ρ and the estimated $\rho^* = 0.1068$ on unbalanced data. Note Fig 5.3b has 2 partition errors.	90
5.4	The clustering results of KKM and DA algorithms on unbalanced data. Where $x/20$ means the according result occurs x times out of 20 runs.	91
5.5	The partition results by applying KDA with different values of ρ and the estimated $\rho^* = 2.8613$ on ring data.	92

5.6	The partition results of KKM and DA algorithms on ring data. KKM is sensitive to data initializations.	93
5.7	The partition results by applying KDA with different values of ρ and the estimated $\rho^* = 0.7743$ on X-O data.	94
5.8	The partition results of KKM and standard DA algorithms on X-O data. KKM is sensitive to data initializations.	95
5.9	Optimal cluster number estimation by contour plot of kernel matrix K	98
5.10	Simple demonstration of KDA against noisy points using the maximization of mutual information. The high value of λ , the more robustness of KDA. Note the detected outliers are not shown. . . .	99
5.11	The image illustration of noisy data, ring data, and X-O data in the feature space, where the topographic ordering of the data in the input space is kept while the probability of the linear separability of the data is increased.	101
6.1	Graphical depiction of the SRM principle. The SRM picks a function f_* which has small training error (empirical risk), and comes from an element of the structure that has low capacity h (confidence interval), thus minimizing a risk bound in (6.3). Figure taken from [93].	106
6.2	The distortion J_c is monotonously decreasing with respect to the increase of cluster number (a). The proposed VB index with respect to cluster number. The minimum is reached at $c^* = 5$ (b).	110
6.3	The simple data set and its three derivations.	113
6.4	Performances of different cluster indices on noisy data (a-g). Clustering result of DA with $c^* = 4$ for noisy data (h).	114

6.5	Performances of different cluster indices on overlapped data (a-g). Clustering result of DA with $c^* = 4$ for overlapped data (h).	115
6.6	Performances of different cluster indices on unbalanced data (a-g). Clustering result of DA with $c^* = 4$ for unbalanced data (h).	116
6.7	Performances of different cluster indices on Iris data (a-g). Clus- tering result of DA with $c^* = 3$ for Iris data by its two dominating features (PL and PW) (h).	117
6.8	Performances of different cluster indices on Wine data (a-g). Clus- tering result of DA with $c^* = 3$ for Wine data by its first two principle components (h).	118
7.1	Segmentation results of different clustering algorithms on the syn- thetic image. DA has 120 misclassified pixels and FCM has 127 misclassified pixels. The minimum number of misclassified pixels of SFCM is 14 at $\alpha = 10$ and $\alpha = 11$. While for the proposed ASDA, the misclassified number is reduced to 7.	132
7.2	Segmentation results of different clustering algorithms on the MR image. The segmentation accuracy (SA) of DA and FCM are 90.52% and 90.34% respectively. The best SA of SFCM is 94.38% obtained at $\alpha = 2.5$, which is slightly better than the result of the proposed ASDA, which leads to 94.31% SA.	133
8.1	Outlier sensitivity problem of standard SVM (a). Weights generated by KPCM for class 1 (b).	141
8.2	The performance comparison between SVM and WSVM while out- liers existing.	147

List of Tables

2.1	An overview of all the reviewed algorithms in chapter 2.	12
3.1	The positions of original and partitioned cluster centers for the data sets in Fig 3.6.	50
3.2	The partition errors of the proposed RDA algorithm and several traditional clustering methods on Iris and Wine data. Note we run NC/FCM/K-means 20 times for each data set with random initialization and show only the best result here.	55
3.3	The numerical results of VC-bound (V_b) with different cluster number ($c = 2, 3, \dots, 10$) on the image segmentation data.	56
4.1	Original and partitioned parameters for the data set in Fig 4.6a. . .	73
4.2	Original and partitioned parameters for the data set in Fig 4.6b. . .	73
4.3	Original and partitioned parameters for the data set in Fig 4.6c. . .	75
4.4	Original and partitioned parameters for the data set in Fig 4.6d. . .	75
5.1	The estimated value ρ^* and the optimal range ρ^o with according partition errors for different data sets. Each cell contains the estimated value ρ^* (or the optimal range ρ^o) followed by partition errors. . . .	94
5.2	The partition errors by applying KDA with different values of ρ and KDA with estimated ρ^* on original Iris data.	96

5.3	The partition errors by applying three methods on original Iris data and its first two principle components (PCs).	96
5.4	The partition errors of four traditional clustering methods on original Iris data and its first two principle components (PCs). Here we run each algorithm 20 times for each data set with random initialization and show only the best results.	97
6.1	Values of c^* preferred by each cluster validity index for the 5 data sets tested in this chapter.	119
7.1	The number of misclassified pixels of four methods on synthetic image.	132
7.2	Segmentation Accuracy of CSF, GM and WM by applying four methods on the simulated MR image.	134
8.1	Comparison of test error between WSVM and standard SVM for “Twonorm” benchmark data set according to different number of mislabelled data points using linear kernel function.	148
8.2	Comparison of test error between WSVM and standard SVM for “Thyroid” benchmark data set according to different number of mislabelled data points using RBF kernel function.	151
9.1	A summary of all the proposed robust clustering algorithms.	156

Chapter 1

Introduction

1.1 Background, Motivation and Objective

Machine Learning and Data Clustering

The advent of the information age has reshaped the world and brings huge amounts of data to our fingertips. In particular the internet, makes it possible to access data from all over the world. These profound changes make it necessary to develop new tools for dealing with the avalanche of available data. After all, the data does not comprise a value in itself: depending on the user's demand, data has to be located, retrieved, processed, visualized, and finally understood in order to be useful. Due to the sheer amount of data and its complexity, we need intelligent methods to extract meaningful information from the data. One interesting approach in the field of data analysis and processing is the idea of machine learning, which can be informally defined as the study of computer algorithms that can be improved automatically through learning. In view of the situation of an abundance of available data, the machine learning is of particular appeal, because it would allow for a relatively automatic acquisition of knowledge by computers which could therefore ideally assist humans in using the full potential of the available data. Machine learning basically falls into two broad categories. One is supervised learning, the

other is unsupervised learning. In supervised learning (like classification), we are provided with a collection of labelled samples; the problem is to label a newly encountered, yet unlabelled, sample. Typically, the given labelled samples are used to learn the descriptions of classes which in turn are used to label a new sample. In the case of unsupervised learning (like clustering), the problem is to group a given collection of unlabelled samples into meaningful clusters. In a sense, labels are associated with clusters also, but these category labels are data driven; that is, they are obtained solely from the data [51].

Clustering is an important unsupervised classification technique. There are at least five basic reasons for interest in unsupervised learning and clustering [33]. Firstly, collecting and labelling a large set of samples can be surprisingly costly. If a classifier can be crudely designed on a small set of labelled samples, and then “tuned up” by allowing it to run without supervision on a large, unlabelled set, much time and trouble can be saved. Secondly, one might wish to proceed in the reverse direction: Train with large amounts of (less expensive) unlabelled data, and only then use supervision to label the groupings found. This may be appropriate for large “data mining” applications, where the contents of a large database are not known beforehand. Thirdly, in many applications the characteristics of the samples can change slowly with time - for example, in automated food classification as the seasons change. If these changes can be tracked by a classifier running in an unsupervised mode, improved performance can be achieved. Fourth, we can use unsupervised methods to find features that will then be useful for categorization. There are unsupervised methods that provide a form of data-dependent “smart preprocessing” or “smart feature extraction”. Lastly, in the early stages of an investigation it may be valuable to perform exploratory data analysis and thereby gain some insight into the nature or structure of the data. The discovery of distinct subclasses - clusters or groups of samples whose members are more similar to each other than they are to other samples - or of major departures from expected characteristics may suggest us significantly alter our approach to design the classifier.

Interest in clustering has been increased recently because of new areas of application, such as data mining, image and speech processing, and bio-informatics. A rich literature on clustering has been developed in the past three decades. Prototype-based clustering algorithms are by far the most popular clustering technique. In prototype-based clustering algorithms, each cluster is represented by a prototype, and the sum of distances from the data samples to the prototypes is usually used as the objective function. These algorithms themselves can be divided into two classes: crisp (or hard) clustering where each data sample belongs to one and only one of the clusters; and fuzzy (or soft) clustering where each data sample belongs to more than one clusters with a certain degree of membership. In general, the performance of fuzzy algorithms is superior to that of the corresponding hard versions, and they have a lower tendency to get stuck in local minima [8]. Prototype-based clustering methods have the advantage of being dynamic in the sense that samples can move from one cluster to another, and they can incorporate knowledge about the global shape or size of clusters by using appropriate prototypes and distance measures in the objective function. These algorithms, however, suffer from some drawbacks (discussed below), such as the sensitivity to noise and outliers, the difficulty in determining the number of clusters, and the sensitivity to initialization of prototypes, which limits their practices in real applications. Hence, it is necessary to address these drawbacks by developing some improved (robust) learning algorithms for data clustering and its extensions.

Major Drawbacks and Existing Solutions

The clustering is usually formulated as an optimization problem by defining a cost criterion to be minimized. The objective function is to evaluate how well the clustering fits to the data. Traditional clustering techniques such as K-means [66] and fuzzy c-means [8] suffer from several major drawbacks. First, traditional clustering algorithms are essentially descent algorithms because the cost is reduced at each iteration. These methods therefore tend to get trapped in local minima

which riddle the cost surface [95], which leads to the sensitivity to the initialization. Second, traditional clustering algorithms tend to assign any data point (even it is a noisy one) to one (hard clustering) or more than one (soft clustering) clusters. In the case of that the given data is contaminated by noisy points (noise and/or outliers), the computed cluster prototypes can be placed far away from the true values. That means these algorithms are sensitive to noisy points. Last, the cluster number is assumed *a priori* in most traditional clustering algorithms, which is generally unknown in real applications. Many solutions have been proposed to overcome the above drawbacks of traditional clustering algorithms. However, the first disadvantage is usually ignored, and most researchers assume that they can provide their clustering algorithms with a suitable initialization. Others use multiple (random) initializations that guarantee (with a given probability) that at least one initialization is good.

The proposed solutions to overcome the problem of sensitivity to noise and outliers can be divided into two classes. In the first class are algorithms that are derived by modifying the objective function of K-means or fuzzy c-means. These algorithms are still sensitive to the initialization and several parameters used. The second class of algorithms are based on robust statistical estimators. These estimators can only find a single cluster in a noisy data. Some of these algorithms have been extended to find multiple clusters progressively. In other words, the data set is classified into a set of inliers (points belonging to the cluster to be extracted) and a set of outliers. Since the set of outliers include points from other clusters, the proportion of outliers can be very high. Therefore, even a breakdown point of 50% is not sufficient to make these algorithms highly robust. To overcome this problem, these algorithms consider the validity of the cluster formed by data points included in the set of inliers. When clusters overlap, the idea of extracting them in a serial fashion will not work, since removing one cluster may partially destroy the structure of other clusters. A unified review of robust clustering methods against noisy points can be found in [29].

Apart from noise and outliers, another issue that clustering algorithms need to deal with is the determination of the cluster number. This problem is closely intertwined with the problem of robustness. Not knowing the number of clusters in a data set complicates the task of separating the good points from the noisy points, and conversely, the presence of noise makes it harder to determine the number of clusters. The traditional approach to determine the optimal number of clusters is to evaluate a certain global validity (performance) measure of the partitions produced by clustering algorithms within a range of cluster number, then to pick the one that optimizes the validity measure. This method for searching an optimal cluster number is dependent on the selected clustering algorithm, whose performance is still dependent on the initial cluster prototypes. An alternative approach to this problem is to do progressive clustering. In this approach, clustering is initially performed with an over-specified number of clusters. After convergence, spurious clusters are eliminated, compatible clusters are merged, and good clusters are identified. Another variation of progressive clustering extracts one cluster at a time. The problem with this method is to define the spurious and compatible clusters. Moreover, although over-specification of the cluster number can reduce the initial cluster center effects, they can not guarantee that all clusters in the data set will be found [113].

Primary Objective

The principal objective of this thesis is to develop robust algorithms to address above challenges (i.e., to overcome drawbacks of traditional clustering algorithms) for the problems of data clustering and its extensions such as image segmentation and pattern classification. This objective is achieved by using some or all of the following techniques (details are given in correlative chapters): 1) the deterministic annealing approach to data clustering, where the annealing process with the phase transitions leads to a natural hierarchical clustering, is independent of the choice of the initial data configuration and has the ability to avoid poor local optima

in addition to the insensitivity to cluster volumes; 2) the kernel methods to data clustering, in which the input data is nonlinearly mapped into a high-dimensional feature space by means of a given kernel function, can reveal structure in the data that may go unnoticed by traditional clustering algorithms in the standard Euclidean space; 3) in terms of the noisy channel coding in information theory, the maximization of mutual information (to achieve channel) with respect to constrained source distribution can be used to phase out the noisy points for data clustering; 4) the model selection criterion VC(Vapnik-Chervonenkis)-bound, estimated based on the structural risk minimization principle in statistical learning theory, can be used as a cluster validity for data clustering algorithms to select the optimal number of clusters.

1.2 Major Contributions of the Thesis

The major contributions of this thesis consist of the formulation, derivation, implementation, and testing of the following robust learning algorithms for data clustering and its extensions. These algorithms overcome some or all of the major drawbacks of traditional clustering methods.

- **The Robust Deterministic Annealing (RDA) Algorithm for Data Clustering.** This method takes advantages of traditional noise clustering (NC) and deterministic annealing (DA) algorithms in terms of independence of data initialization, ability to avoid poor local optima, better performance for unbalanced data, and robustness against noise and outliers. In addition, the model selection criterion VC-bound is specifically extended for RDA to determine the optimal cluster number for a given data set. The superiority of the proposed RDA clustering algorithm is supported by experimental results on artificial and real data sets.
- **The Information Fuzzy C-Spherical Shells (IFCSS) Algorithm for Data Clustering.** The proposed algorithm offers several improved features

over existing spherical shells clustering algorithms: the robustness against noisy points (noise and outliers) is obtained by the maximization of mutual information, where the degree of robustness is controlled through a separate parameter to make a trade-off between rejection of noisy points and preserving of desirable data; the optimal number of spherical shells for a given data is auto-determined by the cluster validity criteria, i.e., VC-bound, which is estimated based on the structural risk minimization principle. The extensive simulation results have demonstrated the effectiveness of the proposed IFCSS algorithm for robust detection and separation of spherical shells.

- The Kernel-based Deterministic Annealing (KDA) Algorithm for Data Clustering.** The proposed method offers several promising features: first, it can reveal high-order correlation structure in the data, that may not be revealed by standard DA approach; second, it is independent of the data initialization and has the ability to avoid poor local optima. Furthermore, several other important issues, namely, the estimation of the optimal cluster number by contour plot of kernel matrix, the robustness against noisy points by maximization of mutual information, and the image illustration of data structure by kernel-induced distance distributions, are preliminarily studied. The experimental results have shown the effectiveness of the proposed KDA method for clustering different kinds of data sets, which may possess different shape, distribution and size.
- The Adaptive Spatially-constrained Deterministic Annealing (ASDA) Algorithm for Image Segmentation.** The proposed algorithm takes into account the spatial continuity constraints by using a dissimilarity index that allows spatial interactions between image pixels. The local spatial continuity constraint reduces the noise effect and the classification ambiguity. More importantly, the strength of spatial constraint for each given image pixel is auto-determined by the scaled variance of its neighbor pixels, which results in the adaptiveness of the proposed algorithm. The effectiveness and efficiency

of the proposed ASDA algorithm for image segmentation are supported by experimental results on both synthetic and magnetic resonance (MR) images.

- **The Weighted Support Vector Machines (WSVM) Algorithm for Pattern Classification.** By assigning different weights to different data points, the proposed WSVM algorithm can learn the decision surface according to the relative importance of data points in the training data set. The weights used in WSVM are generated by a robust fuzzy clustering algorithm, kernel-based possibilistic c-means (KPCM) algorithm, whose partition generates relative high values for important data points but low values for outliers. Experimental results indicate that the proposed method reduces the effect of outliers and yields higher classification rate than standard SVM does when outliers exist in the training data set.

1.3 Organization of the Thesis

The rest of the thesis is organized as follows.

Chapter 2 presents a brief review of the objective function based clustering algorithms. Several standard clustering algorithms are first introduced, followed by the most cited robust methods to solve the drawbacks of standard clustering algorithms. After that, the spherical shells clustering algorithms are briefly reviewed. And finally the basic idea of kernel-based clustering algorithms is explored.

Chapter 3 develops a robust deterministic annealing (RDA) algorithm for data clustering. We first present the formulation and derivation of the proposed clustering algorithm by applying the basic concept of noise clustering (NC) to deterministic annealing (DA) algorithm, then study the VC-bound cluster validity to determine the optimal cluster number for a given data set, and finally test the effectiveness and superiority of RDA clustering algorithm by experimental results on artificial and real-life data sets.

Chapter 4 proposes an algorithm, information fuzzy c-spherical shells (IFCSS), for robust clustering of spherical shells. We start from the standard FCSS algorithm to partition the given data set, then investigate the maximization of mutual information (which induces a robust density estimation to phase out noisy points) and the cluster validity (which is used to determine the number of spherical shells), and finally execute extensive simulations to demonstrate the effectiveness of the proposed clustering method for robust detection and separation of spherical shells.

Chapter 5 investigates data clustering in kernel-induced feature space by developing a kernel-based Deterministic annealing (KDA) algorithm. We first construct a nonlinear version of DA algorithm in the feature space based on a Gaussian kernel-induced distance measure, then put the emphasis on the adaptive selection of Gaussian kernel parameter by the scaled inverse of data covariance, and finally test the proposed method by experimental results on both artificial and benchmark data. Several further issues on KDA algorithm, such as the estimation of the optimal cluster number and the robustness against noisy points are also preliminarily studied.

In chapter 6, the novel cluster validity, i.e., VC-bound (VB) index, which has been used in RDA and IFCSS algorithms, is further investigated in terms of theoretical analysis and experimental comparison. We first introduce the structural risk minimization principle and related VC-bound, then apply it for optimal cluster number estimation, and finally give the pseudo-code of the proposed method. Three numerical examples and two real data sets are used to illustrate the use of VB as a validity index. Its effectiveness is compared to several popular cluster validity indices.

Chapter 7 presents an adaptive spatially-constrained deterministic annealing (ASDA) algorithm for robust image segmentation. We first give a brief review of spatially-constrained fuzzy clustering algorithm, then derive the formulation and derivative of the proposed algorithm, and discuss the adaptive selection of spatial penalty parameter, finally show qualitatively and quantitatively the experimental results

on synthetic and magnetic resonance (MR) data.

Chapter 8 proposes a weighted support vector machine (WSVM) to improve the outlier sensitivity problem of standard support vector machine (SVM) for two-class data classification. We first briefly review the basic theory and outlier sensitivity problem of standard SVM, then derive the formulation of the proposed WSVM, and present the weights generating algorithm, finally report the detailed experimental results. In addition, we present some remarks and discussions on the algorithms, experiments and further investigations.

Chapter 9 summarizes the work in this thesis and provides recommendations for future research in related areas.

Chapter 2

Literature Review of Data Clustering Algorithms

2.1 Introduction

There are basically two unsupervised learning approaches in data clustering, i.e., the parametric and nonparametric algorithms [10] [33]. In the parametric clustering approach, some pre-defined probability density functions are used to calculate the clustering objective function to achieve the optimal performance, which has limited applications in case of unknown distributions of the input data [67]. On the other hand, the nonparametric approach is to partition data without the prior knowledge of the data distribution. This chapter only discusses the nonparametric approach. In most traditional clustering algorithms, the relationship between data points and clusters is represented in one of two ways: each data point may belong to one and only one cluster (hard clustering), as e.g., K-means algorithm [66]; or each data point may belong to more than one clusters with a certain degree of membership (fuzzy clustering), as e.g., fuzzy c-means algorithm [8]. In general, the performance of fuzzy algorithms is superior to that of the corresponding hard versions, and they have a lower tendency to get stuck in local minima [8].

Table 2.1: An overview of all the reviewed algorithms in chapter 2.

Abbr.	Full Name	Category	Specific Use	Reference
FCM	Fuzzy C-Means	Standard clustering algorithm	Spherical, linearly separable	reference [8], section 2.2.1
DA	Deterministic Annealing	Standard clustering algorithm	Spherical, linearly separable	reference [85], section 2.2.2
NC	Noise Clustering	Robust clustering algorithm	Spherical, linearly separable	reference [25], section 2.3.1
PCM	Possibilistic C-Means	Robust clustering algorithm	Spherical, linearly separable	reference [57], section 2.3.1
FCS	Fuzzy C-Shells	Shell-induced clustering algorithm	Spherical Shells	reference [24], section 2.4.1
FCSS	Fuzzy C-Spherical Shells	Shell-induced clustering algorithm	Spherical Shells	reference [56], section 2.4.2
FCR	Fuzzy C-Rings	Shell-induced clustering algorithm	Spherical Shells	reference [68], section 2.4.3
KM	K-Means	Standard clustering algorithm	Spherical, linearly separable	reference [66], section 2.5.3
KKM	Kernel-based K-Means	Kernel-based clustering algorithm	A variety of data structures	reference [89], section 2.5.3

Many fuzzy (and hard) clustering algorithms have been suggested and used in the literature to partition data into clusters. There is a whole class of clustering algorithms in which an objective function based on a distance measure is iteratively minimized to obtain the final partition. The distance measure chosen and the objective function being optimized depend on the geometric structure of the clusters. Different distances have been invented to search for clusters of specific shapes in feature space. For example, the K-means [66] and fuzzy c-means [8] algorithms, using the Euclidean distance, look for clusters that are hyper-spherical in shape; the Gustafson-Kessel [43] and Gath-Geva [37] algorithms, in terms of fuzzy covariance matrix, can deal with hyper-elliptical data structures; shell-based clustering algorithms as in [24] [26] [56], using shell-induced distance measure, work well for circles and/or ellipses in shape; and the kernel-based clustering algorithms as in [89] [39], in terms of kernel-induced distance measure, provide a possible solution for linearly nonseparable data structure.

In this chapter, several standard clustering algorithms, i.e., fuzzy c-means [8] clus-

tering and deterministic annealing [85] clustering, are first introduced, followed by the popular solutions to the drawbacks of standard clustering algorithms, namely, the sensitivity to noisy points and the difficulty in determining the number of clusters. After that, the spherical shells clustering algorithms are briefly reviewed. And finally the basic idea of kernel-based clustering algorithms is explored. For a better understanding of the contents in this chapter, Table 2.1 provides an overview of all the discussed algorithms. The full list of the abbreviations used in the thesis can be found in Appendix A.

2.2 Standard Clustering Methods

The clustering is usually formulated as an optimization problem by defining a cost criterion to be minimized. The objective function is to evaluate how well the clustering fits to the data. Traditional clustering techniques such as K-means [66] and fuzzy c-means [8] are essentially descent algorithms because the cost is reduced at each iteration. They therefore tend to get trapped in local minima which riddle the cost surface [95]. Simulated annealing [55] is a well known technique for avoiding local minima of non-convex optimization problems. But this method requires very slow schedules that are not realistic for many practical applications. Deterministic annealing [85] is developed on the basis of the simulated annealing. It adopts the annealing mechanism that is rooted in the statistical physics, but uses a deterministic relaxation method instead of the probabilistic relaxation method during the iterative procedure. Therefore, it reaches a trade-off between optimal convergence and computation complexity. The deterministic annealing approach to clustering has demonstrated substantial performance as in [46] [86] [87]. In this section, we briefly review the fuzzy c-means clustering and deterministic annealing clustering. In the following discussions, let the input data set be $X = \{x_1, x_2, \dots, x_l\} \subset R^n$, where l is the number of input data points, n is the dimension of input space. Based on a measurement of similarity, the data set is partitioned into c (known a

priori) clusters whose prototypes (centers) denoted by $V = \{v_1, v_2, \dots, v_c\} \subset R^n$.

2.2.1 Fuzzy C-means Clustering

Mathematically, fuzzy c-means algorithm (FCM) is derived to minimize the following objective function with respect to the membership function u_{kj} and the cluster center v_k as given by

$$J_{FCM} = \sum_{j=1}^l \sum_{k=1}^c (u_{kj})^m d_{kj} \quad (2.1)$$

where $m \in (1, \infty)$ is the weighing exponent on fuzzy memberships, and a value of $m = 2$ is known to give good results with the FCM algorithm. The u_{kj} is the membership of point x_j in the k th cluster. And the d_{kj} is the squared Euclidean distance between data point x_j and cluster center v_k as follows,

$$d_{kj} = \|x_j - v_k\|^2 \quad (2.2)$$

By using the above distance measure, the FCM is limited to partition clusters that are hyper-spherical in shape. To take complicated shapes into account, we need to incorporate an appropriate distance measure into the FCM algorithm for different data structures. For example, we may use the distance measure in Gustafson-Kessel algorithm [43] or Gath-Geva algorithm [37] to deal with hyper-elliptical data structures.

The minimization of (2.1) gives the updating equations for membership u_{kj} and cluster center v_k , which are given by

$$u_{kj} = \frac{(\|x_j - v_k\|^2)^{1/(m-1)}}{\sum_{i=1}^c (\|x_j - v_i\|^2)^{1/(m-1)}} \quad (2.3)$$

and

$$v_k = \frac{\sum_{j=1}^N u_{kj}^m x_j}{\sum_{j=1}^N u_{kj}^m} \quad (2.4)$$

The FCM algorithm is summarized as follows,

- Fix the cluster number c , select initial cluster centers $\{v_k\}_{k=1}^c$, and set threshold ε be a small positive value.
- Alternatively update the membership function and cluster center by using (2.3) and (2.4) until $\|v_{new} - v_{old}\| < \varepsilon$.

2.2.2 Deterministic Annealing Clustering

The deterministic annealing (DA) algorithm proposed in [85], in which the annealing process with phase transitions leads to a natural hierarchical clustering, is independent of the choice of the initial data configuration and has the ability to avoid poor local optima (widely believed though not formally proven) in addition to the insensitivity to cluster volumes (ability to partition unbalanced data set due to its mass-constrained property) [87]. Let $d(x_j, v_k)$ be the distance (squared Euclidean distance in [87]) between x_j and v_k , $p(v_k|x_j)$ be the association probability (distribution) relating input point x_j with cluster center v_k , $p(x_j)$ be the source distribution (equal to $\frac{1}{l}$ in [87]). Then, the average expected distortion is given by

$$J_e = \sum_{j=1}^l \sum_{k=1}^c p(x_j) p(v_k|x_j) d(x_j, v_k) \quad (2.5)$$

As no prior knowledge of the data distribution $p(x_j, v_k) = p(x_j)p(v_k|x_j)$ is assumed, of all possible distributions that yield a given value of J_e , we choose the one that

maximizes the conditional Shannon entropy

$$H_s = - \sum_{j=1}^l p(x_j) \sum_{k=1}^c p(v_k|x_j) \log p(v_k|x_j) \quad (2.6)$$

It turns out [87] (according to the maximum entropy principle) that the resultant distribution is the titled distribution, which is given by

$$p(v_k|x_j) = \frac{p(v_k) e^{-\frac{d(x_j, v_k)}{T}}}{Z_{x_j}} \quad (2.7)$$

where $Z_{x_j} = \sum_{i=1}^c p(v_i) e^{-\frac{d(x_j, v_i)}{T}}$ is the partition function, $p(v_k) = \sum_{j=1}^l p(x_j) p(v_k|x_j)$ is the mass probability of k th cluster, and T is the Lagrangian multiplier which is analogical to the temperature in statistical mechanics. Clearly, at limited high T , these are uniform distributions, each data point is equally associated with all clusters. As T is lowered, the distributions become more discriminating. And at limited low T , the clustering becomes hard with each data point assigned to the nearest cluster with probability one.

To estimate the free parameter v_k , the effective cost to be minimized turns out to be the free energy (a well-know concept in statistical mechanics [85]) as follows.

$$F = \min_{\{p(v_k|x_j)\}} (J_e - T H_s) = -T \sum_{j=1}^l p(x_j) \log \sum_{i=1}^c p_i e^{-\frac{d(x_j, v_i)}{T}} \quad (2.8)$$

Based on (2.7), we can get the expression of cluster center v_k by minimizing (2.8), that is

$$v_k = \frac{\sum_{j=1}^l p(x_j) p(v_k|x_j) x_j}{\sum_{j=1}^l p(x_j) p(v_k|x_j)} \quad (2.9)$$

Alternatively updating (2.7) and (2.9) with phase transition (see below) gives the (mass constrained) DA algorithm, which has been shown superiority for data clustering and its extensions as reviewed in [87].

Figure 2.1: The annealing process with phase transitions of DA clustering algorithm on a simple data. The original cluster centers are denoted by “o”, the partitioned cluster centers are denoted by “*”.

During the annealing in T it is observed, that the cluster center remains at the mass center of the related cluster up to a critical value. At that point the representation undergoes a transition and the cluster center splits up in data space [41] [87]. The critical T_k^* for the phase transition for the k th cluster can be approximately calculated as follows [87]

$$T_k^* = 2\lambda_{\max}(C_k(x)) \quad (2.10)$$

where $\lambda_{\max}(C_k(x))$ is the largest eigenvalue of the fuzzy covariance matrix of the

k th cluster, that is

$$C_k(x) = \sum_{j=1}^l p(x_j|v_k)(x_j - v_k)(x_j - v_k)^T \quad (2.11)$$

with the posteriori $p(x_j|v_k)$ obtained by the Bayes formula

$$p(x_j|v_k) = \frac{p(x_j)p(v_k|x_j)}{p(v_k)} \quad (2.12)$$

Fig 2.1 illustrates the annealing process with phase transitions on a simple example. The training set is generated from a mixture of five equal variance Gaussian whose centers are marked by “o”. At high temperature, there is only one effective cluster represented by one cluster center, marked by “x”, at the center of mass of the training set. As the temperature is lowered, the system undergoes phase transitions which increase the number of effective clusters as shown in the figure.

2.3 Robust Clustering Methods

In many applications of data clustering, the real data set often contains noisy points (including outliers and noise). The FCM (so does DA), however, is not robust against noisy points because it tends to associate any data point (even it is a noisy one) in all clusters [29]. If the given data is contaminated by noisy points, the computed cluster centers can be placed far away from the true positions. Many researchers have attempted to solve this problem by modifying the constraint of probability in the membership of FCM. In particular, the noise clustering (NC) proposed in [25] and the possibilistic c-means (PCM) proposed in [57] have been shown to have satisfying ability of handling noisy data. The former defines an extra noise cluster, which attempts to include all the noise with high membership values, such that the good (true) clusters are mostly composed of good (non-noisy) points; The latter is a possibilistic clustering method in which one input point

slightly affects good clusters if its possibility to be noisy point is high, such that the partitioned results are slightly distorted by noisy points. These two methods will be introduced in this section. Apart from noise and outliers, another issue that clustering algorithms need to deal with is the determination of the number of clusters. This problem is closely intertwined with the problem of robustness. Not knowing the number of clusters in a data set complicates the task of separating the good points from the noisy points, and conversely, the presence of noise makes it harder to determine the number of clusters. The most used approach to determine the optimal number of clusters is to evaluate a certain global validity (performance) measure of the partitions produced by clustering algorithms within a range of cluster number, then to pick the one that optimizes the validity measure. The most popularly used cluster validity indices will also be introduced in this section.

2.3.1 Robust Fuzzy C-means Clustering

Noise Clustering

The noise clustering (NC) algorithm proposed in [25] considers noisy points as a separate cluster which is represented by a prototype that has a constant distance δ from all the data points. The membership u_{*j} of point x_j in the noise cluster is defined as

$$u_{*j} = 1 - \sum_{k=1}^c u_{kj} \quad (2.13)$$

Thus, the membership constraint for good clusters is effectively relaxed to

$$0 \leq \sum_{k=1}^c (u_{kj}) \leq 1 \quad (2.14)$$

This makes noisy points have less effect to good clusters. With the relaxed constraint, the objective function of NC is defined as

$$J_{NC} = \sum_{j=1}^l \sum_{k=1}^c (u_{kj})^m d_{kj} + \sum_{j=1}^l \delta (1 - \sum_{k=1}^c u_{kj})^m \quad (2.15)$$

where δ is the constant noise distance, which is recommended in [25] as

$$\delta = \lambda \left[\frac{\sum_{j=1}^l \sum_{k=1}^c d_{kj}}{lc} \right] \quad (2.16)$$

where λ is the value of the multiplier parameter used to obtain δ from the average of distances. Assuming that the distance δ is specified, the minimization of (2.15) gives the updating equation for membership u_{kj} as following:

$$u_{kj} = \frac{1}{\sum_{i=1}^c \left[\frac{d_{kj}}{d_{ij}} \right]^{1/(m-1)} + \left[\frac{d_{kj}}{\delta} \right]^{1/(m-1)}} \quad (2.17)$$

The updating equation for cluster center in NC is the same as that in FCM. The equations (2.17) and (2.4) gives the noise clustering (NC) algorithm. It has been shown in [25] [27] [30] that this method is quite successful in handling noise, and that it is applicable to many practical situations.

