University of Cincinnati

Date: 12/11/2013

I, Mohammad Y. Rawashdeh , hereby submit this original work as part of the requirements for the degree of Doctor of Philosophy in Computer Science & Engineering.

It is entitled:

A Relational Framework for Clustering and Cluster Validity and the Generalization of the Silhouette Measure

Student's name: **Mohammad Y. Rawashdeh**

This work and its defense approved by:

Committee chair: Anca Ralescu, Ph.D.

Committee member: Anil Jegga, D.V.M., M.Res.

Committee member: Traian Marius Truta, Ph.D.

Committee member: Fred Annexstein, Ph.D.

Committee member: Kenneth Berman, Ph.D.

Committee member: Dan Ralescu, Ph.D.

9043

A Relational Framework for Clustering and Cluster Validity and the Generalization of the Silhouette Measure

A Dissertation Submitted to the

Division of Research and Advanced Studies
University of Cincinnati

in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy

In the Department of
Electrical Engineering and Computing Systems
of College of Engineering and Applied Sciences

December 2013

By

Mohammad Y. Rawashdeh

M.S. Computer Science, Yarmouk University, Jordan, August 2005
B.S. Applied Mathematics, Jordan University of Science and Technology, Jordan, 2001

Thesis Advisor and Committee Chair: Anca Ralescu, Ph.D.

ABSTRACT

By clustering one seeks to partition a given set of points into a number of clusters such that points in the same cluster are similar and are dissimilar to points in other clusters. In the virtue of this goal, data of relational nature become typical for clustering. The similarity and dissimilarity relations between the data points are supposed to be the nuts and bolts for cluster formation. Thus, the task is driven by the notion of similarity between the data points. In practice, the similarity is usually measured by the pairwise distances between the data points. Indeed, the objective function of the two widely used clustering algorithms, namely, k-means and fuzzy c-means, appears in terms of the pairwise distances between the data points.

The clustering task is complicated by the choice of the distance measure and estimating the number of clusters. Fuzzy c-means is convenient when there are uncertainties in allocating points, in overlapping areas, to clusters. The k-means algorithm allocates the points unequivocally to clusters; overlooking the similarities between those points in overlapping areas. The fuzzy approach allows a point to be a member in as many clusters as necessary; thus it provides better insight into the relations between the points in overlapping areas.

In this thesis we develop a relational framework that is inspired by the silhouette measure of clustering quality. The framework asserts the relations between the data points by means of logical reasoning with the cluster membership values. The original description of computing the silhouettes is limited to crisp partitions. A natural generalization of silhouettes, to fuzzy partitions is given within our framework. Moreover, two notions of silhouettes emerge within the framework at different levels of granularity, namely, point-wise silhouette and center-wise silhouette. Now by the generalization, each silhouette is capable of measuring the extent to which a crisp, or fuzzy, partition has fulfilled the clustering goal at the level of the individual points, or cluster centers. The partitions are evaluated by the silhouette measure in conjunction with point-to-point or center-to-point distances.

By the generalization, the average silhouette value becomes a reasonable device for selecting between crisp and fuzzy partitions of the same data set. Accordingly, one can find about which partition is better in representing the relations between the data points, in accordance with their pairwise distances. Such powerful feature of the generalized silhouettes has exposed a problem with the partitions generated by fuzzy *c*-means. We have observed that defuzzifying the fuzzy *c*-means partitions always improves the overall representation of the relations between the data points. This is due to the inconsistency between some of the membership values and the distances between the data points. This inconsistency was reported, by others, in a couple of occasions in real life applications.

Finally, we present an experiment that demonstrates a successful application of the generalized silhouette measure in feature selection for highly imbalanced classification. A significant improvement in the classification for a real data set has resulted from a significant reduction in the number of features.

ACKNOWLEDGMENTS

Those years I spent at University of Cincinnati pursuing a Ph.D. have been quite tough years. But it was worth the time and effort since indeed deep down at the bottom of the ocean you find pearls. Below are the pearls I have found. My Ph.D. was made so memorable because of them.

First and foremost, I am forever in the debt of my inspiring advisor, Dr. Anca Ralescu. She is a knowledgeable and a passionate researcher, who is also an amazing cook. I treasure those discussions we had in our weekly, sometimes Starbucks coffee, meetings. The detailed feedback on my work, results and drafts was absolutely instrumental in my progress. She was a friend and true mentor.

I am grateful to Drs. Kenneth Berman, Fred Annexstein, Dan Ralescu, Anil Jegga and Traian Marius Truta for serving on my Ph.D. committee. I was uplifted by their interest in my work, support, and words of encouragement.

Special thanks go to Dr. Kenneth Berman for the wonderful experience I had with him as a teaching assistant. In addition, I would like to thank my department for the trust in appointing me as a teaching assistant for many courses, and the support for attending conferences.

There would be no better opportunity than this one to thank Mary Ellen Collins, Larry Collins, Ann Collins Hamilton and Ryan Hamilton for letting me into their hearts and homes. With them, I met wonderful people, and together, we had wonderful times, not to mention the delicious food.

Last but not least are the crew members of the ship that brought me to Cincinnati. I would like to express my utmost gratitude to Mom, Dad, Rabeah, Ahmad, Omar and Amer. They are the most important people in my world.

Mohammad Y. Rawashdeh

Univeristy of Cincinnati

December 2013

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Change is rampant in our world today, in market, healthcare, education, societies, etc. The discovery of new trends and anomalies followed by proper actions is becoming a competitive differentiator for organizations. In big data there is a bigger potential for such discoveries, and hence for seizing future opportunities. Thus, now and then, data is being collected on customers, workforce, creditors and suppliers. For example, the customer information collected by Amazon.com, the large online retailer, includes[1]:

- Computer IP address, operating system and platform.
- Browser type and version.
- Browser plug-in types and versions.
- The full unifrom resource locator (URL) clickstream to, through and from the web site including date and time, cookie number and products viewed and searched for.
- Purchase history.
- Phone numbers used to call the customer service.

Such practice of data collection is on the rise as more and more are breaking through the big data frontier. The hallmark of this trend is the use of scale-out computing platforms which run on top of commodity hardware, available at affordable costs. Scaling out, as opposed to scaling up, delivers more computing power at lower costs. This process has been pioneered by many of the major web players like Google, Yahoo, eBay and Amazon, to cope with the increasing computational demand [55]. It is also observed as a driving force in the evolution of high-performance computing (HPC) systems.[2] Organizations that wish to take advantage of data scale but are

[1] Amazon.com Help: Amazon.com Privacy Notice. (2012). Retrieved November 30, 2013, from http://www.amazon.com/gp/help/customer/display.html?nodeId=468496

[2] Consider the semi-annual TOP500 lists, ¡ http://www.top500.org/

worried about the operational cost, associated with running their own systems, resort to cloud platforms available. Whether on cloud or on-premises, anybody can take advantage of big data capable platforms.

Descriptive analytics transforms data into hindsight. The hindsight is obtained from querying, reporting, and online analytical processing (OLAP) tools. However, big data big expectations call for new kind of analytics. New avenues are now open for predictive and prescriptive analytics which deliver insight and foresight from data, necessary for informed decisions and for prescribing the best course of action. Machine learning can be applied to predictive analytics, to learn common characteristics, structure and predictive models on data. Machine learning algorithms can be divided into three categories: supervised, semi-supervised, and unsupervised, according to whether they operate on labeled, partially labeled, and unlabeled data, respectively. Some of the core machine learning algorithms are, to name few, cluster analysis, classification, regression, feature selection and similarity learning. Cluster analysis, the focus of this research, is an exploratory data analysis technique with the goal of recovering the group structure of data that best fits some grouping criterion. Clustering is ubiquitous in many applications over a wide range of domains, such as the analysis of gene expressions in bioinformatics [7], image segmentation [73], speech recognition [20], information retrieval [74], web mining [44] and recommendation systems [50].

There are several data analysis packages that support machine learning algorithms such as Matlab, SAS and R. However, they are designed for data that fits a single machine memory and usually they do not scale to larger datasets. Fortunately, several projects, such as Apache Mahout, Apaches Spark, HaLoop, Twister and Daytona, bridge the gap between sophisticated analytics and big data [53]. A quick search for jobs in machine learning and big data on LinkedIn, which purports itself as the world's largest professional network, shows that machine learning and big data analytics is in high demand. At the time of writing, a search using the keywords 'machine learning' and 'big data' returned approximately 450 hits within the United States only. The high demand is evident by comparing the number of hits with another search using the keyword 'teacher', a fairly broad job title, which returned approximately 500 openings in the United States. Thus, it is not surprising that McKinsey & Company, Inc., an American global management consulting firm, projects that by 2018 there will be 140,000 to 190,000 unfilled positions of data analytics experts in the United States and a shortage of 1.5 million managers and analysts with an understanding of how big data can be applied [18]. Driven by such a demand, some universities have started implementing programs in analytics [79].

The Attributes of Big Data

Big data is not just about size, although size matters; there are other defining attributes. Data *velocity* and *variety*, rather than data *volume* alone, contribute to its complexity. The expansion of all these three dimensions causes data complexity to grow beyond the ability of existing tools. That is, data becomes too big, too fast or too hard to process [53]. The collective use of these three attributes, the 3Vs, in defining data complexity has been in use for more than a decade, introduced by Doug Laney [48]. A fourth V can be associated with big data in regard to analysis results rather than to data complexity that is, *value* or the usefulness of the results.

Struggling with large volumes of data caused few respondents in a survey of big data analytics to describe it as the pain-in-the-neck and we-need-to-buy-more-hardware analytics [70]. Nevertheless, the abundance of data has the merit of providing enough, probably reliable, samples for robust results. For example, a study on the early discovery of cancer using predictive lists of genes [29] demonstrates the importance of sample size in generating lists of robust predictive power; robust is in the sense of predicting in an accurate manner if used with patient data from other studies. Published gene lists obtained by different groups showed degraded predictive performance when used on each other's data, on the same clinical types of patients. The lists have very few genes in common due to the small size of the samples used in generating the lists. The main result is that larger samples with expression profiles of several thousand patients are necessary to achieve an overlap of 50% and accurate predictions.

Big data velocity is about matching the speed of decisions to the speed of actions, for example, real-time fraud detection or high quality recommendations at a point of sale. Consider the incident involving the director of one of the agencies behind developing the Electronic Surveillance System for the Early Notification of Community-based Epidemics (ESSENCE). Jay Mansfield, the director, reported to his boss that they detected a gastrointestinal outbreak in Korea. His boss asked him about the time of the outbreak. Since Korea is 13 hours ahead of Washington, Mansfield simply answered: “Tomorrow.” [80]

The range of data sources, the many varying formats and the various degrees of structure contribute to big data variety. Variety also accounts for semantic heterogeneity. In a study aimed at detecting associations between diagnoses, based on the association of gene expression data [38], the group manually reviewed a list of 20,705 unique diagnoses to map terms that refer to the same diagnosis. The terms were extracted from 1.5 million free text problem summary list diagnoses, entered by roughly 2,000 clinicians into the electronic health records (EHR) system.

Uncertainty and Big Data

Big data brings with it some elements of uncertainty. More data means more confusion, false positives in classification tasks [16], noise and errors. Probability theory and fuzzy logic are two different ways of expressing uncertainty. In particular, fuzzy clustering is a clustering approach based on the use of fuzzy sets. It is capable of recovering group structures with overlapping. The overlapping is due to uncertainty in assigning the data point to the groups, or equivalently to the imprecise nature of those groups. Crisp clustering on the other hand, is based on exclusive group assignment of the data points, therefore, overlooking any possible overlapping. Since cluster analysis, or data analysis in a broad sense, is supposed to serve a useful purpose, the quality of its product must be assessed. In case of predictive models, the outcomes are predicted on the basis of numbers, mere probabilities. Probabilities are devices to convey uncertainty. An agreement in the performance of the predictive models on training data and unseen data (test data), is preferred to establish some kind of confidence in them. Bias-variance analysis is the tool to assess such an agreement. In contrast, making an assessment of clustering quality is the subject of cluster validity.

The main contribution of this research is two-fold. First, we develop a relational framework with a unified distance-based perspective on the problems of clustering and cluster validity. Second and within the relational framework, we generalize the silhouette measure, a measure of clustering quality, to the clustering results obtained by any fuzzy algorithm. The framework, and so does the silhouette measure, links together:

- The similarities and dissimilarities between the data points.
- The distances between the data points.
- The cluster assignment of the data points.

In chapter two we discuss the key elements in the clustering task, and give a detailed review of k-means and, its fuzzy variant, fuzzy c-means, the two widely used clustering algorithms. In chapter three we review few of the popular measures used in the validation of crisp and fuzzy clustering results, with a focus on the silhouette measure. In chapter four, the relational framework is presented. In chapter five, and based on concepts defined within the relational framework, the silhouette measure is naturally generalized to fuzzy partitions. The silhouette measure exposes a problem inherent in any fuzzy c-means partition, as explained in chapter six. In chapter seven, we conclude with an experiment that demonstrates a successful application of the generalized silhouette measure in feature selection for classification with real data.

CHAPTER 2
OVERVIEW OF CLUSTERING

Since clustering is a primitive activity necessary to cope with complexity, it is pervasive in analysis. Therefore, various clustering algorithms were developed in different fields. But they are centered on one intuitive goal: *to partition a given set of points into a number of clusters such that points in the same cluster are similar and are dissimilar to points in other clusters.* This is a common definition of the clustering task given in the clustering literature [9, 42, 45, 67, 69, 84]. In machine learning, the perspective is that clustering is an unsupervised approach to learning, due to the lack of target class labels. But in many clustering applications there is a clue, a level of supervision, of what we are looking for. This is exactly the paradoxical *unsupervised-supervised dimension* of clustering stated by Guyon et al. [36]. For example, given a collection of images we might want to cluster them by who is in them or by facial expression. In another application we might seek to cluster a collection of documents by topic, authorship or writing style [83]. In such cases, it is possible to cluster the data points manually, albeit a substantially labor and time intensive task. Such *desired partitions* of the data become the *end goal* of clustering. Few key remarks can be made about clustering in the light of the definition:

- The purpose of clustering is to reduce the similarities and dissimilarities between the data points in the form of clusters. Accordingly, it becomes easier to determine whether a group of points are similar or dissimilar to each other by their cluster membership.