Possibilistic C-means Clustering

Another robust approach, called the possibilistic c-means algorithm, was introduced in [57] to overcome the relative membership problem of the FCM. The primary objective of the possibilistic approach is to achieve membership values that are possibilistic, i.e., the membership value of a point in a cluster represents the typicality of the point in the cluster or the possibility of the point belonging to

the cluster. This was done by modifying the objective function of FCM as follows

$$J_{NC} = \sum_{j=1}^l \sum_{k=1}^c (u_{kj})^m d_{kj} + \sum_{k=1}^c \eta_k \sum_{j=1}^l (1 - u_{kj})^m \quad (2.18)$$

In the above equation, the η_k are suitable positive numbers, which are chosen in [57] as

$$\eta_k = K \frac{\sum_{j=1}^l (u_{kj})^m d_{kj}}{\sum_{j=1}^l (u_{kj})^m} \quad (2.19)$$

Where K is a constant. Typically K is chosen to be 1 [57]. This choice makes η_k proportional to the average fuzzy intra-cluster distance of the k th cluster. There are no constraints on the memberships u_{kj} other than the requirement that they should be in $[0, 1]$. The second term forces u_{kj} to be as large as possible, thus avoiding the trivial solution. Assuming that the distances η_k are specified, the membership update equation in the PCM can be shown to be

$$u_{kj} = \frac{1}{1 + \left(\frac{d_{kj}}{\eta_k} \right)^{1/(m-1)}} \quad (2.20)$$

The updating equation for cluster center in PCM is the same as that in FCM. The equations (2.20) and (2.4) gives the possibilistic c-means (PCM) algorithm. Its ability of handling noise can be found in [57] [58] [59]. More recently, several improved PCM algorithms are developed in the literature to eliminate its tendency to find identical clusters, such as in [76] and [118].

2.3.2 Popular Cluster Validity Indices

There are so many fuzzy cluster validity indices exist in the literature, it is impractical and unnecessary to investigate all of them in this subsection. We here only address several most cited ones.

Bezdek proposed two cluster validity indices for fuzzy clustering, the partition coefficient (PC) and partition entropy (PE) [6] [7], which were defined as

$$PC(c) = \frac{1}{l} \sum_{j=1}^l \sum_{k=1}^c (u_{kj})^2 \quad (2.21)$$

and

$$PE(c) = -\frac{1}{l} \sum_{j=1}^l \sum_{k=1}^c (u_{kj}) \log(u_{kj}) \quad (2.22)$$

The optimal partition (or an optimal value of c^*) is obtained by maximizing PC (or minimizing PE) with respect to $c = 2, 3, \dots, c_{max}$ because this provides compact clusters with higher values of u_{kj} .

The fuzzy hypervolume (FHV) and partition density (PD) validity functions proposed by Gath and Geva [37] were defined by

$$FHV(c) = \sum_{k=1}^c [\det(F_k)]^{1/2} \quad (2.23)$$

and

$$PD(c) = \frac{\sum_{k=1}^c S_k}{\sum_{k=1}^c F_k} \quad (2.24)$$

where the matrix F_k defined by

$$F_k = \frac{\sum_{j=1}^l (u_{kj})^m (x_j - v_k)(x_j - v_k)^T}{\sum_{j=1}^l (u_{kj})^m} \quad (2.25)$$

denotes the fuzzy covariance matrix of cluster k ; and the “sum of central members” S_k defined by

$$S_k = \sum_{j=1}^l (u_{kj})^m, \quad \forall x_j \in \{x_j : (x_j - v_k)F_k^{-1}(x_j - v_k) < 1\} \quad (2.26)$$

takes into account only the points contributing to the core of the cluster, whose radii are the standard deviations of the cluster features. A fuzzy partition can be expected to have a low FHV (or high PD) value if the partition is tight. Thus, we find an optimal c^* by solving $\min_{2 \leq c \leq c_{max}} FHV(c)$ (or $\max_{2 \leq c \leq c_{max}} PD(c)$) to produce a best clustering performance for the data set X . Other indices based on measures of the degree of variance within each cluster are [14] and [82].

Xie and Beni proposed a validity index (XB) [112] that focused on two properties: compactness and separation, defined as

$$XB(c) = \frac{\sum_{j=1}^l \sum_{k=1}^c (u_{kj})^2 \|x_j - v_k\|^2}{l \min_{i \neq k} \|v_i - v_k\|^2} \quad (2.27)$$

The numerator indicates the compactness of the fuzzy partition, while the denominator indicates the strength of the separation between clusters. They stated that a good partition produces a small value for the compactness, and that well-separated cluster centers will produce a high value for the separation. Hence, the most desirable partition is obtained by minimizing XB for $c = 2, 3, \dots, c_{max}$.

Fukuyama and Sugeno also tried to model the cluster validation (FS) [36] by

exploiting the concepts of compactness and separation, defined as

$$FS(c) = J - K = \sum_{j=1}^l \sum_{k=1}^c (u_{kj})^m \|x_j - v_k\|^2 - \sum_{j=1}^l \sum_{k=1}^c (u_{kj})^m \|v_k - \bar{v}\|^2 \quad (2.28)$$

Where $\bar{v}$ is the mean of the cluster centers, J is a compactness measure, and K measures the degree of separation between clusters. The optimal partition is obtained by minimizing FS with respect to $c = 2, 3, \dots, c_{max}$. Based on XB and FS indices, lately developed indices, e.g., [54], [61], [73], [110], and [114], focused on the modifying or extending of the traditional compactness and separation to achieve better performances.

2.4 Spherical Shells Clustering Methods

In the last years, several fuzzy clustering based algorithms have been developed to detect and separate the shell-shapes in data structure. The fuzzy c-shells (FCS) algorithm proposed by Dave has been proved to be successful in clustering spherical shells and is further generalized to be adaptive FCS (AFCS) for the case of elliptical shells [24] [26]. Krishnapuram et al. developed the fuzzy c-spherical shells algorithm (FCSS) [56] to reduce the computational costs of the FCS by introducing an algebraic (non-Euclidean) distance measure. In this way, the prototypes can be calculated directly and the solution of coupled nonlinear equations is not necessarily the same as in the FCS. Specifically for two-dimensional (2D) cases, Man and Gath developed the fuzzy c-rings (FCR) algorithm [68], and Gath and Hoory proposed the fuzzy c-ellipses (FCE) algorithm [38] for clustering ring and ellipse data, respectively. More generically, Krishnapuram et al. developed the fuzzy c-quadric shells (FCQS) algorithm [58], which detects quadrics like circles, ellipses, hyperbolas, or lines. The clustering algorithms for detecting rectangular shells have also been developed in the literature, as e.g., the norm-induced shell prototypes (NISP) algorithm by Bezdek et al. in [9] and the fuzzy c-rectangular

shell (FCRS) algorithm by Hoeppner in [45]. In this section, we give a brief review of the spherical shells clustering algorithms with emphasis on the distance measure used in each algorithm.

Figure 2.2: Schematic drawing for calculation of the cluster prototypes. v_k is the k th cluster center, r_k is the radius of the k th cluster, x_j is the j th data point. Figure taken from [68].

2.4.1 Fuzzy C-shells Clustering Algorithm

In the FCS algorithm, the cluster prototype w_k is an n -dimensional (hyper-)spherical shell which can be characterized by its center v_k and radius r_k , where $k = 1, 2, \dots, c$. The distance from a point x_j to a prototype $w_k := (v_k, r_k)$ is defined as in [24] as

$$(D_{kj})^2 = (\|x_j - v_k\| - r_k)^2 \quad (2.29)$$

In the above equation, $\|\cdot\|$ is the Euclidean norm. Then the objective function of FCS to be minimized is defined as

$$J_{FCS} = \sum_{k=1}^c \sum_{j=1}^l (u_{kj})^m (D_{kj})^2 \quad (2.30)$$

where u_{kj} is the membership of the j th point in the k th cluster, with exponent $m \in (1, \infty)$. The minimization of (2.30) gives the updating equation for membership u_{kj} , which is given by

$$u_{kj} = \frac{1}{\sum_{i=1}^c \left[\frac{(D_{kj})^2}{(D_{ij})^2} \right]^{1/(m-1)}} \quad (2.31)$$

and the following equations are satisfactory for cluster center v_k and cluster radius r_k (the proof can be found in [24])

$$\sum_{j=1}^l (u_{kj})^m \frac{D_{kj}}{d_{kj}} (x_j - v_k) = 0 \quad (2.32)$$

$$\sum_{j=1}^l (u_{kj})^m D_{kj} = 0 \quad (2.33)$$

In the above equations, $d_{kj} = \|x_j - v_k\|$ is the distance of point x_j from cluster center v_k . The implementation of FCS involves the solution of coupled nonlinear equations (2.32) and (2.33). The author in [24] suggests the use of Newton's method with initial guess from fuzzy c-means (FCM) [8].

2.4.2 Fuzzy C-Spherical Shells Clustering Algorithm

FCSS algorithm uses the same prototype definition as that in FCS algorithm. The distance from point x_j to a prototype $w_k := (v_k, r_k)$ is defined as in [56] as

$$(D_{kj})^2 = (\|x_j - v_k\|^2 - r_k^2)^2 = p_k^T M_j p_k + q_j^T p_k + b_j \quad (2.34)$$

where $b_j = (x_j^T x_j)^2$, $q_j = 2(x_j^T x_j)y_j$, $y_j = [x_j^T \ 1]^T$, $M_j = y_j y_j^T$ and $p_k = [-2v_k^T \ v_k^T v_k - r_k^2]^T$. Note that all the variables (vectors or matrix) in (2.34) are only related to the input point x_j except the parameter p_k which is decided by a pair of parameters, i.e., the center v_k and the radius r_k . Then the objective function of FCSS to be minimized is defined as

$$J_{FCSS} = \sum_{k=1}^c \sum_{j=1}^l (u_{kj})^m (p_k^T M_j p_k + q_j^T p_k + b_j) \quad (2.35)$$

In [56] the vectors p_k are assumed to be independent of each other. Then it is easy to show that the vectors p_k that minimize (2.35) are given by

$$p_k = -\frac{1}{2} H_k^{-1} Q_k \quad (2.36)$$

where $H_k = \sum_{j=1}^l (u_{kj})^m M_j$ and $Q_k = \sum_{j=1}^l (u_{kj})^m q_j$. It also can be shown that the updating equation for membership of FCSS is given by (2.31), in which the distance is defined by (2.34) but not by (2.29). Unlike FCS, which needs to solve coupled nonlinear equations, FCSS calculates the prototypes directly, which reduces the computation cost compared to the implementation of FCS. Note although the distance used in FCSS is not correct perpendicular but an algebraic non-Euclidean measure, and is biased toward inside points [26], the results of FCSS are satisfactory.

2.4.3 Fuzzy C-rings Clustering Algorithm

In order to decrease the computational load during the detection of circular ring-shaped clusters, a different approach, the FCR algorithm, that does not require solution of nonlinear equations is suggested in [68] for the characterization of two-dimensional (2D) ring-shaped data. In order to characterize ring-shaped clusters, a vector $[r_1, r_2, \dots, r_c]^T$, where r_k is the radius of the k th ring-shaped cluster, is

incorporated into the optimization of fuzzy c-means [8] as follows:

$$J_{FCS} = \sum_{k=1}^c \sum_{j=1}^l (u_{kj})^m (d_{kj} - r_k)^2 = \sum_{k=1}^c \sum_{j=1}^l (u_{kj})^m (D_{kj})^2 \quad (2.37)$$

where $d_{kj} = \|x_j - v_k\|$ is the distance of point x_j from cluster center v_k , and D_{kj} is the translated Euclidean distance incorporating the radius r_k , which is identical to the distance measure (2.29) used in FCS. Minimization of (2.37) according to v_k , and r_k can be done iteratively, and the solution is given explicitly through the following necessary conditions (the details can be found in [68])

$$v_k = \frac{\sum_{j=1}^l (u_{kj})^m \begin{pmatrix} x_{j1} - r_k \cos \theta \\ x_{j2} - r_k \sin \theta \end{pmatrix}}{\sum_{j=1}^l (u_{kj})^m} \quad (2.38)$$

$$r_k = \frac{\sum_{j=1}^l (u_{kj})^m d_{kj}}{\sum_{j=1}^l (u_{kj})^m} \quad (2.39)$$

where x_{j1} and x_{j2} are the x-coordinate feature and y-coordinate feature of the 2D point x_j , and θ is shown in Fig 2.2. And the updating of u_{kj} is given by the equation (2.31). FCR can be easily implemented by alternative optimization, which is computationally much more efficient than the FCS and does not suffer from a highly non-Euclidean distance measure like the FCSS. FCR, however, is limited to 2D ring-shaped data. In contrast, FCS and FCSS algorithms are general for clustering any dimensional (hyper-)spherical shells.

2.5 Kernel-based Clustering Methods

Most traditional clustering techniques only work well when the underlying clusters exhibit linearly separable shapes. However, they will generate improper results if the underlying data shapes are linearly nonseparable. So far, this kind of clustering problem remains a challenging issue in the literature. The kernel methods, which has drawn considerable attention in recent years due to its good generalization ability for a wide range of applications and better performance compared to other classical leaning machines [23] [53] [72] [93], may offer a possible solution for linearly nonseparable data clustering problem. By nonlinearly mapping the observed data from a low-dimensional input space into a high-dimensional feature space by means of a given kernel function, which aims to increase the probability of the linear separability of the given data, the kernel-based clustering algorithms can reveal structure in the data that may go unnoticed by traditional clustering algorithms in the standard Euclidean space. This section introduces the basic concept of kernel methods and its usage for data clustering.

2.5.1 Nonlinear Mapping

Algorithms in feature spaces make use of the following idea: via a nonlinear mapping

$$\begin{aligned}\phi : \quad R^n &\rightarrow F \\ x_j &\rightarrow \phi(x_j), \quad j = 1, 2, \dots, l\end{aligned}\tag{2.40}$$

the data $x_1, \dots, x_l \in R^n$ is mapped into a potentially much higher dimensional feature space F . For a given learning problem one now considers the same algorithm in F instead of R^n , i.e., one works with the sample

$$\phi(x_1), \dots, \phi(x_l) \in F$$

instead of

$$x_1, \dots, x_l \in R^n$$

This is also implicitly done for neural networks, radial basis networks or boosting algorithms where the input data is mapped to some representation given by the hidden layer.

Figure 2.3: Toy example of a binary classification problem mapped into feature space. In two dimensions a rather complicated nonlinear decision surface is necessary to separate the classes (a), whereas in a feature space the decision surface becomes a hyperplane (b). Figure taken from [93].

Given this mapped representation, a simple clustering or classification in F is to be found if the data structure in F is simplified. All the variability and richness that one needs to have a powerful clustering or classification is then introduced by the mapping ϕ . Intuitively, this idea can be understood from the toy example in Fig 2.3: in two dimensions a rather complicated nonlinear decision surface is necessary to separate the classes (linearly nonseparable case from the view of data clustering), whereas in a feature space of second-order monomials

$$\begin{aligned} \phi : \quad R^2 &\rightarrow R^3 \\ (x_1, x_2) &\rightarrow (z_1, z_2, z_3) := (x_1^2, \sqrt{2}x_1x_2, x_2^2) \end{aligned} \tag{2.41}$$

all one needs for separation is a linear hyperplane (linearly separable case from the view of data clustering). In this simple toy example, we can easily control both: the statistical complexity (by using a simple linear hyperplane classifier) and the algorithmic complexity of the learning machine, as the feature space is only three dimensional. However, it becomes rather tricky to control the latter for large real-world problems: for n -dimensional input patterns, there exist [72]

$$N_F = \binom{d+n-1}{d} = \frac{(d+n-1)!}{d!(n-1)!} \quad (2.42)$$

different monomials of degree d , comprising a feature space F of dimension N_F . For instance, 16×16 pixels and a monomials degree $d = 5$ thus yield a dimension of almost 10^{10} . This problem can be solved by the so-called *kernel trick* as discussed below.

2.5.2 Kernel Trick

For certain feature spaces F and corresponding mappings ϕ there is a highly effective trick for computing the scalar products in feature spaces using kernel functions [22] [90]. For above toy example, the computation of a scalar product between two feature space vectors, can be readily reformulated in terms of a kernel function denoted by k

$$\begin{aligned} (\phi(x) \cdot \phi(y)) &= (x_1^2, \sqrt{2}x_1x_2, x_2^2)(y_1^2, \sqrt{2}y_1y_2, y_2^2)^T \\ &= ((x_1, x_2)(y_1, y_2)^T)^2 = (x \cdot y)^2 =: k(x, y) \end{aligned} \quad (2.43)$$

In other words, the desired kernel k is simply the square of the dot product in input space. This finding generalizes [72]:

- (1) For $x, y \in R^n$, and $d \in N$, the kernel function $k(x, y) = (x \cdot y)^d$ computes a

scalar product in the space of all products of d vector monomials of x and y .

- (2) If k is a continuous kernel of a positive integral operator on a Hilbert space $L_2(C)$ on a compact set $C \in R^n$, i.e.,

$$\forall f \in L_2(C) : \int_{C \times C} k(x, y) f(x) f(y) dx dy \geq 0$$

then there exists a space F and a mapping $\phi : R^n \rightarrow F$ such that $k(x, y) = (\phi(x) \cdot \phi(y))$.

The most used kernel functions are Gaussian Radial Basis Function (*RBF*)

$$k(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\sigma^2} \quad (2.44)$$

and the Polynomial Function

$$k(x_i, x_j) = (x_i \cdot x_j)^d \quad (2.45)$$

the width $2\sigma^2$ in Gaussian *RBF* kernel function and the power d in polynomial kernel function are specified *a priori* by the user.

The interesting point about kernel functions is that the scalar product can be implicitly computed in F , without explicitly using or even knowing the mapping ϕ . So, kernels allow to compute scalar products in feature spaces, where one could otherwise hardly perform any computation. A direct consequence from this finding is [72] [90]: every (linear) algorithm that only uses scalar products can implicitly be executed in F by using kernels, i.e., one can very elegantly construct a nonlinear version of a linear algorithm.

2.5.3 Data Clustering in Feature Space

In the last years, a number of powerful kernel-based learning machines, e.g., support vector machines (SVMs) [22] [106], kernel Fisher discriminant (KFD) [3], [70],

and kernel principal component analysis (KPCA) [90] [91], have been proposed. These approaches have shown practical relevance not only for classification and regression problems (among kernel-based supervised learning, the most relevant one is SVMs, we will discuss it later in chapter 8.) but also, more recently, in unsupervised learning, especially in data clustering as shown below. The existing kernel-based clustering methods can be classified into two categories: one is to perform the standard clustering algorithms in the feature space in terms of kernel-induced distances, such as in [39] [41] [62] [89] [111] [119] [121]; the other is to use the support vectors to define a sphere or multiple spheres (this sphere(s), when mapped back to data space, can be separated into several components, each enclosing a separate cluster of points.), such as in [4] [15] [19] [47]. Both methods have demonstrated superiority over the traditional clustering algorithms. To demonstrate the basic concept of data clustering in feature space, we generalize the standard K-means into kernel-based version by using kernel-induced distance as follows.

Figure 2.4: Comparisons of K-means and kernel-based K-means on a linearly non-separable data set.

K-means Algorithm

K-means is an unsupervised learning algorithm that partitions the data set into a selected number of clusters under some optimization measures. For example,

we often want to minimize the sum of squares of the Euclidean distance between the samples and the centroids (centers). The assumption behind this measure is the belief that the data space consists of isolated elliptical regions. However, such assumption isn't always held on specific applications. To tackle this problem, one idea is to use kernel based method by mapping the data to a higher dimensional space that satisfies the requirement of the optimization measure.

Suppose the data set has l samples $x_1, \dots, x_l$. K-means algorithm aims to partition the l samples into c clusters $V_1, \dots, V_c$, and returns the center of each cluster $v_1, \dots, v_c$ as the representatives of the data set. The steps of the K-means algorithm are described in brief.

Initialization: Choose c initial cluster centers $v_1, \dots, v_c$ randomly from the l points $x_1, \dots, x_l$.

Do loop

step 1. Assign point $x_j, j = 1, \dots, l$ to the cluster V_k with nearest center v_k

$$\|x_j - v_k\|^2 \leq \|x_j - v_i\|^2, \quad i \neq k, \quad i, k \in \{1, 2, \dots, c\} \quad (2.46)$$

step 2. Update cluster centers v_k

$$v_k = \frac{1}{|V_k|} \sum_{x_j \in V_k} x_j, \quad k = 1, \dots, c \quad (2.47)$$

Until cluster belonging no longer changes. where $|V_k|$ is the number of elements belonging to cluster V_k . Note that in case the process does not terminate normally, then it is executed for a maximum fixed number of iterations.

Kernel-based K-means Algorithm

The key issue extending traditional K-means to kernel-based K-means is the computation of distance in the feature space. Let $\phi(x_i)$ denote x_j 's transformation,

and $\phi_k(x), k = 1, \dots, c$ be the cluster centers in feature space. According to [89], we have

$$\phi_k(x) = \frac{1}{|V_k|} \sum_{x_j \in V_k} \phi(x_j) \quad (2.48)$$

then the distance between $\phi(x_j)$ and $\phi_k(x)$ is expressed as

$$\begin{aligned} D_k(x_j) &= \|\phi(x_j) - \phi_k(x)\|^2 \\ &= \phi(x_j)\phi(x_j) - 2\frac{1}{|V_k|} \sum_{x_m \in V_k} \phi(x_j)\phi(x_m) + \frac{1}{|V_k|^2} \sum_{x_m, x_n \in V_k} \phi(x_m)\phi(x_n) \\ &= k(x_j, x_j) - 2\frac{1}{|V_k|} \sum_{x_m \in V_k} k(x_j, x_m) + \frac{1}{|V_k|^2} \sum_{x_m, x_n \in V_k} k(x_m, x_n) \end{aligned} \quad (2.49)$$

where $|V_k|$ is the number of elements belonging to cluster V_k in feature space, $k(x, y) = \phi(x)\phi(y)$ is the kernel function.

Applying the above kernel-induced distance to traditional K-means, we obtain the kernel-based K-means algorithm:

Initialization: Choose c initial cluster centers $\phi_1(x), \dots, \phi_c(x)$ randomly from the l mapped points $\phi(x_1), \dots, \phi(x_l)$.

Do loop:

1. Assign $\Phi(x_j)$ to the cluster V_k with nearest center $\phi_k(x)$

$$\|\Phi(x_j) - \phi_k(x)\|^2 \leq \|\Phi(x_j) - \phi_i(x)\|^2, \quad i \neq k, \quad i, k \in \{1, \dots, c\} \quad (2.50)$$

2. Update cluster centers $\phi_k(x)$

$$\phi_k(x) = \frac{1}{|V_k|} \sum_{x_j \in V_k} \Phi(x_j), \quad k = 1, \dots, c \quad (2.51)$$

Until cluster belonging no longer changes. By using the kernel-induced distance

measure, the kernel-based K-means can reveal structure in the data that may go unnoticed by the traditional K-means in the standard Euclidean space, as demonstrated in Fig 2.4. From the figure, it is obvious that traditional K-means fails to partition this linearly nonseparable data, while the kernel-based K-means correctly reveals the underlying data structure. More comparison results between kernel-based clustering methods and traditional clustering methods can be found in chapter 5.

Chapter 3

A Robust Deterministic Annealing Algorithm for Data Clustering

3.1 Introduction

The deterministic annealing (DA) algorithm, which is originally proposed by K. Rose in [85] and further explored in [87], is independent of the choice of the initial data configuration and has the ability to avoid poor local optima. The DA approach to clustering and its extensions has demonstrated substantial performance improvement over standard supervised and unsupervised learning methods [71] [86] [87]. However, in addition to the requirement of a pre-selected cluster number, the DA algorithm is not robust against noise and outliers because it tends to associate any data point (even it is a noisy one) in all clusters [29]. This chapter aims to improve the robustness of standard DA approach, i.e., robustness against noisy points and auto-determination of optimal cluster number. The basic idea of noise clustering (NC) [25] [30] is used for the standard DA approach to achieve the robustness against noisy data points. Furthermore, the optimal cluster number for a given

data set is auto-determined by a cluster validity, i.e., VC(Vapnik-Chervonenkis)-bound, which is implemented based on the structural risk minimization (SRM) principle [107] in statistical learning theory.

The rest of this chapter is organized as follows. The subsequent section contains the development of the proposed robust deterministic annealing (RDA) algorithm: we first present the formulation and derivation of RDA clustering algorithm, then study the VC-bound cluster validity to determine the optimal cluster number for a given data set, and finally give the detailed pseudo-code of RDA for data clustering. The experimental results of the proposed clustering method are reported in section 3. Finally, conclusion is presented in section 4.

3.2 The Proposed RDA Clustering Algorithm

A brief review of traditional noise clustering (NC) and deterministic annealing (DA) algorithms can be found in the chapter 2, which may be useful for a better understanding of the proposed robust deterministic annealing (RDA) clustering algorithm in this section.

3.2.1 Formulation and Derivation

Assuming there are c good clusters in the data set, we add one more cluster, i.e., the $(c + 1)th$ cluster as the noise cluster. Then the Lagrangian formulation of DA becomes

$$F' = J'_e - TH'_s \quad (3.1)$$

where cost function J'_e is defined by

$$J'_e = \sum_{j=1}^l \sum_{k=1}^{c+1} p(x_j) p(v'_k | x_j) d(x_j, v'_k) \quad (3.2)$$

and Shannon entropy H'_s is defined by

$$H'_s = - \sum_{j=1}^l \sum_{k=1}^{c+1} p(x_j) p(v'_k | x_j) \log p(v'_k | x_j) \quad (3.3)$$

where the distance $d(x_j, v'_k)$ in (3.2) are defined by

$$d(x_j, v'_k) = (x_j - v'_k)^T (x_j - v'_k), \quad k = 1, 2, \dots, c \quad (3.4)$$

and

$$d(x_j, v'_k) = \delta, \quad k = c + 1 \quad (3.5)$$

where δ is the constant noise distance from the noise cluster to each data point. Equation (3.5) means all the points in the data set are at the same distance δ from the noise cluster. Then, the probability distribution $p(v'_k | x_j)$ that minimizing F' is given by

$$p(v'_k | x_j) = \frac{p(v'_k) e^{-\frac{d(x_j, v'_k)}{T}}}{\sum_{i=1}^{c+1} p(v'_i) e^{-\frac{d(x_j, v'_i)}{T}}}, \quad k = 1, 2, \dots, c + 1 \quad (3.6)$$

where $p(v'_k)$ is the mass probability of the k th cluster defined by

$$p(v'_k) = \sum_{j=1}^l p(x_j) p(v'_k | x_j), \quad k = 1, 2, \dots, c + 1 \quad (3.7)$$

The corresponding minimum of F' is obtained by plugging (3.6) back into (3.1), which is given by

$$F^{**} = \min_{\{p(v'_k|x_j)\}} F' = -T \sum_{j=1}^l p(x_j) \log \sum_{k=1}^{c+1} p(v'_k) e^{-\frac{d(x_j, v'_k)}{T}} \quad (3.8)$$

The free parameter v'_k that optimizes F^{**} should satisfy

$$\frac{\partial}{\partial v'_k} F^{**} = 0 \quad (3.9)$$

i.e.,

$$\sum_{j=1}^l p(x_j) \frac{p(v'_k) e^{-\frac{d(x_j, v'_k)}{T}}}{\sum_{i=1}^{c+1} p(v'_i) e^{-\frac{d(x_j, v'_i)}{T}}} (x_j - v'_k) = 0 \quad (3.10)$$

Use (3.6) the above equation can be rewritten as

$$\sum_{j=1}^l p(x_j) p(v'_k|x_j) x_j = \sum_{j=1}^l p(x_j) p(v'_k|x_j) v'_k \quad (3.11)$$

then we get the expression of good cluster centers by

$$v'_k = \frac{\sum_{j=1}^l p(x_j) p(v'_k|x_j) x_j}{\sum_{j=1}^l p(x_j) p(v'_k|x_j)}, \quad k = 1, 2, \dots, c \quad (3.12)$$

Alternatively updating equations (3.6) and (3.12) with phase transition (see Fig 2.1 and related discussions in chapter 2) gives the proposed robust deterministic annealing (RDA) clustering algorithm. The main difference in this formulation is that the constraint on the association probability of good clusters is now equivalent

Figure 3.1: Clustering results of DA and RDA with an added outlier coordinate (50,0), a simple demonstration of the robustness of RDA against noisy points. Note the different locations of the cluster centers (denoted by “*”) obtained by DA and RDA.

to

$$0 \leq \sum_{k=1}^c p(v'_k | x_j) \leq 1 \quad (3.13)$$

instead of the following in the standard DA algorithm

$$\sum_{k=1}^c p(v_k | x_j) = 1 \quad (3.14)$$

This makes noisy points have less effect to good clusters such that the proposed RDA is robust against noise and outliers to some extent [29], as demonstrated in Fig 3.1. Where the positions of cluster centers obtained by DA are obviously distorted by the outlier, while the positions of cluster centers obtained by RDA are hardly affected. More demonstrations are given in the section of experimental results.

RDA optimization begins by determining the minimum of the free energy F^{**} at high values of T and attempts to track the minimum through lower values of T , until the minimum of the free energy at $T \rightarrow 0$ coincides with the minimum of the original cost function J'_e . It is known for deterministic annealing approach that

splits in the cluster representation occur during the annealing process [40] [87]. These splits are related to qualitative changes in the optimization problem and have to be taken into account in the annealing process. The critical temperature for phase transition of the k th ($k = 1, 2, \dots, c$) good cluster can be determined by the maximum eigenvalue of covariance matrix, see equation (2.10) in chapter 2.

The noise distance δ is an important parameter which should be defined in the proposed algorithm. From the derivation of RDA, it can be observed that if the δ is chosen to be very small, then most of the points will get classified as noise, while if δ is very large, then most of the points will be classified into good clusters other than the noise cluster. Since there is no prior information of the underlying data structure, it is normally defined based on the statistic of the distance distributions, as done in [25] [29] [57] [117]. In [25], Dave recommended the use of the scaled average distance over all data points and good clusters as follows,

$$\delta = \lambda \left[\frac{\sum_{j=1}^l \sum_{k=1}^c d(x_j, v'_k)}{lc} \right] \quad (3.15)$$

It was addressed in [25] [27] that the clustering results are not very sensitive to the range of the values of λ , however, how to auto-select an appropriate value of λ for a given data remains undiscussed. In fact, the choice of λ may have a great impact on what is classified into good clusters and what is classified into outliers.

To make the proposed method more practical, a desired definition of the noise distance δ should not introduce additional parameter(s); or if the additional parameter(s) is inevitable, we hope the clustering result is hardly sensitive to it. In practice, we find that the following definition works well,

$$\delta = K \frac{\sum_{j=1}^l \min_{k=1,2,\dots,c} d(x_j, v'_k)}{l} \quad (3.16)$$

This choice makes δ proportional to the average distance of every point to its nearest good cluster. Typically, K is chosen to be a constant 1, which is used in this chapter. The effectiveness of this definition is verified by experimental results, see next section.

3.2.2 Optimal Cluster Number Estimation

A natural question of clustering is how many clusters are appropriate for the description of a given data. In the above discussions, we have assumed the cluster number c is known *a priori*, which may be impractical for real applications. Many cluster validity criteria, for example of the famous Xie-Beni validity [112], have been presented in the literature to determine the optimal cluster number for fuzzy clustering algorithms. These criteria, however, are not originally proposed for robust data clustering and may fail in the cases when there exist noisy points in the data, see example 1 in chapter 6. We follow the statements in [99] to present a practical cluster validity criterion specifically for the proposed RDA algorithm to auto-determine the optimal cluster number for a given data. The cluster validity called VC-bound is implemented based on the structural risk minimization (SRM) principle [107], with the compromise between the minimum empirical risk (noise ratio) and the VC-dimension (model complexity) as discussed below.

According to the information theory [11] [12], a well designed communication channel should have few unreliable input data points to achieve the maximum capacity. This implies that a good clustering algorithm with the correct cluster number should produce few outliers for the underlining nature clusters. However, few outliers can only guarantee small empirical risk but not real risk (generalization) from the view of statistical learning theory [107]. Increasing cluster number (as the temperature is lowered in RDA clustering) normally reduces the empirical error (especially when the cluster number is bigger than the underlying cluster number of the given data) but increases the model complexity, a good cluster validity criteria should make a tradeoff between the empirical risk and model complexity, as

the structural risk minimization (SRM) principle [107] does. The SRM principle finds the function that for the fixed amount of data achieves the minimum of the guaranteed risk. Under the SRM, a set of admissible structures with the nested subsets is defined by $S_1 \subset S_2 \subset \dots \subset S_c, \dots$, with the non-decreasing VC-dimension of the structure $h_1 \leq h_2 \leq \dots \leq h_c, \dots$, where $S_c = (Q^c(x_j, V) : V \in \Lambda_c), \forall j$ denotes the element of the structure at cluster number c , with a set of indicator functions $Q^c(x_j, V)$ of the empirical risk.

The task of the SRM principle is to choose the element S_c of the structure for which the smallest bound on the real risk (the smallest guaranteed risk)

$$V_b \leq R_{ems} + \frac{\varepsilon}{2} \left(1 + \left(1 + R_{ems} \frac{4}{\varepsilon}\right)\right)^{1/2} \quad (3.17)$$

with

$$\varepsilon = \frac{h_c(\log \frac{2l}{h_c} + 1) - \log \frac{\zeta}{4}}{l} \quad (3.18)$$

is achieved. In the above equation, $\zeta < 1$ is a constant. The first term of the right hand side of the bound (3.17) represents the empirical risk, which can be naturally represented by noise ratio as

$$R_{ems} = l_o/l \quad (3.19)$$

where l_o is the number of outliers identified by RDA clustering algorithm in the last subsection. The second term is the confidence interval of the SRM based estimation, whose calculation is discussed below. The bound on the risk is the sum of the empirical risk and of the confidence interval. The smallest bound of the risk is achieved on some appropriate element of the structure.

Assume the given data is partitioned by the RDA clustering algorithm, which results in certain partitions with according probabilities $p(v'_k|x_j)$ ($k = 1, 2, \dots, c+1$

and $j = 1, 2, \dots, l$) within the cluster number range ($c = 2, 3, \dots, c_{max}$). The key issue of applying (3.17) for cluster number selection is to construct a set of indicator functions specifically for RDA clustering algorithm, which is given by

$$Q^c(x_j, V) = \sum_{k=1}^{c+1} \lim_{T \rightarrow 0} p(v'_k | x_j) = \sum_{k=1}^{c+1} \lim_{T \rightarrow 0} \frac{p(v'_k) e^{-d(x_j, v'_k)/T}}{\sum_{i=1}^{c+1} p(v'_i) e^{-d(x_j, v'_i)/T}} \quad (3.20)$$

At limited low T , the RDA clustering becomes hard with each data point assigned to the nearest cluster with probability one, such that the indicator $Q^{c+1}(x_j, V)$ can be approximated by the step function, which is linear in parameters. Please refer to chapter 6 for the detailed derivations. Then according to [107], the VC-dimension (denoted by h_c) of the model complexity is equal to the number of parameters, i.e.,

$$h_c = (c + 1) \times n \quad (3.21)$$

for each nested subset S_c , where n is the dimensionality of the input data. From the above equation, it can be seen that the increase of the estimated VC-dimension is proportional to the increase of the cluster number. To obtain good generalization performance one has to use the admissible structure based on the set of indicator functions to search for an optimal cluster number that minimizes a VC-bound (denoted by V_b) similar to the support vector machine [107], except that we are looking for the strongest data points in the input space instead of seeking the weakest data points in the feature space. In conclusion, the cluster validity is stated as: by evaluating the estimated VC-bound for each chosen cluster number by equation (3.17), we select the one that yields the minimum value of V_b as the optimal cluster number.

3.2.3 Pseudo-code of RDA algorithm

According to the above discussions, we give a detailed pseudo-code of RDA for data clustering as follows.

- Step 1) Set the maximum number of clusters c_{max} , initial temperature $T_{ini} > 2T_1^*$ (see (2.10)), minimum temperature $T_{min} = T_{ini}/1000$, cluster center $v_1 = \frac{1}{l} \sum_{j=1}^l x_j$ with mass probability $p(v_1) = 1$, source distribution $p(x_j) = \frac{1}{l}$ ($j = 1, 2, \dots, l$), and $c = 1$.
- Step 2) Alternatively update titled distribution (3.6), mass distribution (3.7) and cluster centers (3.12) for $k = 1, 2, \dots, c$ (fixed point iterations) until the maximum change in the cluster centers between consecutive iterations is less than the given threshold ε , where ε is small positive value.
- Step 3) Calculate and verify the value of the VC-bound (3.17) within the range of maximum cluster number c_{max} . If the minimum of V_b is found, then identify noise cluster, set $T \rightarrow 0$ and perform the last iteration, then stop. Otherwise, go to next step.
- Step 4) If $T < T_{min}$, set $T \rightarrow 0$ and perform the last iteration, then stop. Otherwise, let $T = \eta T$ ($0 < \eta < 1$), and check condition for phase transition for $k = 1, 2, \dots, c$, if critical T_k^* is reached for cluster k (see (2.10)), add a new cluster center by $v_{k+1} = v_k + \Delta$ with $p(v_{k+1}) = p(v_k)/2$ and $p(v_k) = p(v_k)/2$, where Δ is a small disturbance, let $c \leftarrow c + 1$, and go to step 2.

Here we mention several remarks for the implementation of RDA clustering algorithm:

- 1) In practical applications, the data points are usually assumed to be independent of each other. In other words, $p(x_j) = \frac{1}{l}$ is necessarily assumed for the implementation of RDA.
- 2) The choice of annealing step η depends on the users priority: smaller values of η lead to faster computation at the risk of lower quality of the results and vice

versa. In practice, the values of η in the range of $[0.8, 0.95]$ seem to get the best clustering results.

3) The critical temperature T_1^* in step 1 and T_k^* in step 4 can be obtained through a simple perturbation method. In this case we always keep two centers in one cluster and perturb them when we update T . Until the critical temperature is reached they will be merged together by the iterations. At phase transition they will move further apart.

Figure 3.2: The plot of VC-bound with different cluster number for data sets in example 1. V_b reaches the minimum at $c^* = 3$ for both data sets.

3.3 Experimental Results

This section demonstrates the effectiveness and superiority of the the proposed RDA for robust data clustering by using both artificial and real data sets. We firstly discuss the effectiveness of VC-bound for determining the optimal cluster number, then compare the performances between DA and RDA, followed by the comparison between NC and RDA, and finally investigate two real data sets: Iris and Wine. In all experimental results, the original data points are marked by symbol “o”, the original centers of good cluster (assume the noise is artificially

Figure 3.3: The partitioned results and values of VC-bound of RDA with different cluster number (from 2 to 6) for the first data set in example 1. V_b reaches the minimum at $c^* = 3$ corresponding to the most reasonable partitioned result.

removed) are marked by symbol “ $\star$ ”, the partitioned clusters are randomly marked by different symbols (“o”, “+”, “x”, “ ∇ ”, “ $\triangle$ ”, and “ $\diamond$ ”), the partitioned cluster centers are marked by big symbol “ $\star$ ”, and the partitioned noisy points are marked by symbol “.”.

Before showing the experimental results, we give the values of the parameters for three clustering methods. In all examples, we set $\epsilon = 0.001$ (convergence threshold for all methods), $\eta = 0.9$ (cooling rate for DA and RDA), and $m = 2$ (exponent for NC). Specifically for NC algorithm, the value of multiplier parameter $\lambda = 0.10$ used in example 3 is determined by experimentations, namely, we investigate different values of λ and select the one that yielded the best (or most reasonable) results. The cluster number c is known *a priori* for DA and NC algorithms but auto-determined by VC-bound for RDA algorithm.

Figure 3.4: The partitioned results and values of VC-bound of RDA with different cluster number (from 2 to 6) for the second data set in example 1. V_b reaches the minimum at $c^* = 3$ corresponding to the most reasonable partitioned result.

3.3.1 Example 1: Optimal Cluster Number

As discussed in last section, the optimal cluster number for a given data set is auto-determined by VC-bound in RDA algorithm: the minimum of V_b corresponds to the optimal cluster number. The first data set considered in this example is generated from three Gaussian distributions as shown in Fig 3.3a. We calculate and plot the values of V_b by equation (3.17) with different cluster number (from 2 to 6) for this data set in Fig 3.2a. The partitioned results of RDA for different cluster number are plotted in Fig 3.3b-f for visualization. As observed from Fig 3.2a and Fig 3.3, the VC-bound successfully finds the optimal cluster number for this data set: V_b reaches the minimum at $c^* = 3$, which corresponds to the most reasonable partitioned result. The second data set tested in this example is the noisy version of the first one as shown in Fig 3.4a, where lots of noisy points exist

Figure 3.5: The plot of VC-bound with different cluster number for data sets in example 2. V_b reaches the minimum at $c^* = 2$ for noisy data set 1 and $c^* = 4$ for noisy data set 2, respectively.

in the data set. The calculated values of VC-bound with different cluster number are shown in Fig 3.2b, and the partitioned results of RDA for different cluster number are plotted in Fig 3.4b-f. Similarly, the VC-bound successfully reveals the optimal cluster number for this noisy data set: V_b reaches the minimum at $c^* = 3$, the underlying cluster number of this noisy data set.

Table 3.1: The positions of original and partitioned cluster centers for the data sets in Fig 3.6.

	Original Positions	Partitioned Positions	
		DA	RDA
Center 1 in Fig 3.6a	(4.0000, 4.0000)	(4.1625, 4.1846)	(4.0000, 4.0000)
Center 2 in Fig 3.6a	(8.0000, 4.0000)	(7.8375, 4.1845)	(8.0000, 4.0000)
Center 1 in Fig 3.6d	(1.5000, 1.5000)	(1.5053, 1.9142)	(1.4820, 1.4079)
Center 2 in Fig 3.6d	(1.5000, -1.5000)	(1.6655, -2.0192)	(1.5403, -1.4977)
Center 3 in Fig 3.6d	(-1.5000, 1.5000)	(-1.2415, 2.4991)	(-1.3658, 1.6073)
Center 4 in Fig 3.6d	(-1.5000, -1.5000)	(-1.2463, -2.1721)	(-1.3725, -1.5774)

Figure 3.6: Clustering noisy data sets, comparison between DA and RDA. The values of original and partitioned cluster centers are shown in Table 3.1.

3.3.2 Example 2: DA vs RDA

Example 2 aims to show the robustness of RDA against noisy points (including noise and outliers) through the comparison of the performances between DA and RDA. The first data set tested here contains two apparent clusters and three noisy points as shown in Fig 3.6a. The second data set consists of two assumed clusters surrounded by many noisy points as shown in Fig 3.6d. We plot the values of VC-bound with different cluster number for both data sets in Fig 3.5. It is obvious that V_b reaches the minimum at $c^* = 2$ for the first data set and $c^* = 4$ for the second data set, such that reveals the correct cluster number for both data sets. With the optimal cluster number, the partitioned results of DA and RDA for these two data sets are shown in Fig 3.6b-c and Fig 3.6e-f, respectively. We can see that the partitioned cluster centers of DA are distorted by the noisy points as shown

Figure 3.7: The plot of VC-bound with different cluster number for data sets in example 3. V_b reaches the minimum at $c^* = 3$ for both data sets.

in Fig 3.6b and Fig 3.6e, which implies that DA is sensitive to noise (or outliers). While the proposed RDA has better performance than DA, the partitioned cluster centers of RDA is hardly distorted by the noisy points as shown in Fig 3.6c and Fig 3.6f. For quantitative investigation, the positions of original and partitioned cluster centers for both data sets are concluded in Table 3.1. It can be seen that the proposed RDA algorithm not only correctly partitions the data, but also reveals the positions of cluster centers with good accuracy. It should be noted that for clear (no noisy points exist) data set, DA and RDA have similar performances and the partitioned cluster centers of these two methods almost coincide.

3.3.3 Example 3: NC vs RDA

This example aims at verifying that RDA is insensitive to data initialization and has better performance for unbalanced data set through the comparison of the performances between NC¹ and RDA. One balanced data (that means each cluster has similar number of points) as shown in Fig 3.8a and one unbalanced data set

¹We run NC algorithm 20 times for each data set with random initialization due to its insensitivity to data initialization.

Figure 3.8: Clustering balanced and unbalanced data sets, comparison between NC and RDA. Where $x/20$ means the according result occurs x times out of 20 runs.

(that means some clusters have much more points than that of other clusters) as shown in Fig 3.8e are tested in this example. We plot the values of VC-bound with different cluster number for both data sets in Fig 3.7. It can be observed that V_b reaches the minimum at $c^* = 3$ for both data sets. With the estimated cluster number, the partitioned results of NC and RDA for each data set are shown in Fig 3.8b-d and Fig 3.8f-h respectively. For the balanced data set, most runs (12 out of 20) of NC can obtain a satisfactory result as shown in Fig 3.8c, but the result of other runs (8 out of 20) is unreasonable as shown in Fig 3.8b. When considering the unbalanced data set, only few runs (1 out of 20) of NC can obtain a satisfactory result as shown in Fig 3.8g, but the results of most runs (19 out of 20) are unreasonable, one of them is shown in Fig 3.8f: the dominating cluster is divided into three clusters and the two small clusters are classified into the noise cluster. While for the proposed RDA algorithm, all runs reach a certain result for each given data set with independence of the data initialization and the cluster volumes: the partitioned result for the balanced data set is shown in Fig 3.8d and the partitioned result for the unbalanced data set is shown Fig 3.8h. Note in

Figure 3.9: RDA on Iris and Wine data sets. V_b reaches the minimum at $c^* = 3$ for both data. The number of misclassified points are 11 for Iris data and 5 for Wine data.

this example the value of multiplier parameter $\lambda = 0.10$ for NC is determined by experimentations.

3.3.4 Example 4: RDA on Real Data Sets

The purpose of this subsection is to examine the usefulness of RDA for real-life data. The Iris and Wine data investigated here are standard benchmarks in the pattern recognition literature and can be obtained from the UCI repository [105]. The large image segmentation data from the same repository is also investigated to test the RDA's performance on large data sets.

The Iris data set contains three classes of 50 points each, where each class refers to a type of iris plant. One class is linearly separable from the other two; the

Table 3.2: The partition errors of the proposed RDA algorithm and several traditional clustering methods on Iris and Wine data. Note we run NC/FCM/K-means 20 times for each data set with random initialization and show only the best result here.

	NC	FCM	K-means	DA	RDA
Iris Data	11	16	16	16	11
Wine Data	7	5	5	5	5

latter are not linearly separable from each other. The plot of Iris data by its two dominating features (i.e., PL - petal length and PW - petal width) is shown in Fig 3.9a for visualization. The VC-bound reaches the minimum at $c^* = 3$ for this data as shown in Fig 3.9b, such that correctly reveals the underlying cluster number. The partitioned result of RDA algorithm with $c^* = 3$ is shown in Fig 3.9c by Iris's two dominating features. From the view of classification, the partition of RDA has 11 errors, which is a little better than the results of other nonparametric clustering algorithms: the support vector clustering algorithm [4] leads to 14 misclassifications and the alternative fuzzy c-means algorithm [109] algorithm has 13 misclassifications.

As a second real data set we present results from a wine recognition problem. The Wine data consists of 178 13-dimensional samples which are a set of chemical analysis of three types of wine with number of samples 59, 71, and 48 respectively. For visualization, the Wine data is shown in Fig 3.9d by its first two principal components (taken from [84]). The calculated values of VC-bound with different cluster number is plotted in Fig 3.9e, where the minimum of v_b is reached at $c^* = 3$, such that correctly revealing the underlying cluster number. The partitioned result of RDA algorithm with $c^* = 3$ is shown in Fig 3.9f by Wine's first two principal components. From the view of classification, the partition of RDA has 5 errors. Our result compared favorably with other unsupervised clustering algorithms: the minimum-entropy data partitioning method [84] leads to 5 misclassifications and the Mercer kernel-based clustering method [39] has 4 misclassifications.

We also compare the partition errors of the proposed algorithm to DA, NC, and

Table 3.3: The numerical results of VC-bound (V_b) with different cluster number ($c = 2, 3, \dots, 10$) on the image segmentation data.

c	2	3	4	5	6	7	8	9	10
V_b	1.1652	0.9549	0.9384	0.9163	0.9057	0.8873	0.9108	0.9182	0.9893

several other traditional clustering methods, like fuzzy c-means (FCM) and K-means, on the above two real life data. The numerical results are concluded in Table 3.2. Please note that we run NC/FCM/K-means 20 times for each data set with random initialization and show only the best result in the table. For the Iris data, the result of RDA is much better than that of DA, FCM and K-means, and the same as that (the best result) of NC. For the Wine data, RDA achieves the same result as that of DA, FCM, and K-means, and a little better result than that of NC. That indicates: 1) there are no noisy points in the Wine data, so some robust clustering methods, such as NC, is not recommended for this data set; 2) for the clean data set (no noisy points exist), RDA has the same (or similar) performance as the traditional DA algorithm, so the proposed RDA is suitable for both noisy and clean data sets.

The final effort aims to test the performance of the proposed RDA on large data sets. The image segmentation data set from UCI repository [105] is considered here. The image segmentation data was created and donated to the UCI repository by Vision Group, University of Massachusetts. There are 2310 (Training data: 210; Test data: 2100) instances in the data set. The instances were drawn randomly from a database of 7 (brickface, sky, foliage, cement, window, path, grass) outdoor images. The images were hand-segmented to create a classification for every pixel. Each instance has 19 continuous attributes. The numerical values of V_b by using the proposed RDA with different cluster number (from 2 to 10) on this data set is reported in Table 3.3. It can be seen that the minimum of $V_b = 0.8873$ is reached at $c^* = 7$, such that correctly finding the optimal cluster number for the image segmentation. The number of misclassified instances of RDA with the optimal cluster number $c^* = 7$ on the image segmentation data is not reported here. We

think it is not meaningful to investigate the clustering accuracy of the proposed RDA on the image segmentation data because there are none comparable results reported in the literature.

3.4 Conclusion

In this chapter, we have proposed a robust deterministic annealing (RDA) algorithm to improve the performance of standard deterministic annealing (DA) algorithm for data clustering: the robustness against noise and outliers is obtained by using the basic idea of noise clustering algorithm, and the optimal cluster number for a given data is auto-determined by the cluster validity criteria, i.e., VC-bound. Therefore, the proposed RDA exhibits several robust clustering characteristics: 1) robust to the initialization (cluster number and initial guesses), 2) robust to cluster volumes (ability to detect different volumes of clusters), and 3) robust to noise and outliers (the given data are contaminated by noisy points). In order to demonstrate the effectiveness of the proposed RDA clustering algorithm in providing optimal clustering results, several artificial and real data sets have been considered. The results showed that RDA clustering algorithm provides a performance that is superior to that of traditional NC and DA algorithms. In other words, the proposed method takes advantages of noise clustering and deterministic annealing algorithms in addition to the auto-selection of the optimal cluster number for a given data.

Chapter 4

An Information Fuzzy C-Spherical Shells Clustering Algorithm

4.1 Introduction

In the fuzzy clustering literature, the fuzzy c-means (FCM) clustering algorithm, proposed by Dunn [34] and extended by Bezdek [8] has been successfully applied to a wide variety of problems. By using some specific shell-induced distance measures, the FCM algorithm has been modified to partition shell-shaped data structures such as lines, circles, ellipses and rectangles, as in [9] [24] [26] [38] [45] [56] [58] [68]. Although several specific algorithms, such as fuzzy c-shells (FCS) [24], fuzzy c-spherical shells (FCSS) [56] and fuzzy c-rings (FCR) [68], have been developed to detect and separate spherical shells (ring-shaped data) in the clustering literature. These methods, however, suffer from one or all of the following problems: 1)sensitive to the initialization of cluster prototypes, 2)requiring the optimal cluster number *a priori*, and 3)lack of the robustness against noisy points (noise and outliers). This chapter aims to improve the last two issues of traditional spherical shells clustering algorithms. The robustness of the proposed method against noisy points is obtained by the maximization of mutual information (MI), where the de-

gree of robustness is controlled through a separate parameter to make a trade-off between rejection of noisy points and preserving of desired points. The optimal number of spherical shells for a given data is auto-determined by the cluster validity criteria, i.e., VC(Vapnik-Chervonenkis)-bound, based on the structural risk minimization (SRM) principle [107]. In this chapter, we specifically take FCSS algorithm into account and develop an information fuzzy c-spherical shells (IFCSS) algorithm for robust clustering of spherical shells.

The rest of this chapter is organized as follows. Section 2 contains the derivation of the proposed IFCSS algorithm. We start from the standard FCSS algorithm to partition the given data set, then investigate the maximization of MI (which induces a robust density estimation to phase out noisy points) and the cluster validity (which is used to determine the number of spherical shells), and finally give the detailed pseudo-code of IFCSS for robust detection and separation of spherical shells. The experimental results of the proposed method are reported in section 3. Finally, conclusion and discussion are presented in section 4.

4.2 The Proposed Clustering Method

The proposed information fuzzy c-spherical shells (IFCSS) algorithm is basically a two step approach. In the clustering phase, the given data is clustered by the standard fuzzy c-spherical shells (FCSS) algorithm; in the robust pruning phase, the noisy points are phased out through the maximization of mutual information. Furthermore, the VC-bound is estimated to determine the optimal cluster number for a given data. The formulation of standard FCSS algorithm is rewritten here for discussion convenience.

4.2.1 FCSS Algorithm for Spherical Shells

Let the input data set be $X = \{x_1, x_2, \dots, x_l\} \subset R^n$, where l is the number of input data points, n is the dimension of input space. Assume the cluster number c is known *a priori* for the given data. In the FCSS algorithm, the cluster prototype w_k is a spherical shell which can be characterized by its center v_k and radius r_k , where $k = 1, 2, \dots, c$. The distance from a point x_j to a prototype $w_k := (v_k, r_k)$ is defined as in [56] as

$$(D_{kj})^2 = (\|x_j - v_k\|^2 - r_k^2)^2 = p_k^T M_j p_k + q_j^T p_k + b_j \quad (4.1)$$

where $b_j = (x_j^T x_j)^2$, $q_j = 2(x_j^T x_j)y_j$, $y_j = [x_j^T \ 1]^T$, $M_j = y_j y_j^T$ and $p_k = [-2v_k^T \ v_k^T v_k - r_k^2]^T$. Note that all the variables (vectors or matrix) in (4.1) are only related to the input point x_j except the parameter p_k which is decided by a pair of parameters, i.e., the center v_k and the radius r_k . Then the objective function¹ of FCSS to be minimized is defined as

$$E_{FCSS} = \sum_{k=1}^c \sum_{j=1}^l u_j (u_{k|j})^m (p_k^T M_j p_k + q_j^T p_k + b_j) \quad (4.2)$$

where u_j is the source distribution (equal to $1/l$ in the clustering phase), $u_{k|j}$ is the membership of point x_j in the k th cluster, and $m \in (1, \infty)$ is the exponent controlling fuzziness. The minimization of (4.2) with respect to $u_{k|j}$ gives the updating equation for the membership, which is given by

$$u_{k|j} = \frac{1}{\sum_{i=1}^c \left[\frac{(D_{kj})^2}{(D_{ij})^2} \right]^{1/(m-1)}} \quad (4.3)$$

¹Note we add the source data distribution (denoted by u_j) in the formulation of the objective function. As discussed in the next subsection, u_j can be used to phase out noisy points through the maximization of mutual information.

In [56] the vectors p_k are assumed to be independent of each other. Then it is easy to show that the vectors p_k that minimize (4.2) are given by

$$p_k = -\frac{1}{2}H_k^{-1}Q_k \quad (4.4)$$

where

$$H_k = \sum_{j=1}^l u_j(u_{k|j})^m M_j \quad (4.5)$$

and

$$Q_k = \sum_{j=1}^l u_j(u_{k|j})^m q_j \quad (4.6)$$

Unlike fuzzy c-shells (FCS) in [24], which needs to solve the coupled nonlinear equations, FCSS calculates the prototypes directly, which reduces the computation cost compared to the implementation of FCS. And unlike fuzzy c-rings (FCR) in [68], which is limited to two-dimensional ring-shaped data, FCSS is general for clustering any dimensional (hyper-)spherical shells. However, like FCS and FCR, FCSS suffers from the sensitivity to noisy points in addition to the pre-selection of cluster number². These two issues will be improved in the proposed IFCSS algorithm through the study of the following two subsections: the maximization of mutual information (MI) leads to the robustness of IFCSS against noisy points, and the optimal cluster number of shells for a given data is auto-determined by VC-bound cluster validity.

²[56] and [68] have preliminarily discussed the cluster number determination problem, however, the former can only deal with simple cases, and the latter is too heuristic to be reproduced. They may fail in many instances, especially when there exist noisy points in the data.

4.2.2 Robustness through Maximization of Mutual Information

From the clustering phase in the last subsection, we obtain an optimal set of membership $\bar{u}_{k|j}$ ($k = 1, \dots, c$ and $j = 1, \dots, l$), which minimize the distortion function (4.2) for a given data. As discussed in [29], the fuzzy membership tends to associate any data point (even it is a noisy one) in all clusters, such that the standard FCSS algorithm is not robust against outliers and noise or disturbance of the input data. We here use the capacity maximization [11] [99] of information theory to achieve the robustness for the standard FCSS. For the purpose of information theory, let a discrete channel be described by a probability transition matrix $Q = [\bar{u}_{k|j}]$, and the set of the probability distributions on the channel input be denoted by $U = [u_j]$. Then the clustering step can be regarded as an encoding procedure by putting $X = \{x_1, x_2, \dots, x_l\}$ into the channel then obtaining the output $W = \{w_1, w_2, \dots, w_c\}$. The input data reliability is fundamentally derived from the following *converse theorem* (the proof is given in Appendix B):

- Let p_e denote the probability of error of decoding X from W , such that if the input uncertainty

$$H(X) = - \sum_{j=1}^l u_j \ln u_j \quad (4.7)$$

is greater than the capacity

$$C = \max_{u_j \in U} I(X, W) \quad (4.8)$$

then the probability of error p_e will be bounded away from zero as

$$p_e \geq \frac{1}{\log(l-1)} (H(X) - C - 1) \quad (4.9)$$

The above theorem gives an important indication that any input information (input data) beyond the capacity C will generate unreliable data transmission. Hence, the capacity maximization is desirable to achieve the minimum of upper bound of empirical risk, which is implemented through the maximization of mutual information against the input distribution u_j , refer to (4.8) and discussions below.

$I(X, W)$ used in the above theorem is known as the mutual information between channel input and channel output as [11] [12]

$$I(X, W) = \sum_{j=1}^l \sum_{k=1}^c u_j \bar{u}_{k|j} \log \frac{\bar{u}_{k|j}}{\sum_{q=1}^l u_q \bar{u}_{k|q}} \quad (4.10)$$

The average mutual information (4.10) is nonnegative. However, the individual item $I(j, k) = \log \frac{\bar{u}_{k|j}}{\sum_{q=1}^l u_q \bar{u}_{k|q}}$ can be negative [12]. If the j th point x_j is taken into account and $\bar{u}_{k|j} < \sum_{q=1}^l u_q \bar{u}_{k|q}$, then the probability of the k th prototype is decreased by the observed point and gives a negative information about x_j . Then the particular input point may be considered as an unreliable (noisy) point and its negative effect must be offset by other input points. Therefore, the maximization of mutual information (i.e., to achieve the capacity (4.8)) provides a good robust estimation of the noisy points (outliers or noise). For any value of j , if $\sum_k u_{j|k} = 0$ (given by (4.14)), then u_j should be set equal to zero in order to obtain the maximum, such that the corresponding point x_j can be deleted and dropped from further consideration in the optimization procedure as an outlier or noise. In practical implementation, we treat a point as an outlier or noise if its according u_j is less than or equal to a small positive value.

The capacity of unconstrained discrete channel is achieved by the following statement [11] [12]: A probability distribution u_j ($j = 1, \dots, l$) on the channel input achieves capacity of the channel with transition matrix Q if and only if there exists

a number C such that

$$\sum_{k=1}^c \bar{u}_{k|j} \log \frac{\bar{u}_{k|j}}{\sum_{j=1}^l u_j \bar{u}_{k|j}} = C, \quad \forall u_j \neq 0 \quad (4.11)$$

$$\sum_{k=1}^c \bar{u}_{k|j} \log \frac{\bar{u}_{k|j}}{\sum_{j=1}^l u_j \bar{u}_{k|j}} < C, \quad \forall u_j = 0 \quad (4.12)$$

and the capacity is then achieved by

$$u_j = \frac{\exp \left(\sum_{k=1}^c \bar{u}_{k|j} \log u_{j|k} \right)}{\sum_{j=1}^l \exp \left(\sum_{k=1}^c \bar{u}_{k|j} \log u_{j|k} \right)} \quad (4.13)$$

with the posteriori $u_{j|k}$ is obtained through the Bayes formula

$$u_{j|k} = \frac{u_j \bar{u}_{k|j}}{\sum_{j=1}^l u_j \bar{u}_{k|j}} \quad (4.14)$$

For the IFCSS algorithm, we need to investigate the capacity of the constrained discrete channel with the requirement that the average expense is less than or equal to a specified number $E(u^*)$. Let's specify a vector e_j , the expense of using the j th input point, as

$$e_j = \sum_{k=1}^c \bar{u}_{k|j} (D_{kj})^2 \quad (4.15)$$

The capacity at expense $E(u)$ is then defined by

$$C(E(u)) = \max_{u_j \in U} I(X, W) \quad (4.16)$$

with the constraint

$$E(u) = \sum_{j=1}^l u_j e_j = \sum_{j=1}^l \sum_{k=1}^c u_j \bar{u}_{k|j} (D_{kj})^2 \leq E(u^*) \quad (4.17)$$

where

$$E(u^*) = \sum_{j=1}^l u_j^* e_j \quad (4.18)$$

and u_j^* achieves the above maximum. The constrained capacity maximization can be rewritten as an optimization problem with a Lagrangian multiplier $\lambda \geq 0$ as

$$C(E(u)) = \max_{u_j \in U} [I(X, W) - \lambda(E(u) - E(u^*))] \quad (4.19)$$

The capacity of constrained discrete channel is achieved by the following statement [11] [99]: A probability vector u_j ($j = 1, \dots, l$) achieves capacity at some expense $E(u)$ for the channel with transition matrix Q and expense vector e_j ($j = 1, \dots, l$) if and only if there exists a number V such that

$$\begin{aligned} \sum_{k=1}^c \bar{u}_{k|j} \log \frac{\bar{u}_{k|j}}{\sum_{j=1}^l u_j \bar{u}_{k|j}} - \lambda e_j &= V, \quad \forall u_j \neq 0 \\ \sum_{k=1}^c \bar{u}_{k|j} \log \frac{\bar{u}_{k|j}}{\sum_{j=1}^l u_j \bar{u}_{k|j}} - \lambda e_j &< V, \quad \forall u_j = 0 \end{aligned} \quad (4.20)$$

and the constrained capacity $C(E(u))$ is then achieved by

$$u_j = \frac{\exp \left(\sum_{k=1}^c \bar{u}_{k|j} \log u_{j|k} - \lambda e_j \right)}{\sum_{j=1}^l \exp \left(\sum_{k=1}^c \bar{u}_{k|j} \log u_{j|k} - \lambda e_j \right)} = \frac{u_j c_j}{\sum_{j=1}^l u_j c_j} \quad (4.21)$$

with the specific distortion measure³ $E(u) = E(u^*)$ for $u_j \geq 0$ of all $1 \leq j \leq l$. The last equality in (4.21) is obtained by using (4.14) together with the definition of c_j as

$$c_j = \exp \left(\sum_{k=1}^c \bar{u}_{k|j} \log \frac{\bar{u}_{k|j}}{\sum_{j=1}^l u_j \bar{u}_{k|j}} - \lambda e_j \right) \quad (4.22)$$

The degree of the robustness of IFCSS algorithm against the outlier (noisy input points) with maximum capacity is controlled by selecting a proper value of the parameter λ to pruning each input point by the conditions (4.20). The Kuhn-Tucker conditions tell that the number of noisy points can be controlled through the parameter λ . If a bigger value of $\lambda > 0$ is selected, then more points possess $u_j = 0$ to satisfy the Kuhn-Tucker conditions, which implies that more outliers (noisy points) are detected in the robust estimation procedure, i.e., the robustness of the IFCSS algorithm is increased. Normally, if we set $\lambda = 0$, the IFCSS algorithm is limited to the rejection of the most unreliable noisy points; if we let $\lambda > 0$, the algorithm is robust against any possible noisy points depending on the constraint term. The higher the value of λ is, the more the input points become outliers.

4.2.3 Optimal Cluster Number by VC-Bound

Several cluster validity criteria specific for shell-type clusters, as e.g., shell thickness and shell hyper-volume, have been presented in the literature such as in [56] and [68]. However, as discussed in [28], they may fail in some instances, especially when there exist noisy points in the data. This makes them impractical for real

³The capacity curve $C(E(u))$ is continuous, nondecreasing and concave on $E(u)$ [11], the maximum of capacity under the constraint (4.17) is reached at $E(u) = E(u^*)$ based on the premise of that the Kuhn-Tucker conditions (4.20) are satisfied.

applications. In chapter 3, the model selection criterion, i.e., the VC-bound

$$V_b \leq R_{ems} + \frac{\varepsilon}{2} \left(1 + \left(1 + R_{ems} \frac{4}{\varepsilon}\right)\right)^{1/2} \quad (4.23)$$

with

$$\varepsilon = \frac{h_c(\log \frac{2l}{h_c} + 1) - \log \frac{\zeta}{4}}{l} \quad (4.24)$$

has been used as a cluster validity to estimate the optimal cluster number for RDA clustering algorithm. The first term of the right hand side of the VC-bound (4.23) represents the empirical risk and the second term is the confidence interval of the SRM based estimation.