- The definition clearly establishes the *notion of cluster* on some constituent *notion of similarity.* Cluster formation is essentially driven by the selected notion of similarity. Different clusters are formed by different notions of similarity.

- Usually in practice, the similarities between the data points are measured by a *distance.*

- The distance measure is considered poor for the clustering task if the formed clusters overlap each other to a great extent. Setting the clustering parameters improperly is also responsible for *cluster overlapping*, for example, the *number of clusters*.

- The clustering quality of a partition is a function of

  Cluster *compactness* and *separation*, if using a distance measure.

  Cluster homogeneity and heterogeneity, if using an actual similarity measure.

- Compactness is associated with the *within-cluster* pairwise distances. Smaller distances indicate that similar points were successfully assigned to the same cluster.

- Separation is associated with the *between-cluster* distances. Larger distances indicate that dissimilar points were successfully separated by assigning them to different clusters.

- Independent from any distance or similarity, the number of all possible partitions of a set of n points is given by the n^{th} Bell number[3]. This is inclusive of the two trivial partitions of assigning all points to the same cluster or assigning each point to its own cluster (n singletons). For instance, there are 115975 possible ways to partition a set of 10 points.

The desired partition is the third element in Blums triple, which defines the clustering problem [15]. The triple is (X, d, P^*) where X is the set of data points, d is the distance measure and P^* is the unknown desired partition. The triple is part of a framework that addresses the properties of the relation between d and P^* sufficient for an algorithm to produce P^* with a low error, therefore generating a meaningful partition. For example, consider the *single cutoff* property that is for some cutoff value c we have $d(x, y) < c$ for all pairs x and y that should be in the same cluster and $d(x, y) > c$ for all pairs that should be in different clusters. Clearly this makes the

[3] The Bell numbers satisfy the recurrence relation $B_{n+1} = \sum_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) B_k$, where $B_0 = B_1 = 1$.

job of the clustering algorithm trivial; recovering P^* by a single greedy scan of the pairwise distances. However, when it comes to other properties which correspond to a more or less poor distance measure, the task is not straightforward. Therefore, clustering algorithms define some criterion in terms of the distance measure to be optimized on the entire set of points, in the presence of any cluster overlapping. A review of distance measures and criteria used in clustering is given in [67]. It is worthwhile noting that the single cutoff property is similar in essence to Dunns definition of *compact separated clusters*[4] [28].

Input consisting of *feature representation* of the data and a *distance measure*, or alternatively the set of *pairwise distances*, is typical for hierarchical, partitioning and density-based clustering algorithms. Consider [9, 67, 84] for reviews on these methods of clustering. Such input defines a *graph structure* on the data. Changing the features or the distance measure opens the doors for new clustering possibilities; as it redefines the similarity relations between the data points, the graph structure. Regardless, fixing the input, the goal of clustering can be restated as *the search for optimally compact and separated clusters*. In the best case scenario, the clustering algorithm converges to an optimal partition that approximates well the desired, possibly unknown, partition. For instance, an optimal partition of a collection of documents really reflects a set of meaningful topics. A scenario that becomes possible only if the graph structure of the data set is relevant to the desired partition. A partition is optimal with respect to the graph structure (distances); whereas it is meaningful if it approximates the desired partition. So it is reasonable to ask for the features and the distance measure to be selected in the context of the desired partition. The answer is provided by methods for learning a distance measure on data as in [5,83] or a set of features [86], in a supervised setting. Similar in spirit is the problem of learning a *kernel* [47,93]; considering the fact that kernels are generalized distance measures [71]. All of these learning problems, and by embedding the data in a different space, manipulate the relations between the data points in order for the relations to be relevant for other learning tasks, for instance classification.

[4] Informally speaking, the clusters (subsets) of a set of points are said to be compact separated clusters if the distance between each pair of points within the same cluster is smaller than every distance between points in different clusters.

Recognizing the critical role played by the distance measure in clustering, Blum, Bezdek and Dunn, respectively, make the following claims:

- The distance measure is of the same level of importance as (if not more than) the clustering algorithm [15].
- The definition of a mathematical measure of similarity is the fundamental problem of cluster analysis [14].
- The existence of a partition into k compact separated clusters is an intrinsic property of the pair (X, d) that is the set of points and the distance measure [28].

The key steps involved in the clustering process are shown in Figure 2.1 and they are summarized below. They are based on outlines given in [37, 39, 42, 84].

- *Feature representation*: the set of data points are described by a set of features which is refined, if necessary, by feature selection, feature extraction, or normalization (for example by computing the z-scores).
- *Distance measure*: the measure must be meaningful to the task i.e. relevant to the desired partition; since it quantifies the relations between the data points. Clustering algorithms critically rely on the chosen measure in achieving the desired partition. Clustering, and so cluster formation, is driven by the measure. Rather than a component of the clustering algorithm, it is reasonable to treat the distance measure as an input. Many clustering algorithms accept different measures where the clustering results vary with the distance measure. The input is either the set of pairwise distances or a combination of data feature representation and a distance measure.
- *Clustering algorithm*: at the core of the clustering algorithm is a criterion defined in terms of the distance measure. A reasonable criterion must be a measure of clustering quality that is cluster compactness, cluster separation or both. The task then translates into an optimization problem: to optimize the criterion over the space of all possible partitions. However, as noted earlier,

this space on a given set might be huge. Searching it entirely enumerating all possible partitions seems computationally infeasible. Thus the search is parameterized to prune the search space, for example, parameterized by the number of clusters, or by a distance threshold which locally defines a neighborhood for each point.

- *Validation*: under certain conditions, only local optimality is guaranteed by some algorithms. Besides, other optimal solutions are obtained if we change the algorithm parameter values, most frequently the number of clusters. Therefore clustering is repeated for multiple times and then the best partition is chosen among a pool of candidate partitions. The clustering quality of each partition is assessed by evaluating measures of compactness and separation. This evaluation, as explained later, takes place at the level of the individual points, or clusters. Validation criteria are no different than clustering criteria, in the sense of involving measures of compactness and separation, but a validation criterion tend to be more thorough since it is meant to be evaluated rather than optimized.

- *Interpretation*: the partition filtered by validation is then interpreted in the context of the end goal. In the process, the cluster allocation of the points is examined to see if it really conveys similarity and dissimilarity. For example, in a topic-driven clustering application, the partition is examined to verify that documents in the same cluster are similar in the sense of representing the same topic. In case of unsatisfactory results, the analysis is repeated possibly on different input.

Next we review the widely used algorithm of k-means which appears among the top 10 algorithms in data mining [81]. The algorithm conveniently serves the purpose of introducing basic concepts associated with clustering. Its fuzzy variant, fuzzy c-means, also serves as a good example of fuzzy clustering algorithms. The popularity of k-means is due to its intuitive clustering criterion, ease of implementation and a linear complexity which makes it computationally attractive.

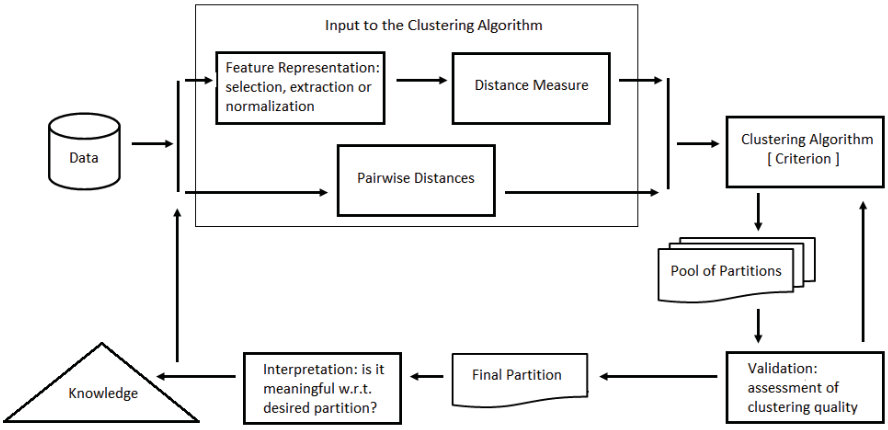
Input to the Clustering Algorithm
Data
Feature Representation: selection, extraction or normalization
Distance Measure
Pairwise Distances
Clustering Algorithm [Criterion]
Pool of Partitions
Validation: assessment of clustering quality
Final Partition
Interpretation: is it meaningful w.r.t. desired partition?
Knowledge

Figure 2.1: The outline of the clustering process.

2.1 The k-Means Algorithm

The term k-means was first used to refer to a problem rather than an algorithm, coined by MacQueen [52]. The following is a description of the k-means problem [54]: given a set of n points (p-dimensional vectors)

$$X = \{x_j \mid x_j = [x_{j1}, \dots, x_{jp}]^T \in \Re^p\}_{j=1}^n$$

find a set of k points, called centers

$$V = \{v_i \mid v_i = [v_{i1}, \dots v_{ip}]^T \in \Re^p\}_{i=1}^k$$

which minimizes

$$J(V) = \sum_{j=1}^{n} d^2(V, x_j) \tag{2.1}$$

where $d(V, x_j)$ denotes the distance between x_j and the nearest center in V. The 2-means problem is NP-hard in the Euclidean space [2, 21]. For a general number of clusters but in the Euclidean plane ($p = 2$), k-means is also NP-hard [54]. The given proofs of the problem hardness assume d the Euclidean distance.

Reformulated in the context of a partition, the problem reads as follows: partition

X into k subsets $U = \{u_i \mid u_i \subset X\}_{i=1}^{k}$ by finding a solution for the following [72]

$$\begin{aligned}
&\text{minimize} && J(\mathrm{U}, V) = \sum_{i=1}^{k}\sum_{j=1}^{n} u_{ij} d^2(v_i, x_j) && && \text{(2.2a)}\\
&\text{subject to} && \sum_{i=1}^{k} u_{ij} = 1, && 1 \le j \le n && \text{(2.2b)}\\
& && 0 < \sum_{j=1}^{n} u_{ij} < n, && 1 \le i \le k && \text{(2.2c)}\\
& && u_{ij} \in \{0, 1\}, && 1 \le i \le k,\ 1 \le j \le n && \text{(2.2d)}
\end{aligned}$$

The partition is characterized by the matrix $\mathrm{U} = [u_{ij}] \in \Re^{kn}$; each u_{ij} is interpreted as the value of an indicator function, $u_i : X \to \{0, 1\}$, associated with the i^{th} cluster u_i. V denotes the set of cluster centers. The criterion, J, a sum of squared errors formula, is a measure of compactness; the smaller the sum is the tighter the points are to their cluster center. For the same number of clusters, minimizing J is equivalent to maximizing a measure of separation [94], given by

$$\grave{J}(V) = \sum_{i=1}^{k} n_i d^2(v_i, v) \tag{2.3}$$

where

$$n_i = \sum_{j=1}^{n} u_{ij} \tag{2.4}$$

$$v = \frac{1}{n}\sum_{j=1}^{n} x_j \tag{2.5}$$

Equation (2.2a) can be also rewritten explicitly in terms of the pairwise distances as [21]:

$$\ddot{J}(\mathrm{U}) = \sum_{i=1}^{k} \frac{1}{2n_i} \sum_{r=1}^{n}\sum_{s=1}^{n} u_{ir} u_{is} d^2(x_r, x_s) \tag{2.6}$$

So minimizing the distances between points and their cluster centers is equivalent to minimizing average cluster diameters. The average cluster diameter is the average distance between points within the same cluster.

Lloyd's algorithm [51] is an iterative heuristic method that became the standard for solving the k-means problem, hence known by the k-means algorithm. It iterates the application of the two rules below to minimize (2.2a):

$$v_i = \frac{\sum_{j=1}^{n} u_{ij} x_j}{\sum_{j=1}^{n} u_{ij}} \tag{2.7}$$

$$u_{ij} = \begin{cases} 1, & i = \operatorname{argmin}_{1 \le h \le k} d(v_h, x_j) \\ 0, & \text{otherwise} \end{cases} \tag{2.8}$$

The pseudo-code is given in Algorithm 2.1. The algorithm halts if there is neither significant improvement in J, nor change to the cluster centers. It also stops when the maximum number of iterations is exceeded, in case the algorithm does not converge in a reasonable time. A proof of the finite convergence of k-means type algorithms for any given metric is given in [72]. It is shown that the algorithm converges to partial optimal solutions[5] which are Kuhn-Tucker points under certain conditions.

Algorithm 2.1: k-MEANS

```
Parameters: $k$ (number of clusters)
Input     : $X$ (set of $n$ $p$-dimensional vectors)
Output    : U (the indicator matrix a partition of $k$ clusters)
            $V$ (the associated set of cluster centers)
$V \leftarrow V_0$                                        // random $V_0$

while the stopping conditon is not satisfied do
    Update U: $\forall i, \forall j$ compute $u_{ij}$ using (2.8).
    Update $V$: $\forall i$ compute $v_i$ using (2.7).
return U, $V$
```

We conclude this section with a couple of examples to illustrate the clustering

[5] A point (U^*, V^*) is a partial optimal solution if $\forall V,\ J(\mathrm{U}^*, V^*) \le J(\mathrm{U}^*, V)$ and $\forall \mathrm{U},\ J(\mathrm{U}^*, V^*) \le J(\mathrm{U}, V^*)$.

by k-means of a simple data set and the iris data set. A data set of 10 points and its k-means partition are given in Figure 2.2. The indicator matrix, U, of the partition is given in Table 2.1. The output of k-means on UCI iris data is shown in Figure 2.3. Note the following in the two examples:

- Although x_8 is assigned to u_2, the cluster depicted in circles in Figure 2.2, it seems closer to x_2 and x_4 than to the points x_6 and x_7. Nevertheless, such cluster assignment of x_8 achieves better overall separation and compactness; it is better to separate x_8 from x_3 and x_5 than to group them together in the same cluster. Quantified by a distance measure, separation and compactness need to be maximized and minimized, respectively.
- By unequivocally grouping a point in an overlapping area in one cluster, its apparent similarity to other points not in the cluster is lost. Thus, such clustering overlooks some of the relations between the data instead of representing them, a sacrifice offered for the goodness of the overall clustering. In particular, the similarities between x_8, x_2 and x_4 are not totally represented by the partition.
- The overlapping between iris versicolor and iris virginica flowers in Figure 2.3.

Table 2.1: The characteristic matrix of the k-means partition in Figure 2.2b.

U	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
u_1	1	1	1	1	1	0	0	0	0	0
u_2	0	0	0	0	0	1	1	1	1	1

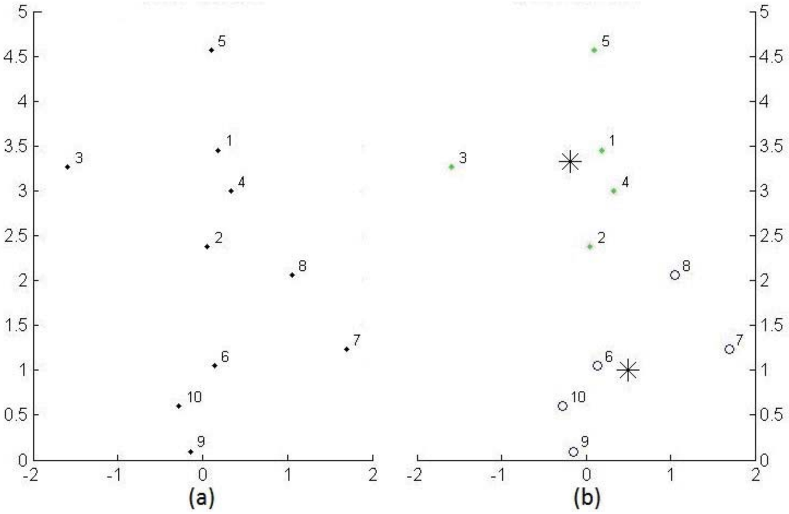

Figure 2.2: The k-means partition of a data set of 10 points: (a) the data set and (b) its partition into two clusters.

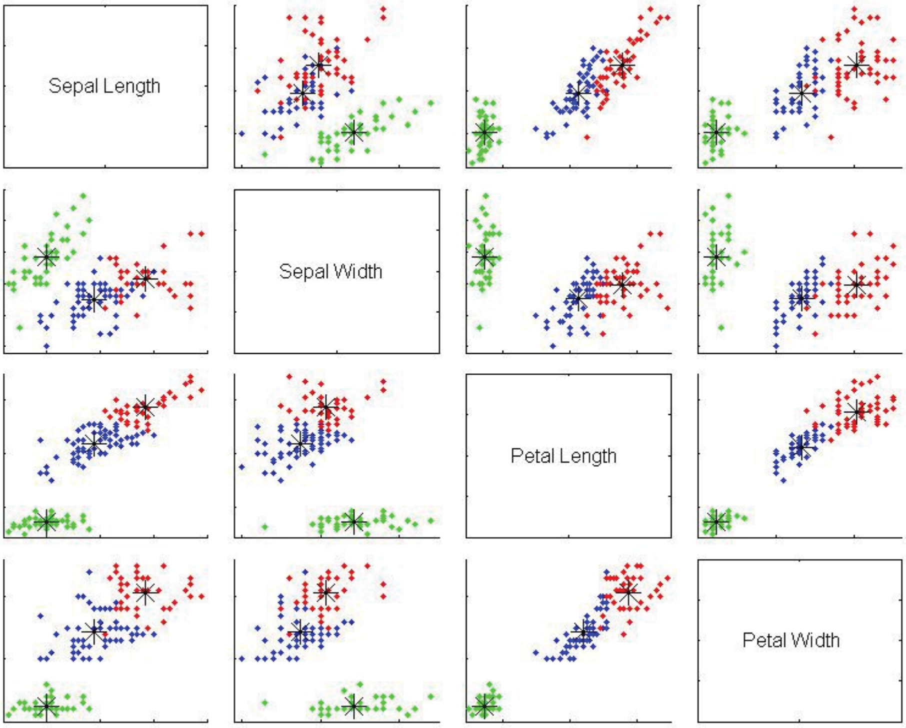

Figure 2.3: Iris data and its k-means partition into 3 clusters: the upper triangular plots show iris species, setosa in green, versicolor in blue and virginica in red. The lower triangular plots show the k-means clusters. Species and cluster centers are shown in black asterisks.

2.2 The Fuzzy c-Means Algorithm

The exclusive cluster assignment carried by k-means is restrictive for some applications. For example, a page returned by a search engine might fit many categories, or an investigator who might belong to multiple communities in a co-authorship network. Even in applications that assume disjoint clusters, there is the uncertainty about a point membership in a particular cluster, probably an outlier. For example, in the segmentation of MRI images into different tissues, the images always present overlapping gray-scale intensities for different tissues due to noise and blur in image acquisition [85]. In regards to a subset of pixels, a distance computed on image intensity profiles fails in reasoning about their cluster (tissue) assignment. Incorporating the spatial characteristics of the images in the distance measure does not resolve the situation. To cope with these elements of structural uncertainty i.e. the uncertainties in the relations between the data, a new framework is needed.

In a seminal paper which introduced fuzzy sets, Lotfi Zadeh generalized the notion of set by extending the concept of set membership [88]. The membership of a point in a set is allowed to assume any value in many, possibly a continuum of, grades of membership. More specifically, a fuzzy set is characterized by a membership function which takes values in the interval $[0, 1]$. To characterize an ordinary set, the membership function reduces to the two-valued indicator function, assuming the two values 0 and 1. Ruspini was among the first to suggest using the concept of fuzzy sets in clustering [69]. Dunn subsequently [28] proposed a variant of the ISODATA algorithm [4], called fuzzy ISODATA, to solve the following relaxed version of the k-means problem

$$\text{minimize} \qquad J_2(\mathrm{U}, V) = \sum_{i=1}^{k} \sum_{j=1}^{n} u_{ij}^2 d^2(v_i, x_j) \tag{2.9a}$$

$$\text{subject to} \qquad \sum_{i=1}^{k} u_{ij} = 1, \qquad 1 \leq j \leq n \tag{2.9b}$$

$$u_{ij} \in [0, 1], \qquad 1 \leq i \leq k,\ 1 \leq j \leq n \tag{2.9c}$$

Fuzzy ISODATA, employs the necessary conditions, for any local minimum of J_2, as update rules in an alternating optimization process. The resulting fuzzy partitions are characterized by the membership matrix $\mathrm{U} = [u_{ij}]$. Bezdek generalized J_2 to an infinite family of objective functions $\{J_m(\mathrm{U}, V) \mid 1 \leq m < \infty\}$ [13]. He obtained update rules similar to Dunn's but in the parameter m, used in fuzzy ISODATA to iteratively optimize J_m for any $m > 1$. The new parameter m introduced into the problem and its role warrant further discussion, which follows soon. Later in his book [10], Bezdek called the algorithm fuzzy c-means (FCM). The parameter c replaces k in denoting the number of clusters. Below is the relaxed generalized problem of k-means [11]:

$$\text{minimize} \quad J_m(\mathrm{U}, V) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d^2(v_i, x_j) \tag{2.10a}$$

$$\text{subject to} \quad \sum_{i=1}^{c} u_{ij} = 1, \qquad 1 \leq j \leq n \tag{2.10b}$$

$$0 < \sum_{j=1}^{n} u_{ij} < n, \qquad 1 \leq i \leq c \tag{2.10c}$$

$$u_{ij} \in [0, 1], \qquad 1 \leq i \leq c,\ 1 \leq j \leq n \tag{2.10d}$$

The necessary conditions for any optimal solution are:

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m} \tag{2.11}$$

$$u_{ij} = \left(\sum_{h=1}^{c} \left(\frac{d(v_i, x_j)}{d(v_h, x_j)} \right)^{\frac{2}{m-1}} \right)^{-1} \tag{2.12}$$

Algorithm 2.2 gives the pseudo-code of fuzzy c-means. A proper matrix norm on U can be used in the assessment of insignificant change in cluster assignments, as an additional stopping condition. Fuzzy c-means converges to local minima or saddle points of (2.10a) [12]. As in k-means, the dual of the fuzzy c-means problem in terms of the pairwise distances is formulated in [40]. But more interestingly, a dual of the algorithm itself is delivered that solves the dual problem. The criterion function, equivalent to (2.10a), becomes

$$\ddot{J}_m(\mathrm{U}) = \sum_{i=1}^{c} \frac{1}{2n_i} \sum_{r=1}^{n} \sum_{s=1}^{n} u_{ir}^m u_{is}^m d^2(x_r, x_s) \tag{2.13}$$

The fuzzy c-means algorithm and its dual, both, converge to the exact same partitions. The dual algorithm only requires the pairwise distances for its operation i.e. fuzzy c-means clustering is applicable in the absence of data feature representation, provided the availability of the pairwise distances.

```
Algorithm 2.2: FUZZY c-MEANS
    Parameters: $c$ (number of clusters)
    Input     : $X$ (set of $n$ $p$-dimensional vectors)
    Output    : $\mathrm{U}$ (the membership matrix a partition of $c$ clusters)
                $V$ (the associated set of cluster centers)
$\mathrm{U} \leftarrow \mathrm{U}_0$        // random $\mathrm{U}_0$ which satisfies the constraints in (2.10)

while the stopping conditon is not satisfied do
    Update $V$: $\forall i$ compute $v_i$ using (2.11).
    Update $\mathrm{U}$: $\forall i, \forall j$ compute $u_{ij}$ using (2.12).
return $\mathrm{U}, V$
```

As mentioned earlier, there are two trivial partitions of no interest to us. On one hand, there is the assignment of all points to one cluster. On the other hand, there is the partition of a data set of n points into n singleton clusters. Now, consider a fuzzy partition into c clusters with each point assuming the membership value of $\frac{1}{c}$ in each cluster. Such partition into c clusters is in the *fuzziest* state possible. By fuzzy set containment[6], one can see that the partition into singleton clusters and the fuzziest partition reduce to a partition of one cluster i.e. no clustering.

[6] As defined by Zadeh [88], a fuzzy set A is said to be a subset of the fuzzy set B, denoted by $A \subset B$, if and only if, $\forall x,\ u_A(x) \leq u_B(x)$.

The Fuzzifier m

Denoting $d_{ij} = d(v_i, x_j)$, and expanding (2.12) obtains for u_{ij}

$$u_{ij} = \frac{1}{\left(\frac{d_{ij}}{d_{1j}}\right)^{\frac{2}{m-1}} + \left(\frac{d_{ij}}{d_{2j}}\right)^{\frac{2}{m-1}} + \ldots + \left(\frac{d_{ij}}{d_{cj}}\right)^{\frac{2}{m-1}}} \tag{2.14}$$

Ratios of center-to-point distances raised to $\frac{2}{m-1}$, constitute the c terms that appear in the denominator. Define the following family of functions

$$\mathcal{F} = \{f_m(x) = x^{\frac{2}{m-1}} \mid 1 < m < \infty\} \tag{2.15}$$

Function plots for selected values of m are shown in Figure 2.4. The next example demonstrates how the fuzzy membership values are shaped by m.

Suppose that we are about to compute the membership values of x_t in 4 clusters. Also, assume that the distance between x_t and v_1, v_2, v_3 and v_4 are respectively: 5, 2, 7 and 10. Since v_2 is the nearest center, x_t its highest grade of membership should be to cluster u_2. Using (2.14), u_{2t} and u_{3t}, were computed for different values of m. Their computations are given in Table 2.2 and Table 2.3, in the detail of the denominator terms that appear in (2.14). In the tables, the values obtained using $m = 3$ are distinguished to facilitate a discussion that follows soon. When $m = 1.01$, u_{2t} and u_{3t} evaluate respectively to the crisp values of 1 and 0. As m increases, the computed values of u_{2t} and u_{3t} approach 0.25. This is observed in the last columns of Table 2.2 and Table 2.3.

Indeed, such change in the membership values driven by m, is formally stated by the limiting properties of (2.11) and (2.12) given in [58]. More specifically, as $m \to 1$, equations (2.11) and (2.12) become respectively (2.7) and (2.8) in page 13. Thus, by setting $m \approx 1$, the fuzzy c-means algorithm carries a k-means clustering. Conversely, as $m \to \infty$, the cluster centers converge to the grand mean of the entire data set, which becomes the only optimal solution for v_i, therefore (2.11) becomes (2.5) in page 12. Since the formed clusters share the same center; (2.12) computes the membership values to $\frac{1}{c}$. Such a partition is in its fuzziest state, U = $[\frac{1}{c}]$. Since, larger values of m have a blurring effect, by which the membership values become

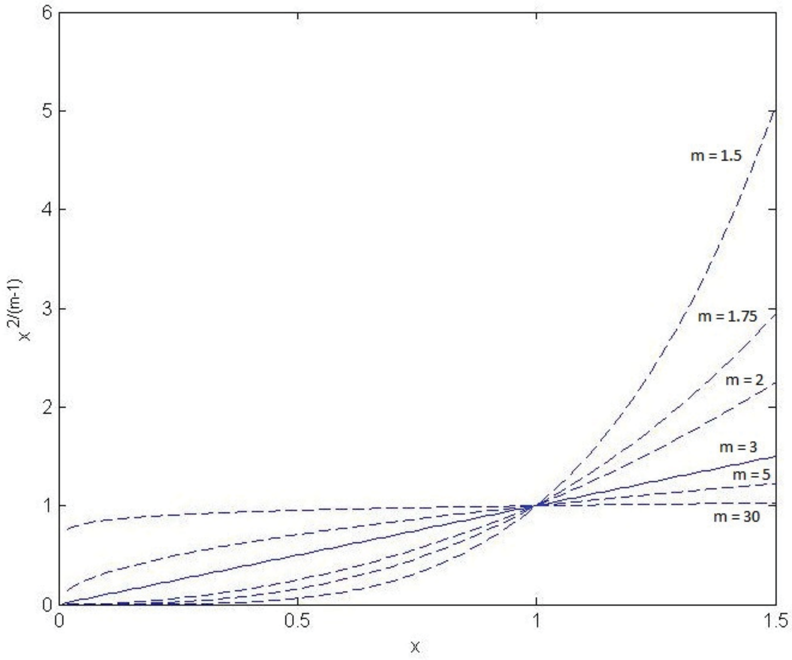

Figure 2.4: The blur effect of the fuzzifier: plots of $f_m(x)$ defined in (2.15), for selected values of m.