In this subsection, we specifically apply the VC-bound to the proposed IFCSS algorithm to estimate the number of spherical shells in a given data. Assume the number of noisy points identified through the maximization of mutual information in the pruning phase is l_o , then the first term (empirical risk) in (4.23) can be represented by

$$R_{ems} = \frac{l_o}{l} \quad (4.25)$$

For numerical calculation of the second term in (4.23), we need to construct a set of indicator functions with calculable VC-dimension h_c . From the clustering phase, we have got a set of distances D_{kj} , where $k = 1, 2, \dots, c$ and $j = 1, 2, \dots, l$. Let's consider the function

$$U_k = \lim_{m \rightarrow 1} u_{k|j} = \lim_{m \rightarrow 1} \sum_{i=1}^c \left[\frac{(D_{kj})}{(D_{ij})} \right]^{-2/(m-1)} \approx \begin{cases} 1 & x_j \in w_k \\ 0 & otherwise \end{cases} \quad (4.26)$$

where $x_j \in w_k$ means the point j is belonged to cluster k . Note the independent

variable D_{kj} can be presented as an inner product of two $(n + 1)$ -dimensional vectors of the input space as

$$(D_{kj}) = (\|x_j - v_k\|^2 - r_k^2) = [(d_{kj}^1)^2 + (d_{kj}^2)^2 + \dots + (d_{kj}^n)^2 - r_k^2] \quad (4.27)$$

where d_{kj}^r ($r = 1, 2, \dots, n$) denotes the r th element of the vector $x_j - v_k$, and n is the dimensionality of the input data. Then the function (4.26) can be approximated by the step function (denoted by θ) as

$$U_k \approx \theta\left(\sum_{i=1}^{n+1} \gamma_{ki} \phi_{ki}(x_j)\right) \quad (4.28)$$

Where γ_{ki} is the diameter parameter and ϕ_{ki} is linearly independent function. We can now construct the indicator functions for the IFCSS algorithm as

$$Q^c(x_j, W) = \sum_{k=1}^c U_k \approx \sum_{k=1}^c \theta\left(\sum_{i=1}^{n+1} \gamma_{ki} \phi_{ki}(x_j)\right) \quad (4.29)$$

It can be seen that the above constructed indicator functions, i.e., step functions (4.29), are linear in their parameters. According to [107], the VC-dimension of a set of functions linear in their parameters is equal to the number of parameters, i.e.,

$$h_c = c \times (n + 1) \quad (4.30)$$

Then the cluster validity criteria is stated as: by evaluating the estimated VC-bound for each chosen cluster number by equation (4.23), where R_{ems} is defined by (4.25) and h_c is defined by (4.30), we select the one that yields the minimum value of V_b as the optimal cluster number. Detailed discussions of VC-bound as a cluster validity can be found in chapter 6.

4.2.4 Pseudo-Code of the Proposed Method

According to the above discussions, we give a detailed pseudo-code of the proposed IFCSS for robust clustering of spherical shells as follows.

- Step 1) set the maximum number of clusters c_{max} , convergence parameter $\epsilon = 0.001$, source distribution $u_j = \frac{1}{l}$ ($j = 1, 2, \dots, l$), and initial cluster number $c = 2$ (assume the given data set contains at least two clusters)
- Step 2) fixed point iterations with cluster number c
 - perform clustering step: set exponent $1 < m < \infty$ and initialize membership $u_{k|j}$ ($k = 1, 2, \dots, c; j = 1, 2, \dots, l$), then alternatively update the cluster prototypes and fuzzy membership by using (4.4) and (4.3) until the maximum change in the membership between consecutive iterations is less than the given threshold value ϵ .
 - Perform pruning step: set the control parameter $0 \leq \lambda < \infty$, then iteratively update the prior distribution by using (4.22) and (4.21) until⁴

$$\log \max_j c_j - \log \sum_{j=1}^l u_j c_j < \epsilon$$

is satisfied.

- Step 3) calculate and verify the value of the VC-bound (4.23) within the range of maximum cluster number c_{max} . If the minimum of V_b is found, then delete noisy points, and recalculate the membership and cluster prototypes, then stop. Otherwise, let $c = c + 1$, and initialize distribution $u_j = \frac{1}{l}$ ($j = 1, 2, \dots, l$), then go to step 2.

⁴The termination is based on the following theorem [11]: Let the parameter λ be given, and suppose u_j ($j = 1, \dots, l$) is any probability vector. Then at the point $E(u) = \sum_j u_j c_j$, the following hold: a) $C(E(u)) \geq \lambda E(u) + \log \sum_j u_j c_j$; and b) $C(E(u)) \leq \lambda E(u) + \log \max_j c_j$.

4.3 Experimental Results

This section presents a few simulation examples to show the improvement and superiority of the proposed IFCSS method for clustering of spherical shells. Although the algorithm developed in the previous section is applicable to feature spaces of any dimension, here we present only results of two-dimensional data sets. In all simulation results, the original data sets are marked by “o” and the partitioned clusters are displayed by using different marks. The circumferences (determined by centers and radii) are also plotted by dotted lines “...” for partitioned results visualization. The cluster number c is known *a priori* for standard FCSS, while it is auto-determined by VC-bound for the proposed IFCSS algorithm⁵.

Figure 4.1: Clustering results of standard FCSS and the proposed IFCSS with different values of λ for concentric shell-shaped clusters with some outliers. The larger value λ is, the more outliers are eliminated.

4.3.1 Example 1: Robustness Against Noisy Points

This example demonstrates the robustness of IFCSS against noisy points (outliers), where the degree of robustness is controlled by the separate parameter λ . The given data consists of two concentric spherical shells (containing 60 and 30 data points respectively) contaminated with some outliers. The partitioned results of the standard FCSS (with $c^* = 2$) and the proposed IFCSS are shown in Fig 4.1. It can be seen that the result of FCSS is distorted by outliers (see the plotted

⁵In all experiments, we run IFCSS thirty times for each data set with random initializations. The minimum value of V_b and its according result are used in our discussions.

Figure 4.2: VC-bound vs cluster number with different values of λ for the data set in example 1. V_b reaches the minimum at $c^* = 2$ with independence of λ .

circumference) as shown in Fig 4.1a, which implies that FCSS is not robust against noisy points. In contrast, IFCSS obtains much better performances: when λ is set to be 0, the outliers between shells are eliminated through the MI maximization as shown in Fig 4.1b; when λ is increased to be 0.05 and 0.10, the outliers located outside the outer shell and inside the inner shell are eliminated as shown in Fig 4.1c and Fig 4.1d, respectively. The optimal cluster number $c^* = 2$ used in this example is determined by the VC-bound as shown in Fig 4.2: different values of λ lead to the same optimal cluster number $c^* = 2$ for this data set. That implies that the selection of λ hardly affects the determination of the optimal cluster number (since the VC-bound only depends on the relative values of R_{ems} and h_c over different cluster number). As a general rule of thumb, if eliminating more noisy points is interested we can gradually increase λ and redo the maximization optimization with the optimal cluster number.

4.3.2 Example 2: The Optimal Cluster Number

As discussed above, the optimal cluster number of spherical shells is auto-determined by VC-bound in IFCSS algorithm. In most cases, the value of λ hardly affects the determination of the optimal cluster number. We set λ as a constant 0.05 in all

Figure 4.3: Partitioned results and VC-bound of IFCSS with different cluster number c for data set 1 in example 2. V_b reaches the minimum at $c^* = 3$, which is corresponding to the most reasonable result.

Figure 4.4: Partitioned results and VC-bound of IFCSS with different cluster number c for data set 2 in example 2. V_b reaches the minimum at $c^* = 3$, which is corresponding to the most reasonable result.

experiments below if it is not specified. Two data sets are considered in this example. Data set 1 consists of three disturbed concentric spherical shells as shown in Fig 4.3a. We calculate and plot the values of V_b by equation (4.23) with different cluster number (from 2 to 6) for this data set as in Fig 4.5. The partitioned results of IFCSS with $c = 2, 3, 4$ are also plotted in Fig 4.3 for visualization. As observed from Fig 4.3 and Fig 4.5, the VC-bound finds the underlying cluster number for this data set: V_b reaches the minimum at $c^* = 3$, which corresponds to the most reasonable result. Data set 2 contains three disturbed intersected spherical shells as shown in Fig 4.4a. The VC-bound by equation (4.23) with different cluster number (from 2 to 6) is plotted in Fig 4.5 and the partitioned results of IFCSS with $c = 2, 3, 4$ are shown in Fig 4.4. Similarly, the VC-bound correctly reveals the optimal cluster number for this data set: V_b reaches the minimum at $c^* = 3$,

Figure 4.5: VC-bound vs cluster number for the data sets in example 2. V_b reaches the minimum at $c^* = 3$ for both data sets.

which corresponds to the most reasonable result.

Table 4.1: Original and partitioned parameters for the data set in Fig 4.6a.

	Original Parameters		Partitioned Parameters	
	Center	Radius	Center	Radius
Cluster 1	(1.0000, 1.0000)	2.0000	(1.0656, 1.0688)	2.0912
Cluster 2	(1.0000, 1.0000)	5.0000	(0.9843, 1.0058)	5.1524

Table 4.2: Original and partitioned parameters for the data set in Fig 4.6b.

	Original Parameters		Partitioned Parameters	
	Center	Radius	Center	Radius
Cluster 1	(0.0000, -0.4000)	1.5000	(-0.0039, -0.4113)	1.5096
Cluster 2	(1.0000, 0.0000)	1.5000	(0.9987, -0.0384)	1.4757
Cluster 3	(2.0000, -0.4000)	1.5000	(2.0023, -0.4072)	1.5052

4.3.3 Example 3: Several Additional Specific Cases

We show the effectiveness of IFCSS for robust clustering spherical shells by investigating several complicated cases. Four data sets are considered: data 1 contains two seriously disturbed ring data with identical center but different radii as shown in Fig 4.6a; data 2 consists of three partial ring data with several noisy points as shown in Fig 4.6b, data 3 and data 4 contain four and five slightly disturbed ring

Figure 4.6: First row: four original data sets used in example 3; Second row: according VC-bound revealing the optimal cluster for each data set; Third row: according partitioned results of IFCSS with optimal cluster number for each data set.

data contaminated by many noisy points as shown in Fig 4.6c and Fig 4.6d, respectively. The values of VC-bound for each data set with different cluster number are plotted in Fig 4.6e-h. As we can see, the optimal cluster number for all data sets are correctly revealed by VC-bound, V_b reaches the minimum at $c^* = 2$ for data 1, $c^* = 3$ for data 2, $c^* = 4$ for data 3 and $c^* = 5$ for data 4, respectively. With the optimal cluster number, IFCSS correctly partitions each given data into the original data structure in addition to removing the noisy points as shown in Fig 4.6i-l. For quantitative investigation, the original and partitioned cluster centers and radii for each data set are concluded in Table 4.1-4.4. It can be seen that

Table 4.3: Original and partitioned parameters for the data set in Fig 4.6c.

	Original Parameters		Partitioned Parameters	
	Center	Radius	Center	Radius
Cluster 1	(0.0000, 0.0000)	2.0000	(0.0007, 0.0029)	2.0043
Cluster 2	(0.0000, 0.0000)	4.0000	(-0.0051, 0.0078)	3.9925
Cluster 3	(-2.5000, -2.5000)	3.0000	(-2.4925, -2.5013)	3.0028
Cluster 4	(2.5000, 2.5000)	3.0000	(2.4839, 2.4902)	2.9992

Table 4.4: Original and partitioned parameters for the data set in Fig 4.6d.

	Original Parameters		Partitioned Parameters	
	Center	Radius	Center	Radius
Cluster 1	(0.0000, 0.0000)	1.5000	(0.0016, 0.0013)	1.4946
Cluster 2	(-1.0000, 0.0000)	1.5000	(-1.0014, 0.0043)	1.4945
Cluster 3	(0.0000, -1.0000)	1.5000	(0.0069, -0.9966)	1.5024
Cluster 4	(1.0000, 0.0000)	1.5000	(1.0028, -0.0037)	1.5005
Cluster 5	(0.0000, 1.0000)	2.0000	(0.0013, 0.9984)	1.4996

the proposed algorithm not only correctly partitions the data, removes the noisy points, and reveals the underlying cluster number, but also finds the parameters of the shell clusters with good accuracy.

4.4 Conclusion and Discussion

We have proposed an information fuzzy c-spherical shells (IFCSS) clustering algorithm for robust detection and separation of spherical shells: the robustness against noisy points is obtained by the maximization of mutual information and the optimal cluster number is auto-determined by the VC-bound. The superiority of the proposed clustering method has been tested by experimental results. The selection of fuzzier m remains undiscussed in this chapter. It is actually an open problem in the literature of fuzzy clustering. For fuzzy clustering algorithms, as e.g., FCM algorithm, increasing the value of m represents increased sharing of points among all clusters, which may affect the result of maximization of mutual information in IFCSS algorithm. Though a value of $m = 2$ is known to give good results with the FCM algorithm, we find from the experiments that the value of

$m = 1.5$ (this value is used in all experiments in this chapter) is more suitable for IFCSS algorithm. This is an interesting topic worthy of further investigation.

Chapter 5

Data Clustering in Mercer Kernel-Induced Feature Space

5.1 Introduction

Most traditional clustering techniques, such as hard c-means algorithm [66], fuzzy c-means algorithm [8], simulated annealing algorithm [96] and deterministic annealing algorithm [85], by means of Euclidean distance measure, only work well when the underlying clusters exhibit linearly separable shapes. However, they will generate improper results if the underlying data shapes are linearly nonseparable. So far, few attempts have been proposed to deal with this challenging issue in the data clustering literature. This chapter makes an effort to improve the capacity of standard deterministic annealing (DA) algorithm for more complicated data structures by means of kernel methods [72] [93]: By nonlinearly mapping the observed data from a low-dimensional input space into a high-dimensional feature space by means of a given kernel function, which increases the probability of the linear separability of the given data, the kernel-based clustering algorithms can reveal structure in the data that may go unnoticed by traditional clustering algorithms in the standard Euclidean space. For this purpose, a kernel-based deterministic

annealing (KDA) algorithm is proposed for data clustering by using Gaussian kernel function. In addition, we put out a simple but effective method to select a reasonable value for the Gaussian kernel parameter (width), whose selection is so far remains unsolved in the literature. The experiments focus on revealing that the scaled inverse of data covariance is a reasonable (optimal or at least suboptimal) estimation of the Gaussian kernel parameter (width), and demonstrating the effectiveness of the proposed KDA for clustering different kinds of data sets, which may possess different shape, distribution and size.

The rest of this chapter is organized as follows. Section 2 contains the derivation and implementation of the proposed KDA algorithm. We first construct a nonlinear version of DA algorithm in the feature space based on a Gaussian kernel-induced distance measure, then put the emphasis on the adaptive selection of Gaussian kernel parameter by the scaled inverse of data covariance, and finally give the detailed pseudo-code of KDA algorithm for data clustering. The experimental results on both artificial and benchmark data are shown in section 3. In section 4, several further investigations on KDA algorithm are preliminarily studied. Finally, the conclusion is given in section 5.

5.2 Derivation and Implementation

A kernel-based deterministic annealing (KDA) clustering algorithm with adaptively selected Gaussian kernel parameter is proposed in this section. A brief review of the basic idea of kernel methods can be found in the chapter 2, which may be helpful for a better understanding of the discussions in this section.

5.2.1 Kernel-induced Distance Measure

Let the input data set be $X = \{x_1, x_2, \dots, x_l\} \subset R^n$, where l is the number of input data points, n is the dimension of input space. Suppose that the input space

$X \in R^n$ is mapped into a Hilbert space F through a nonlinear mapping function ϕ

$$\begin{aligned} \phi : \quad R^n &\rightarrow F \\ x_j &\rightarrow \phi(x_j), \quad j = 1, 2, \dots, l \end{aligned} \quad (5.1)$$

Where F is referred to as a feature space, which could have an arbitrarily large, possible infinite, dimensionality. And $\phi(x_j)$ is the transformed point of x_j . Assume all the points are partitioned into c clusters in feature space, the set of cluster centers denoted by $\{\phi_1(x), \phi_2(x), \dots, \phi_c(x)\} \subset F$ has a similar presentation like the K-means clustering in feature space as in [89] as

$$\phi_k(x) = \sum_{j=1}^l \gamma_{kj} \phi(x_j), \quad k = 1, 2, \dots, c \quad (5.2)$$

Where γ_{kj} is the parameters to be decided by KDA algorithm as shown below. Note that unlike the *hard* clustering, where each data point is assigned to one certain cluster, the proposed method is a *soft* clustering approach. It means that all the training data with a total number l could have relationship with all the c clusters as shown in (5.2). Using (5.2) the squared Euclidian distance between a cluster center $\phi_k(x)$ and a mapped point $\phi(x)$ (it is called the *kernel-induced distance* in this chapter) can be stated as

$$\begin{aligned} D_k(x) &= \left\| \phi(x) - \sum_{i=1}^l \gamma_{ki} \phi(x_i) \right\|^2 \\ &= \phi(x)\phi(x) - 2\phi(x) \sum_{i=1}^l \gamma_{ki} \phi(x_i) + \sum_{i,j=1}^l \gamma_{ki} \phi(x_i) \gamma_{kj} \phi(x_j) \end{aligned} \quad (5.3)$$

Clearly, if the dimensionality of F is arbitrarily large or even infinite, it would be impossible to calculate (5.3) directly. To overcome this limitation we use the *kernel trick* as in kernel principle component analysis [90] and Support Vector Machines

[22]. Rather than directly calculate the dot product $\phi(x)\phi(y)$ in feature space, we may get this dot product implicitly in input space using kernel function $k(x, y) = \phi(x)\phi(y)$. One of the most popularly used kernel functions is the Gaussian kernel

$$k(x, y) = e^{-\rho\|x-y\|^2} \quad (5.4)$$

where ρ is the Gaussian kernel parameter. The Gaussian kernel function is used in our method and the adaptive selection of the parameter ρ for a given data is discussed later. Using the kernel trick, (5.3) can be rewritten as

$$D_k(x) = k(x, x) - 2 \sum_{i=1}^l \gamma_{ki} k(x, x_i) + \sum_{i,j=1}^l \gamma_{ki} \gamma_{kj} k(x_i, x_j) \quad (5.5)$$

5.2.2 The KDA Clustering Algorithm

Similar to the notations used in [87], here we let $p(\phi_k(x)|\phi(x_j))$ be the association probability relating mapped point $\phi(x_j)$ with cluster center $\phi_k(x)$. Using the kernel-induced distance measure (5.5), the distortion function in feature space is stated as

$$J_\phi = \sum_{j=1}^l \sum_{k=1}^c p(\phi(x_j)) p(\phi_k(x)|\phi(x_j)) D_k(x_j) \quad (5.6)$$

As no prior knowledge of the data distribution is assumed, of all possible distributions that yield a given value of J_ϕ , we choose the one that maximizes the conditional Shannon entropy in feature space

$$H_\phi = - \sum_{j=1}^l \sum_{k=1}^c p(\phi(x_j)) p(\phi_k(x)|\phi(x_j)) \log p(\phi_k(x)|\phi(x_j)) \quad (5.7)$$

i.e., the problem of clustering by KDA is formulated as minimizing

$$F_\phi = J_\phi - TH_\phi \quad (5.8)$$

Where T is the Lagrangian multiplier which is analogical to the temperature in statistical mechanics. It turns out [87] (according to the maximum entropy principle) that the resultant distribution is the titled distribution and is given by

$$p(\phi_k(x)|\phi(x_j)) = \frac{p(\phi_k)e^{-\frac{D_k(x_j)}{T}}}{\sum_{i=1}^c p(\phi_i)e^{-\frac{D_i(x_j)}{T}}} \quad (5.9)$$

where $p(\phi_k)$ is the kernel-induced mass probability of the k th cluster in feature space, that is

$$p(\phi_k) = \sum_{j=1}^l p(\phi(x_j))p(\phi_k(x)|\phi(x_j)) \quad (5.10)$$

Clearly, at limited high T , these are uniform distributions, each data point is equally associated with all clusters. As T is lowered, the distributions become more discriminating. And at limited low T , the clustering becomes hard with each data point assigned to the nearest cluster with probability one. Plugging (5.9) back into (5.8), we get the free energy function in the feature space, that is

$$F_\phi^* = \min_{\{p(\phi_k(x)|\phi(x_j))\}} (J_\phi - TH_\phi) = -T \sum_{j=1}^l p(\phi(x_j)) \log \sum_{k=1}^c p(\phi_k) e^{-\frac{D_k(x_j)}{T}} \quad (5.11)$$

Minimizing F_ϕ^* with respect to $\phi_k(x)$ yields

$$\frac{\partial(F_\phi^*)}{\partial(\phi_k(x))} = 0 \quad (5.12)$$

i.e.,

$$\sum_{j=1}^l p(\phi(x_j))p(\phi_k)e^{-\frac{D_k(x_j)}{T}}[\phi(x_j) - \phi_k(x)] = 0 \quad (5.13)$$

Divide the above equation by the normalized factor

$$Z_{\phi(x_j)} = \sum_{i=1}^c p(\phi_i)e^{-\frac{D_i(x_j)}{T}} \quad (5.14)$$

which leads to

$$\sum_{j=1}^l \frac{p(\phi(x_j))p(\phi_k)e^{-\frac{D_k(x_j)}{T}}}{Z_{\phi(x_j)}}\phi(x_j) = \sum_{j=1}^l \frac{p(\phi(x_j))p(\phi_k)e^{-\frac{D_k(x_j)}{T}}}{Z_{\phi(x_j)}}\phi_k(x) \quad (5.15)$$

Use (5.9) the above equation can be rewritten as

$$\sum_{j=1}^l p(\phi(x_j))p(\phi_k(x)|\phi(x_j))\phi(x_j) = \sum_{j=1}^l p(\phi(x_j))p(\phi_k(x)|\phi(x_j))\phi_k(x) \quad (5.16)$$

i.e.,

$$\phi_k(x) = \frac{\sum_{j=1}^l p(\phi(x_j))p(\phi_k(x)|\phi(x_j))\phi(x_j)}{\sum_{j=1}^l p(\phi(x_j))p(\phi_k(x)|\phi(x_j))} = \sum_{j=1}^l \gamma_{kj}\phi(x_j) \quad (5.17)$$

From (5.17), we can obtain the explicit expression of γ_{kj} which has been used to calculate the kernel-induced distance in (5.3) and (5.5), that is

$$\gamma_{kj} = \frac{p(\phi(x_j))p(\phi_k(x)|\phi(x_j))}{\sum_{i=1}^l p(\phi(x_i))p(\phi_k(x)|\phi(x_i))} \quad (5.18)$$

Alternatively updating (5.9) and (5.5) with phase transition gives the so-called KDA algorithm. The adaptiveness of KDA, i.e., the adaptive selection of Gaussian kernel parameter ρ for a given data set, will be discussed in next subsection. The key idea of KDA approach is to do the optimization on the so-called free energy F_ϕ^* , which is parameterized by a parameter T . The parameter T determines the amount of smoothing that is done on the cost function J_ϕ . The minimization of the free energy (5.11) is achieved in two nested loops. In the inner loop (5.9) and (5.17) can be solved by fixed point iteration for a given value of T . This procedure constitutes an expectation maximization (EM) [32] and can be shown to converge to a local minimum of the free energy. In order to find the optimal minimum of the cost function, T is varied in the outer loop according to an annealing schedule. Starting from high values of T and fuzzy associations, T is stepwise decreased ($T = \eta T$, $0 < \eta < 1$) until a good minimum of the original cost function J_ϕ is found at low values of T , where the previously mentioned fuzzy associations become unique.

Figure 5.1: Plot of the r.h.s of (5.20) as a function of distance $\|x_i - x_j\|$ in data space for four different values of ρ . At distances $\|x_i - x_j\| \gg \rho$ and $\|x_i - x_j\| \ll \rho$ the assignment costs are constant.

5.2.3 Adaptive Selection of Gaussian Kernel Parameter

The selection of kernel parameter(s) is an important issue for the implementation of data clustering in the feature space because it determines the nonlinear mapping together with the given kernel function. Unfortunately, it remains an open problem for kernel methods so far in the literature of machine learning. Extensive experiments are normally needed to decide the optimal value of kernel parameter(s) according to certain criterion from clustering solution, e.g., the criterion to guarantee a minimum number of support vectors in [4] or reach a minimum value of extended Xie-Beni index in [62]. However, this investigation suffers from the high computation complexity and dependence of the accuracy of criterion. We here put out a simple but efficient method for the Gaussian kernel parameter selection as follows.

Let us consider the *kernel-induced distance* between two transformed points $\phi(x_i)$ and $\phi(x_j)$, that is

$$D_{ij} = \|\phi(x_i) - \phi(x_j)\|^2 = \phi(x_i)\phi(x_i) - 2\phi(x_i)\phi(x_j) + \phi(x_j)\phi(x_j) \quad (5.19)$$

After plugging Gaussian kernel function into (5.19), we get

$$D_{ij} = 2(1 - e^{-\rho\|x_i - x_j\|^2}) \quad (5.20)$$

The r.h.s of (5.20) is plotted in Fig 5.1 as a function of the distance $\|x_i - x_j\|$ for four different values of ρ . It can be seen that for a large value of ρ only very closed points influence the kernel-induced distance, while for points further away the kernel-induced distance is almost constant. For a small value of ρ , there is a significant influence for all data points. From the plot, we see that the kernel-induced distance is bounded and monotone increasing with distance measure zero as $\|x_j - v_k\| \rightarrow 0$ and distance measure two as $\|x_j - v_k\| \rightarrow \infty$. If the kernel parameter ρ tends to infinite, each data point will have no neighborhood. That is, the distance between

every point will attach its maximum distance measure, and each point become an individual because the distances between them are all the farthest (constant 2). If ρ tends to zero, the space will degenerate to a point. The distances between them are the smallest distance measure zero. In each of those two extreme cases, the assignment costs are constant and the data can not be partitioned in the feature space. For the data with well-separated clusters (large data covariance), the kernel-induced distance tends to reach maximum, such that a small value of ρ is suitable to separate the clusters. For the data with undistinguished clusters (small data covariance), the kernel-induced distance tends to reach minimum, so a large value of ρ is effective to separate the clusters. Inspired by the above observations, we select the ρ as the scaled inverse of data covariance, which is given by

$$\rho^* = \left(\frac{\frac{1}{n} \sum_{j=1}^l \|x_j - \bar{x}\|^2}{l} \right)^{-1} \quad (5.21)$$

where $\bar{x}$ is the mean of input data points, which is given by

$$\bar{x} = \frac{1}{l} \sum_{j=1}^l x_j \quad (5.22)$$

and n is the dimensionality of input data. This choice makes ρ proportional to the scaled inverse of data covariance in input space. The data covariance is divided by n such that the dot products in the Gaussian kernel function is scaled with the dimensionality of the input space to make the ρ independent of it. This method is in accordance with the statement that the parameter(s) of the kernel function has to be chosen in accordance with the range and dimensionality of the data [88]. The experimental results in next section reveal that (5.21) is a reasonable estimation of ρ . In other words, the KDA with the estimated value ρ^* can generate a reasonable partition for a given data set.

5.2.4 Pseudo-code of KDA Implementation

Two issues related to KDA implementation are necessarily discussed before we give the pseudo-code. First, since the cluster centers $\phi_k(x)$ ($k = 1, 2, \dots, c$) in the high-dimensional space F can hardly be visualized, we plot each cluster center by its approximate *projection*

$$\hat{\phi}_k = \sum_{j=1}^l \gamma_{kj} x_j \quad (5.23)$$

in the case of partition visualization is needed. Second, it is well known that for deterministic annealing approach phase transitions (splits in the cluster representation) occur during annealing process. The critical value T^* for the first phase transition is determined in terms of the largest eigenvalue of covariance matrix [87]

$$T^* = 2\lambda_{\max}(C_{\phi(x)}) \quad (5.24)$$

where $C_{\phi(x)}$ is the kernel-induced covariance matrix

$$C_{\phi(x)} = \frac{1}{l} \sum_{j=1}^l \bar{\phi}(x_j) \bar{\phi}(x_j)^T \quad (5.25)$$

of centered images $\bar{\phi}(x_j) = \phi(x_j) - \frac{1}{l} \sum_{i=1}^l \phi(x_i)$ of the data points. The calculation of eigenvalue of $C_{\phi(x)}$ is extremely not practical due to the high or infinite dimension of feature space. From singular value decomposition (SVD) it can be seen that the matrix of dot products $D_{\phi(x)}$ with elements $D_{ij} = \frac{1}{l} \bar{\phi}(x_i) \bar{\phi}(x_j)^T$ has the same nonzero eigenvalues as $C_{\phi(x)}$ [40]. Thus $\lambda_{\max}(C_{\phi(x)}) = \lambda_{\max}(D_{\phi(x)})$, where $D_{\phi(x)}$ can be calculated using the kernel function $k_{ij} = k(x_i, x_j) = \phi(x_i) \phi(x_j)$ as follows

$$D_{ij} = \frac{1}{l} \bar{\phi}(x_i) \bar{\phi}(x_j)^T \quad (5.26)$$

$$\begin{aligned}
&= \frac{1}{l} [\phi(x_i) - \frac{1}{l} \sum_{p=1}^l \phi(x_p)] [\phi(x_j) - \frac{1}{l} \sum_{q=1}^l \phi(x_q)] \\
&= \frac{1}{l} [\phi(x_i)\phi(x_j) - \frac{1}{l} \sum_{p=1}^l \phi(x_i)\phi(x_p) - \frac{1}{l} \sum_{q=1}^l \phi(x_j)\phi(x_q) + \frac{1}{l^2} \sum_{p,q=1}^l \phi(x_p)\phi(x_q)] \\
&= \frac{1}{l} [k_{ij} - \frac{1}{l} \sum_{p=1}^l k_{pi} - \frac{1}{l} \sum_{q=1}^l k_{qj} + \frac{1}{l^2} \sum_{p,q=1}^l k_{pq}]
\end{aligned}$$

We now give a detailed pseudo-code of KDA for data clustering as follows.

- Step 1) Select the Gaussian kernel parameter ρ^* by (5.21);
- Step 2) If the selection of optimal cluster number c^* is desirable, then estimate it by the contour plot of kernel matrix, see section 4;
- Step 3) Initial temperature $T_{ini} > 2\lambda_{max}(C_{\phi(x)})$, minimum temperature $T_{min} = T_{ini}/1000$, convergence parameter $\varepsilon = 0.001$, source distribution $p(\phi(x_j)) = \frac{1}{l}$ ($j = 1, 2, \dots, l$), mass probability $p(\phi_1) = 1$, and $k = 1$;
- Step 4) Alternatively update (5.18), (5.5), (5.9) for $i = 1, 2, \dots, k$ (fixed point iteration) until the maximum change in the kernel-induced distance between consecutive iterations is less than the given threshold value ε ;
- Step 5) If $T < T_{min}$ then stop, else let $T = \eta T$ ($0 < \eta < 1$) and go to next step;
- Step 6) If $k \geq c^*$ then stop, else check the condition of phase transition for $r = 1, 2, \dots, k$, if critical T_r^* is reached for cluster r (see remarks below), implicitly add a new cluster center by $D_{k+1}(x) = D_r(x) + \delta$ with $p(\phi_{k+1}) = p(\phi_r)/2$ and $p(\phi_r) = p(\phi_r)/2$, where δ is a small disturbance, and let $k = k + 1$, then go to step 4;
- Step 7) If robust pruning of noisy points is desirable, then set the control parameter $0 \leq \lambda < \infty$, and iteratively update capacity and prior distribution

by using (5.31) and (5.30) until $\log \max_j c_j - \log \sum_{j=1}^l u_j c_j < \varepsilon$ is satisfied, identify the outliers and re-partition the data without them, see section 4;

- Step 8) For partition visualization, calculate and plot the approximate *projection* of cluster centers by (5.23);
- Step 9) If the image illustration of the data structure in feature space is desirable, then use the kernel-induced distance to illustrate the locations of the data points, see section 4.

Here we mention several remarks for the implementation of KDA clustering algorithm:

- As mentioned above, the experiments in this chapter will focus on revealing that the scaled inverse of data covariance is a reasonable estimation of the Gaussian kernel parameter (width), and demonstrating the effectiveness of the proposed KDA for clustering different kinds of data sets. This is the reason why we make step 2, step 7 and step 9 optional, the preliminary study of these issues will be addressed in section 4.
- In practical applications, the data points are usually assumed to be independent of each other. In other words, $p(\phi(x_j)) = \frac{1}{l}$ is necessarily assumed for the implementation of KDA, as standard DA does.
- The cluster center $\phi_k(x)$ is implicitly updated by updating the kernel-induced distance $D_k(x)$ in step 4 and step 6. The reason to do this is that we do not know the explicit expression of the nonlinear mapping ϕ , so we can not calculate the cluster center directly in feature space by using (5.17).
- The critical T_r^* in step 6 can be obtained through a simple perturbation method. In this case we always keep two centers in one cluster and perturb them when we update T . Until the critical temperature is reached they will be merged together by the iterations. At phase transition they will move further apart.

Figure 5.2: Three artificial data sets tested in the experiments.

5.3 Experimental Results

The superiority of the proposed KDA algorithm for data clustering is tested on some artificial and benchmark data sets. Three artificial data (unbalanced data, ring data and X-O data) and one benchmark data (Iris data) are considered. All data sets are also tested by standard DA approach (see chapter 2.2) and kernel-based K-means (KKM)¹ algorithm (see chapter 2.5). We don't aim to provide a complete comparison between KDA and other clustering methods, but focus on: 1) revealing that the scaled inverse of data covariance is a reasonable estimation of the Gaussian kernel parameter (width); 2) demonstrating the effectiveness of KDA for clustering different kinds of data sets, which may possess different shape, distribution and size. For all examples, we set $\eta = 0.9$ (cooling rate for DA and KDA) and $\epsilon = 0.001$ (convergence threshold for all methods); for 2D data, the positions of cluster centers of KDA and KKM in input space are approximately determined by (5.23); the selection problem of cluster number c is not the focus of this chapter and we assume it is *a priori* for each given data set, though it can be estimated by the contour plot of kernel matrix as discussed in section 4.

¹In all experiments, we run KKM twenty times for each data set with different initialization due to its insensitivity to data initialization. For fair comparison, the Gaussian kernel function with estimated ρ^* by (5.21) is used for KKM.

Figure 5.3: The partition results by applying KDA with different values of ρ and the estimated $\rho^* = 0.1068$ on unbalanced data. Note Fig 5.3b has 2 partition errors.

5.3.1 Artificial Data

Three two-dimensional (2D) data sets are considered in this chapter as shown in Fig 5.2. For each given data set, we first investigate the influence of kernel parameter ρ on the partition results, then compare the performance of KDA to KMM and standard DA. The positions of cluster centers are marked by big “*” in all figures.