Table 2.2: The membership of x_t in u_2 and the terms involved in the computation for different values of m.

m	$T_{2i} = (d_{2t}/d_{it})^{2/(m-1)}$				$u_{2t} = 1/\sum_i T_{2i}$
1.01	0.000	1.000	0.000	0.000	**1.000**
1.5	0.026	1.000	0.007	0.002	**0.967**
1.75	0.087	1.000	0.035	0.014	**0.880**
2	0.160	1.000	0.082	0.040	**0.780**
3	0.400	1.000	0.286	0.200	**0.530**
5	0.632	1.000	0.535	0.447	**0.383**
30	0.939	1.000	0.917	0.895	**0.267**
50	0.963	1.000	0.950	0.936	**0.260**
70	0.974	1.000	0.964	0.954	**0.257**

Table 2.3: The membership of x_t in u_3 and the terms involved in the computation for different values of m.

m	$T_{3i} = (d_{3t}/d_{it})^{2/(m-1)}$				$u_{3t} = 1/\sum_i T_{3i}$
1.01	1.6812e+029	6.5104e+108	1.000	0.000	**0.000**
1.5	3.842	150.063	1.000	0.240	**0.006**
1.75	2.453	28.239	1.000	0.386	**0.031**
2	1.960	12.250	1.000	0.490	**0.064**
3	1.400	3.500	1.000	0.700	**0.152**
5	1.183	1.871	1.000	0.837	**0.204**
30	1.023	1.090	1.000	0.976	**0.245**
50	1.014	1.052	1.000	0.986	**0.247**
70	1.010	1.037	1.000	0.990	**0.248**

fuzzier; m is known sometimes as the *fuzzifier*. Other than the limit analysis given in [58], there is no theoretically justified rule about setting m. There is one exception, the two rules given in [87] based on stability analysis of fuzzy c-means. The rules, when they apply, only impose upper bounds on m. In practice, the values of 1.5, 2 and 3 are frequently used.

Informally speaking, it is recommended to use $m = 3$. With respect to x_t, the membership values $\{u_{it}\}_{i=1}^{c}$ are computed in terms of ratios of the distances from the same set $\{d_{it}\}_{i=1}^{c}$. The linearity of $f_3(x) = x$, results in no distortion to the distance ratios as u_{it} is being computed. For this reason, the values that correspond to $m = 3$ are distinguished in Table 2.2 and Table 2.3 as base values. The role of m is then simply described by the deviation of the distance ratios, and the resultant membership values, from the base values. As m decreases from 3, $f_m(x)$ tends to amplify those ratios greater than 1 and to suppress ratios less than 1, to the extent of nullifying them at $m \approx 1$. Accordingly, the membership of x_t in u_i evaluates to 1, if v_i is the nearest center to x_t, and 0 otherwise. In contrast, as m increases unboundedly, $f_m(x)$ brings the ratios closer and closer to 1, to the point of evaluating u_{it} to $\frac{1}{c}$. Thus, x_t assumes a membership of $\frac{1}{c}$ in every cluster regardless the distance between x_t and the cluster centers.

In summary, the resulting partitions have the highest fidelity in representing the relations between the data points, under the uncertainty inherent in the associated distances, if the fuzzifier is set to 3. The algorithm becomes more or less aware of the fuzziness implied by the distances between x_t and the cluster centers, as m is set to values in the interval $(1, 3)$. Accordingly, the generated membership values become fuzzier or crispier. Unnecessary fuzziness, not related to the distances, is introduced to the computation of the membership values when $m > 3$. The larger the value of m is, the fuzzier the algorithm becomes.

2.3 Defuzzification and the Visualization of Fuzzy Partitions

In the context of fuzzy partitions of 2- or 3-dimensional data sets, each point is a member of each cluster, to some degree of membership. Visualizing such fuzzy partitions is not as straightforward as in the visualization of crisp partitions. One might suggest marking each point as a member of the cluster in which it attains its highest membership value; in the consequence, the point membership in the remaining clusters is ignored. This describes a procedure to the defuzzification of fuzzy partitions. More specifically, the fuzzy partition $\mathrm{U} = [u_{ij}]$ is reduced to the crisp partition $\acute{\mathrm{U}} = [\acute{u}_{ij}]$ by computing

$$\acute{u}_{ij} = \begin{cases} 1, & i = \operatorname{argmax}_{1 \leq h \leq c} u_{hj} \\ 0, & \text{otherwise} \end{cases} \tag{2.16}$$

The fuzzy partition is visualized on the basis of the membership values in $\acute{\mathrm{U}}$. Alternatively, it can be visualized over several intensity plots, where each plot is dedicated to one of the fuzzy clusters. To illustrate these ideas to the visualization of fuzzy partitions, consider the data set in Figure 2.5a. A fuzzy c-means partition of the data set into three clusters was obtained using $m = 3$. A plot of the partition generated using the first approach, of defuzzification, is shown in Figure 2.5b. Although, in the figure, $\{u_1, u_2, u_3\}$ is used in denoting the fuzzy clusters, it is more accurate to use $\{\acute{u}_1, \acute{u}_2, \acute{u}_3\}$. The alternative intensity plots of the same partition, occupy the entire left column of Figure 2.6. To demonstrate how the fuzzifier affects the membership values, therefore the intensities, another partition into the same number of

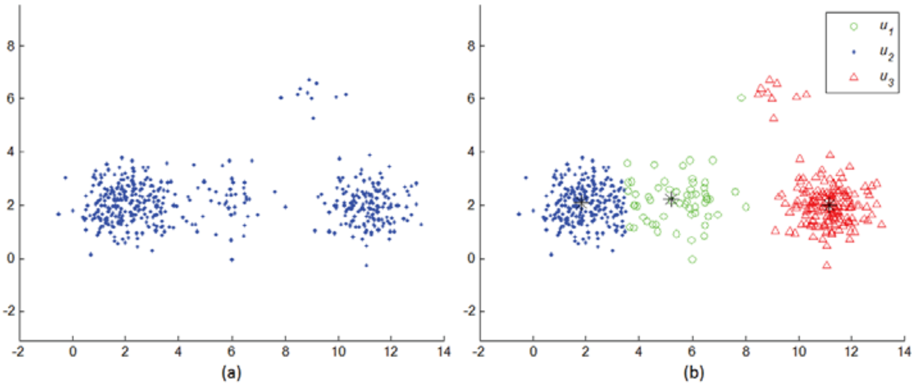

Figure 2.5: A data set shown in (a) and the plot of its (b) fuzzy c-means partitions into 3 clusters generated by means of defuzzification.

clusters was obtained using $m = 2$, shown by the intensity plots on the right side of the same figure.

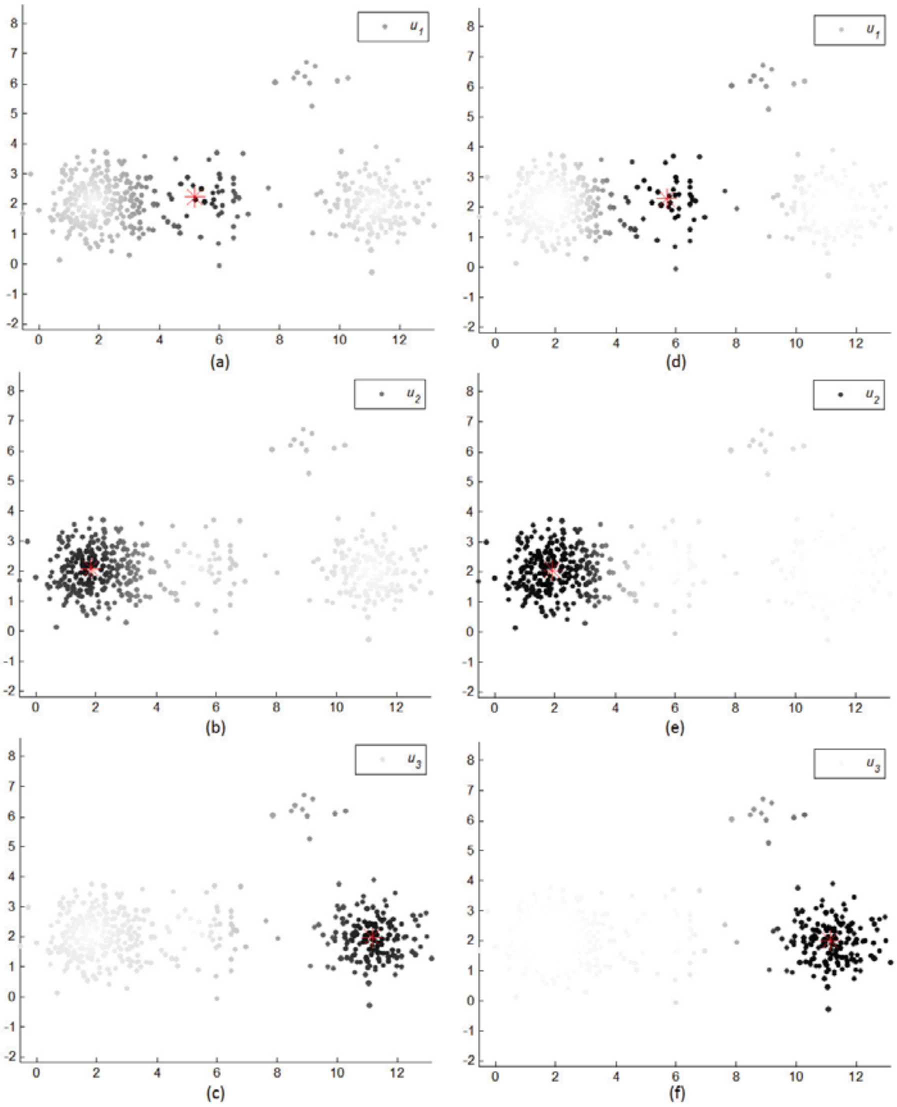

Figure 2.6: The intensity plots, per cluster, for two partitions of the data set in Figure 2.5. The clusters in (a), (b) and (c) were obtained using $m = 3$. The clusters in (d), (e) and (f) were obtained using $m = 2$. The cluster centers are shown in red asterisks. Note the intensity level of points farther from the cluster centers.

2.4 The Evaluation of Clustering Results

Any clustering algorithm is capable of partitioning an input data set into clusters. An arbitrary cluster assignment of the data points is also a partition of the data set. The available possibilities in clustering require reliable measures able to distinguish the partition best in clustering quality, among many others. Different partitioning results are obtained by using:

- Crisp or fuzzy clustering approaches.
- The same approach, but algorithms with different clustering criteria.
- The same algorithm but varying its parameter values.

It is even possible for one algorithm to produce different outputs in the same exact settings. For instance, k-means may converge to different, locally optimal, partitions of the same data into the same number of clusters due to variation in the initialization step. Cluster validity addresses the evaluation of clustering results. The next chapter reviews some of the popular cluster validity measures.

CHAPTER 3
CLUSTER VALIDITY

It seems reasonable, before turning into any validity measure, to investigate which clustering results are tractable for validation by the validity measures.

3.1 Validity for Distance-Based Clustering

The literature on cluster validity has validation measures as many as clustering criteria there used in clustering. The goal of validation is to evaluate the clustering quality of given partitions by incorporating measures of compactness and separation. However by validation, some refer only to the problem of determining the number of clusters, viewed as the fundamental problem of cluster validity [84]. Developing a validity measure around this goal is subjective since the common practice is to use two dimensional data sets to guide the measure development. Consider, for example, the data set shown in Figure 3.1 [60], deciding on the number of clusters just by visual inspection is controversial. Shall we trust a measure that picks the partition in Figure 3.1d over another which picks the partition in Figure 3.1b? Assume that there is an agreement on the number of clusters in the data sets used in the measure development. And, the measure successfully detects these numbers. Is it useful in comparing partitions of the same data sets but into any number of clusters, relative to clustering quality? Moreover in regards to data sets that can

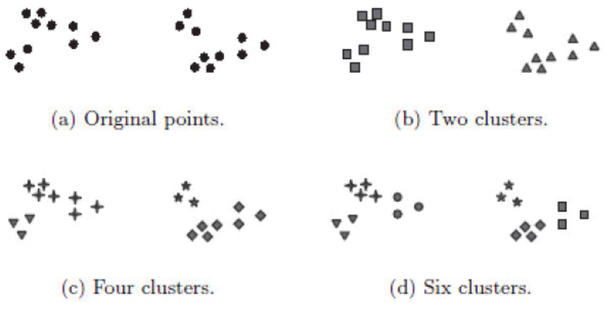
(a) Original points.
(b) Two clusters.
(c) Four clusters.
(d) Six clusters.

Figure 3.1: The number of clusters: (a) A set of points and three possible partitions into (b) two clusters, (c) four clusters and (d) six clusters.

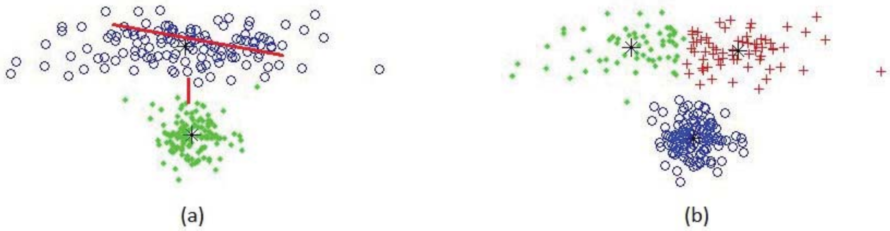

Figure 3.2: Cluster shapes versus pairwise distances: two k-means partitions of a set into (a) 2 and (b) 3 clusters. The red lines in (a) help in visualizing the relations between the points.

be visualized, there is the tendency of conceptualizing the clusters on the basis of their shape rather than the distance between the data points. Consider the two k-means partitions of a data set into two and three clusters, shown respectively in Figure 3.2a and Figure 3.2b. The data set consists of 250 points sampled from two bivariate Gaussians, 125 points from each distribution. It is tempting to assume that the data has two clusters by mere visual inspection. However, suppose the data is based on customer purchase history, is the partition in Figure 3.2a more useful in recommending products to customers than the partition in Figure 3.2b? Notice the similarities and dissimilarities suggested by the red lines in Figure 3.2a. If the partition in Figure 3.2a is the preferred one, then a model-based clustering approach [31] rather than distance-based might be appropriate for the target application, for instance, the expectation-maximization (EM) algorithm [23]. The EM algorithm carries a maximum likelihood estimation of the parameters of a mixture of densities. Accordingly, two points are assigned to the same cluster, therefore perceived similar, if their highest probabilities are by the associated mixture component.

In another example where the clusters are not distance-based rather shape-based: the Euclidean distance is obviously inconsistent with the ‘two ring’ formation of the data points in Figure 3.3a; note the similarities and dissimilarities between the points emphasized by the red lines in the same figure. Thus, k-means using the Euclidean distance, on data 2-dimesnional representation, does not produce a partition into clusters that align with the two rings, as shown in Figure 3.3b. But the Euclidean distance works well for k-means if applied on the right feature representation. Figure 3.3c shows the k-means partition of the data using the Euclidean distance, but applied on a 1-dimensional feature representation extracted by means of an RBF function. A distance defined on a vector of the RBF distances perfectly establishes the similarities and dissimilarities between the points, relevant to the two ring formation. It is possible to achieve a clustering of the points into the two rings directly on their 2-dimensional representation, but using the single linkage algorithm [34]. Single linkage is a member of a family of agglomerative algorithms in which clusters separated by the shortest link are merged. The definition of 'link' is what characterizes each member algorithm. Single linkage defines a link between two clusters as the shortest distance between point pairs, one in each cluster. Clearly, any apparent dissimilarity between pairs of points, one in each cluster, is not taken into account during the merges. The relations between the data points are not considered in a full scale by the algorithm. It is possible for two points that are farther apart to belong to the same cluster by a series of intermediate cluster merges, a phenomenon referred to as *chaining* [24]. Although it is perceived as a disadvantage, chaining served the algorithm in obtaining the two ring partition; see Figure 3.3d.

The rationale of identifying clusters of 'arbitrary shape' renders any distance-based similarity measure obsolete, in a global sense, for the task. Two points of a large distance from each other are still perceived similar for being members of the same shape-based cluster. Some algorithms allow untraditional measures of similarity, for instance, the Rock clustering algorithm [35][7]. Rock utilizes a distance measure to establish neighborhood relations between the data points. The similarity, called a *link*, is defined then by the number of common neighbors. Thus, the algorithm employs two similarity functions, at two different levels:

- *Locally*: neighbor(x_r, x_s) that determines if the points x_r and x_s are considered neighbors, denoting a similarity between the points.

- *Globally*: link(w, z) that gives the number of common neighbors, defined for

[7] The unsuitability of the Euclidean distance used in k-means to cluster a set of four points (transactions as binary vectors over a set of items), in Example 1 in [35], can be argued. It might seems a mistake to merge [100100] and [000001] since they do not have items in common but what they have in common is the absence of items 2, 3 and 5. Nevertheless, the target applications benefit from defining the similarity measure on the basis of common items.

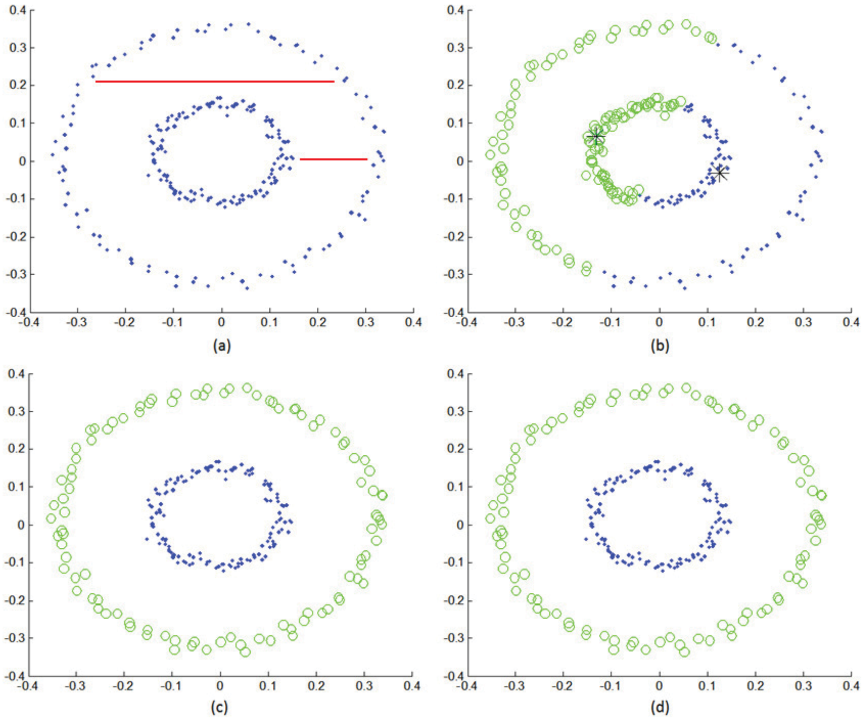

Figure 3.3: A data set of points taking the form of two concentric rings: (a) the data set and its (b) k-means partition using the same 2-dimesnional representation, (c) k-means partition using 1-dimesnional feature representation extracted by means of an RBF function and (d) single linkage partition using the 2-dimensional representation. The red lines in (a) help in visualizing the similarities and dissimilarities between the points.

points and clusters as well. This function is used in merging clusters.

Generally speaking however, it is natural to assume a unified binding criterion when we think about a group of objects. To quote Thomas Jefferson, one of the founding fathers of the United States [43]:

> *It is strangely absurd to suppose that a million of human beings, collected together, are not under the same moral laws which bind each of them separately.*

Nevertheless, arbitrary shape driven clustering, with no unified notion of similarity, has a merit if used with data of a visual nature, as in images and video. Visual inspection is the only reliable means of confirming the quality of clustering results.

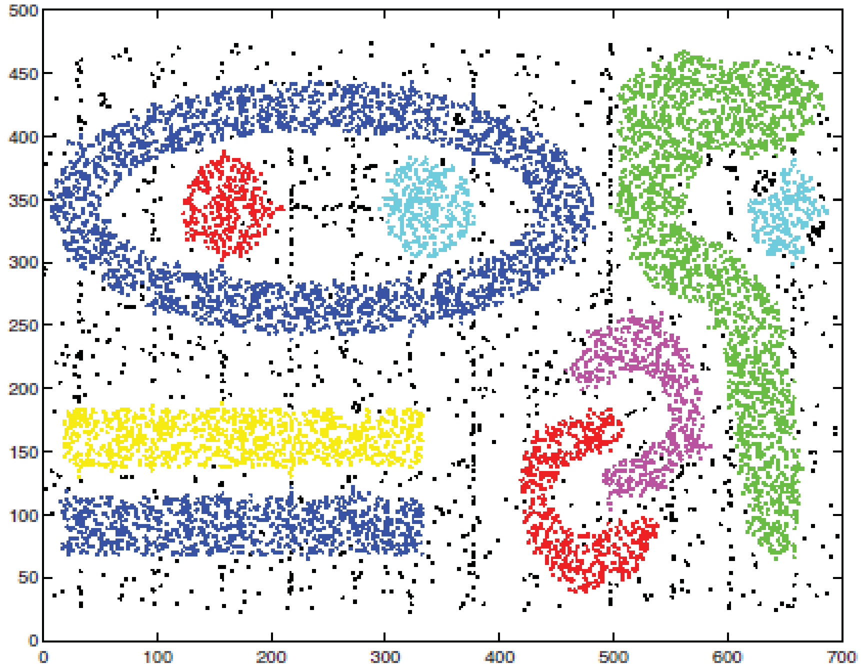

Figure 3.4: Arbitrary shape clusters: DBSCAN clusters, shown in different colors, obtained using $\epsilon = 5.9$ and MinPts = 4 [92].

Consider the partition into the colorfully characterized clusters shown in Figure 3.4. This example is from [92]. The partition was obtained by DBSCAN [30] with $\epsilon = 5.9$ and MinPts = 4. Neighborhood and density are key concepts in DBSCAN defined by the two previous parameters. Despite the success of DBSCAN in detecting the clusters and noisy points in Figure 3.4, it is very sensitive to the selection of parameter values (also true for ROCKS and other algorithms used in the experiments), see [92]. Other than by visual inspection, we do not see how the parameter values could be selected to achieve the clustering in Figure 3.4.

In the development of DBSCAN, data relevant to Earth Science tasks was used as real data benchmark in testing the algorithm, see [30]. It contains raster data, point data, polygon data and directed graph data. The benchmark results are just mere running times with no mention of the actual clustering results; no assessment of clustering quality is possible. No data abstraction, the aspects of the cluster learned from its members, is possible from clustering which assumes arbitrary shape clusters. Abstraction and generalization are the basic two operations in most schemes relevant to the classifications of patterns into a finite number of categories, according to Bellman, Kalaba, and Zadeh [6].To seek a partition as in Figure 3.4, in data of four dimensions and more is similar to ask for the bizarre classification of *Celestial Emporium of Benevolent Knowledge* [17]. In this classification, claimed to be from an ancient Chinese encyclopedia, animals are divided into:

(a) those that belong to the Emperor,

(b) embalmed ones,

(c) those that are trained,

(d) suckling pigs,

(e) mermaids,

(f) fabulous ones,

(g) stray dogs,

(h) those included in the present classification,

(i) those that tremble as if they were mad,

(j) innumerable ones,

(k) those drawn with a very fine camelhair brush,

(l) others,

(m) those that have just broken a flower vase,

(n) and those that from a long way off look like flies.

Here, ‘others’ could represent noise. There is no one unified criterion that helps in differentiating all of those categories, thus arbitrary. Cluster analysis that seeks arbitrary shape clusters lacks the criterion for validation, thus out of this research’s scope. Detecting arbitrary shape clusters, and the efficient tools proposed for the task, are undoubtedly useful in their domain of applications. It is natural to think about the pairwise relations between the data points in the context of a partition. The two widely used clustering algorithms, of k-means and fuzzy c-means, indeed, optimize measures of compactness in terms of the pairwise distances. So, we are concerning ourselves only with data consisting of relations, measured by distances.

3.2 Validity for Crisp and Fuzzy Clustering

Clustering aims at fitting the relations between the data points by a set of clusters that is a partition. The quality of the fit is determined by "the extent that similar objects are placed in the same partition class and dissimilar objects are placed in distinct partition classes" [41]. If the input relations lack the precision to dictate a partition into mutually exclusive clusters then it is convenient to use fuzzy clustering. This imprecision pertains to complexity as stated by Zadeh in his principle of incompatibility [89]:

> *Stated informally, the essence of this principle is that as the complexity of a system increases our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics.*

Incompatibility is also an issue with highly complex data, which applies to relationship data, in clustering problems. The fuzzy approach, as opposed to crisp, is tolerant of imprecision and partial membership. Consequently, it is possible for the transition from one cluster to another, over overlapping areas, to be gradual rather than abrupt. This feature is well illustrated in Figure 3.5 [3] which shows the fuzzy membership functions that map any temperature measurement to one or more of the following clusters: too cold, cold, warm, hot and too hot. Such trapezoidal-shaped membership functions are common in the practice of fuzzy logic. However in regards to clustering quality, one might ask if there is any gain from using such trapezoidal functions as opposed to the rectangular-shaped functions that characterize crisp clusters. This research addresses the two questions of:

- How to reason about the clustering quality of fuzzy partitions?

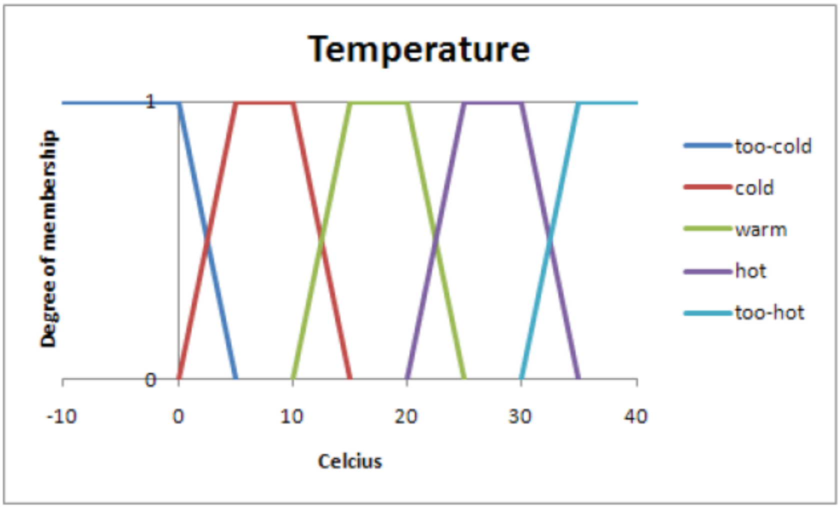

Figure 3.5: A partition of temperature measurements into fuzzy categories.

- Is there an overall gain in clustering quality by using the fuzzy approach as opposed to crisp?

By means of defuzzification, using (2.16), it is possible to apply measures of clustering quality devised for crisp partitions. However, the loss in similarity and dissimilarity information represented by U, defeats the purpose of performing fuzzy clustering. In our review of some measures of clustering quality, mostly applicable to fuzzy partitions, we point out how each measure accounts for compactness and separation in the partitions, if possible.

3.3 Measures of Clustering Quality

The measures that appear next in the review were selected for the reasons below:

- To emphasize a distance-based evaluation of distance-based clustering results.

- To give examples of some of the possible measures of compactness and separation.

- To emphasize the accuracy aspect in selecting proper measures of compactness and separation.

- To highlight a fact that the measures operate at different level of granularity, accordingly becoming rough or sophisticated.
- To demonstrate potential artifacts in the evaluation of clustering results due to those aspects of the measure irrelevant to the task. Any change in the measure values should purely reflect a change in clustering quality.
- The silhouette measure, in specific, appealed to us because it complies with the relational perspective of the clustering task.