Unbalanced Data

Fig 5.2a shows an unbalanced data with great difference in the number of points in different clusters. The 2D data set is generated from five Gaussian distributions where 120 points from one distribution and the other four distributions contain 20

Figure 5.4: The clustering results of KKM and DA algorithms on unbalanced data. Where $x/20$ means the according result occurs x times out of 20 runs.

points each. To investigate the influence of kernel parameter on data clustering in feature space, we plot the partition results of KDA (kernel-based DA without adaptive selection of kernel parameter, see section 2.2) with different values of ρ in Fig 5.3. It can be seen that the value of ρ significantly affects the partition results. Only reasonable values, as e.g., $\rho = 0.01$, $\rho = 0.1$ and $\rho = 0.3$, can generate a satisfactory result for this data set as shown in Fig 5.3a. From the view of classification, it has 0 partition errors (the number of misclassified points). However, when ρ is increased to certain larger values, the partition results turn out to be unsatisfied as shown in Fig 5.3b-e. This demonstration shows the necessity of adaptively selecting a reasonable value of ρ for a given data. The partition result of KDA with estimated $\rho^* = 0.1068$ by (5.23) is shown in Fig 5.3f, which correctly partition the original data structure with 0 partition errors. Fig 5.4 shows the partition results of KKM and standard DA algorithms. For KKM, only few runs (2 out of 20) can obtain a satisfactory result, as shown in Fig 5.4a, but the results of other runs (18 out of 20) are obviously unreasonable, one of them is shown in Fig 5.4b: the dominating cluster is divided into two smaller clusters and the two small clusters are merged into one bigger clusters. For this well-separated Gaussian distributed data set, the standard DA algorithm correctly partitions the given data as shown in Fig 5.4c.

Figure 5.5: The partition results by applying KDA with different values of ρ and the estimated $\rho^* = 2.8613$ on ring data.

Ring Data

Fig 5.2b shows an linearly nonseparable data set - ring data, which has been widely used in the clustering literature [39] [62]. This data set is drawn from two disturbed ring distributions. The total number of points is 150 with 100 points from outside ring and 50 points from inside ring. The partition results of KDA with different values of ρ are plotted in Fig 5.5. For this case, only the values $\rho = 2, 5, 10$ correctly reveal the original data structure with 0 partition errors as shown in Fig 5.5c. While smaller values, as e.g., $\rho = 0.1$ and $\rho = 1$, and larger values, as e.g., $\rho = 20$ and $\rho = 50$, generate a certain number of partition errors as shown in Fig 5.5a-b and Fig 5.5d-e, respectively. The proposed *KDA* with estimated $\rho^* = 2.8613$ by (5.21) correctly partitions this data with 0 partition errors as shown in Fig 5.5f. The partition results of KKM and standard DA algorithms

Figure 5.6: The partition results of KKM and DA algorithms on ring data. KKM is sensitive to data initializations.

are shown in Fig 5.6. It is obvious that the standard DA algorithm can not deal with this linearly nonseparable data, as shown in Fig 5.6c. KKM suffers from local minimum solution: good initializations induce a reasonable result as shown in Fig 5.6a, while bad ones lead to unreasonable results, one of them is shown in Fig 5.6b.

X-O Data

The non-identical distributed X-O data is shown in Fig 5.2c. The data set consists of three well-separated groups of data, however, two of them are X-O distributed [19] and one possesses an ellipsoidal-shaped structure. The total number of points is 150 with 50 points from each distribution. The partition results of KDA with different values of ρ are plotted in Fig 5.7. Similarly, only reasonable values, as e.g. $\rho = 0.5$ and $\rho = 1.0$, can correctly reveal the original data structure with 0 partition errors as shown in Fig 5.7b. While smaller values and larger values can not correctly partition the data set as shown in Fig 5.7a and Fig 5.7c-e, respectively. The partition result of KDA with estimated $\rho^* = 0.7743$ is shown in Fig 5.7f, which generates 0 partition errors. Fig 5.8 plots the partition results by standard DA and KKM algorithms. It can be seen that the standard DA algorithm can not deal with this non-identical distributed data as shown in Fig 5.8c. KKM can reveal the original data structure in most (14 out of 20) runs, as shown in Fig 5.8a, however,

Figure 5.7: The partition results by applying KDA with different values of ρ and the estimated $\rho^* = 0.7743$ on X-O data.

it is sensitive to data initialization, which may not be practical in real applications.

Fig 5.8b shows an unsatisfied result of KKM.

Table 5.1: The estimated value ρ^* and the optimal range ρ^o with according partition errors for different data sets. Each cell contains the estimated value ρ^* (or the optimal range ρ^o) followed by partition errors.

	Unbalanced		Ring		X-O		Iris	
	Value	Errors	Value	Errors	Value	Errors	Value	Errors
Estimated ρ^*	0.1068/0		2.8613/0		0.7743/0		0.8813/10	
Optimal ρ^o	[0.01,0.35]/0		[1.9,15]/0		[0.4,1.8]/0		[0.7,1.2]/10	

Several Observations

- The value of kernel parameter significantly affects the partition results of kernel-based data clustering algorithm, fortunately, the scaled inverse of data covariance provides a reasonable estimation of the Gaussian kernel parameter

Figure 5.8: The partition results of KKM and standard DA algorithms on X-O data. KKM is sensitive to data initializations.

for the proposed KDA algorithm. Table 5.1 shows the optimal range ρ^o and the estimated ρ^* for each data set. Where ρ^* refers to the estimated value of ρ by using (5.21), ρ^o refers to the experimental range of ρ by applying KDA with different values of ρ and selecting the range that yielded the lowest partition errors. Obviously, the estimated ρ^* is located inside the optimal range ρ^o . This reveals the effectiveness of the proposed Gaussian kernel parameter selection method.

- The proposed KDA is suitable for partitioning different kinds of data sets, which possess different size, shape and distribution as shown in Fig 5.2. For each data, KKM can generate a satisfactory result in some runs, however, it is sensitive to data initialization, which may not be practical in real applications. In contrast, KDA partitions the data through annealing process such that it is independent of data initialization.
- The proposed KDA is mass constrained such that it can tackle the problem which arises when there are great difference in the number of points in different clusters [33], as observed from the partitions of unbalanced data.
- The proposed KDA is a generalization of DA by using a kernel-induced distance measure, which can reveal structure in the data that may go unnoticed if DA is performed in the original input space, as observed from the partitions

of ring data and X-O data.

Table 5.2: The partition errors by applying KDA with different values of ρ and KDA with estimated ρ^* on original Iris data.

$\rho =$	0.01	0.1	0.5	0.8	1.0	2.0	5.0	$\rho^* = 0.8813$
Errors	17	14	12	10	10	23	31	10

5.3.2 Real-life Data

The purpose of this subsection is to examine the usefulness of KDA for real-life data. Iris data is a standard benchmark in the pattern recognition literature and can be obtained from the UCI repository [105]. The data set contains three classes (Iris Setosa, Iris Versicolor and Iris Virginica) of 50 points each, where each class refers to a type of iris plant. One class is linearly separable from the other two; the latter are not linearly separable from each other. In all experiments the number of clusters is equal to 3. To investigate the influence of kernel parameter on data clustering in the feature space, we test this data set by applying KDA with different values of ρ , whose partition results are shown in Table 5.2. It is obvious that different ρ generates different partition errors. Certain values, as e.g., $\rho = 0.8$ and $\rho = 1.0$ have the minimum errors 10, while others have much more errors. The partition result of KDA with estimated $\rho^* = 0.8813$ is shown in the last column of Table 5.2. As we can see it has only 10 partition errors, which verifies that the scaled inverse of data covariance is a reasonable estimation of the Gaussian kernel parameter.

To provide the performance comparison of KDA to KKM and standard DA, we test the original Iris data and its first two principle components by the three meth-

Table 5.3: The partition errors by applying three methods on original Iris data and its first two principle components (PCs).

	DA	KKM	KDA
Original Iris Data	16	minimum:7, modal:15, maximum:23	10
Iris' First Two PCs	5	minimum:3, modal:4, maximum:22	3

Table 5.4: The partition errors of four traditional clustering methods on original Iris data and its first two principle components (PCs). Here we run each algorithm 20 times for each data set with random initialization and show only the best results.

	FCM	K-Means	NC	PCM
Original Iris Data	16	16	11	11
Iris' First Two PCs	8	6	8	8

ods. The clustering results are shown in Table 5.3. From the view of classification, the partition of DA has 16 errors in 150 points when the original data is tested. In contrast, the proposed KDA has only 10 errors. Our result compared favorably with other non-parametric clustering algorithms: the support vector clustering algorithm [4] leads to 14 misclassifications and the alternative fuzzy c-means algorithm [109] algorithm has 13 misclassifications. Though the best result of KKM has 7 errors, however, KKM is sensitive to data initialization and the result of 7 errors occurs only once in total 20 runs, which may not be practical in real applications. We also perform KDA on its first two principal components (PCs), the number of misclassified points is reduced to only 3, this is the best result obtained by unsupervised clustering algorithms so far as reported in [83]. In contrast, the result of DA has 5 errors for the first two principal components. Note the best result of KKM has 3 errors, which occurs only 5 times in total 20 runs. The comparisons are concluded in Table 5.3. The partition errors of four traditional algorithms (i.e., fuzzy c-means (FCM), K-means, noise clustering (NC), and possibilistic c-means (PCM)) on original Iris data and its first two principle components are also listed in Table 5.4 for further comparisons. It is obvious that the results (the best ones out of 20 runs) of the traditional clustering methods are worse than the proposed KDA algorithm.

5.4 Several Further Investigations

Several important issues, i.e., the estimation of optimal cluster number, the robust pruning of noisy points and the image illustration of data structure in feature

space, are addressed in this section.

Figure 5.9: Optimal cluster number estimation by contour plot of kernel matrix K .

5.4.1 Optimal Cluster Number Estimation

In chapter 3 and chapter 4, the VC-bound has been used to select the optimal cluster number for RDA and IFCSS clustering algorithms. However, the feature space induced by Gaussian kernel has infinite VC-dimension [93], which leads to the infinity of VC-bound independence of the cluster number, such that VC-bound can not be used for KDA algorithm. As addressed in [39], the kernel matrix may reveal the underlying structure of the data. Let the $l \times l$ kernel matrix denote by $K = [k(x_i, x_j)]$ ($i, j = 1, 2, \dots, l$). As each element of the kernel matrix defines a dot-product distance in the kernel-induced feature space the matrix will have a block diagonal structure when there are definite groupings or clusters within the data sample. This can be demonstrated by the contour plot of kernel matrix for the four data sets tested in last section as shown in Fig 5.9. The block structure of the matrix is apparent: there are 5 clusters within unbalanced data, 2 clusters within ring data, 3 clusters within both X-O and Iris data. Hence, the contour plot of kernel matrix provides a good estimation of the optimal cluster number. Note block structure of kernel matrix is determined by the Gaussian kernel parameter when Gaussian kernel function is used. A suitable value of Gaussian kernel parameter is the premise of the success of above optimal cluster number estimation. Fortunately, we have proposed a method to adaptively select a reasonable value

for Gaussian kernel parameter in section 2.3 and its effectiveness has been verified by the experimental results in section 3, which in result assures the practicality of above optimal cluster number estimation.

Figure 5.10: Simple demonstration of KDA against noisy points using the maximization of mutual information. The high value of λ , the more robustness of KDA. Note the detected outliers are not shown.

5.4.2 Robust Pruning of Noisy Points

It has been shown in chapter 4 that the maximization of mutual information against the prior probability can be used to phase out the outliers. Here we utilize this method to achieve the robustness of KDA against noisy points, which is implemented by maximizing the following objective function (all discussions below are in the feature space.)

$$F_{MI} = \max_{P(\Phi(X))} \left(I(\Phi(X), \Phi(W)) - \lambda \sum_{j=1}^l p(\phi(x_j)) e_j \right) \quad (5.27)$$

where $\lambda \in [0, +\infty)$ is a separate parameter to control the degree of robustness against outliers. $I(\Phi(X), \Phi(W))$ is the mutual information given by

$$I(\Phi(X), \Phi(W)) = \sum_{j=1}^l \sum_{k=1}^c p(\phi(x_j)) \bar{p}(\phi(w_k) | \phi(x_j)) \log \frac{p(\phi(x_j) | \phi(w_k))}{p(\phi(x_j))} \quad (5.28)$$

and e_j is the expense of using the j th input point, which is given by

$$e_j = \sum_{k=1}^c \bar{p}(\phi(w_k)|\phi(x_j)) D_k(x_j) \quad (5.29)$$

the resultant prior that maximize F_{MI} is given by

$$p(\phi(x_j)) = \frac{p(\phi(x_j))c_j}{\sum_{j=1}^l p(\phi(x_j))c_j} \quad (5.30)$$

where c_j is the capacity of the j th input point, which is defined by

$$c_j = \exp \left(\sum_{k=1}^c \bar{p}(\phi(w_k)|\phi(x_j)) \log \frac{\bar{p}(\phi(w_k)|\phi(x_j))}{\bar{p}(\phi(w_k))} - \lambda e_j \right) \quad (5.31)$$

As discussed in chapter 4, the average mutual information (5.28) is not negative. However, the individual item in the sum of the capacity can be negative. If the j th point x_j is taken into account and $p(\phi(w_k)|\phi(x_j)) < \sum_{i=1}^l p(\phi(x_i))p(\phi(w_k)|\phi(x_i))$, then the probability of the k th prototype (cluster center) is decreased by the observed point and gives a negative information about x_j . Then the particular input point may be considered as an unreliable point (outlier) and its negative effect must be offset by other input points. Therefore, the maximization of the mutual information (5.27) provides a good robust estimation of the noisy points (outliers or noise), where the degree of robustness is controlled by the value of λ : the higher value λ is, the more robustness KDA achieves, as demonstrated by a simple noisy data in Fig 5.10.

5.4.3 Image Illustration in Feature Space

As discussed above, the nonlinearly mapping by means of Gaussian kernel function can increase the probability of the linear separability of the given data but preserve

Figure 5.11: The image illustration of noisy data, ring data, and X-O data in the feature space, where the topographic ordering of the data in the input space is kept while the probability of the linear separability of the data is increased.

the topographic ordering of the given data in the feature space, such that KDA can reveal structure in the data that may go unnoticed if DA is performed in the standard Euclidean space. However, the feature space induced by Gaussian kernel has infinite dimensionality, which makes the data visualization impossible in the feature space. To further investigate the data clustering in feature space, we provide a method to illustrate the image of the data points in the feature space by the kernel-induced distance distributions. Let the cluster centers (5.17) represent the coordinates, then the position of each data point can be determined by the kernel-induced distances (5.5) from the given point to the cluster centers. Fig 5.11a plots the image illustration of the noisy data (see Fig 5.10a) in the feature space by above method. We can see there are two obvious separated clusters (the map of the two rings) in the feature space with some outliers (marked from 1 to 4) lying between the two clusters and others (marked from 5 to 8) lying outside the two clusters. From the illustration we obtain an interesting observation: the topographic ordering of the data in the data space is preserved in the feature space, while the data structure is simplified from linearly nonseparable to linearly separable. We also illustrate the data structures of ring data and X-O data in the feature space in Fig 5.11b and Fig 5.11c, respectively. Similarly, the original data structure in the input space is kept while the probability of the linear separability

of the data is increased in the feature space. This explain to some extent why the kernel-based clustering algorithms, e.g., KDA, in the feature space, normally perform better than the traditional clustering algorithms, e.g., DA, in the standard Euclidean space.

5.5 Conclusion

In this chapter, we have extended and generalized the standard DA algorithm into feature space using a kernel approach and developed an kernel-based deterministic annealing (KDA) algorithm for data clustering. This follows the same basic idea of Support Vector Machines (SVM) and kernel principal component analysis (KPCA) by employing the nonlinear data mapping and the kernel trick. More importantly, a reasonable value of Gaussian kernel parameter (width) is auto-determined by the scaled inverse of data covariance, which induces the adaptiveness of the proposed clustering method. In order to demonstrate the effectiveness of the KDA clustering algorithm in providing optimal clustering results, several artificial and benchmark data sets have been considered. The results showed that: 1) the scaled inverse of data covariance is a reasonable estimation of the Gaussian kernel parameter (width); 2) the proposed KDA clustering method is capable of clustering different kinds of data with different size, shape and distribution; 3) the proposed KDA clustering method provides a performance that is superior to that of standard DA clustering algorithm and kernel-based K-means (KKM) clustering algorithm.

Chapter 6

Further Investigation on VC-bound as Cluster Validity

6.1 Introduction

Clustering plays an important role in many engineering fields such as pattern recognition, system modelling, image processing, communication, data mining, and so on. Most traditional clustering methods, such as fuzzy c-means (FCM) and deterministic annealing (DA), need to pre-select the optimal cluster number of clusters, which is generally unknown in practical applications. Thus, an evaluation methodology is required to validate each of the partitions (according to $c = 2, 3, \dots, c_{max}$) and to obtain an optimal partition (or optimal number of clusters c^*). This quantitative evaluation is the subject of cluster validity. The mathematical formula used to compute the validation is referred to as a cluster validity index. In chapter 3 and chapter 4, the cluster validity, i.e., VC-bound, has been presented for RDA and IFCSS algorithms to estimate the optimal cluster number, where the empirical risk was represented by the noise ratio. In this chapter, we will further investigate its usage as a cluster validity for standard clustering algorithms, especially for DA algorithm, where the empirical risk will be represented by the scaled objective

function. This is the major objective of this chapter.

In most traditional clustering algorithms, the relationship between data points and clusters is represented in one of two ways: each data point may belong to one and only one cluster (hard clustering); or each data point may belong to more than one clusters with a certain degree of membership (fuzzy clustering). The latter type of representation is more general and more precise in principle, although its practical value is still not clearly proven [5]. As discussed in [75], the fuzzy and probabilistic label vectors (i.e., memberships by FCM or probabilities by DA) are mathematically identical, having entries between zero and one that sum to one over each column: Assume l data points are partitioned into c clusters, and let the partitioned label vectors be denoted by $P = [p_{k|j}]$ ($k = 1, 2, \dots, c$ and $j = 1, 2, \dots, l$), then the following conditions hold: 1) $p_{k|j} \in [0, 1] \forall j, k$; 2) $\sum_k p_{k|j} = 1 \forall j$; 3) $0 \leq \sum_j p_{k|j} \leq l \forall k$. From the view of this point, the DA clustering algorithm can be categorized into fuzzy clustering, though a unique hard clustering partition can be obtained as a byproduct in the limited low temperature. It is therefore interesting to investigate the performance of the existing fuzzy cluster validity indices on the partitions produced by DA clustering algorithm. This is another important objective of this chapter.

Many indices have been proposed in the literature which are used to measure the fitness of the partitions produced by clustering algorithms. The first fuzzy cluster-validity functions associated with the fuzzy c -partitions introduced by Bezdek are the partition coefficient (PC) and the partition entropy (PE) [6] [7]. The major drawback of these indices is that they use only the fuzzy membership degrees for each cluster without considering the data structure of the clusters [54] [75]. To overcome this weakness, Xie and Beni [112] and Fukayama and Sugno [36] have introduced new fuzzy validity criteria, denoted by XB and FS indices, respectively, for evaluating fuzzy c -partitions by exploiting the concept of compactness and separation. They combined, with a unique function, the properties of the fuzzy membership degrees and the structure of data, and therefore took into account the

geometrical properties of the input data. Kwon extended the XB index to eliminate its tendency to monotonically decrease when the number of clusters approaches to the number of data points [61]. Most recently developed indices focused on the modification or extension of the traditional compactness and separation to achieve better performance, e.g., [54], [73], [110], [114], etc.. Different from the indices described above, which tend to focus on compactness and separation, the fuzzy hypervolume (FHV) and partition density (PD) validity functions proposed by Gath and Geva [37] are based on the measure of the degree of variance within each cluster. The lately proposed indices using this kind of measure can be found in [14] [82].

In this chapter, a new approach for clustering validity is proposed from the view of statistical learning theory. The proposed VB index is estimated based on the structural risk minimization (SRM) principle [107], which optimizes the bound simultaneously over both the distortion function (empirical risk) and the VC-dimension (model complexity) to achieve the minimum of the guaranteed risk. The empirical risk is monotonically decreasing with the increase of the cluster number c . While the model complexity is monotonically increasing with the index of cluster number. The VB index aims to find a minimum value of the VC-bound (within the cluster number range) to make a trade-off between the minimizations of the empirical risk and model complexity. The corresponding cluster number and its partition validates the best description of the data structure. To construct the VB index, the empirical risk is represented by the scaled distortion function of DA clustering algorithm, and the VC-dimension of system structure is found to be equal to the number of parameters of a set of specifically defined indicator functions. The validity indices are considered to be independent of clustering algorithms. Though the proposed VB index is applicable to any fuzzy and crisp clustering algorithms, we only investigate its performance on the partitions produced by the DA clustering algorithm.

The rest of this chapter is organized as follows. The subsequent section contains

the derivation of the proposed new cluster validity, i.e., VB index. We first introduce the SRM principle and VC-bound, then apply VC-bound for optimal cluster number estimation, and finally give the pseudo-code of the proposed method. In section 3, the experimental results demonstrating the superiority of the VB index in appropriately determining the number of clusters, as compared to other well-known validity indices, are provided on several numerical examples and real data sets. Conclusion and discussion are given in Section 4.

6.2 The Proposed New Cluster Validity

In this section, we propose a novel practical cluster validity called VB index for data clustering from the view of statistical learning theory [107]. For this purpose, we need firstly to introduce the basis of structural risk minimization (SRM) principle and related VC-bound as below.

Figure 6.1: Graphical depiction of the SRM principle. The SRM picks a function f_* which has small training error (empirical risk), and comes from an element of the structure that has low capacity h (confidence interval), thus minimizing a risk bound in (6.3). Figure taken from [93].

6.2.1 SRM Principle and VC-bound

The principle of structural risk minimization (SRM) is a key issue to obtain good generalization performances for a variety of learning machines, as e.g., the well-known support vector machine (SVM). The SRM principle finds the function that for the fixed amount of data achieves the minimum of the guaranteed risk. To find the guaranteed risk, one has to use bounds, e.g., VC-bound, on the actual risk. Under the SRM, a set of admissible structures with the nested subsets is defined as follows

$$S_1 \subset S_2 \subset \dots \subset S_c \dots \quad (6.1)$$

with the non-decreasing VC-dimension of the structure

$$h_1 \leq h_2 \leq \dots \leq h_c \dots \quad (6.2)$$

where $S_c = (Q^c(x_j, V) : V \in \Lambda_c), \forall j$ denotes the element of the structure at cluster number c , with a set of indicator functions $Q^c(x_j, V)$ of the empirical risk defined according to the problems under investigation [107]. Here $\Lambda(c)$ denotes a set, to which the V belongs, at the specific cluster number c . The task of the SRM principle is to choose the element S_c of the structure for which the smallest bound on the real risk (the smallest guaranteed risk)

$$V_b \leq R_{ems} + \frac{\varepsilon}{2} \left(1 + \left(1 + R_{ems} \frac{4}{\varepsilon}\right)\right)^{1/2} \quad (6.3)$$

with

$$\varepsilon = \frac{h_c (\log \frac{2l}{h_c} + 1) - \log \frac{\zeta}{4}}{l} \quad (6.4)$$

is achieved. Above $\zeta < 1$ is a constant (we set $\zeta = 0.01$ in all experiments). The first term of the right hand side of the bound (6.3) represents the empirical risk and the second term is the confidence interval of the SRM based estimation. The bound on the risk is the sum of the empirical risk and of the confidence interval. The empirical risk is decreasing with the index of element of the structure (determined by cluster number in data clustering, refer to the next subsection), while the confidence interval is increasing. The smallest bound of the risk is achieved on some appropriate element of the structure, as illustrated in Fig 6.1. The bound (6.3) above is called the VC-bound (VC is the abbr. of Vapnik-Chervonenkis) in the literature. We apply it for data clustering to select the optimal cluster number in the next subsection. The induced cluster validity criteria is called VB index in this chapter.

6.2.2 Cluster Number Selection by VB index

Assume the given data is partitioned by the DA clustering algorithm, which results in certain partitions with according probabilities $p(v_k|x_j)$ ($k = 1, 2, \dots, c$ and $j = 1, 2, \dots, l$) within the cluster number range ($c = 2, 3, \dots, c_{max}$). The key issue of applying (6.3) for cluster number selection is to construct a set of indicator functions specifically for DA clustering algorithm. Let's consider the function

$$P_k = \lim_{T \rightarrow 0} p(v_k|x_j) = \lim_{T \rightarrow 0} \frac{p(v_k)e^{-d(x_j, v_k)/T}}{\sum_{i=1}^c p(v_i)e^{-d(x_j, v_i)/T}} \approx \begin{cases} 1 & \text{if } x_j \in v_k \\ 0 & \text{otherwise} \end{cases} \quad (6.5)$$

where $x_j \in v_k$ means the point j is belonged to cluster k . Note the independent variable $d(x_j, v_k)$ can be presented as an inner product of two (n) -dimensional vectors of the input space as

$$d(x_j, v_k) = \|x_j - c_k\|^2 = [(d_{kj}^1)^2 + (d_{kj}^2)^2 + \dots + (d_{kj}^n)^2] \quad (6.6)$$

where $d_{kj}^r (r = 1, 2, \dots, n)$ denotes the r th element of the vector $x_j - v_k$, and n is the dimensionality of the input data. Then the function (6.5) can be approximated by the step function (denoted by θ) as

$$P_k \approx \theta\left(\sum_{i=1}^n \gamma_{ki} \phi_{ki}(x_j)\right) \quad (6.7)$$

Where γ_{ki} is the diameter parameter and ϕ_{ki} is linearly independent function. We can now construct the indicator functions for the DA clustering algorithm as

$$Q^c(x_j, V) = \sum_{k=1}^c P_k \approx \sum_{k=1}^c \theta\left(\sum_{i=1}^n \gamma_{ki} \phi_{ki}(x_j)\right) \quad (6.8)$$

It can be seen that the above constructed indicator functions (6.8) are linear in their parameters. According to [107], the VC-dimension of the structure is equal to the number of parameters of (6.8), i.e.,

$$h_c = c \times n \quad (6.9)$$

for each nested subset S_c , such that the increase of cluster number is proportional to the increase of the estimated VC-dimension. To obtain good generalization performance one has to use the admissible structure (6.1) based on the set of indicator functions (6.8) to search for an optimal cluster number that minimizes the VC-bound (6.3).

Another important issue of applying (6.3) for cluster number selection is to represent the empirical risk in the VC-bound for DA clustering algorithm. It is well-known for the pattern recognition problem, the empirical risk in SRM estimation is defined based on the misclassified error as follows,

$$R_{ems} = \frac{1}{l} \sum_{j=1}^l \text{sign}(y_j - \bar{y}_j) \quad (6.10)$$

Figure 6.2: The distortion J_c is monotonously decreasing with respect to the increase of cluster number (a). The proposed VB index with respect to cluster number. The minimum is reached at $c^* = 5$ (b).

where y_j is the actual class label of x_j and $\bar{y}_j$ is the estimated class label of x_j . Inspired by (6.10), One of the natural representations¹ of empirical risk for clustering problem can be obtained based on the distortion measures. Hence, we define it as

$$R_{ems}(c) = J_c/J_{var} \quad (6.11)$$

where J_c is the distortion obtained by DA at cluster number c ($c = 2, 3, \dots, c_{max}$), which is defined by

$$J_c = \sum_{k=1}^c \sum_{j=1}^l p(x_j) p(v_k|x_j) d(x_j, v_k) \quad (6.12)$$

¹Another natural representation of empirical risk is the noise ratio as in the robust clustering methods, as e.g., RDA algorithm in chapter 3 and IFCSS algorithm in chapter 4.

J_{var} is the variance of the input data, which is defined by

$$J_{var} = \frac{1}{l} \sum_{j=1}^l \|x_j - \bar{x}\|^2 \quad (6.13)$$

where $\bar{x} = \frac{1}{l} \sum_{j=1}^l x_j$ is the mean of the input data. The distortion J_c is scaled by the variance such that the induced $R_{ems}(c)$ is located inside the range of $[0, 1]$. It has been shown in [85] [87], the distortion J_c is monotonously decreasing with respect to the increase of cluster number in the annealing process. In Fig 6.2a, we plot the distortion measures J_c with respect to the cluster number c according to the phase transitions in Fig 2.1 in chapter 2. For a given data, the variance J_{var} is a constant, so the empirical risk $R_{ems}(c)$ is monotonously decreasing with respect to the increase of cluster number, i.e., the quality of approximation of the data structure is increasing, which is in accordance with the description of SRM estimation, see Fig 6.1. As shown in the experimental results, this representation normally leads to the convexity of the curve of VB index with the minimum value at the optimal cluster number. In conclusion, the proposed VB index is stated as: by evaluating the estimated VC-bound for each chosen cluster number by equation (6.3), where h_c is defined by (6.9) and $R_{ems}(c)$ is defined by (6.11), we select the one that yields the minimum value of V_b as the optimal cluster number. Fig 6.2b shows the values of V_b with respect to the cluster number c from 2 to 10 for the data set in Fig 2.1 in chapter 2. It can be observed that the minimum of V_b is reached at $c^* = 5$, which correctly reveals the underlying data structure.

6.2.3 Pseudo-Code of the Proposed Method

According to the above discussions, we now give a detailed pseudo-code of the DA clustering algorithm with the proposed VB index for cluster number selection as follows.

- Step 1) Set the maximum number of clusters c_{max}^2 , initial temperature $T_{ini} > 2T_1^*$ (see (2.10)), minimum temperature $T_{min} = T_{ini}/1000$, convergence parameter $\varepsilon = 0.001$, cluster center $v_1 = \frac{1}{l} \sum_{j=1}^l x_j$ with mass probability $p(v_1) = 1$, source distribution $p(x_j) = \frac{1}{l}$ ($j = 1, 2, \dots, l$), and $c = 1$.
- Step 2) Alternatively update the titled distribution (2.7), and the cluster centers (2.9) for $k = 1, 2, \dots, c$ (fixed point iterations) until the maximum change in the cluster centers between consecutive iterations is less than the given threshold value ε .
- Step 3) Save the induced partition, and calculate the value of V_b (6.3) for the current cluster number c . If $c = c_{max}$, set $flag = 0$ then go to step 5; otherwise, go to the next step.
- Step 4) If $T < T_{min}$, set $flag = 1$ then go to step 5; Otherwise, let $T = \eta T$ ($0 < \eta < 1$), and check condition of phase transition for $k = 1, 2, \dots, c$, if critical T_k^* is reached for cluster k (see (2.10)), add a new cluster center by $v_{k+1} = v_k + \delta$ with $p(v_{k+1}) = p(v_k)/2$ and $p(v_k) = p(v_k)/2$, where δ is a small disturbance, and let $c \leftarrow c + 1$, then go to step 2.
- Step 5) If $flag = 0$, then select the optimal cluster number c^* by $c^* = \arg \min_c V_b$, where $c = 2, 3, \dots, c_{max}$, and recover the saved partition according to the optimal cluster number c^* . If $flag = 1$, then indicate that the clustering procedure is failed.

6.3 Experimental Results

To demonstrate the effectiveness and superiority of the proposed VB index for determining the optimal cluster number, we conducted extensive comparisons with

²There is no general agreement on what value to use for c_{max} , a rule of thumb that many investigators use is $c_{max} \leq (l)^{1/2}$ [75]. Strict adherence to this rule is not recommended., however, in most situations there will be some practical limit for c_{max} that is known to the user. In this chapter, we set $c_{max} = 8$ in all experimental results.

Figure 6.3: The simple data set and its three derivations.

other cluster validity indices on three numerical examples and two real data sets. The proposed VB index was compared with six well-known fuzzy cluster validity indices introduced in chapter 2.3: Bezdek's PC and PE [6] [7], Xie and Beni's XB [112], Fukuyama and Sugeno's FS [36], and Gath and Geva's FHV and PD [37].

6.3.1 Numerical Examples

For the purpose of data structure visualization, we only consider two-dimensional (2D) data sets in this subsection. We test the effectiveness and robustness of VB index for optimal cluster number selection on three specific data sets. The three data tested are noisy data shown in Fig 6.3b, overlapped data shown in Fig 6.3c, and unbalanced data shown in Fig 6.3d. They are all derived from a simple data set shown in Fig 6.3a, which is generated from four Gaussian distributions with means $[(-3, -3), (-3, 3), (3, -3), (3, 3)]$, variances $[0.6, 0.6, 0.6, 0.6]$, and number $[30, 30, 30, 30]$. All the cluster validity indices can find the optimal cluster number $c^* = 4$ for this simple data (the results are not shown in this chapter).

Example 1: noisy data

The noisy data, as shown in Fig 6.3b, is generated by contaminating the simple data set with 60 random noisy points. Robust clustering are commonly referred to two respects: outlier and cluster number that are closely linked to each other [29]. Not knowing the number of clusters in a data set complicates the task of separating

Figure 6.4: Performances of different cluster indices on noisy data (a-g). Clustering result of DA with $c^* = 4$ for noisy data (h).

the good data points from the noisy points, and conversely, the presence of noise makes it harder to determine the number of clusters. A good validity index should work well in the noisy situations. We plot the values of V_b (6.3) with respect to the cluster number $c = 2, 3, \dots, 8$ in Fig 6.4a. It can be seen that the minimum of V_b is reached at $c^* = 4$, so that the proposed VB index correctly reveals the underlying cluster number for this noisy data. We also plot the values of PC, PE, XB, FS, FHV and PD with respect to cluster number c in Fig 6.4b-g, respectively. Where only FHV and PD indices can also find the correct cluster number $c^* = 4$ for this data set as shown in Fig 6.4f and Fig 6.4g. Among the unsuccessful indices, PC, PE, and FS indices indicate the optimal value of $c^* = 7$. XB index points to the optimal number $c^* = 6$; however, it indicates that $c^* = 4$ may be another good cluster number estimation as shown in Fig 6.4b. For clustering result visualization, Fig 6.4h shows the partition obtained by DA clustering algorithm with $c^* = 4$.

Figure 6.5: Performances of different cluster indices on overlapped data (a-g). Clustering result of DA with $c^* = 4$ for overlapped data (h).

Example 2: overlapped data

The overlapped data, as shown in Fig 6.3c, is derived by moving the means of the four clusters in the simple data set from $[(-3, -3), (-3, 3), (3, -3), (3, 3)]$ to $[(-1, -1), (-1, 1), (1, -1), (1, 1)]$ but keeping other parameters unchanged. In practical applications, the data sets are normally overlapped. Hence, it is interesting to investigate the performances of existing validity indices including the proposed VB on this overlapped data set. The plots of different validity indices with different cluster number $c = 2, \dots, 8$ are shown in Fig 6.5. Four of them correctly point to the optimal cluster number $c^* = 4$ for this overlapped data set: VB, XB and FHV indices reach the minimum and PD gets the maximum at the point $c^* = 4$. The corresponding partition from DA clustering algorithm at $c^* = 4$ is shown in Fig 6.5h, which correct reveals the underlying data structure. On the other hand, the unsuccessful indices, PC, PE and FS present a monotonic tendency of the cluster number c where they give a largest optimal cluster number estimation $c^* = 8$.

Figure 6.6: Performances of different cluster indices on unbalanced data (a-g). Clustering result of DA with $c^* = 4$ for unbalanced data (h).

Example 3: unbalanced data

The unbalanced data, as shown in Fig 6.3d, is derived by changing the number of one cluster in the simple data set from 30 to 240 and the according variance from 0.6 to 1.2 but keeping other parameters unchanged. The situation that there are great difference in the number of samples in different clusters may occur in the practical applications, so it is also interesting to investigate this unbalanced data set. Fig 6.6 shows all the validity indices with respect to the cluster number $c = 2, \dots, 8$. It can be observed that the unbalance of the samples does not obviously affect the performances of validity indices, the optimal number of clusters $c^* = 4$ for this unbalanced data set is correctly recognized by all validity indices except the PD. Though PD points to $c^* = 6$ as the optimal value, it also indicates that $c^* = 4$ is a possible estimation. The partition of DA clustering algorithm with $c^* = 4$ is shown in Fig 6.6h for clustering result visualization. It provides a good description of the underlying data structure, only two points are misclassified.

Figure 6.7: Performances of different cluster indices on Iris data (a-g). Clustering result of DA with $c^* = 3$ for Iris data by its two dominating features (PL and PW) (h).

6.3.2 Two Real Data Sets

The Iris and Wine data are investigated in this subsection for validity index comparisons. They are standard benchmarks in the machine learning literature and can be obtained from the UCI repository [105].

Example 4: Iris data

The Iris data set contains three classes (Iris Setosa, Iris Versicolor and Iris Virginica) of 50 points each, where each class refers to a type of iris plant. One class is linearly separable from the other two; the latter are not linearly separable from each other. Thus, one can argue $c^* = 2$ or $c^* = 3$ for the Iris data set. The validity indices of the Iris data set are shown in Fig 6.7. The proposed VB index reaches the minimum at $c^* = 3$ so that correctly reveals the underlying cluster number. The partition of DA clustering algorithm with $c^* = 3$ is shown in Fig 6.7h by Iris' two dominating features (i.e., PL - petal length and PW - petal width). XB index

Figure 6.8: Performances of different cluster indices on Wine data (a-g). Clustering result of DA with $c^* = 3$ for Wine data by its first two principle components (h).

shows that $c^* = 2$, argued as another correct number for Iris [75], is the optimal cluster number estimation. PD index regards $c^* = 8$ as the optimal number and indicates that $c^* = 4$ may be another optimal number. The results of other indices (including PC, PE, FS, and FHV) are unexpected since they present a monotonic tendency of the cluster number c where they give a largest optimal cluster number estimate $c^* = 8$.

Example 5: Wine data

As a final data set we present results from a wine recognition problem. The Wine data consists of 178 13-dimensional samples which are a set of chemical analysis of three types of wine with number of samples 59, 71, and 48 respectively. Fig 6.8 shows the cluster validity results for the normalized Wine data set. Two of them find the correct cluster number: both VB and XB indices reach the minimum at the point of $c^* = 3$. The partition of DA clustering algorithm with $c^* = 3$ is shown in Fig 6.8h by Wine' first two principal components (taken from [84]). FHV indicates that $c^* = 3$ may be an optimal cluster number, though it reaches the

minimum at $c^* = 8$. PC, PE, and FS indices failed in this case, they show that $c^* = 7$ is the optimal cluster number. PD index doesn't work in this case since it possesses a constant value with independence of cluster number c .

Table 6.1: Values of c^* preferred by each cluster validity index for the 5 data sets tested in this chapter.

Data Set	c_{opt}	VB	XB	FS	PC	PE	FHV	PD	AIC	BIC
Noisy	4	4	6(4)	7	7	7	4	4	6	5
Overlapped	4	4	4	8	8	8	4	4	6	4
Unbalanced	4	4	4	4	4	4	4	7	7	6
Iris	2/3	3	2	8	8	8	8	8	3	3
Wine	3	3	3	7	7	7	8(3)	—	3	2

2/3 means both $c^* = 2$ and 3 are argued as optimal number for Iris; (x) means the number x inside the bracket is indicated as another good estimation; — means PD index doesn't work on Wine.

6.4 Conclusion and Discussion

Interest in clustering has increased recently because of new areas of application, such as data mining, image and speech processing, and bio-informatics. A central issue in these and other applications of clustering is how many clusters provide an appropriate description of the data. This is the main issue of cluster validity. In this chapter, we have proposed a new validity index, called VB index, for data clustering. It is estimated based on the structural risk minimization (SRM) principle in statistical learning theory, which optimizes the bound simultaneously over both the distortion function (empirical risk) and the VC-dimension (model complexity) to reach the minimum of the guaranteed risk, i.e., the best generalization performance. The smallest value of VB index on some appropriate cluster number validates the best description of the data structure. Its effectiveness and superiority are demonstrated by comparing to several popular cluster-validity indices on three numerical examples and two real data sets. And the results of comparative study show that the proposed VB index has high ability in producing a good cluster number estimation.

Table 6.1 summarizes the results obtained when the 7 different validity indices were applied to the 5 different data sets tested. The column c_{opt} in Table 6.1 gives the optimal number of clusters for each data set, and the other columns show the optimal cluster numbers obtained using each validity index. The proposed VB index is the only index that correctly recognizes the number of clusters for all data sets; hence VB is the most effective of the indices considered. The XB index correctly identifies the optimal c^* in all data sets except the noisy data. It indicates that the correct number $c^* = 4$ may be another good cluster number estimation for the noisy data; hence XB is also an effective index. FHV and PD work very well on most of the numerical examples. However, they fail in most of the real data sets. Though they have been widely used for c-shells cluster number estimation, as in [56] and [68], the practical applications of these two indices need to be further investigated. On the other hand, PC, PE and FS indices present a monotonic tendency of the cluster number c on most of the data sets. They can not find the correct cluster number for all data sets except the unbalanced data set; hence these three indices are proved as the most ineffective of the indices considered. Note the direction of the monotonic tendency of PC (and PE) on FCM clustering algorithm is the inverse of that on DA clustering algorithm. This is induced by the fact that they use only the membership from clustering algorithms. The fuzzy membership degree of FCM is increasing with the increase of the cluster number assume that the fuzziness exponent keeps a constant, while for DA the probabilistic membership degree is decreasing with the increase of the cluster number (as the temperature decreases from high value to low value).

The VB induced cluster validity index is basically a model selection criterion [107]. In the literature of statistical learning theory, there exist many model selection criteria, such as the Akaike information criterion (AIC) [2] and Bayesian information criterion (BIC) [94]³, which have been widely used in supervised classifications and regressions for model selection. In recent years, the AIC and BIC have also been

³(please refer to Appendix C for a brief introduction of these two information criterion rules for cluster number selection)

extended to determine the cluster number for unsupervised clustering, especially in biological and biomedical engineering. To provide a comparison of the proposed VB to AIC and BIC for optimal cluster number selection, we list the numerical results of AIC and BIC on the 5 different data sets in the last two columns in Table 6.1. It can be seen that VB has better performance than AIC and BIC on the tested data sets: AIC can only find the correct cluster number for Iris and Wine data, but failed in other cases; and BIC can only reveal the correct cluster number for Iris and overlapped data, but failed in other cases; in contrast, VB indicates the correct cluster number for all of data sets. We think the optimal cluster number selection based on model selection criterion is an interesting topic worthy of further investigation.

Chapter 7

Image Segmentation By Deterministic Annealing Algorithm With Adaptive Spatial Constraints

7.1 Introduction

Image segmentation is a process which divides images into several meaningful areas such that the segmented image can be further analyzed and interpreted. It plays an important role in a variety of applications such as robot vision, object recognition, and medical imaging [74] [78]. In the last decades, fuzzy segmentation methods, especially the fuzzy c-means algorithm (FCM) [8], have been widely used in the image segmentation and such a success chiefly attributes to the introduction of the fuzziness for belongingness of each image pixel. Unlike hard segmentation methods which force pixels to belong exclusively to one class, fuzzy segmentation methods allow pixels belong to multiple classes with varying degrees of membership, which makes the fuzzy segmentation methods be able to retain more information from

the original image than the hard segmentation methods [79]. Although the original intensity-based FCM algorithm performs well on the segmentation of most noise-free images, it fails to classify the images corrupted by noise, outliers, and other image artifacts, such as the intensity inhomogeneity induced by the radio-frequency coil in magnetic resonance imaging (MRI), and thus, makes accurate segmentation difficult. In addition, many image pixels in a real image are ambiguous and cannot be classified consistently based on feature attribute(s) alone. As observed in a real image, pixels of the same object usually form coherent patches. Thus, the incorporation of local spatial information in the clustering process could filter out noise and other image artifacts and reduce classification ambiguities such that yield a more accurate segmentation result [63].

Many attempts have been made to introduce spatial contextual information into image segmentation procedures by modifying the objective function of conventional fuzzy clustering algorithm (mainly on FCM) to improve the segmentation performance. In [103], Tolia et al. proposed a fuzzy rule-based scheme called the rule-based neighborhood enhancement system to impose spatial continuity by postprocessing on the clustering results obtained using the FCM algorithm. In another paper [104], they incorporated a spatial continuity constraint into the FCM algorithm by either the addition of a small positive constant to, or subtraction of a small positive constant from, the membership value of the center pixel in a 3×3 window. In [1], Ahmed et al. proposed a modified FCM algorithm to compensate for the intensity inhomogeneity by allowing the label of a pixel to be influenced by the labels of its immediate neighborhood. The similar formulation is also used by Liew et al. in [64] for the segmentation of color lip images, and in another paper [63], they combined the spatial continuity constraints with a multiplier field containing the first- and second-order regularization terms for the segmentation of three-dimensional magnetic resonance (MR) images. More recently, Zhang et al. [116] proposed a modified FCM algorithm by using both a spatial penalty term and a kernel-induced distance measure for medical image segmentation, and Vovk et al. [108] incorporated spatial image features in addition to commonly used in-

tensity features, which give the method enough information to successfully correct even highly dynamic bias fields.

Spatially-constrained fuzzy c-means clustering (SFCM) algorithm can raise insensitivity to noise to some extent, however, there remains several problems to be improved: first, it is well known that FCM kind algorithms are sensitive to initialization and converges to local optimal solutions; second, how to automatically determine (but not set *a priori*) the strength of the spatial term is still a problem for most spatially-constrained fuzzy clustering algorithms. In this chapter, we present an adaptive spatially-constrained deterministic annealing algorithm (ASDA) by adaptively incorporating the local spatial information into the formulation of standard deterministic annealing (DA) algorithm [85] [87] to deal with these existing problems. The proposed algorithm offers several improved features compared to existing image segmentation methods. First, it is less prone to noise and has better convergence due to the incorporation of spatial information. Second, it is independent of the data initialization and has the ability to avoid many poor local optima due to the mass-constrained DA process. Lastly, the strength of the spatial constraint for each pixel is auto-determined by the scaled variance of its neighbor pixels to improve the adaptiveness of the proposed algorithm.

The rest of this chapter is organized as follows. Section 2 briefly review the conventional SFCM clustering algorithms for image segmentation. In section 3, we first derive the formulation and implementation of the proposed ASDA algorithm, then discuss the adaptive selection of spatial penalty parameter, and finally list the pseudo-code of ASDA for image segmentation. The experimental results on synthetic and magnetic resonance (MR) data are shown qualitatively and quantitatively in section 4. Finally, conclusion and discussion are presented in section 5.

7.2 Review of Spatially-Constrained Fuzzy C-means Algorithm

Mathematically, Fuzzy c-means algorithm (FCM) is derived to minimize the following objective function with respect to the membership function u_{jk} and the cluster center v_k as given by

$$J_{FCM} = \sum_{j=1}^N \sum_{k=1}^c u_{kj}^m \|x_j - v_k\|^2 \quad (7.1)$$

where c is the number of clusters, and N is the number of pixels, m is the weighing exponent on fuzzy memberships, and x_j is the observation at pixel j , in image segmentation, the most commonly used feature is the gray-level value, or intensity of image pixel. The minimization of (7.1) gives the updating equations for membership u_{kj} and cluster center v_k as follows:

$$u_{kj} = \frac{(\|x_j - v_k\|^2)^{1/(m-1)}}{\sum_{i=1}^c (\|x_j - v_i\|^2)^{1/(m-1)}} \quad \text{and} \quad v_k = \frac{\sum_{j=1}^N u_{kj}^m x_j}{\sum_{j=1}^N u_{kj}^m} \quad (7.2)$$

It is apparent from (7.1) that the FCM objective function does not take into consideration of any spatial dependence between observations. Thus the computed membership function and cluster center will exhibit a sensitivity to noise in the observed image. To overcome this problem, many spatially-constrained FCM (SFCM) algorithms with slightly different formulations have been made to improve the robustness of FCM against noise in image segmentation. Here we follow the approach proposed in [1] [64] to derive the formulation of SFCM algorithm by

modifying the objective function defined in (7.1) as follows:

$$J_{SFCM} = \sum_{j=1}^N \sum_{k=1}^c u_{kj}^m \|x_j - v_k\|^2 + \frac{\alpha}{N_R} \sum_{j=1}^N \sum_{k=1}^c u_{kj}^m \sum_{r \in N_j} \|x_r - v_k\|^2 \quad (7.3)$$

where N_j stands for the set of neighbors falling into a (normally 3×3) window around x_j (not including x_j) and N_R is its cardinality. The spatial penalty term is regularized by the parameter α . The relative importance of the regularizing term is inversely proportional to the signal-to-noise (SNR) ratio of the observed image. Low SNR would require a higher value of α , and vice versa.

An iterative algorithm for minimizing (7.3) can be derived by taking the first derivatives of (7.3) with respect to the membership function u_{jk} and cluster center v_k . The necessary conditions on u_{jk} and v_k for (7.3) to be at a local minimum are given by the following equations.

$$u_{kj} = \frac{(\|x_j - v_k\|^2 + \frac{\alpha}{N_R} \sum_{r \in N_j} \|x_r - v_k\|^2)^{1/(m-1)}}{\sum_{i=1}^c (\|x_j - v_i\|^2 + \frac{\alpha}{N_R} \sum_{r \in N_j} \|x_r - v_i\|^2)^{1/(m-1)}} \quad (7.4)$$

and

$$v_k = \frac{\sum_{j=1}^N u_{kj}^m (x_j + \frac{\alpha}{N_R} \sum_{r \in N_j} x_r)}{\sum_{j=1}^N (1 + \alpha) u_{kj}^m} \quad (7.5)$$

The SFCM algorithm is summarized as follows.

- Fix the cluster number c , select initial cluster centers $\{v_k\}_{k=1}^c$, and set threshold ε be a small positive value.
- Alternatively update the membership function and cluster center by using (7.4) and (7.5) until $\|v_{new} - v_{old}\| < \varepsilon$.

7.3 The proposed ASDA Clustering Algorithm

The derivation of the proposed adaptive spatially-constrained deterministic annealing (ASDA) algorithm is presented in this section.

7.3.1 Formulation and Derivation

As discussed above, pixels of the same object in a observed image usually form coherent patches, thus, the incorporation of local spatial information in the clustering process could filter out noise and other image artifacts such that yield a more accurate segmentation result. In this subsection, we modify the probability distribution of standard DA to incorporate the spatial contextual information by introducing a spatial penalty term. The penalty term acts as a regularizer and biases the solution toward piecewise-homogeneous labelling [1]. Such regularization is helpful in classifying images corrupted by noise. The spatial information is incorporated by considering the neighborhood pixels effect for the distance measure (dissimilarity) definition, that is

$$D_{jk} = d(x_j, v_k) + \frac{\alpha_j}{N_R} \sum_{r \in N_j} d(x_r, v_k) \quad (7.6)$$

where N_j stands for the set of neighbors falling into a (normally 3×3) window around x_j (not including x_j) and N_R is its cardinality. The spatial penalty term of image pixel j is regularized by the parameter α_j . Note unlike the standard spatially-constrained methods, where all the pixels are regularized by a constant parameter α . While they are regularized by dynamical values α_j in ASDA, which will be adaptively selected by the scaled variance of its neighbor pixels. The relative importance of the regularizing term α_j is proportional to the variance of a given pixel j against its neighbor pixels. We will discuss the adaptive selection of α_j later. Using the spatially-constrained distance measure (7.6), the probability

distribution of the proposed ASDA is given by

$$p_s(v_k|x_j) = \frac{p(v_k)e^{-\left(d(x_j,v_k)+\frac{\alpha_j}{N_R}\sum_{r \in N_j} d(x_r,v_k)\right)/T}}{\sum_{i=1}^c p(v_i)e^{-\left(d(x_j,v_i)+\frac{\alpha_j}{N_R}\sum_{r \in N_j} d(x_r,v_i)\right)/T}} \quad (7.7)$$

Accordingly, the free energy function becomes

$$F_s = -T \sum_{j=1}^N p(x_j) \log \sum_{i=1}^c p(v_i) e^{-\left(d(x_j,v_i)+\frac{\alpha_j}{N_R}\sum_{r \in N_j} d(x_r,v_i)\right)/T} \quad (7.8)$$

The cluster center v_k is obtained by minimizing (7.8) with respect to v_k as follows.

$$\frac{\partial(F_s)}{\partial(v_k)} = 0 \quad (7.9)$$

which leads to (detailed derivation is omitted)

$$\sum_{j=1}^N p(x_j) p_s(v_k|x_j) \left(x_j + \frac{\alpha_j}{N_R} \sum_{r \in N_j} x_r\right) = \sum_{j=1}^N (1 + \alpha_j) p(x_j) p_s(v_k|x_j) v_k$$

i.e.,

$$v_k = \frac{\sum_{j=1}^N p(x_j) p_s(v_k|x_j) \left(x_j + \frac{\alpha_j}{N_R} \sum_{r \in N_j} x_r\right)}{\sum_{j=1}^N (1 + \alpha_j) p(x_j) p_s(v_k|x_j)} \quad (7.10)$$

Alternatively updating (7.7) and (7.10) with the adaptive selection of α_j (discuss later) gives the proposed ASDA algorithm.

7.3.2 A Complexity Reduced Version of ASDA

As observed from (7.7) (7.10), the computation of the neighborhood terms takes much more time than the standard non-spatially-constrained algorithm. Here, we present a complexity reduced version of ASDA by equivalently replace $\sum_{r \in N_j} x_r$ by $\bar{x}_j$ and approximately replace $\frac{1}{N_R} \sum_{r \in N_j} d(x_r, v_i)$ by $d(\bar{x}_j, v_i)$, where $\bar{x}_j$ is the mean of neighborhood pixels at location j . This approximation keeps the spatial information such that is robust against noise as the original version does. Unlike the original formulation, $\bar{x}_j$ can be computed in advance, thus the clustering time can be significantly reduced. To obtain the complexity reduced version of ASDA, the only modification is to replace (7.7) by

$$p_r(v_k|x_j) = \frac{p(v_k)e^{\{-d(x_j, v_k) + \alpha_j d(\bar{x}_j, v_k)\}/T}}{\sum_{i=1}^c p(v_i)e^{\{-d(x_j, v_i) + \alpha_j d(\bar{x}_j, v_i)\}/T}} \quad (7.11)$$

and accordingly, (7.10) becomes

$$v_k = \frac{\sum_{j=1}^N p(x_j)p_r(v_k|x_j)(x_j + \alpha \bar{x}_j)}{\sum_{j=1}^N (1 + \alpha_j)p(x_j)p_r(v_k|x_j)} \quad (7.12)$$

Equations (7.11) and (7.12) is just the complexity reduced version of ASDA algorithm, which is used in our experiments.

7.3.3 Adaptive Selection of Spatial Penalty α_j

Proper selection of the α (dynamic α_j in our method or constant α in conventional methods) parameter is important for obtaining optimal or near-optimal image segmentation performance. When α is set to zero, the spatially-constrained algorithm is equivalent to the non-spatially-constrained algorithm, while α approaches too

larger, the blurring phenomenon may occur. Determination of an appropriate value of α is dependent on the image being clustered. Theoretically, the relative importance of the regularizing term (determined by α) is inversely proportional to the signal-to-noise (SNR) ratio of the observed image. Low SNR would require a higher value of α , and vice versa. However, in practical cases, SNR may not be known *a priori*, which makes the selection of α difficult, in result, few attempts have been presented to deal with it. To our knowledge, only [79] so far discussed the adaptive selection of α by using a cross-validation method, however, that method suffers from extreme computation cost, which makes it impractical for real world applications.

As observed from a real image, homogenous pixels are normally clustered together such that the intensity value x_j of a given image pixel j is normally similar to the values of its neighbor pixels, that means the variance of a given image pixel x_j against its neighbors is small. While if the given image pixel is contaminated, the variance becomes large. From this observation, we select the spatial strength α_j for each image pixel j by the scaled variance as follows,

$$\alpha_j = \text{var}(x_j)/\bar{v}ar \quad (7.13)$$

where $\text{var}(x_j)$ is the variance of x_j against its neighbors, which is defined by

$$\text{var}(x_j) = \frac{1}{N_R} \sum_{r \in N_j} d(x_r, x_j) \quad (7.14)$$

and $\bar{v}ar$ is the mean variance of all pixels, which is defined by

$$\bar{v}ar = \frac{1}{N} \sum_{j=1}^N \text{var}(x_j) \quad (7.15)$$

where N_j stands for the set of neighbors falling into a (normally 3×3) window

around x_j (not including x_j) and N_R is its cardinality. From (7.13), the value of α_j can be calculated directly from the given image itself, which is computationally practical in really applications. It will be seen from the experiments that (7.13) is a reasonable estimation of the spatial strength, i.e., the ASDA with estimated α_j by (7.13) can produce satisfied segmentation results for the given images.

7.3.4 Pseudo-code of ASDA for Image Segmentation

Based on above discussions, we here give a detailed pseudo-code of ASKA for image segmentation as follows.

Step 1) Set the number of clusters c , calculate the spatial strength α_j by (7.13), initialize temperature T_{ini} , minimum temperature $T_{min} = T_{ini}/1000$, $k = 1$, and $v_k = \frac{1}{N} \sum_{j=1}^N x_j$ with $p(v_k) = 1$.

Step 2) Alternatively update (7.11) (7.12) for $r = 1, 2, \dots, k$ until convergence is reached.

Step 3) If $T < T_{min}$ then stop, else let $T = \eta T$ ($0 < \eta < 1$ is cooling rate) and go to next step.

Step 4) If $k < c$, check condition for phase transition for $r = 1, 2, \dots, k$, if critical T is reached for cluster r , add a new cluster center by $v_{k+1} = v_r + \delta$ (δ is a small positive value) with $p(v_{k+1}) = p(v_r)/2$ and $p(v_r) = p(v_r)/2$, and let $k = k + 1$, go to next step.

Step 5) Go to step 2.

7.4 Experimental Results

To demonstrate and evaluate the performance of the proposed approach, we applied it to synthetic and magnetic resonance (MR) images. To provide quantitative comparison, all the images were also performed by the conventional Fuzzy C-means (FCM) [8], mass-constrained deterministic annealing (DA) [87], and spatially-

constrained FCM (SFCM) [1] [64] clustering algorithms. In all examples, we set $N_R = 9$ (a 3×3 window centered around each pixel), $m = 2$ (weighing exponent for FCM and SFCM), $\eta = 0.9$ (cooling rate for DA and ASDA), $\epsilon = 0.001$ (convergence threshold), the value of α_j for ASDA is determined by (7.13), while the value of α for SFCM is manually set *a priori*.

Figure 7.1: Segmentation results of different clustering algorithms on the synthetic image. DA has 120 misclassified pixels and FCM has 127 misclassified pixels. The minimum number of misclassified pixels of SFCM is 14 at $\alpha = 10$ and $\alpha = 11$. While for the proposed ASDA, the misclassified number is reduced to 7.

Table 7.1: The number of misclassified pixels of four methods on synthetic image.

	DA	FCM	Opt SFCM	Ave SFCM	ASDA
Misclassified Pixels	120	127	14	15.5	7

Opt SFCM means the best result of SFCM (at $\alpha = 10$ and $\alpha = 11$), Ave SFCM means the average result of SFCM (α within $[1 : 1 : 13]$, see Fig 7.1).

Figure 7.2: Segmentation results of different clustering algorithms on the MR image. The segmentation accuracy (SA) of DA and FCM are 90.52% and 90.34% respectively. The best SA of SFCM is 94.38% obtained at $\alpha = 2.5$, which is slightly better than the result of the proposed ASDA, which leads to 94.31% SA.

7.4.1 Synthetic Image

This example simply demonstrate the effectiveness of (7.13) for estimation of the spatial strength α_j . Fig 7.1a shows a synthetic image. This image contains a two-class pattern corrupted by hybrid 1% “Gaussian” and 1% “pepper and salt” (randomly occurring white and black pixels) noise. The spatial strength α_j calculated from (7.13) is shown in Fig 7.1b, where the value of α_j has been scaled from 0 to 255 for visualization. It is obvious that the estimated spatial strength reflects the distribution of noise to some extent: the clear pixels have lower (darker) spatial penalty values and the noisy pixels have higher (lighter) spatial penalty values. This is in accordance with the statement that the relative importance of the spatial penalty term is inversely proportional to the signal-to-noise (SNR) ratio of the observed image. This reveals the effectiveness of the scaled variance for deter-

mining the spatial strength. The segmented result of ASDA is shown in Fig 7.1c, which leads to 7 misclassified pixels. For comparison, the segmentation results of DA and FCM are shown in Fig 7.1d and Fig 7.1e respectively, where DA has 120 misclassified pixels and FCM leads to 127. The number of misclassified pixels of SFCM according to varying values of α is shown in Fig 7.1d, where the value of α is set *a priori*. It can be seen that the minimum number 14 is reached by SFCM at $\alpha = 10$ and $\alpha = 11$. From this example we can see that because of the noise effect presents in the data, the performances of both FCM and DA are seriously deteriorated. In contrast, spatially constrained methods ASDA and SFCM obtain much better results, with ASDA achieving a little better performance than SFCM. Table 7.1 summarizes the number of misclassified pixels of the four methods on the synthetic image.

Table 7.2: Segmentation Accuracy of CSF, GM and WM by applying four methods on the simulated MR image.

	DA	FCM	Opt SFCM	Ave SFCM	ASDA
CSF	91.74 %	92.68 %	93.55%	93.10%	93.06%
GM	84.29%	87.41 %	92.48%	92.41%	92.22%
WM	92.93%	90.59 %	95.46%	95.03%	95.47%
Overall	90.52%	90.34%	94.38%	94.13%	94.31%

Opt SFCM means the best result of SFCM (at $\alpha = 2.5$), Ave SFCM means the average result of SFCM (α within $[0.5 : 0.5 : 5]$, see Fig 7.2.).

7.4.2 Simulated MR Data

The simulated MR images are obtained from BrainWeb Simulated Brain Database [69]. Simulated brain data of varying noise are used to perform quantitative assessment of the proposed algorithm since ground truths are known for these data. Here in our experiments, we use a high-resolution T1-weighted phantom with slice thickness of $1mm$, 7% noise. The number of tissue classes in the segmentation was set to three, which corresponds to gray matter (GM), white matter (WM) and cerebrospinal fluid (CSF). Background and other tissues are ignored in the compu-

tation. Fig 7.2a shows the true partial volume model¹ of the slice 75. Accordingly, Fig 7.2c-e show the segmented results obtained by applying DA, FCM, and ASDA respectively. Fig 7.2f shows the results of SFCM with respect to the different given values of α . The DA and FCM have only 90.52% and 90.34% segmentation accuracy (SA) respectively, and the best result of SFCM leads to 94.38% SA at $\alpha = 2.5$, where SA is defined as the sum of the total number of pixels divided by the sum of number of correctly classified pixels [1] [77]. In contrast, the ASDA achieves a satisfied result with 94.31% SA, which is slightly worse than the best result of SFCM (94.38% SA at $\alpha = 2.5$). However, The SA obtained by ASDA is better than the average result of SFCM (94.13% SA when α within the range [0.5 : 0.5 : 5.0]). In addition, ASDA adaptively determine the values of α_j by (7.13) (the illustration is shown in Fig 7.2b), but SFCM requires α *a priori*, which may be not practical for real applications. The numerical results of these four methods on the simulated MR image are shown in Table 7.2 for quantitatively comparisons.

7.5 Conclusion and Discussion

In this chapter, we present an adaptive spatially-constrained DA algorithm for image segmentation. The key to the algorithm is a new dissimilarity measure, that adaptively takes into account the influence of the neighboring pixels on the center pixel in a 3×3 window, is incorporated into the formulation of the standard DA approach. The proposed algorithm offers several improved features compared to existing image segmentation methods. First, it is less sensitive to noise and has better convergence due to the incorporation of spatial information. Second, it is independent of the data initialization and has the ability to avoid many poor local optima due to the mass-constrained DA process. Lastly, the reasonable strength of the spatial constraints is auto-determined by the scaled variance of each image pixel against its neighbor pixels to improve the adaptiveness of the proposed algorithm.

¹The reference (ground truth) of the simulated MR images is available at BrainWeb [http : //www.bic.mni.mcgill.ca/brainweb/anatomic_normal.html](http://www.bic.mni.mcgill.ca/brainweb/anatomic_normal.html)

The superiority of the proposed algorithm has been demonstrated by both synthetic and MR images. It is obviously that the proposed ASDA needs more execution time than DA due to the computation of spatial information. Typically, the standard DA runs 15 – 25 times faster than the normal spatially-constrained ASDA, i.e., (7.7) and (7.10). Fortunately, the computation reduced version of ASDA, i.e., (7.11) and (7.12), significantly reduces the execution time, which typically runs 5 – 10 faster than normal ASDA algorithm and 2 – 5 slower than standard DA algorithm. This makes ASDA practical for real applications.

Chapter 8

A Weighted Support Vector Machine for Data Classification

8.1 Introduction

Support vector machine (SVM) was first introduced to solve the pattern classification and regression problems by Vapnik and his colleagues [13], [22]. It can be seen as a new training algorithm for the traditional polynomial, radial basis function (RBF) and multi-layer perceptron (MLP) classifiers by defining relevant kernel functions. The interesting property of SVM is that it is an approximate implementation of the structural risk minimization principle in statistical learning theory [106] [107] rather than the empirical risk minimization method. In recent years, SVM has drawn considerable attentions due to its high generalization ability for a wide range of applications and better performance than other traditional leaning machines [23], [72], [93].

In many practical engineering applications, the obtained training data is often contaminated by noise. Furthermore, some points in the training data set are misplaced far away from main body or even on the wrong side in input space. These *atypical* points are known as outliers. As noted in [13] [120], outliers tend

to become support vectors with large Lagrangian coefficients during the training process. Since the classifier obtained by SVM only depends on the support vectors. When there exist outliers or noise in training data set, the standard SVM training algorithm makes the decision boundary deviate severely from the optimal hyperplane, such that, the SVM is very sensitive to outliers and noise.

Some techniques have been found to tackle this problem for SVM in the literature of data classification¹. In [120] a central SVM (CSVM) is proposed by using the class centers to build the support vector machine. In [44] an adaptive margin SVM (AMSVM) is developed based on the utilization of adaptive margins for each training point. Motivated by these two methods, a robust SVM (RSVM) proposed in [98] [48] presents a general way to use the distance between the center of each class of the training data and the data point to form an adaptive margin. This approach, however, has the drawback that the penalty parameter λ is difficult to be tuned. Different from the methods using the *averaging* algorithm to reduce the effect of outliers, the proposed approach in [65] [49], resulting in a fuzzy SVM (FSVM), is to apply fuzzy memberships to the training data to relax the effect of the outliers and noise. However, the selection of membership function remains a problem for FSVM so far. Another interesting method, which is called support vector novelty detector (SVND) (or support vector data description (SVDD)) [92] [17] [102], has been shown effectiveness to detect outliers from the normal data points, however, it is well-known that SVND (or SVDD) is particular in the case of one-class classification problem. More recently, a hybrid technique is proposed in [60] involving symbolization of data to remove the noise (and outliers) and use of entropy minima to feature selection in conjunction with support vector machines to obtain a more robust classification algorithm.