3.3.1 The Partition Coefficient

Dunn defined a measure, known as the partition coefficient, of the amount of overlapping i.e. the fuzziness, in partitions [27]. The coefficient is defined by:

$$PC_c(\mathrm{U}) = \mathrm{trace}\left(\frac{\mathrm{U}\mathrm{U}^{\mathrm{T}}}{n}\right) = \frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^2 \tag{3.1}$$

In a theorem stating the extreme values of the measure, which appears to be a strictly convex function on a convex domain of partitions satisfying the constraints in (2.10) in page 17, Bezdek showed that the partition coefficient attains a unique global minimum, and local maxima at the extreme points, that is every crisp partition into c clusters. Note that any arbitrary crisp partition belongs to this set of extreme points. The coefficient has its minimum at the fuzziest possible partition into c clusters that is $\mathrm{U} = [\frac{1}{c}]$. The extreme values are given by the inequality [14]:

$$\tfrac{1}{c} \leq PC_c(\mathrm{U}) \leq 1, \; 2 \leq c \leq n \tag{3.2}$$

Bezdek proposed the use of the coefficient in the selection of the number of clusters c, hence assuming a role as a measure of clustering quality [14]. The proposition is justified by an indirect relationship between a presumed partition into compact separated clusters and the partition coefficient. The property of compact separated cluster was introduced by Dunn [28], readily quantified by his *separation index*. The separation index is defined for a crisp partition of X into $W = \{w_i \mid w_i \subset X\}_{i=1}^{k}$ as:

$$\alpha(k, W) = \frac{\min_{1\leq h\leq k} \min_{1\leq i\neq h\leq k} \ddot{d}(w_h, w_i)}{\max_{1\leq i\leq k} \operatorname{diam}(w_i)} \tag{3.3}$$

where

$$\ddot{d}(w_h, w_i) = \min_{x_r\in w_h,\, x_s\in w_i} d(x_r, x_s) \tag{3.4}$$

$$\operatorname{diam}(w_i) = \max_{x_r, x_s\, \in\, w_i} d(x_r, x_s) \tag{3.5}$$

Then, $\hat{W}$ is a set of compact separated clusters relative to d if and only if $\alpha(k, \hat{W}) > 1$. Despite its name, the separation index incorporates a measure of compactness which appears in the denominator in (3.3). The claim is that an optimal fuzzy partition which minimizes J_m, represented by the pair (U, V), is a good approximation[8] of such $\hat{W}$ only if $PC_c(\mathrm{U})$ is close to 1. Let $\hat{\mathrm{W}}$ be the characteristic matrix of $\hat{W}$. The proposition is based on the two following implications

$$\|\hat{\mathrm{W}} - \mathrm{U}\| \leq \epsilon \implies PC_c(\mathrm{U}) \geq 1 - \frac{\epsilon}{n}(2\sqrt{n} - \epsilon), \qquad 0 \leq \epsilon \leq \infty \tag{3.6}$$

$$PC_c(\mathrm{U}) = z \implies \|\hat{\mathrm{W}} - \mathrm{U}\| \geq \sqrt{n(1 - \sqrt{z})^2}, \qquad \frac{1}{c} \leq z \leq 1 \tag{3.7}$$

In (3.6) if ϵ is very small then the coefficient is very close to 1, especially for large n. According to (3.7) a value of the coefficient very close to $\frac{1}{c}$ implies a poor approximation of $\hat{W}$; the matrix norm is relatively large. However, the existence of $\hat{W}$ is not guaranteed, in the first place. And in order for the coefficient to be a valid measure of clustering quality, one needs to prove that large values of the coefficient implies good partitions, as argued by Trauwaert [75]. Also, one can easily obtain a fuzzy c-means partition U that almost resembles a crisp partition by setting m near 1; in the consequence, $PC_c(\mathrm{U})$ evaluates very close to 1, regardless the clustering quality of U. Trauwaert showed that larger coefficient values are not necessary an indication of better clustering quality. This is evident in the following example.

[8] The proposition is actually stated on the more strict property of *compact well separated* clusters (CWS) that involves the convex hulls of the clusters. Nevertheless, this does not change the argument.

Consider the fuzzy c-means partitions of a data set shown in Figure 3.6. The partitions were obtained using $m = 3$. The data set was sampled from three bivariate Gaussians. The sample sizes of the bottom left corner, the top one, and the bottom right corner are 200, 150 and 100, respectively. According to the partition coefficient, the partition into $c = 2$ in Figure 3.6a, $PC_2 = 0.701$, is better than the partition into $c = 3$ in Figure 3.6b, $PC_3 = 0.678$. But, it is obvious that the latter partition makes a better approximation of a partition into compact separated clusters, than the former one.

In summary, the absence of a distance measure as part of the coefficient means that it does not carry any measurement of compactness and separation. For this reason, the coefficient is dependent on the use of the fuzzy c-means algorithm, to assure that the partition in hands is meaningful, in regards to data relations. The partition coefficient remains an efficient measure of fuzziness, serving well its original purpose.

3.3.2 Xie-Beni Index

Xie and Beni addressed the problem of cluster validity for fuzzy partitions in [82], where they proposed their measure. First, they pointed out the lack of any direct connection of the partition coefficient to the geometric relations between the data. Then, they defined a number of measures of compactness and separation. A measure of compactness, called the total variation, is given by

$$\sigma = \sum_{i=1}^{x} \sigma_i \tag{3.8}$$

where σ_i is the variation of cluster u_i, defined as

$$\sigma_i = \sum_{j=1}^{n} u_{ij}^2 d^2(v_i, x_j) \tag{3.9}$$

The total variation σ is exactly the objective function of the fuzzy c-means algorithm J_2, given by (2.10a) in page 17 for $m = 2$. Another measure of compactness, called

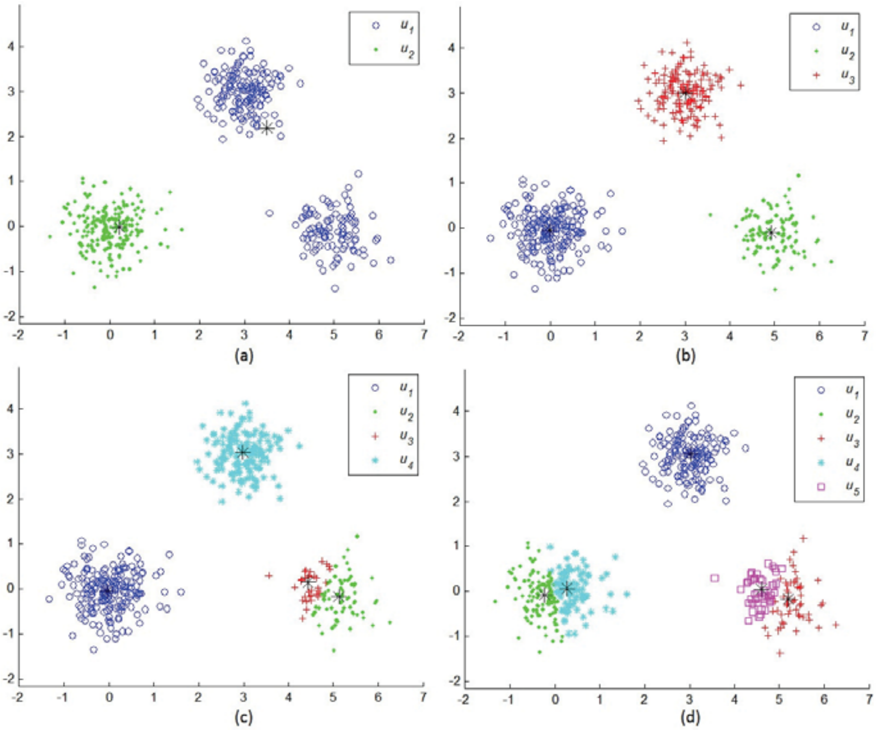

Figure 3.6: The fuzzy c-means partitions of a data set into (a) 2, (b) 3, (c) 4 and (d) 5 clusters. The partitions were obtained using $m = 3$. The plots were generated by means of defuzification.

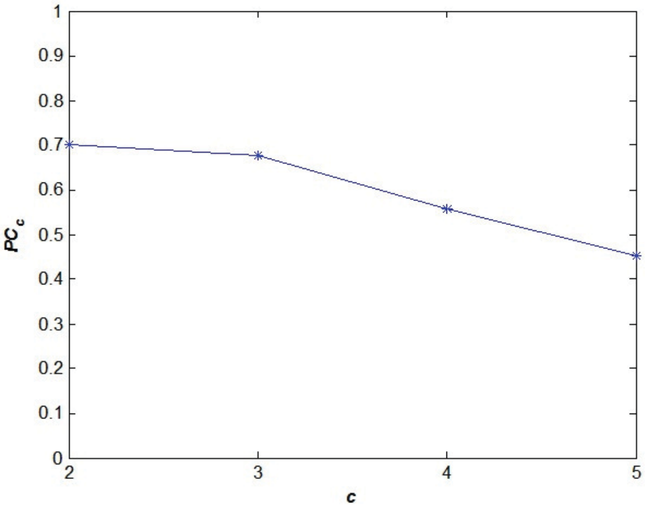

Figure 3.7: The partitions in Figure 3.6, represented by their number of clusters c, versus the partition coefficient.

the compactness of the fuzzy c-partition, is defined as the ratio

$$\pi = \frac{\sigma}{n} \tag{3.10}$$

Similarly, the compactness of cluster u_i is defined as

$$\pi_i = \frac{\sigma_i}{n_i} \tag{3.11}$$

Where n_i is computed by (2.4) in page 12. For measuring the overall separation in a fuzzy partition they use the minimum distance between the cluster centers. The separation of the fuzzy c-partition is therefore defined as

$$d_{\min} = \min_{1 \le i \ne h \le c} d(v_i, v_h) \tag{3.12}$$

The compactness of the c-partition and the separation of the c-partition combine to produce a measure, called the *compactness and separation validity function*, given by

$$XB(X, \mathrm{U}, V) = \frac{\pi}{d_{\min}^2} = \frac{\sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^2 d^2(v_i, x_j)}{n \ \min_{1 \le i \ne h \le c} d^2(v_i, v_h)} \tag{3.13}$$

However, the measure in (3.13) is better known as Xie-Beni index [19, 58]. The authors suggest replacing u_{ij}^2 by u_{ij}^m, in (3.13), in evaluating the output of the fuzzy c-means algorithm, where m is the value used for the fuzzifier in the algorithm. Denote the resulting variant of the index by XB_m. Their only justification is for the total variation to be 'compatible' with the final value of J_m obtained by the algorithm. Unless the final value of J_m is substituted for the numerator in (3.13), hence reducing the computation time for evaluating the index, it is unclear if this benefits the assessment of clustering quality. Also, this recommendation only affects the index measure of compactness, with no effect on its measure of separation. There is a question, then, about the consistency of these measures under such a modification. In a coming example, see Figure 3.12 in page 43, XB and XB_m disagree in their evaluation of the same set of partitions. In particular, according

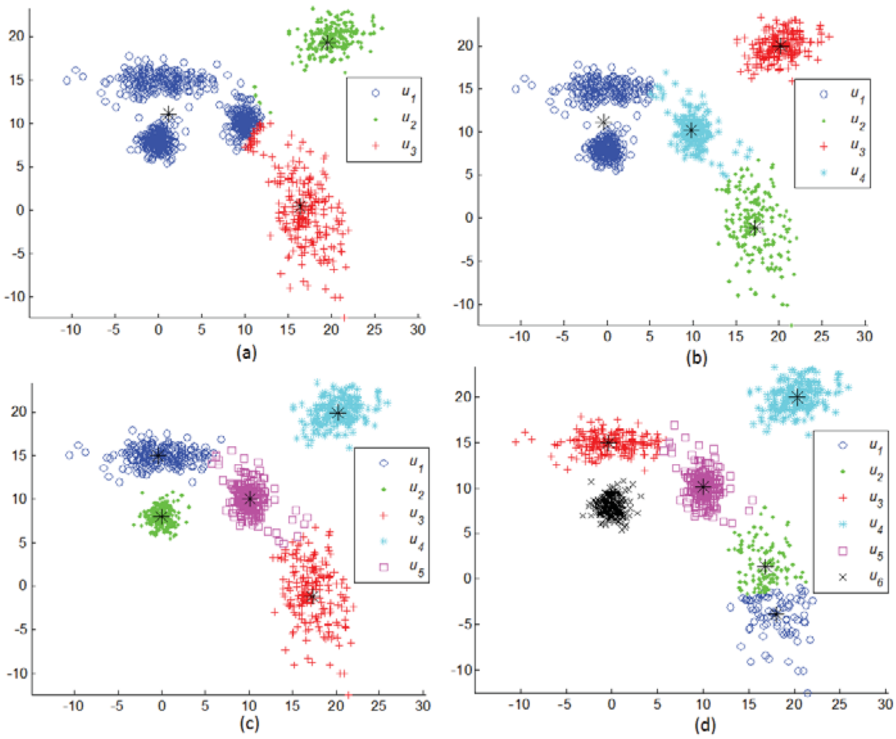

Figure 3.8: The fuzzy c-means partitions of a data set into (a) 3, (b) 4, (c) 5 and (d) 6 clusters. The plots were generated by means of defuzification. The partitions were obtained using $m = 2$.

to XB the partition into $c = 2$ is assumed to have the best clustering quality; it has the minimum XB value. Meanwhile, the partition into $c = 4$ has the minimum XB_m value.

Moreover, the evaluation of the fuzzy c-means partitions shown in Figure 3.8 by XB leaves a question about its ability to detect nuances in clustering quality. It ranks the partition into $c = 3$ as the best in clustering quality. Although, the partitions into $c = 4$ and $c = 5$ seem better in compactness and slightly better in separation. Note that the increase in c from 3, to 4 and to 5 only affected cluster u_1 in Figure 3.8a, causing it to split into 2 and 3 clusters.

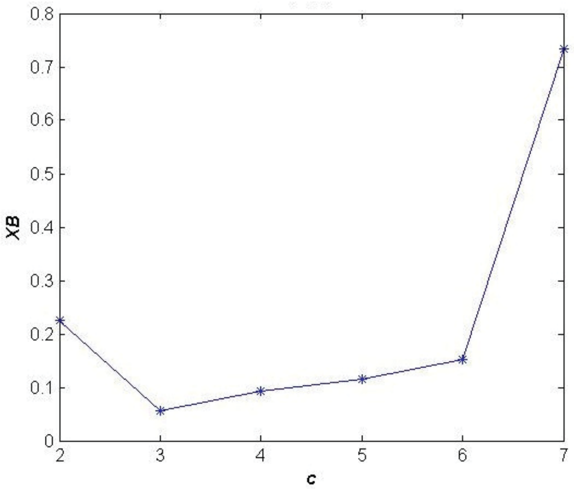

Figure 3.9: The fuzzy c-means partitions of the data set in Figure 3.8 into different number of clusters, represented by their number of clusters c, versus Xie-Beni index. The partitions into 3, 4, 5 and 6 are shown in Figure 3.8.