In this chapter, we aim to improve the outlier sensitivity problem of standard support vector machine (SVM) by using a robust fuzzy clustering technique. The basic idea is to assign different weights to different data points by the robust

¹This chapter focuses on SVM classification, interested readers can refer to [20] [100] [101] for discussions of outlier problems for SVM regression.

fuzzy clustering technique such that the induced weighted support vector machine (WSVM) learns the decision surface according to the relative importance of data point in the training set. As a result, the effect of the less important points (outliers and noise) to the decision surface is reduced during the training process. We use the kernel-based possibilistic c-means (KPCM) algorithm to generate the weights for WSVM. The original PCM algorithm [57] [59] has been shown to have the satisfying ability of handling noise and outliers. In addition, the induced partition assigns each data point a relative value according to the compatibility of the points with the class prototypes: important data points have high values but outliers and noise have low values. In this chapter, we extend the original PCM algorithm from input space into a kernel-induced feature space by using kernel methods and develop the KPCM algorithm whose partitioned relative values are served as the weights for the proposed WSVM training.

The rest of this chapter is organized as follows. Section 2 briefly reviews the basic theory and outlier sensitivity problem of standard SVM. In section 3, we derive the formulation of the proposed WSVM, and present the weights generating algorithm. The detailed experimental results are reported in section 4. In section 5, we present some remarks and discussions on algorithms, experiments and further investigations. Finally, the conclusion is given in section 6.

8.2 Standard SVM

8.2.1 Formulation of SVM

The main idea behind SVM technique is to derive a unique separating hyperplane (i.e., the optimal margin hyperplane) that maximizes the margin between the two classes. Given l training data points

$$\{(x_i, y_i)\}_{i=1}^l, \quad x_i \in R^N, y_i \in \{-1, 1\}$$

the support vector technique requires the solution of the following optimization problem:

$$\text{Minimize } \Phi(w) = \frac{1}{2}w^T w + C \sum_{i=1}^l \xi_i \quad (8.1)$$

subject to

$$\begin{aligned} y_i(\langle w, \phi(x_i) \rangle + b) &\geq 1 - \xi_i, \quad i = 1, \dots, l \\ \xi_i &\geq 0, \quad i = 1, \dots, l \end{aligned} \quad (8.2)$$

where the training vectors x_i are mapped into a higher dimensional space by the function ϕ . C is a user-specified positive parameter, which controls the tradeoff between classification violation and margin maximization. The existing common method to solve (8.1) is through its dual, a finite quadratic programming problem:

$$\begin{aligned} W(\alpha) &= \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j \langle \phi(x_i), \phi(x_j) \rangle \\ &= \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j K(x_i, x_j) \end{aligned} \quad (8.3)$$

subject to

$$\sum_{i=1}^l y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, l \quad (8.4)$$

where α is the Lagrangian parameter. Note the kernel trick $K(x, y) = \langle \phi(x), \phi(y) \rangle$ is used in the last equality in (8.3). The Kuhn-Tucker conditions of SVM are defined by

$$\alpha_i [y_i(\langle w, \phi(x_i) \rangle + b) - 1 + \xi_i] = 0, \quad i = 1, \dots, l$$

$$(C - \alpha_i)\xi_i = 0, \quad i = 1, \dots, l \quad (8.5)$$

The point x_i with the corresponding $\alpha_i > 0$ is called a support vector. The optimal value of weight vector w_0 is obtained by $w_0 = \sum_{i=1}^l \alpha_i y_i \phi(x_i) = \sum_{i=1}^{l_s} \alpha_i y_i \phi(x_i)$, where l_s is the number of support vectors. The optimal value of bias b_0 can be computed from the Kuhn-Tucker conditions (8.5). Once the optimal pair (w_0, b_0) is determined, the decision function is obtained by

$$f(x) = \text{sign}(\langle w_0, \phi(x) \rangle + b_0) = \text{sign}\left(\sum_{i=1}^{l_s} \alpha_i y_i K(x, x_i) + b_0\right) \quad (8.6)$$

Figure 8.1: Outlier sensitivity problem of standard SVM (a). Weights generated by KPCM for class 1 (b).

8.2.2 Outlier Sensitivity Problem of SVM

It can be observed from equation (8.6) that the decision boundary is decided by only the support vectors: the points with $\alpha_i > 0$. In many practical engineering applications, the training data is often affected by noise or outliers, which tend to

become the support vectors during the SVM training process [13] [120]. In this case, the decision boundary will be deviated from the optimal boundary severely. This phenomenon is well-known as the outlier sensitivity problem of standard SVM algorithm.

Fig 8.1a gives an illustration of the outlier sensitivity problem of SVM. When the training data is clear (no outlier exists), SVM can find an optimal hyperplane (denoted by solid line) with maximum margin to separate the two classes. If there exists two outliers (located on the wrong side and denoted by number 1 and 2), for example, in class 1, the decision hyperplane (denoted by dotted line) is severely deviated from the optimal one due to the over-fitting problem. Therefore, the training process of standard SVM is very sensitive to outliers. This chapter aims to improve the SVM training, especially focuses on the the over-fitting problem with outliers that make the two classes nonseparable.

8.3 The Proposed Weighted SVM

The basic idea of weighted support vector machine (WSVM) is to assign each data point a different weight according to its relative importance in the class such that different data point has different contribution to the learning of the decision surface. Suppose the weights are given, then the training data set becomes

$$\{(x_i, y_i, W_i)\}_{i=1}^l, \quad x_i \in R^N, y_i \in \{-1, 1\}, W_i \in R$$

where the scalar $0 \leq W_i \leq 1$ is the weight assigned to data point x_i by kernel-based possible c-means (KPCM) algorithm, which will be discussed later in this section.

8.3.1 Formulation of WSVM

Starting with the construction of a cost function, the WSVM wants to maximize the margin of separation and minimize the classification error such that a good

generalization ability can be achieved. Unlike the penalty term in standard SVM, where the value of C is fixed and all training data points are equally treated during the training process, WSVM weighs the penalty term in order to reduce the effect of less important data points (such as outliers and noise). The constrained optimization problem is formulated as

$$\text{Minimize } \Phi(w) = \frac{1}{2}w^T w + C \sum_{i=1}^l W_i \xi_i \quad (8.7)$$

subject to

$$\begin{aligned} y_i(\langle w, \phi(x_i) \rangle + b) &\geq 1 - \xi_i, \quad i = 1, \dots, l \\ \xi_i &\geq 0, \quad i = 1, \dots, l \end{aligned} \quad (8.8)$$

Note that we assign the weight W_i to the data point x_i in above formulation. Accordingly, the dual formulation becomes

$$W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (8.9)$$

subject to

$$\sum_{i=1}^l y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C W_i, \quad i = 1, \dots, l \quad (8.10)$$

and the Kuhn-Tucker conditions of WSVM become

$$\begin{aligned} \alpha_i [y_i(\langle w, \phi(x_i) \rangle + b) - 1 + \xi_i] &= 0, \quad i = 1, \dots, l \\ (C W_i - \alpha_i) \xi_i &= 0, \quad i = 1, \dots, l \end{aligned} \quad (8.11)$$

It is clear that the only difference between SVM and the proposed WSVM is the upper bounds of Lagrangian multipliers α_i in the dual problem. In SVM, the upper bounds of α_i are bounded by a constant C while they are bounded by dynamical boundaries that are weight values CW_i in WSVM. Suppose a data point x_i is an outlier that is located on the wrong side in input space. Then the outlier may become a support vector and the upper bound of Lagrangian multiplier α_i is reached. For WSVM training, we have $\alpha_i = CW_i$. The weight W_i assigned by KPCM algorithm to this support vector (outlier) is much smaller than other support vectors (non-outliers), such that the effect of this support vector to the training procedure will be reduced and the decision boundary will be less distorted by the outlier. While for the standard SVM training, the value of α_i is equal to C , which will affect the decision surface more significantly than WSVM does. We have to point out that in the cases of $C \rightarrow \infty$ and $C \rightarrow 0$, we have $CW_i = C$, then the performance of the proposed WSVM becomes identical to the standard SVM. That means the effect of the proposed WSVM for reducing outlier effects can only be observed for the cases where the value of C is finitely positive.

8.3.2 Weights Generating Algorithm

The weights used in WSVM are generated by kernel-based possibilistic c-means (KPCM) algorithm. Possibilistic c-means (PCM) algorithm, which was first proposed in [57] and further explored in [29] [59] to overcome the relative membership problem of fuzzy c-means (FCM) algorithm, has been shown to have the satisfying ability of handling noise and outliers. In addition, the induced partition assigns each data point a relative value (possibility) according to the compatibility of the points with the class prototypes: important data points have high values but outliers and noise have low values. Therefore, we can choose the partitioned relative values as the weight values in the proposed WSVM training process. However, the WSVM training procedure is processed in kernel-induced feature space by using different kernel functions. The weights generated by PCM in input space may

be not suitable for WSVM training. To make use of the PCM algorithm for the proposed WSVM, we have to extend the original PCM into feature space by using kernel methods and develop the KPCM algorithm to generate weights in feature space instead of in input space.

Similar to the objective function of PCM (which is formulated by modifying the objective function of standard FCM, details referring to [29] [57] or discussions in chapter 2), the objective function of KPCM is formulated as follows

$$J = \sum_{k=1}^c \sum_{j=1}^l (p_{kj})^m D_k(x_j) + \sum_{k=1}^c \eta_k \sum_{j=1}^l (1 - p_{kj})^m \quad (8.12)$$

where $D_k(x_j)$ is the squared Euclidian distance between the transformed point $\phi(x_j)$ and cluster prototype $\phi_k(x)$ in feature space (please refer to (5.2) and (5.3) in chapter 5 for details), m ($1 < m < \infty$) determines the fuzziness, and p_{kj} is the possibility of transformed point $\phi(x_j)$ in the k th cluster, and there are no constraint on the possibility p_{kj} other than the requirement that they should be in $[0, 1]$. Note the second term forces p_{kj} to be as large as possible, thus avoiding the trivial solution of standard FCM, more details can be found in [29].

Assuming that the parameter η_k is specified, the minimization of (8.12) gives the updating equations for possibility p_{kj} and cluster prototype $\phi_k(x)$ as follows:

$$p_{kj} = \frac{1}{1 + \left[\frac{D_k(x_j)}{\eta_k} \right]^{1/(m-1)}} \quad (8.13)$$

$$\phi_k(x) = \sum_{j=1}^l \gamma_{kj} \phi(x_j) = \frac{\sum_{j=1}^l (p_{kj})^m \phi(x_j)}{\sum_{j=1}^l (p_{kj})^m} \quad (8.14)$$

Note the explicit expression of γ_{kj} is given in (8.14). The positive parameter η_k is

a critical parameter which should be determined in the above algorithm. Referring to the recommendation in [57], η_k can be obtained from the average possibilistic intra-cluster distance of cluster k as

$$\eta_k = K \frac{\sum_{j=1}^l (p_{kj})^m D_k(x_j)}{\sum_{j=1}^l (p_{kj})^m} \quad (8.15)$$

The KPCM algorithm is implemented by iteratively updating (8.13) and (8.14) until the stopping criterion is satisfied. The partition results are not very sensitive to the range of the values of K [57]. A typical value is the constant 1, as used in the experimental results in this chapter.

We can now use the above KPCM algorithm to generate weights for the training of WSVM algorithm. Let the set of possibility values of positive class ($y_i = 1$) and negative class ($y_i = -1$) be denoted by p_i^+ and p_i^- after partitioning respectively. Then the weights of WSVM are finally obtained by setting $W_i^+ = p_i^+$ and $W_i^- = p_i^-$ (Here W^+ denote the weights of points in positive class and W^- denote the weights of points in negative class). After partitioning, the weights of outliers and noise will be very small or even nearly to zero, such that the effect of outliers and noise to decision hyperplane is significantly reduced. Fig 8.1b shows the weights of a given class (class 1 from Fig 8.1a, where two outliers exist) generated by KPCM. As we can see, the weights of training data points in the main body are much larger (nearly one) and the weights of the outliers are much smaller (nearly zero). Accordingly, the effect of the outliers to the decision surface will be significantly reduced, similar observation can be obtained in the section of experimental results.

Figure 8.2: The performance comparison between SVM and WSVM while outliers existing.

8.4 Experimental Results

The effectiveness of the proposed WSVM algorithm for classification problems is tested on some artificial and benchmark data sets. The WSVM program is developed based on the standard SVM training program [42].

8.4.1 Artificial Data Set

We generate a two class data set in R^2 (represented by crosses and pluses), where positive class has two outliers (marked by arrows nearby) located on the wrong side. The training results of SVM and WSVM using linear kernel function with different values of C are shown in Fig 8.2. Where “x”, and “+” denote the positive class and negative class respectively, the data points with circle symbols surrounding

them are support vectors, the distance between the two dotted lines is the margin of separation, the line between the two dotted lines is the optimal hyperplane i.e., the decision surface. As we can see, the decision surface obtained by standard SVM is significantly distorted by outliers (as shown in (a1), (a2) and (a3)), which will lead to poor generalization performance, i.e., the test error is large. In contrast, the proposed WSVM has much better performance (as shown in (b1), (b2) and (b3)), the obtained decision surface is almost not affected by outliers compared to SVM. Actually, the decision boundary obtained by WSVM with $C = 100$ is almost identical to the solid line in Fig 8.1a, which is the optimal hyperplane assuming the outliers are removed. Thus, the proposed method successfully reduces the effect of outliers and accordingly yields good generalization performance than standard SVM does. In addition, it can be obtained from Fig 8.2 that the proposed WSVM is less sensitive to the value of the penalty parameter C than standard SVM does for the data set. Note if the outliers are removed, SVM and WSVM will have similar performances.

Table 8.1: Comparison of test error between WSVM and standard SVM for “Twonorm” benchmark data set according to different number of mislabelled data points using linear kernel function.

	$N_o = 0$	$N_o = 10$	$N_o = 20$	$N_o = 30$	$N_o = 40$	$N_o = 50$	$N_o \geq 90$
SVM	2.50%	2.69%	3.06%	3.51%	4.14%	4.90%	$\geq 28.7\%$
WSVM	2.52%	2.54%	2.67%	2.83%	3.01%	3.30%	$\geq 15.1\%$

8.4.2 Benchmark Data Set

The benchmark data set used to test the proposed method is “Twonorm” from IDA benchmark repository [50], which has been used to evaluate kernel and boosting methods in recent years [72] [81]. This data set is originally from DELVE benchmark repository [31]. It is a 20 dimensional, 2 class classification example. Each class is drawn from a multivariate normal distribution with unit variance. Class 1 has mean $(a, a, \dots, a)$ while Class 2 has mean $(-a, -a, \dots, -a)$. Where $a = 2/\sqrt{20}$. The number of data points is 7400 (400 training data points and 7000 testing data

points). Here we use linear kernel to classify this benchmark data set. The value of C is determined through cross validation. The classification error of standard SVM and the proposed WSVM are 2.50% and 2.52% (see the first column of numerical results in Table 8.1) respectively. We have to point out that WSVM (and other outlier reduction methods mentioned in the introduction section) is originally proposed to solve the outlier sensitivity problem of standard SVM when outliers exist in the training data. However, the benchmark data may not contain outliers, in this case, the performance of WSVM will be similar to that of standard SVM.

To show the effectiveness of WSVM algorithm to reduce the effect of outliers, we gradually *corrupt* the “Twonorm” benchmark data set by mislabelling some data points (the number is denoted as N_o): we select N_o data points, for example, from the positive (+1) class, then artificially set the label of these points to -1 . These mislabelled data points can be regarded as *outliers*. Table 8.1 compares the percentage test error of WSVM to standard SVM according to the different number of mislabelled data points. From the table, we can see the performance of standard SVM is severely distorted by the mislabelled data points: the test error of standard SVM increases obviously from 2.50% to 4.90% as the number of mislabelled data points N_o increases from 0 to 50. While the proposed WSVM appears to be more robust and has better performance: the test error increases only from 2.52% to 3.30%. This indicates that the proposed method successfully reduces the effect of outliers (mislabelled data points in this case) and yields higher classification rate than standard SVM does when the training data set is contaminated by outliers (mislabelled data points in this case). Note when N_o is beyond a large value (such as 90), the test error of both SVM and WSVM classifiers become undesirable (see the last column of numerical results in Table 8.1).

8.5 Remarks and Discussions

On Algorithms

1) For KPCM algorithm, we define convergence to be reached when the maximum change in the possibility values p_{kj} over all points between consecutive iterations is less than a given threshold value. In practice, we used a threshold value of 0.001. Please note we can not use the cluster center $\phi_k(x)$ to tell whether or not the convergence is reached. This is because we do not know the explicit expression of the nonlinear mapping ϕ , such that we can not directly compute the cluster center by using (8.14). In practice, the cluster center is implicitly updated by updating the distance in feature space, i.e., using (5.3).

2) KPCM algorithm requires an initial estimation of possibility values p_{kj} . Proper selection will generally improve accuracy and convergence of the algorithm. We use standard FCM to generate initial possibility values for KPCM as recommended in [57]. This method seems to be suitable and we find the KPCM algorithm is always convergent in several iterations. We also find that KPCM algorithm generally converges faster than the standard PCM algorithm does. This observation may be due to the relative simpler data structure in feature space than that in input space provided that suitable kernel function and kernel parameter(s) are used, as mentioned in [39].

3) We use KPCM to partition positive class and negative class respectively by setting the cluster number $c = 1$ for each class. The partition time is quite acceptable and can be neglectable compared to the training time of SVM algorithm such that the total computation time (including partitioning and training) of WSVM is similar to that of standard SVM. The computation complexity of WSVM may be further reduced through a simple pruning method when the training data set is contaminated by outliers, as discussed later in this section.

Table 8.2: Comparison of test error between WSVM and standard SVM for “Thyroid” benchmark data set according to different number of mislabelled data points using RBF kernel function.

	$N_o = 0$	$N_o = 5$	$N_o = 10$	$N_o = 15$	$N_o = 20$	$N_o = 25$
SVM	2.67%	6.67%	10.7%	14.7%	20.0%	24.1%
WSVM	2.67%	2.67%	2.67%	4.00%	5.33%	14.7%

On Experiments

1) In our experiments, we only compare the performances between SVM and WSVM by using linear kernel function for “Twonorm” benchmark data set. In fact, the proposed method is universal and the similar observations can be obtained by using any kernel function for most benchmark data sets if we *corrupt* (artificially adding or mislabelling some data points) the original data sets. For instance, Table 8.2 shows the percentage test error between WSVM and standard SVM by using RBF kernel function for “Thyroid” data set from IDA benchmark repository [50] (originally taken from UCI benchmark repository [105]) according to the different number of mislabelled data points. Note that the advantage of the proposed WSVM over standard SVM shown in Table 8.2 is much more obvious: the test error of WSVM retains unchanged (2.67%) if the number of mislabelled data points keeps small ($N_o \leq 10$).

2) According to equation (8.6), the decision surface is determined by all the support vectors. If the number of support vectors increases but the number of outliers (which become support vectors after training process) remain unchanged, then the contribution of the outliers to the decision surface will accordingly decreases. In Fig 8.2a1 and Fig 8.2b1, the proportion of the number of outliers to total number of support vectors is quite small such that the reduction of outliers can not significantly affect the learning of the decision surface. This just explains why the decision surface of WSVM (shown in Fig 8.2b1) is only a bit better than that of SVM (shown in Fig 8.2a1). Please note that the sparse property is sacrificed in this case.

3) In our experiments, we set K (the parameter of KPCM algorithm, see (8.15)) a constant with the value 1 for all data sets. This indicates tuning of K could give even better performance for WSVM though the improvement is not significant. As a general rule, the value of K should be decreased when the number of outliers in training data increases.

On Further Investigations

1) As mentioned above, the WSVM (or other outlier reduction techniques mentioned in the introduction section) is originally proposed to improve the outlier sensitivity problem of standard SVM when outliers exist in the training data, our future work will focus on the contaminated data set to further investigate the ability of WSVM in reducing the effect of outliers.

2) Choosing C effectively is still an opening problem. Our further research will investigate how the WSVM formulation affects the ability to choose the penalty parameter C . Unlike the penalty term in standard SVM, where the value of C is fixed and all training data points are equally treated during the training process, WSVM weighs the penalty term in order to reduce the effect of outliers if they exist in the training set. Though the preliminary experiments reveal that the proposed WSVM is less sensitive to the value of C than the standard SVM for the contaminated training data set, further fundamental research is needed in this interesting direction.

3) The training and testing complexities of WSVM can be greatly reduced through a simple pruning method when the training data set is contaminated by outliers. While in pruning methods for classical MLP the procedure requires the computation of a Hessian matrix or its inverse, the pruning in WSVM can be done based upon the weighted data points themselves. Less meaningful data points as indicated by their (smaller) weights are removed and the WSVM is trained on the basis of the remaining points, this procedure may significantly reduce both the training time and the effect of outliers. The testing time can similarly be reduced

by removing the less meaningful support vectors with smaller weights. Though the pruning procedure may induce the loss of training and testing accuracy, this topic is desirable to be further investigated.

8.6 Conclusion

In this chapter, a weighted support vector machine (WSVM) is proposed to deal with the outlier sensitivity problem in traditional support vector machine (SVM) for two-class data classification. The robust fuzzy clustering algorithm, i.e., possibilistic c-means (PCM), is extended into kernel-induced feature space to generate different weight values for main training data points and outliers according to their relative importance in the training set. Experimental results show that the proposed WSVM can reduce the effect of outliers and yield higher classification rate than standard SVM does when the training data set is contaminated by outliers.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

In this thesis, we have presented several robust clustering algorithms to overcome the major drawbacks associated with traditional clustering methods. The first clustering approach, robust deterministic annealing (RDA) algorithm, achieves the robustness by incorporating the basic idea of noise clustering into the deterministic annealing algorithm. This implementation is simple but efficient: its insensitivity to initial data configuration and cluster volumes is assured by the deterministic annealing process; its robustness against noise and outliers is obtained by partitioning the noisy data points into an extra noise cluster. Furthermore, a VC-bound induced cluster validity has been presented to determine the optimal cluster number. As a result, the proposed RDA exhibits several robust clustering characteristics: 1) robust to the initialization; 2) robust to cluster volumes; and 3) robust to noise and outliers. The experimental results on artificial and real data have shown that the proposed clustering approach provides a performance that is superior to that of traditional noise clustering and deterministic annealing algorithms: compared to the former, RDA is independent of initialization of cluster prototypes and has better performance for unbalanced data set; compared to the latter, RDA is robust against noise and outliers. By using the Euclidian distance, the RDA is currently

only suitable for partitioning clusters that are hyper-spherical in shape. This is a limitation of the proposed clustering method.

The second clustering approach, information fuzzy c-spherical shells (IFCSS) algorithm, is specifically proposed to improve the robustness of existing traditional spherical shells clustering methods. Compared to the traditional spherical shells clustering algorithms, two robust clustering characteristics have been achieved in the proposed IFCSS algorithm: 1) the robustness against noisy points is obtained by the maximization of mutual information; and 2) the optimal cluster number is auto-determined by the VC-bound induced cluster validity. The superiority of the proposed clustering method has been tested by experimental results: the proposed algorithm can correctly partition the spherical shells, remove the noisy data points, reveal the underlying cluster number, and find the parameters of the shell clusters with good accuracy. The current limitations of the proposed IFCSS algorithm could be: 1) the sensitivity to data initialization is not considered, we assume that at least one initialization is good by using multiple random initializations in the experimental results; 2) the selection of fuzzier m remains undiscussed in the thesis, we find out from the experiments that the value of $m = 1.5$ works well for the IFCSS algorithm.

The third clustering approach, kernel-based deterministic annealing (KDA) algorithm, is developed to discover some complicated (e.g., linearly nonseparable) data structures which can not be revealed by traditional clustering methods in the standard Euclidean space. we have generalized a nonlinear version of the standard deterministic annealing algorithm by means of a Gaussian kernel-induced distance measure. A simple but effective method to select a reasonable value for the Gaussian kernel parameter (width), whose selection is so far remains unsolved in the literature, has been worked out in the thesis. The results on artificial and real data showed that: 1) the scaled inverse of the input data covariance is a reasonable estimation of the Gaussian kernel parameter (width); 2) the proposed KDA clustering method is capable of clustering a variety of data structures with different

Table 9.1: A summary of all the proposed robust clustering algorithms.

	The proposed algorithm
	RDA
Specific Use	Robust clustering of spherical and linearly-separable data
Advantages	1)Insensitivity to data initialization; 2)Insensitivity to cluster volumes; 3)Robustness against noise and outliers; 4)Auto-determination of cluster number
Disadvantages	1)Need to specify an additional parameter; 2)Only suitable for spherical and linearly-separable data
Comparisons	Comparison results showed the performance of RDA was better than that of noise clustering and deterministic annealing algorithms
Contributions	1)Apply the basic idea of noise clustering to deterministic annealing; 2)VC-bound cluster validity for optimal cluster number selection
	IFCSS
Specific Use	Robust clustering of spherical shells
Advantages	1)Robustness against noise and outliers; 2)Auto-determination of cluster number
Disadvantages	1)Sensitivity to data initialization and cluster volumes; 2)Need to specify an additional parameter
Experiments	The superiority of the proposed IFCSS algorithm was supported by simulation results
Contributions	1)Mutual information maximization to phase out noise and outliers; 2)VC-bound cluster validity index to determine the optimal cluster number
	KDA
Specific Use	Robust clustering of a variety of data structures
Advantages	1)Insensitivity to data initialization; 2)Insensitivity to cluster volumes; 3)Capacity of revealing complicated data structures
Disadvantages	Expensive computation cost
Comparisons	Comparison results showed the performance of KDA was better than that of deterministic annealing and kernel-based K-means algorithms
Contributions	1)Extend the standard deterministic annealing into the Mercer kernel-induced feature space; 2)Work out a simple but efficient method to select a reasonable value for Gaussian kernel parameter (width)

Tabel 9.1 Continues...	
	ASDA
Specific Use	Robust image segmentation
Advantages	1)Robustness against noise; 2)Adaptive spatial strength selection
Disadvantages	Need more execution time than traditional image segmentation methods
Comparisons	Comparison results showed the performance of ASDA was better than that of deterministic annealing, fuzzy c-means, and spatially-constrained fuzzy c-means algorithms
Contributions	Adaptive selection of the spatial penalty parameter
	KPCM (used for WSVM)
Specific Use	Generate weights for the proposed WSVM
Advantages	Assign relative high values for important data points but low values for outliers
Disadvantages	Need to specify an additional parameter
Comparisons	By using the weights from KPCM, the proposed WSVM reduces the effect of outliers and yields higher classification rate than standard support vector machine does
Contributions	Improve the robustness of support vector machine

size, shape and distribution; 3) the proposed KDA clustering method provides a performance that is superior to that of traditional clustering algorithms. Like most of the kernel-based learning algorithms, the proposed KDA clustering algorithm suffers from the expensive computation time for large data samples. To speed up the kernel-based learning algorithms is currently an open problem in the literature of kernel methods.

Finally, robust clustering methods have been developed for image segmentation and pattern classification. First, we have proposed an adaptive spatially-constrained deterministic annealing (ASDA), that can perform unsupervised clustering for robust image segmentation. The key to the algorithm is a new dissimilarity measure that takes into account the influence of the neighboring pixels on the center pixel. The strength of spatial constraint for each given image pixel is auto-determined by the scaled variance of its neighbor pixels in a 3×3 window. The adaptiveness and robustness of the presented method for image segmentation have been

demonstrated on synthetic and MR images. Second, a robust clustering algorithm, kernel-based possibilistic c-means (KPCM) algorithm, has been developed to generate relative high values for important data points but low values for outliers, such that different data point can make different contribution to the learning of the decision surfaces according to its relative importance in the training process of the proposed weighted support vector machine (WSVM) algorithm. Experimental results has indicated that the proposed method reduces the effect of outliers and yields higher classification rate than standard support vector machine does when outliers exist in the training data set.

Table 9.1 provides a summary of all the proposed robust clustering algorithms with their specific use, benefits, disadvantages, and other related issues.

9.2 Future Work

Further research work should be carried out to make the robust learning algorithms more acceptable to data clustering and its extensions. Possible directions of future work that emerge from this thesis are briefly discussed in this subsection.

The adaptive selection of the additional parameter(s) for the proposed robust clustering methods.

The robust clustering methods, on one hand, achieve the robustness against noisy points; on the other hand, introduce additional parameter(s), such as the parameter K in RDA algorithm and the parameter λ in IFCSS algorithm, to control the degree of robustness. As reported in the former chapters, the estimation of the cluster number is hardly sensitive to the values of the additional parameter(s). However, the choice of the additional parameter(s) may have a great impact on what is classified into good clusters and what is classified into noisy points. Take the parameter K in RDA as an example: if the K is chosen to be very small, then most of the points will get classified as noisy points; while if K is very large, then most of the points will be classified into good clusters other than the noisy

points. Unfortunately, how to auto-select an appropriate value of K (or λ) for a given data remains a difficulty. Due to the fact that there is no prior information of the underlying data structure for data clustering, a common method in the literature is to select the value based on the statistic of the distance distributions. If the clusters possess similar distributions, a constant value of K (or λ) normally works well, as demonstrated by the experimental results in the thesis. However, If the clusters have quite different distributions, a constant value may fail to phase out noisy points for all clusters. Such mixture distributions may require variable parameter techniques [97].

Further investigation of the optimal cluster number selection based on model selection criterion.

The VC-bound (VB) induced cluster validity index, which has been shown to be effective in producing a good cluster number estimation for RDA and IFCSS algorithms, is basically a model selection criterion in statistical learning theory [107]. As addressed in the thesis, the VB index is implemented based on the structural risk minimization (SRM) principle, an induction principle for controlling the generalization ability of learning machines that use small size samples. The sample size l is considered to be small if the l/h_c (ratio of the number of samples to the VC dimension of the set of functions) is small, say $l/h_c < 20$ [107]. If this condition is not satisfied, for an extreme example, $l/h_c \gg 20$, then the second term in the VC-bound (see equation (6.3)) will make no contribution to the real risk, such that the VB index will be determined completely by the empirical risk. Problem(s) may occur in this case since the empirical risk (the first term in the VC-bound) basically presents a monotonic tendency of the cluster number, which induces the monotonic decrease of the VB index with the increase of cluster number. This problem may be solved by normalizing the second term in VC-bound into a suitable range, e.g., $[0, 1]$, or using other bounds rather than VC-bound in the literature of statistical learning theory. On the other hand, if $l/h_c < 20$ is satisfied, to avoid the domination of the second term in the contribution to VC-

bound, the cluster number for a given data should not exceed some pre-specified maximum number. In other words, the maximum cluster number c_{max} may be selected by $l/(c_{max} \times n) \rightarrow 20$. We leave this issue to our further research work.

Several further investigations on Mercer kernel-based data clustering algorithms.

It is well known that the kernel-based learning algorithms suffer from the expensive computation time and storage memory for large data samples. For example, a prohibitive amount of memory is required to store the kernel matrix $l \times l$ if l is a large number. In the past years, several efficient methods, especially the decomposition algorithms [18] [52] [80], have been proposed to tackle this problem for the well-known support vector machines (SVMs). The key idea of decomposition is to freeze all but a small number of optimization variables, and to solve a sequence of constant-size problems. This method breaks large quadratic programming (QP) problem into a series of smaller QP subproblems such that significantly reduces the storage and computation. But so far, seldom attempts have been proposed for the unsupervised learning in feature space. The development of a speed-up method for data clustering in feature space, especially for the proposed KDA (and KPCM) algorithm, is desirable for further investigation. For the proposed KDA (and KPCM) algorithm, the bottleneck of computation is to calculate the kernel-induced distance, see (5.5). We need to put out that the third term of the right-hand in these equations is to be calculated only once for each cluster during each iteration. And for Gaussian kernel function, the first term of the right-hand is a constant for all points, such that it can be omitted from the calculation. Another problem related to kernel-based clustering algorithms is the estimation of the optimal cluster number. In our preliminary study, we have used the contour plot of kernel matrix to solve this issue. This method only provides a visual but not numerical solution. The latter one is more desirable in real applications. Our current effort is to seek a solution for this issue by using some suitable model selection criteria in the statistical learning theory.

Appendix A: Abbreviations' List

Abbr.	Full Name	Remarks
AFCS	Adaptive Fuzzy C-Shells	
AMSVM	Adaptive Margin Support Vector Machine	
ASDA	Adaptive Spatially-Constrained Deterministic Annealing	Proposed in chapter 7
CSF	Cerebrospinal Fluid	
CSVM	Central Support Vector Machine	
DA	Deterministic Annealing	Reviewed in section 2.2.2
FCE	Fuzzy C-Ellipses	
FCM	Fuzzy C-Means	Reviewed in section 2.2.1
FCQS	Fuzzy C-Quadric Shells	
FCR	Fuzzy C-Rings	Reviewed in section 2.4.3
FCRS	Fuzzy C-Rectangular Shells	
FCS	Fuzzy C-Shells	Reviewed in section 2.4.1
FCSS	Fuzzy C-Spherical Shells	Reviewed in section 2.4.2
FHV	Fuzzy Hyper-Volume	Discussed in section 2.3.2
FS	Fukuyama and Sugeno	Discussed in section 2.3.2
GM	Gray Matter	
IFCSS	Information Fuzzy C-Spherical Shells	Proposed in chapter 4
KDA	Kernel-Based Deterministic Annealing	Proposed in chapter 5
KFD	Kernel Fisher Discriminant	
KM	K-Means	Reviewed in section 2.5.3
KKM	Kernel-Based K-Means	Reviewed in section 2.5.3
KPCA	Kernel Principle Component Analysis	
KPCM	Kernel-Based Possibilistic C-Means	Proposed in section 8.3.2
MI	Mutual Information	
MLP	Multi-Layer Perceptron	
MR	Magnetic Resonance	
NC	Noise Clustering	Reviewed in section 2.3.1
NISP	Norm-Induced Shell Prototypes	

Continues...

Abbr.	Full Name	Remarks
PC	Partition Coefficient	Discussed in section 2.3.2
PCM	Possibilistic C-Means	Reviewed in section 2.3.1
PD	Partition Density	Discussed in section 2.3.2
PE	Partition Entropy	Discussed in section 2.3.2
RBF	Radial Basis Function	
RDA	Robust Deterministic Annealing	Proposed in chapter 3
RSVM	Robust Support Vector Machine	
SFCM	Spatially-Constrained Fuzzy C-Means	Reviewed in section 7.2
SNR	Signal-to-Noise	
SRM	Structural Risk Minimization	Discussed in section 6.2.1
SVDD	Support Vector Data Description	
SVM(s)	Support Vector Machine(s)	Reviewed in section 8.2
SVND	Support Vector Novelty Detector	
VB	Vapnik-Chervonenkis Bound	Proposed in chapter 6
VC	Vapnik-Chervonenkis	Discussed in section 6.2.1
WM	White Matter	
WSVM	Weighted Support Vector Machine	Proposed in chapter 8
XB	Xie and Beni	Discussed in section 2.3.2

Appendix B: The Proof of Fano's Inequality

In accordance with our previous discussions, let a discrete channel be described by a forward probability transition matrix $Q = [\bar{u}_{k|j}]$; and assume the input of the channel is $X = [x_j]$ with probability distributions $U = [u_j]$ ($j = 1, 2, \dots, l$) and the output of the channel is $W = [w_k]$ with probability distributions $[u_k = \sum_j u_j \bar{u}_{k|j}]$ ($k = 1, 2, \dots, c$). Then the input uncertainty can be defined by Shannon's entropy as

$$H(X) = - \sum_{j=1}^l u_j \log u_j \quad (9.1)$$

After having observed a feature vector W , the uncertainty becomes the conditional entropy

$$H(X|W) = - \sum_{j=1}^l \sum_{k=1}^c u_{kj} \log u_{j|k} \quad (9.2)$$

where u_{kj} is the joint probability given by $u_{kj} = u_j \bar{u}_{k|j}$, and $u_{j|k}$ is the posteriori probability given by Bayes formula (4.14). The amount by which the uncertainty is reduced is called the mutual information $I(X, W)$ as

$$I(X, W) = H(X) - H(W|X) \quad (9.3)$$

Then we have the following properties:

- upper bound of entropy: $H(X) \leq \log(|X|)$, with equality holding if and only if X is uniformly distributed, where $|X|$ is the length of X .
- conditioning reduces entropy: $H(W|X) \leq H(W)$, with equality if and only if X and W are independent.
- chain rule: $H(X, W) = H(X) + H(W|X) = H(W) + H(X|W)$.
- property of conditional entropy: $H(X|W) = 0$ if and only if X is a function of W .

Now we are ready to give the proof of the converse theorem:

Define an error random variable,

$$E = \begin{cases} 0 & \text{if } W \Rightarrow X \\ 1 & \text{otherwise} \end{cases} \quad (9.4)$$

where $W \Rightarrow X$ means X is decoded from W without error. Let p_e denote the probability of error of decoding X from W , then we have $P_r(E = 1) = p_e$ and $P_r(E = 0) = 1 - p_e$. Using the chain rule, we can expand $H(E, X|W)$ in two different ways:

$$H(E, X|W) = H(X|W) + H(E|X, W) = H(E|W) + H(X|E, W) \quad (9.5)$$

Since

- A1: conditioning reduces entropy, so $H(E|W) \leq H(E)$;
- A2: E is a binary-valued random variable, so $H(E) = H(p_e)$;
- A3: E is a function of X and W , so the conditional entropy $H(E|X, W) = 0$;

- A4: given $E = 0$, that means X can be decoded from W without error, there is no uncertainty of X given W is observed, then the conditional entropy $H(X|W, E = 0) = 0$;
- A5: given $E = 1$, then the conditional entropy $H(X|W, E = 1)$ is bounded by the number of all symbols except the correct one in X , i.e., $H(X|W, E = 1) \leq \log(|X| - 1)$.

Combining A1 and A2, we have

$$H(E|W) \leq H(p_e) = -p_e \log p_e - (1 - p_e) \log(1 - p_e) \leq 1 \quad (9.6)$$

Combining A4 and A5, we have

$$\begin{aligned} H(X|E, W) &= P_r(E = 0)H(X|W, E = 0) + P_r(E = 1)H(X|W, E = 1) \\ &\leq (1 - p_e)0 + p_e \log(|X| - 1) = p_e \log(l - 1) \end{aligned} \quad (9.7)$$

Plugging A3, (9.6), and (9.7) back to (9.5), we obtain

$$H(X|W) \leq 1 + p_e \log(l - 1) \quad (9.8)$$

i.e.,

$$p_e \geq \frac{1}{\log(l - 1)}(H(X|W) - 1) \quad (9.9)$$

Since $H(X|W) = H(X) - I(X, W)$, we have

$$p_e \geq \frac{1}{\log(l - 1)}(H(X) - I(X, W) - 1) \geq \frac{1}{\log(l - 1)}(H(X) - C - 1) \quad (9.10)$$

The last inequality is obtained according to the definition of channel capacity in (4.8). This completes the proof.

Appendix C: AIC and BIC

Model Selection aims to estimate the performance of different models in order to choose the best one. Akaike information criterion (AIC) [2] and Bayesian Information Criterion (BIC) [94], the information criterion for model selection, have been widely used in supervised classifications and regressions, and recently extended to determine the cluster number in some cases, especially in biological and biomedical engineering. Here we give a brief introduction of these two information criterion rules for cluster number selection.

Let the training sample be $X = \{x_1, x_2, \dots, x_l\} \subset R^n$, where l is the number of data points, n is the dimension of sample space. Suppose we have candidate models G_c ($c = 1, 2, \dots, c_{max}$) with corresponding model parameters θ_c . The AIC and BIC can be formulated as

$$AIC = -2\log[L(\hat{\theta}_c|X, G_c)] + 2K_c \quad (9.11)$$

and

$$BIC = -2\log[L(\hat{\theta}_c|X, G_c)] + K_c \log(l) \quad (9.12)$$

Where $\log[L(\hat{\theta}_c|X, G_c)]$ is the log of the likelihood when $G_c(X|\hat{\theta}_c)$ is maximized, $\hat{\theta}_c$ is the estimated model parameters, and K_c is the number of the parameters in the model (or the degrees of freedom used by the model). The log-likelihood loss

function in AIC and BIC can be calculated by

$$\log[L(\hat{\theta}_c|X, G_c)] = E_x[(G_c(X|\hat{\theta}_c))] \quad (9.13)$$

When applying AIC and BIC to select the optimal cluster number for the deterministic annealing (DA) clustering algorithm in chapter 6, the following simple expressions are used in the experiments

$$AIC(c) = l \log(R_{ems}(c)) + 2K_c \quad (9.14)$$

and

$$BIC(c) = l \log(R_{ems}(c)) + K_c \log(l) \quad (9.15)$$

where $R_{ems}(c)$ is given by (6.11) and K_c is given by (6.9), please refer to chapter 6 for details. The best model is chosen giving smallest AIC (or BIC) over the set of models considered, i.e., the optimal partition (or an optimal value of c^*) is obtained by minimizing AIC (or BIC) with respect to $c = 2, 3, \dots, c_{max}$. The numerical results of AIC and BIC for the cluster number selection on several artificial and real data sets are listed in Table 6.1.

Author's Publications

- (1) Q. Song, W.J. Hu, X.L. Yang, "An accelerated robust support vector machine algorithm", book chapter in Wang, Lipo (Ed.) Support vector machines: theory and applications, Pages:219-232, Springer-Verlag 2005.
- (2) A.Z. Cao, Q. Song, X.L. Yang, Z.M. Wang, S. Liu, "Mammographic mass detection by robust learning algorithms" book chapter in J.S. Suri and R.M. Rangayyan (Eds.) Recent advances in mammography, breast imaging, and computer-aided diagnosis of breast cancer, Pages:165-202, SPIE Press 2006.
- (3) Q. Song, X.L. Yang, Y.C. Soh, "An information c-spherical shells clustering algorithm", IEEE Trans. on Fuzzy System, Revised.
- (4) X.L. Yang, Q. Song, "A new cluster validity for data clustering", Neural Processing Letters, In Press, 2006.
- (5) X.L. Yang, A.Z. Cao, Q. Song, "A modified deterministic annealing algorithm for robust image segmentation", Image and Vision Computing, Revised.
- (6) X.L. Yang, Q. Song, "A robust deterministic annealing algorithm for data clustering", Data and Knowledge Engineering, In Press, 2006.
- (7) X.L. Yang, "A weighted support vector machine for data classification", International Journal of Pattern Recognition and Artificial Intelligence, Accepted, 2006.
- (8) X.L. Yang, Q. Song, "A new kernel based clustering method", IEE Vision, Image and Signal Processing, In Press, 2006.

- (9) X.L. Yang, Q. Song, A.Z. Cao, "Clustering nonlinearly separable and unbalanced data set", Intelligent Systems, Proceedings. International IEEE Conference on, vol.2, pp.491-496, June 22-24, 2004.
- (10) X.L. Yang, Q. Song, A.Z. Cao, S. Liu, C.Y. Guo, "Robust c-shells based deterministic annealing clustering algorithm", Fuzzy Systems, Proceedings. IEEE International Conference on, vol.3, pp.1413-1417, July 25-29, 2004.
- (11) X.L. Yang, Q. Song, S. Liu, "Pre-selection of working set for SVM decomposition algorithm", IEEE International Joint Conference on Neural Networks, vol.2, pp.883-888, Jul 31 - Aug 4, 2005.
- (12) X.L. Yang, Q. Song, W.B. Zhang, Z.M. Wang, "Clustering of spherical shells by a mini-max information algorithm", ACCV-2006, Lecture Notes in Computer Science, vol.3852, pp.224-233, Springer, 2006.
- (13) X.L. Yang, A.Z. Cao, Q. Song, "Image segmentation by deterministic annealing algorithm with adaptive spatial constraints", Lecture Notes in Computer Science, vol.3972, pp.504-510, Springer, 2006.
- (14) X.L. Yang, Q. Song, M.J. Er, "Robust data clustering in kernel space", Lecture Notes in Computer Science, vol.3971, pp.1231-1237, Springer, 2006.
- (15) A.Z. Cao, Q. Song, X.L. Yang, L. Wang, "Breast mass segmentation based on information theory", Pattern Recognition, Proceedings of the 17th International Conference on, vol.3, pp.758-761, Aug 23-26, 2004.
- (16) A.Z. Cao, Q. Song, X.L. Yang, L. Wang, "Breast mass segmentation on digital mammogram by a combined deterministic annealing method", IEEE International Symposium on Biomedical Imaging, pp.1303-1306, 2004.
- (17) A.Z. Cao, Q. Song, X.L. Yang, S. Liu, C.Y. Guo; "Mammographic mass detection by vicinal support vector machine", Neural Networks, Proceedings. 2004 IEEE International Joint Conference on, vol.3, pp.1953-1958, 2004.

Bibliography

- [1] Ahmed, M.N., Yamany, S.M., Mohamed, N., Farag A.A. and Moriarty, T., *A modified fuzzy C-means algorithm for bias field estimation and segmentation of MRI data*, IEEE Trans. on Medical Imaging, vol.21, no.3, pp.193-199, 2002.
- [2] Akaike, H., *A new look at the statistical model identification*, IEEE Trans. on Automatic Control, vol.19, no.6, pp.716-723, 1974.
- [3] Baudat, G. and Anouar, F., *Generalized discriminant analysis using a kernel approach*, Neural Computation, vol.12, no.10, pp.2385-2404, 2000.
- [4] Ben-Hur, A., Horn, D., Siegelmann, H.T. and Vapnik, V.N., *Support vector clustering*, Journal of Machine Learning Research, vol.2, pp.125-137, 2001.
- [5] Beni, G. and Liu, X., *A least biased fuzzy clustering method*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.16, no.9, pp.954-960, 1994.
- [6] Bezdek, J.C., *Numerical taxonomy with fuzzy sets*, Journal of Mathematical Biology, vol.1, no.1, pp.57-71, 1974.
- [7] Bezdek, J.C., *Cluster validity with fuzzy sets*, Journal of Cybernetics, vol.3, no.3, pp.58-72, 1974.
- [8] Bezdek, J.C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

- [9] Bezdek, J.C., Hathaway, R.J. and Pal, N.R., *Norm-induced shell-prototypes (NIPS) clustering*, Neural, Parallel and Scientific Computations, vol.3, pp.431-450, 1995.
- [10] Bishop, C.M., *Neural Networks for Pattern Recognition*, Oxford University Press, New York, 1995.
- [11] Blahut, R.E., *Computation of channel capacity and rate-distortion functions*, IEEE Trans. on Information Theory, vol.18, no.4, pp.460-473, 1972.
- [12] Blahut, R.E., *Princinple and Practice of Information Theory*, Addison-Wesley, 1987.
- [13] Boser, B.E., Guyon, I.M. and Vapnik, V.N., *A training algorithm for optimal margin classifiers*, The Fifth Annual Workshop on Computational Learning Theory, pp.144-152, 1992.
- [14] Boudraa, A.O., *Dynamic estimation of number of clusters in data sets*, Electronics Letters, vol.35, no.19, pp.1606-1607, 1999.
- [15] Camastra, F. and Verri, A., *A novel kernel method for clustering*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.27, no.5, pp.801-805, 2005.
- [16] Cao, A.Z., *Mammographic Mass Detection Based on Robust Learning Algorithms*, Ph.D Theis, Electrical and Electronics Engineering School, Nanyang Technological University, 2005.
- [17] Cao, L.J., Lee, H.P. and Chong, W.K., *Modified support vector novelty detector using training data with outliers*, Pattern Recognition Letters, vol.24, no.14, pp.2479-2487, 2003.
- [18] Chang, C.C. and Lin, C.J., *LIBSVM: a library for support vector machines*, the software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

- [19] Chiang, J.H. and Hao, P.Y., *A new kernel-based fuzzy clustering approach: support vector clustering with cell growing*, IEEE Trans. on Fuzzy Systems, vol.11, no.4, pp.518-527, 2003.
- [20] Chuang, C.C, Su, S.F., Jeng, J.T. and Hsiao, C.C., *Robust support vector regression networks for function approximation with outliers*, IEEE Trans. on Neural Networks, vol.13, no.6, pp.1322-1330, 2002.
- [21] Cocosco, C.A., Kollokian, V., Kwan, R.K.S. and Evans, A.C., *BrainWeb: online interface to a 3D MRI simulated brain database*, NeuroImage, vol.5, no.4, S245, 1997.
- [22] Cortes, C. and Vapnik, V.N., *Support vector networks*, Machine Learning, vol.20, no.3, pp.273-297, 1995.
- [23] Cristianin, N. and Shawe-Taylor, J., *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, Cambridge, 2000.
- [24] Dave, R.N., *Fuzzy shell-clustering and applications to circle detection in digital images*, Int.J.General Systems, vol.16, no.4, pp.343-355, 1990.
- [25] Dave, R.N., *Characterization and detection of noise in clustering*, Pattern Recognition Letters, vol.12, no.11, pp.657-664, 1991.
- [26] Dave, R.N. and Bhaswan, K., *Adaptive fuzzy c-shells clustering and detection of ellipse*, IEEE Trans. on Neural Networks, vol.3, no.5, pp.643-662, 1992.
- [27] Dave, R.N. and Fu, T., *Robust shape detection using fuzzy clustering: practical applications*, Fuzzy Sets and Systems, vol.65, no.2-3, pp.161-185, 1994.
- [28] Dave, R.N., *Validating fuzzy partitions obtained through c-shells clustering*, Pattern Recognition Letters, vol.17, no.6, pp.613-623, 1996.
- [29] Dave, R.N. and Krishnapuram, R., *Robust clustering methods: a unified view*, IEEE Trans. on Fuzzy Systems, vol.5, no.2, pp.270-293, 1997.

-
- [30] Dave, R.N. and Sen, S., *Robust fuzzy clustering of relational data*, IEEE Trans. on Fuzzy Systems, vol.10, no.5, pp.713-727, 2002.
 - [31] *DELVE benchmark repository : a collection of artificial and real world data sets*, Available at <http://www.cs.utoronto.ca/delve/data/datasets.html>.
 - [32] Dempster, A.P., Laird, N.M. and Rubin, D.B., *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society (B), vol.39, pp.1-38, 1977.
 - [33] Duda, R.O., Hart, P.E. and Stork, D.G., *Pattern Classification—Second Edition*, John Wiley and Sons, Inc, New York, 2001.
 - [34] Dunn, J.C., *A fuzzy relative of the ISODATA process and its use in detecting compact well-separated cluster*, Journal of Cybernetics, vol.3, pp.32-57, 1974.
 - [35] Fogel, D.B., *An information criterion for optimal neural network selection*, IEEE Trans. on Neural Networks, vol.2, no.5, pp.490-497, 1991.
 - [36] Fukuyama, Y. and Sugeno, M. *A new method of choosing the number of clusters for the fuzzy c-means method*, in: Proceedings of the Fifth Fuzzy Systems Symposium, pp.247-250, 1989.
 - [37] Gath, I. and Geva, A.B., *Unsupervised optimal fuzzy clustering*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.11, no.7, pp.773-781, 1989.
 - [38] Gath, I. and Hoory, D., *Fuzzy clustering of elliptic ring-shaped clusters*, Pattern Recognition Letters, vol.16, pp.727-741, 1995.
 - [39] Girolami, M., *Mercer kernel-based clustering in feature space*, IEEE Trans. on Neural Networks, vol.13, no.3, pp.780-784, 2002.
 - [40] Graepel, T., Burger, M. and Obermayer K., *Self-organizing maps: generalizations and new optimization techniques*, Neurocomputing, vol.21, no.1-3, pp.173-190, 1998.
-

- [41] Graepel, T. and Obermayer, K., *Fuzzy topographic kernel clustering*, In W. Brauer, editor, Proceedings of the 5th GI Workshop Fuzzy Neuro Systems, pp.90-97, 1998.
- [42] Gunn, S., *Support vector machines for classification and regression*, Technical Report, Image, Speech and Intelligent Systems Research Group ISIS, University of Southampton, 1998.
- [43] Gustafson, E.E. and Kessel, W.C., *Fuzzy clustering with a fuzzy covariance matrix*, in Proc. IEEE Conf. Decision Contr., San Diego, CA, pp.761-766, 1979.
- [44] Herbrich, R. and Weston, J., *Adaptive margin support vector machines for classification*, The Ninth International Conference on Artificial Neural Network (ICANN 99), vol.2, pp.880-885, 1999.
- [45] Hoepfner, F., *Fuzzy shell clustering algorithms in image processing: fuzzy c-rectangular and 2-rectangular shells*, IEEE Trans. on Fuzzy Systems, vol.5, no.8, pp.599-613, 1997.
- [46] Hofmann, T. and Buhmann, J.M., *Pairwise data clustering by deterministic annealing*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.19, no.1, pp.1-14, 1997.
- [47] Horn, D., *Clustering via hilbert space*, Physica A, vol.302, pp.70-79, 2001.
- [48] Hu, W.J. and Song, Q., *An accelerated decomposition algorithm for robust support vector machines*, IEEE Trans. on Circuits and Systems, vol.51, no.5, pp.234-240, 2004.
- [49] Huang, H.P. and Liu, Y.H., *Fuzzy support vector machines for pattern recognition and data mining*, International Journal of Fuzzy Systems, vol.4, no.3, pp.826-835, 2002.
- [50] *IDA benchmark repository used in several boosting, KFD and SVM papers*, available at <http://ida.first.gmd.de/ratsch/data/benchmarks.htm>.

-
- [51] Jain, A.K., Murty, M.N. and Flynn, P.J., *Data clustering: a review*, ACM Computing Surveys, vol.31, no.3, pp.264-323, 1999.
 - [52] Joachims, T., *Making large-scale support vector machine learning practical*, In Scholkopf et al. (Eds.). *Advances in Kernel Methods-Support Vector Learning* (pp.169-184). Cambridge, MA: MIT Press, 1999.
 - [53] *Kernel machines: a collection of literature, software and web pointers dealing with SVM and Gaussian processes*, Available at <http://www.kernel-machines.org>.
 - [54] Kim, D.W., Lee, K.H. and Lee, D., *On cluster validity index for estimation of the optimal number of fuzzy clusters*, Pattern Recognition, vol.37, no.10, pp.2009-2025, 2004.
 - [55] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P., *Optimization by simulated annealing*, Science, vol.220, no.4598, pp.671-680, 1983.
 - [56] Krishnapuram, R., Nasraoui, O. and Frigui, H., *The fuzzy c-spherical shells algorithm: a new approach*, IEEE Trans. on Neural Networks, vol.3, no.5, pp.663-671, 1992.
 - [57] Krishnapuram, R. and Keller, J.M., *A possibilistic approach to clustering*, IEEE Trans. on Fuzzy System, vol.1, no.2, pp.98-110, 1993.
 - [58] Krishnapuram, R., Frigui, H. and Nasraoui, O., *Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation Part I and Part II*, IEEE Trans. on Fuzzy Systems, vol.3, no.1, pp.29-60, 1995.
 - [59] Krishnapuram, R. and Keller, J.M., *The possibilistic c-means algorithm: insights and recommendations*, IEEE Trans. on Fuzzy System, vol.4, no.3, pp.385-393, 1996.

- [60] Kumar, R., Jayaraman, V.K. and Kulkarni, B.D., *An SVM classifier incorporating simultaneous noise reduction and feature selection: illustrative case examples*, Pattern Recognition, vol.38, pp.41-49, 2005.
- [61] Kwon, S.H., *Cluster validity index for fuzzy clustering*, Electronics Letters, vol.34, no.22, pp.2176-2177, 1998.
- [62] Leski, J.M., *Fuzzy c-varieties/elliptotypes clustering in reproducing kernel Hilbert space*, Fuzzy Sets and Systems, vol.141, pp.259-280, 2004.
- [63] Liew, A.W.C. and Yan, H., *An adaptive spatial fuzzy clustering algorithm for 3-D MR image segmentation*, IEEE Trans. on Medical Imaging, vol.22, no.9, pp.1063-1075, 2003.
- [64] Liew, A.W.C., Leung, S.H. and Lau, W.H., *Segmentation of color lip images by spatial fuzzy clustering*, IEEE Trans. on Fuzzy Systems, vol.11, no.4, pp.542-549, 2003.
- [65] Lin, C.F. and Wang, S.D., *Fuzzy support vector machines*, IEEE Trans. on Neural Networks, vol.13, no.2, pp.464-471, 2002.
- [66] MacQueen, S., *Some methods for classification and analysis of multivariate observations*, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, vol.1, pp.281-297, 1967.
- [67] McLachlan, G.J. and Basford, K.E., *Mixture Models: Inference and Applications to Clustering*, Marcel Dekker Inc., New York, 1988.
- [68] Man, Y. and Gath, I., *Detection and separation of ring-shaped clusters using fuzzy clustering*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.16, no.8, pp.855-861, 1994.
- [69] McGill University, Available: <http://www.bic.mni.mcgill.ca/brainweb>.

-
- [70] Mika, S., *Kernel Fisher Discriminants*, Ph.D Thesis, University of Technology, Berlin, 2002.
 - [71] Miller, D. and Rose K., *Hierarchical, unsupervised learning with growing via phase transitions*, Neural Computation, vol.8, no.2, pp.427-452, 1996.
 - [72] Muller, K.R., Mika, S., Ratsch, G., Tsuda, K. and Scholkopf, B., *An introduction to kernel based learning algorithms*, IEEE Trans. on Neural Networks, vol.12, no.2, pp.181-201, 2001.
 - [73] Pakhira, M.K., Bandyopadhyay, S. and Maulik, U., *Validity index for crisp and fuzzy clusters*, Pattern Recognition, vol.37, no.3, pp.487-501, 2004.
 - [74] Pal, N.R. and Pal, S.K., *A review on image segmentation techniques*, Pattern Recognition, vol.26, no.9, pp.1277-1294, 1993.
 - [75] Pal, N.R. and Bezdek, J.C., *On cluster validity for the fuzzy cmeans model*, IEEE Trans. on Fuzzy Systems, vol.3, no.3, pp.370-379, 1995.
 - [76] Pal, N.R., Pal, K., Keller, J.M. and Bezdek, J.C., *A possibilistic fuzzy c-means clustering algorithm*, IEEE Trans. on Fuzzy Systems, vol.13, no.4, pp.517-530, 2005.
 - [77] Pham, D.L. and Prince, J.L., *Adaptive fuzzy segmentation of magnetic resonance images*, IEEE Trans. on Medical Imaging, vol.18, no.9, pp.737-752, 1999.
 - [78] Pham, D.L., Xu, C. and Prince, J.L., *Current methods in medical image segmentation*, Annual Review of Biomedical Engineering, vol.2, pp.315-337, Palo Alto, CA, 2000.
 - [79] Pham, D.L., *Spatial models for fuzzy clustering*, Computer Vision and Image Understanding, vol.84, pp.285-297, 2001.

-
- [80] Platt, J.C., *Fast training of support vector machines using sequential minimal optimization*, In Scholkopf et al. (Eds.). *Advances in Kernel Methods-Support Vector Learning* (pp.185-208). Cambridge, MA: MIT Press, 1999.
 - [81] Ratsch, G., and Onoda, T. and Muller, K.R., *Soft margins for AdaBoost*, Machine Learning, vol.42, no.3, pp.287-320, 2001.
 - [82] Rezaee, M.R., Lelieveldt, B.P.F. and Reiber, J.H.C., *A new cluster validity index for the fuzzy c-mean*, Pattern Recognition Letters, vol.19, no.3-4, pp.237-246, 1998.
 - [83] Roberts, S.J., Everson, R. and Rezek, I., *Maximum certainty data partitioning*, Pattern Recognition, vol.33, no.5, pp.833-839, 2000.
 - [84] Roberts, S.J., Holmes, C. and Denison D., *Minimum-entropy data partitioning using reversible jump Markov chain Monte Carlo*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.23, no.8, pp.909-914, 2001.
 - [85] Rose, K., Gurewitz, E. and Fox, G.C., *Statistical mechanics and phase transitions in clustering*, Physical Review letters, vol.65, no.8, pp.945-948, 1990.
 - [86] Rose, K., Gurewitz, E. and Fox, G.C., *Constrained clustering as an optimization method*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol.15, no.8, pp.785-794, 1993.
 - [87] Rose, K., *Deterministic annealing for clustering, compression, classification, regression, and related optimization problems*, Proceedings of the IEEE, vol.86, no.11, pp.2210-2239, 1998.
 - [88] Scholkopf, B., Burges, C.J.C. and Vapnik, V.N., *Extracting support data for a given task*, in: Fayyad, U.M., Uthurusamy, R. (Eds.), Ist Int. Conf. on Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, CA, pp.252-257, 1995.

-
- [89] Scholkopf, B., Smola, A.J. and Muller, K.R., *Nonlinear component analysis as a kernel eigenvalue problem*, Technical Report, Max Planck Institute for Biological Cybernetics, Tubingen, Germany, 1996.
 - [90] Scholkopf, B., Smola, A.J. and Muller, K.R., *Nonlinear component analysis as a kernel eigenvalue problem*, Neural Computation, vol.10, no.5, pp.1299-1319, 1998.
 - [91] Scholkopf, B., Mika, S., Burges, C.J.C., Knirsch, P., Muller, K.R., Ratsch, G. and Smola, A.J., *Input space versus feature space in kernel based methods*, IEEE Trans. on Neural Networks, vol.10, no.5, pp.1000-1017, 1999.
 - [92] Scholkopf, B., Platt, J., Shawe-Taylor, J. and Smola, A.J., *Estimating the support of a high-dimensional distribution*, Neural Computation, vol.13, no.7, pp.1443-1470, 2001.
 - [93] Scholkopf, B. and Smola, A.J., *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.
 - [94] Schwarz, G., *Estimating the dimension of a model*, The Annals of Statistics, vol.6, no.2, pp.461-464, 1978.
 - [95] Selim, S.Z. and Ismail, M.A., *K-means-type algorithms: a generalized convergence theorem and characterization of local optimality*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.6, no.1, pp.81-86, 1984.
 - [96] Selim, S.Z. and Alsulta, K.S., *A simulated annealing algorithm for the clustering problem*, Pattern Recognition, vol.24, no.10, pp.1003-1008, 1991.
 - [97] Silverman, B.W., *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London, 1986.
 - [98] Song, Q., Hu, W.J. and Xie, W.F, *Robust support vector machine with bullet hole image classification*, IEEE Trans. on Systems, Man and Cybernetics, vol.32, no.4, pp.440-448, 2002.
-

-
- [99] Song, Q., *A robust information clustering algorithm*, Neural Computation, vol.17, no.12, pp.2672-2698, 2005.
 - [100] Suykens, J.A.K., De Brabanter, J., Lukas, L. and Vandewalle, J., *Weighted least squares support vector machines: robustness and sparse approximation*, Neurocomputing, vol.48, no.1-4, pp.85-105, 2002.
 - [101] Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B. and Vandewalle, J., *Least Squares Support Vector Machines*, World Scientific Pub. Co., Singapore, 2002
 - [102] Tax, D.M.J. and Duin, R.P.W., *Support vector data description*, Pattern Recognition Letters, vol.20, no.11-13, pp.1191-1199, 1999.
 - [103] Tolias, Y.A. and Panas, S.M., *On applying spatial constraints in fuzzy image clustering using a fuzzy rule-based system*, IEEE Signal Processing Letters, vol.5, no.10, pp.245-247, 1998.
 - [104] Tolias, Y.A. and Panas, S.M., *Image segmentation by a fuzzy clustering algorithm using adaptive spatially constrained membership functions*, IEEE Trans. on Systems, Man and Cybernetics, vol.28, no.3, pp.359-369, 1998.
 - [105] *UCI benchmark repository : a huge collection of artificial and real world data sets*, University of California Irvine, <http://www.ics.uci.edu/mllearn>.
 - [106] Vapnik, V.N., *The Nature of Statistical Learning Theory*, Springer, New York, 1995.
 - [107] Vapnik, V.N., *Statistical Learning Theory*, John Wiley and Sons, Inc, New York, 1998.
 - [108] Vovk, U., Pernus, F. and Likar, B., *MRI intensity inhomogeneity correction by combining intensity and spatial information*, Phys. Med. Biol., vol.49, pp.4119-4133, 2004.

- [109] Wu, K.L. and Yang, M.S., *Alternative c-means clustering algorithms*, Pattern Recognition, vol.35, no.10, pp.2267-2278, 2002.
- [110] Wu, K.L. and Yang, M.S., *A cluster validity index for fuzzy clustering*, Pattern Recognition Letters, vol.26, no.9, pp.1275-1291, 2005.
- [111] Wu, Z.D., Xie, W.X. and Yu, J.P., *Fuzzy c-means clustering algorithm based on kernel method*, Fifth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'03), pp.49-55, 2003.
- [112] Xie, X.L. and Beni, G., *A validity measure for fuzzy clustering*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.13, no.8, pp.841-847, 1991.
- [113] Yang, M.S. and Wu, K.L., *A similarity-based robust clustering method*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.26, no.4, pp.434-448, 2004.
- [114] Zahid, N., Limouri, M. and Essaid, A., *A new cluster-validity for fuzzy clustering*, Pattern Recognition, vol.32, no.7, pp.1089-1097, 1999.
- [115] Zhang, D.Q. and Chen, S.C., *A novel kernelized fuzzy C-means algorithm with application in medical image segmentation*, Artificial Intelligence in Medicine, vol.32, pp.37-50, 2004.
- [116] Zhang, D.Q. and Chen, S.C., *Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance Measure*, IEEE Trans. on Systems, Man and Cybernetics, vol.34, no.4, pp.1907-1916, 2004.
- [117] Zhang, J.S. and Leung, Y.W., *Robust clustering by pruning outliers*, IEEE Trans. on Systems, Man, and Cybernetics, vol.33, no.6, pp.983-999, 2003.
- [118] Zhang, J.S. and Leung, Y.W., *Improved possibilistic c-means clustering algorithms*, IEEE Trans. on Fuzzy Systems, vol.12, no.2, pp.209-217, 2004.

-
- [119] Zhang, R. and Rudnický, A.I., *A large scale clustering scheme for kernel K-means*, IEEE Proceedings, 16th International Conference on Pattern Recognition, pp.289-292, 2002.
- [120] Zhang, X.G., *Using class-center vectors to build support vector machines*, Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop, pp.3-11, 1999.
- [121] Zhou, S.M. and Gan, J.Q., *An unsupervised kernel based fuzzy c-means clustering algorithm with kernel normalization*, International Journal of Computational Intelligence and Applications, vol.4, no.4, pp.355-373, 2004.