3.3.3 Pakhira-Bandyopadhyay-Maulik Index

The index was proposed for purpose of measuring the clustering quality of crisp partitions [57], the index is defined as

$$PBM(X, \mathrm{U}, V) = \left(\frac{E_1 D_k}{k E_k}\right)^2 \tag{3.14}$$

where

$$E_k = \sum_{i=1}^{k} e_i \tag{3.15}$$

$$e_i = \sum_{j=1}^{n} u_{ij} d(v_i, v_h) \tag{3.16}$$

$$D_k = \max_{1 \leq i \neq h \leq k} d(v_i, v_h) \tag{3.17}$$

$$E_1 = \sum_{j=1}^{n} d(v, x_j) \tag{3.18}$$

Here, U, V and v denote, respectively, the crisp membership matrix, the set of the k cluster centers and the grand mean of the entire data set, defined in (2.5) in page 12. E_k is a measure of the overall compactness while e_i measures the cluster compactness of u_i. D_k is the separation measure used as part of the index. $\frac{E_1}{k}$ is constant over all partitions of the same data set into the same number of clusters. The authors propose a variant of the index for the evaluation of fuzzy partitions, denoted by the fuzzy *PBM* index, or just *PBMF*, obtained by replacing E_k by J_m as follows

$$PBMF(X, \mathrm{U}, V) = \left(\frac{E_1 D_k}{k J_m}\right)^2 \tag{3.19}$$

To illustrate a serious shortcoming of the index, consider the data set in Figure 3.10. It was sampled from four bivariate Gaussians. The sample sizes of the top left corner, bottom left corner, the bottom right corner and the middle one are respectively 100, 60, 80 and 20. A number of fuzzy c-means partitions of the data set into $c = 2, \ldots, 10$ clusters were obtained using $m = 2$. Only two partitions, into $c = 3$ and $c = 4$, are shown in Figure 3.10. Denote them respectively by P3 and P4. One might argue about 3, 4 or 5 clusters, as plausible partitions of the set, but not more. The fuzzy index, *PBMF*, disagrees with such claim since it ranks partitions into more than 5 clusters, as better in clustering quality; see Figure 3.11b. The effect of allowing more and more clusters in a partition is for smaller and smaller clusters to form, hence an improvement in compactness. It is up to the separation measure to detect the degradation in clustering quality due to the associated bad cluster separation. The penalty factor in *PBMF* that is $\frac{1}{k}$, fails to cover for the shortcoming of using the maximum center-to-center distance as a measure of separation. Xie-Beni index does a better job by employing the minimum center-to-center distance, which covers situations where clusters become more compact but not well separated, for example, clusters u_1 and u_2 in Figure 3.8d. it is more realistic than just penalizing an increase in the number of clusters k with no change to the maximum center-to-center distance. Moreover, Xie-Beni index has more consistency since only squared distances appear in its measures of compactness and separation, as opposed to *PBMF* which uses squared distances in J_m and a non-squared distance in D_k. Even *PBM*, with only non-squared distances, ranks P3 higher than P4, as shown in Figure 3.11a. P4 seems slightly better than P3. Refer to Figure 3.10 to compare between the two partitions. If one proposes

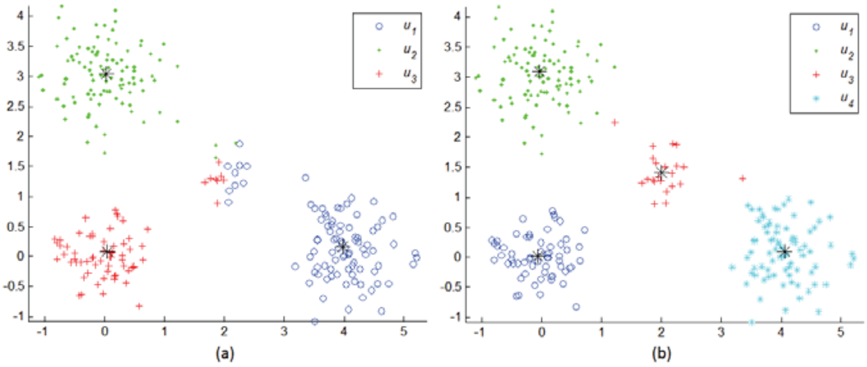

Figure 3.10: The fuzzy c-means partitions of a data set into (a) 3 and (b) 4 clusters. The partitions were obtained using $m = 2$.

two measures of clustering quality that differ in their ranking of the same set of partitions, an explanation is owed to justify the use of each measure. This was encountered with XB and PBM, in the next example.

The data set in Figure 3.10 was partitioned again by fuzzy c-means into $c = 2, \ldots, 10$, but using $m = 5$. The evaluation results by XB, XB_m, PBM and $PBMF$ of the partitions are shown in Figure 3.12. The evaluation by $PBMF$ is the complete opposite of PBM's. Such inconsistency is much less apparent among XB and XB_m. The detailed evaluation results, the index values, are given in Table 3.1. In the table, the values which supposedly determine the partition best in clustering quality are highlighted in bold. Notice the conflicting evaluation among the measures. Also observe each index range of values, for example, $0.148 \leq XB \leq 54.872$. It is understood that values near zero indicate good clustering but how bad is it when $XB = 54.872$? Similarly, how good is it when $PBMF = 18810000$?

Table 3.1: The detailed results that correspond to the plots in Figure 3.12. Values that are supposed to determine the partition best in clustering quality are highlighted in bold.

c	2	3	4	5	6	7	8	9	10
$\boldsymbol{XB}$	**0.148**	0.181	0.221	4.718	5.083	54.872	25.83	20.172	25.59
$\boldsymbol{XB_m}$	0.014	0.004	**0.002**	0.024	0.016	0.101	0.031	0.018	0.016
$\boldsymbol{PBM}$	**5.325**	3.553	2.074	1.414	0.969	0.779	0.589	0.458	0.394
$\boldsymbol{PBMF}(\times 10^7)$	0.000	0.001	0.006	0.023	0.065	0.200	0.459	0.854	**1.881**

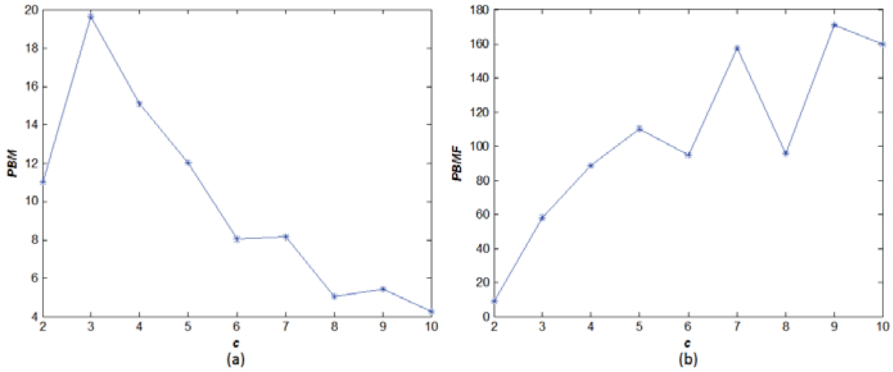

Figure 3.11: The fuzzy c-means partitions of the data set shown in Figure 3.10 into different number of clusters, represented by their number of clusters c, versus (a) *PBM* index and (b) *PBMF* index. The partitions into 3 and 4 clusters are shown in Figure 3.10. The partitions were obtained using $m = 2$.

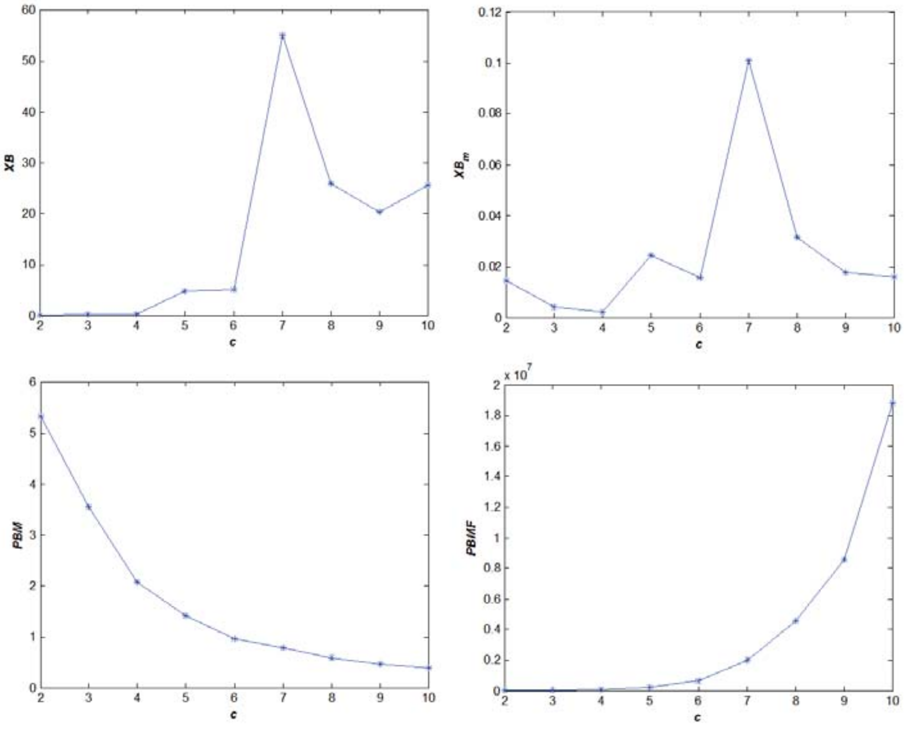

Figure 3.12: The fuzzy c-means partitions of the data set shown in Figure 3.10 into different number of clusters, represented by their number of clusters c, versus (a) *XB* index, (b) XB_m index, (c) *PBM* index and (d) *PBMF* index. The partitions were obtained using $m = 5$.

3.3.4 Average Silhouette Index

Rousseeuw introduced the interesting notion of silhouettes in [68]. The framework that we propose in this thesis, has been inspired by the rationale behind silhouettes, hence the extensive review. Silhouette is an attractive point-wise measure of clustering quality, defined in the context of crisp partitions. In order to illustrate the computation of silhouettes, consider the set of points $X = \{x_j\}_{j=1}^n$ and a crisp partition $U = \{u_i\}_{i=1}^c$, and U $= [u_{ij}]$ is the characterizing matrix. The average distance between x_t and all the points in cluster u_i is computed using

$$\bar{d}(u_i, x_t) = \frac{\sum_{j=1, j \neq t}^n u_{ij} d(x_t, x_j)}{\sum_{j=1, j \neq t}^n u_{ij}} \tag{3.20}$$

The average distance of x_t with respect to each cluster is computed first. Then, two measures of compactness and separation relative to x_t, are defined respectively as

$$a_t = a(x_t) = \bar{d}(u_h, x_t), \text{ where } u_{ht} = 1 \tag{3.21}$$

$$b_t = b(x_t) = \min_i \{\bar{d}(u_i, x_t) \mid u_{it} = 0\} \tag{3.22}$$

In words, the measure of compactness a_t is the average distance between x_t and all the points in cluster u_h, to which x_t has been assigned i.e. $u_{ht} = 1$. The measure of separation b_t is the minimum average distance over the remaining clusters $u_i \in U \setminus \{u_h\}$; one can see that the remaining clusters have $u_{it} = 0$. The cluster at which this minimum is attained is called the *neighbor cluster* of x_t. Since, both of a_t and b_t are average distances, we refer to them respectively as the *compactness distance* and the *separation distance*, for the point x_t. Figure 3.13 helps visualize how these distances relate to the cluster assignment of x_t; imagine moving cluster u_1 closer and closer to x_t and how this affects the preferences to assign x_t to u_2. How a_t and b_t relate to the clustering of x_t, is reasonably explained as follows:

- x_t is *well-clustered*, realized by $b_t \gg a_t$: the neighbor cluster of x_t is not nearly as close as the cluster to which it has been assigned. This indicates a good clustering since the algorithm successfully, and to a great extent, grouped x_t with similar points in one cluster. In Figure 3.13a, u_2 is closer, hence more

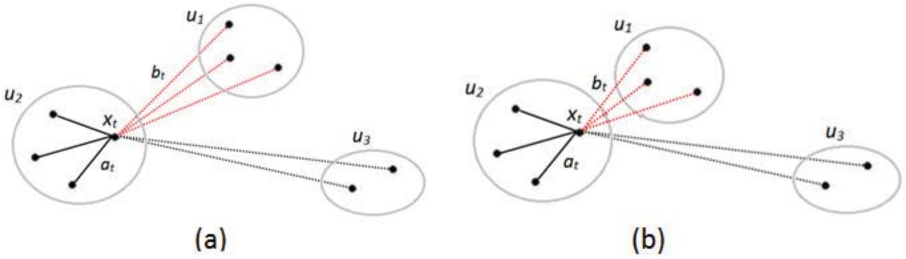

Figure 3.13: The compactness distance and the separation distance with respect to x_t: a_t is the compactness distance computed as an average distance over the solid lines while b_t is the separation distance computed as an average distance over the red dashed lines. The assignment of x_t to cluster u_2 rather than u_1 is more reasonable in (a) with $b_t > a_t$ than in (b) with $b_t \approx a_t$.

similar, to x_t than the neighbor cluster u_1 is.

- The *intermediate* case of $b_t \approx a_t$: x_t is almost of equal distance from its cluster and its neighbor cluster. Thus, it is not clear which cluster x_t should be assigned to. In Figure 3.13b, x_t seems equidistant from the members of u_1 and the members of u_2.

- x_t is *misclustered*, realized by $b_t \ll a_t$: this defies the goal of clustering, recalling its definition. Instead of grouping x_t with the similar points in the same cluster, it has been separated from them, even worse; it has been mischievously grouped with dissimilar points. The neighbor cluster seems the actual best-choice to accommodate x_t; since it is the nearest cluster to x_t, as implied by the inequality.

To facilitate straightforward reasoning using one quantity, the silhouette of x_t, denoted by s_t, combines both of the compactness distance and the separation distance as follows

$$s_t = s(x_t) = \frac{b_t - a_t}{\max\{a_t, b_t\}} \tag{3.23}$$

The silhouette measure is scaled by the maximum distance, and one can easily see that

$$-1 \leq s_t \leq 1 \tag{3.24}$$

It is much easier to interpret the measure values, knowing that a value near 1 implies near perfect clustering of the point. The three cases, listed above, of being well-clustered, misclustered and in an intermediate position between clusters, map respectively to near 1, -1 and 0, on the silhouette scale. Our perception of the similarities and dissimilarities in a given data set is special to the set, rather than to a general reference. A distance of 5, for instance, might imply similarity in one data set and dissimilarity in another. Since it is scaled by the maximum distance; the silhouette measure acts as a general reference of clustering quality, and one can reasonably compare between:

- Two distance measures, in the sense of which results in a better clustering quality i.e. larger silhouette values.
- The clustering of two points in the same data set. For example, based on the silhouette value of a document, in a document clustering application, it might be claimed as a better representative of the subject than another with a smaller silhouette.
- The clustering of two points in two different data sets.

It is worth noting that the silhouette measure can be simply modified to accommodate actual similarities, as opposed to the distances given by the function d in (3.20), see [68]. On the coarser cluster level, the clustering quality of u_i is measured by the average silhouette over its member points, that is

$$S_i = S(u_i) = \frac{\sum_{j=1}^{n} u_{ij} s_j}{\sum_{j=1}^{n} u_{ij}} \tag{3.25}$$

Similarly, the overall clustering quality $\bar{S}$ is defined as the average silhouette over the entire set

$$\bar{S} = \frac{1}{n}\sum_{j=1}^{n} s_j \tag{3.26}$$

Alternatively, one can take the average over $\{S_i\}_{i=1}^{c}$ i.e.

$$\bar{\bar{S}} = \frac{1}{c}\sum_{i=1}^{c} S_i \tag{3.27}$$

Equation (3.26) is more useful if we prefer to attribute the clustering quality to the clusters rather than to the individual points; it defines a cluster-wise measure. This is significant in imbalanced classification problems; if the accuracy measure does not value the minority class, the learned classifier will be biased toward the majority class. Such preferences, to seek good clusters regardless the size, is plausible in clustering. To illustrate how the two alternatives of the overall measure compare with each other, consider the example in Figure 3.14. The figure shows two k-means partitions of the same data set into 3 and 4 clusters. Note the distinguishable small group of 10 points located at the top right corner in both partitions. The different average silhouette values using (3.25), (3.26) and (3.27), are given in Table 3.2. The partition in Figure 3.14a has $\bar{S} > \bar{\bar{S}}$. The 10 points have a relatively small average silhouette value of 0.1. Such small values do more harm to the average value over the clusters as opposed to the entire set whose larger size mediates such harm. In contrast, in Figure 3.14b, the 10 points occupy their own cluster, namely, cluster u_3. The cluster has the largest average silhouette value; $S_3 = 0.781$. Incorporating this relatively large value into $\bar{\bar{S}}$, results in $\bar{\bar{S}} > \bar{S}$.

According to Rousseeuw, the main promoting feature of the silhouette measure is its graphical display. The constructed silhouette plot targets the audience of hierarchical clustering who rely on dendrograms to read and reason about clustering results. The plot resembles a horizontal bar graph grouped by clusters. Its construction proceeds after finding the silhouette of each individual point. Each silhouette is represented by a horizontal bar whose width is determined by the silhouette value

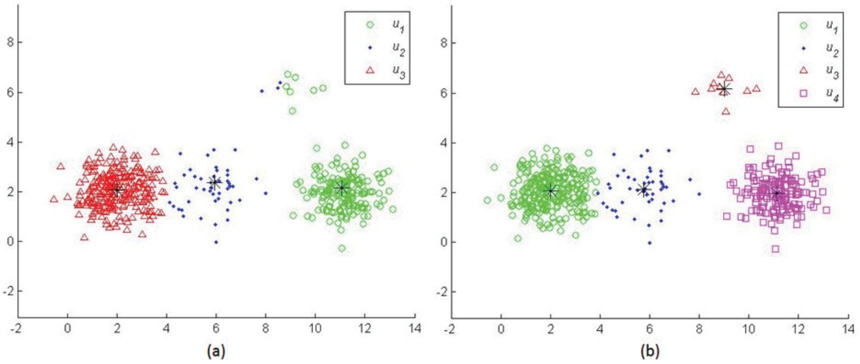

Figure 3.14: Two k-means partitions of the same data set into (a) 3 clusters and (b) 4 clusters. See Table 3.2 for the corresponding silhouette results.

Table 3.2: The different average silhouette measures for the two k-means partitions shown in Figure 3.14a ($c = 3$) and Figure 3.14b ($c = 4$). n_i denotes the size of cluster u_i.

c	$\bar{S}$	$\bar{\bar{S}}$	S_1	n_1	S_2	n_2	S_3	n_3	S_4	n_4
4	0.650	**0.671**	0.618	247	0.567	48	0.781	10	0.719	155
3	**0.648**	0.615	0.702	162	0.5	50	0.642	248		

of the associated point. The silhouette of each cluster is formed by aggregating the silhouettes, the associated bars, of its individual members in a decreasing order of width. Accordingly, the silhouette width, either point-wise or cluster-wise, becomes an indicator of clustering quality. Figure 3.15, gives a k-means partition of a data set and its silhouette plot. The same partition was shown before in Figure 2.2 in page 15; recall to the related discussion in page 14 about the standing of x_8 in the partition. The silhouette plot clearly states the weak clustering of x_8. The partition has an average silhouette value of $\bar{S} = 0.423$. The smallest point-wise silhouette value belongs to x_8; $s_8 = 0.115$. This small value corresponds to the narrow bar at the bottom of the silhouette plot in Figure 3.15b. The silhouette plot of the partition in Figure 3.14a, of a larger data set, is given in Figure 3.16.

To summarize the properties that recommend silhouette as a measure of clustering quality:

1. Its two measures of compactness and separation, a_t and b_t, are the proxies to

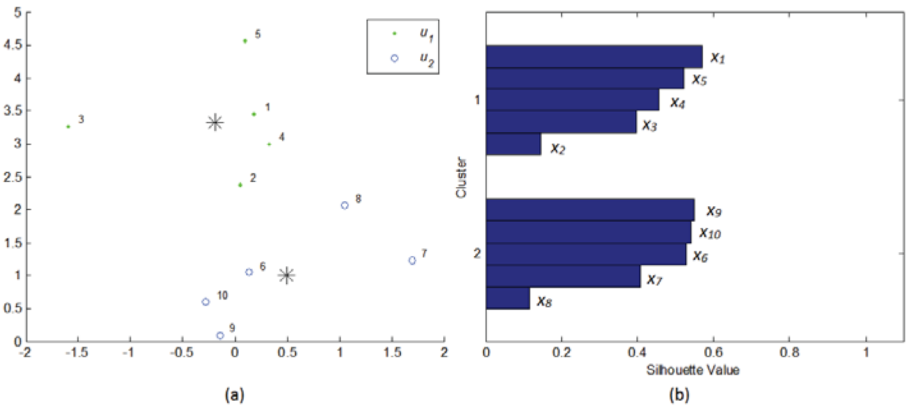

Figure 3.15: The succinct silhouette plot: (a) the k-means partition of data set of 10 points and its (b) silhouette plot.

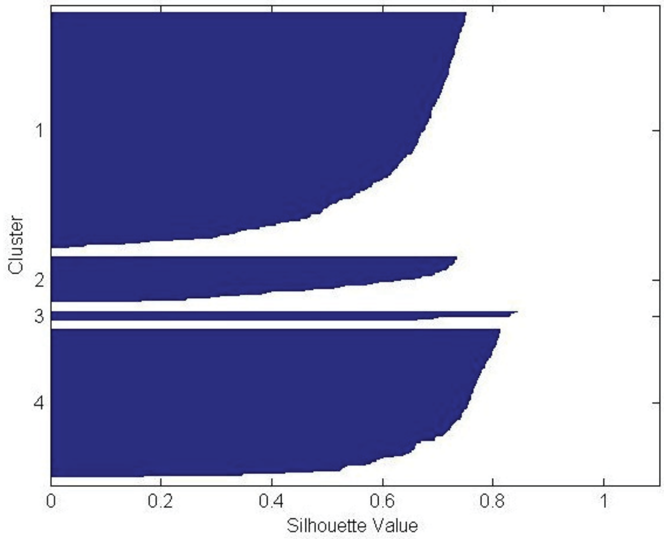

Figure 3.16: The silhouette plot of the partition into 4 clusters shown in Figure 3.14b.

verify the execution of the two folds of clustering, at the level of the individual points. Recall that the aim of clustering is two-fold, to

- Group similar points in the same cluster.
- Separate dissimilar points by assigning them to different clusters.

2. The consistency between a_t and b_t: both are weighted means of point-to-point distances, that measure compactness and separation with respect to an individual

point.

3. The flexibility in the assessment of clustering quality at different levels of granularity: point-level, cluster-level, and overall. Two overall measures are computed by (3.26) and (3.27), with the latter being cluster-wise. This flexibility also holds, if one seeks cluster, or overall, measures of compactness and separation. Such measures are obtained by replacing s_j by a_j, or b_j, in (3.25) and (3.26).

4. It is independent of, and no assumptions about, the clustering algorithm. It accepts any crisp partition of a given data set, to be verified against the relations between the data points, supplied as distances or similarities.

5. Ease of interpretation due to the bounded scale stated in (3.24).

6. The graphical display.

Given the above advantages, it seems unfortunate that silhouettes are restricted to crisp partitions. By means of defuzzification, one might prepare a given fuzzy partition for silhouette evaluation. But, several fuzzy partitions might reduce to the same crisp partition using the same defuzzification technique, so, how should they be differentiated on the basis of clustering quality? Such fuzzy partitions are easily obtained by repeating the application of fuzzy c-means on the same data with, say, using $m = 1.5$ and $m = 2$. Not to mention, defuzzification is responsible for a loss in similarity and dissimilarity information represented by the fuzzy partition, as noted before.

3.3.5 Extended Average Silhouette Index

In an attempt to explicitly incorporate the fuzzy membership values in the evaluation of fuzzy partitions, Campello and Hruschka proposed an extension to the average silhouette index in [19]. Defuzzification is a necessary step in the computation of the extended index. Suppose that the fuzzy partition, in consideration, is characterized by the membership matrix $\mathrm{U} = [u_{ij}]$. Using (2.16), U is defuzzified into the characteristic matrix $\acute{\mathrm{U}} = [\acute{u}_{ij}]$. Then, it becomes possible to compute the individual silhouette values s_j by (3.23) on $\acute{\mathrm{U}}$. Recall that, the average silhouette value is an overall measure of clustering quality, $\bar{S}$ given by (3.26). The extension transforms $\bar{S}$ from an arithmetic mean into a weighted arithmetic mean. Each silhouette is weighted by the difference between the two largest cluster membership values of the associated point. Let $p(j)$ and $q(j)$ denote, respectively, the cluster indices in which x_j has its first and second largest membership values. Then, the associated weight w_j is simply computed by

$$w_j = u_{p(j)j} - u_{q(j)j} \tag{3.28}$$

The extended average silhouette index is computed by

$$ES = \frac{\sum_{j=1}^{n} w_j s_j}{\sum_{j=1}^{n} w_j} \tag{3.29}$$

Clearly, the fuzzy partition U is involved in the computation of w_j, and its defuzzification into the crisp partition into Ú is involved in the computation of s_j.

One can anticipate that the good properties attributed to silhouettes are also passed to the extended measure. Other than the explicit use of the fuzzy membership values, what else do the weights add to the assessment of clustering quality? By definition of the weighted mean, the silhouettes contribute unequally to the final average value, in a manner determined by the associated weights. The authors believe that the weights stress the role of the points in dense areas in the computation of the measure. From their perspective, this is an improvement to the average silhouette index; it becomes better in detecting areas of high densities. Fuzzy c-means produces partitions in which the points near the vicinity of cluster centers assuming larger weights, therefore importance, than other points. Points in overlapping areas fall under this category of 'other points'. However, treating the points differentially, in the evaluation of clustering quality, works against the extended index rather than in its favor, as illustrated by the following example.

Consider the data set shown in Figure 3.17. A number of fuzzy c-means partitions of the data set into $c = 2, \ldots, 7$ clusters were obtained using $m = 2$. Only the partitions into $c = 3$ and $c = 4$ are shown respectively in Figure 3.17a and Figure 3.17b. Denote the partitions respectively by P3 and P4. The weights computed by (3.28) on P3 and P4, are visualized by means of intensity plots. The weight intensity plots obtained on P3 and P4 are given respectively in Figure 3.17c and Figure 3.17d. All partitions were evaluated by both of the average silhouette index, and the extended average silhouette index; see the evaluation results in Figure 3.18. Both measures almost agree in their relative evaluation of the partitions, except for the ranking of P3 and P4 relative to each other. The points that occupy cluster u_1 in P3 could be allocated to 2 or 3 cluster to obtain a partition that is better in compactness and separation. This is met by clusters u_1 and u_3 in P4. Thus, P4, is claimed as a better clustering than P3. The average silhouette index is in support of this cliam but is not, the extended index. This disagreement is definitely due to the extra weighting terms, since both meassures incorporate the exact same silhouette values. Figure 3.17c shows that the points in the middle of the figure, since lightly shaded, have relatively small wieghts. Their low silhouette values were not fully taken into account in the computation of the weighted mean, hence the relatively large weighted mean. As if the evaluation of the extended index was blinded by the weights. In contrast, the same points have larger weights in the contetxt of P4, the darkley shaded points in the middle of Figure 3.17d, but unfortunately they did not help in achieving a weighted mean whose value is larger.

The example above highlights the risk of being loose in the assessment of clustering quality rather than thorough. Similar to the partition coefficient, the weights are not necessary connected to the clustering quality of the partition. The extended measure does not have the same senstivity to the clustering of each point; therefore it seems somehow a naïve measure. Moreover, the extension is limited to the average measures. The point-wise silhouettes are still disconnected from the fuzzy membership values. The clustering of each individual point can be evaluated by the silhouette measure only in the context of a crisp partition. Fortunately, it is possible to reason about the clustering of each point using fuzzy membership values within the framework developed in the next chapter.

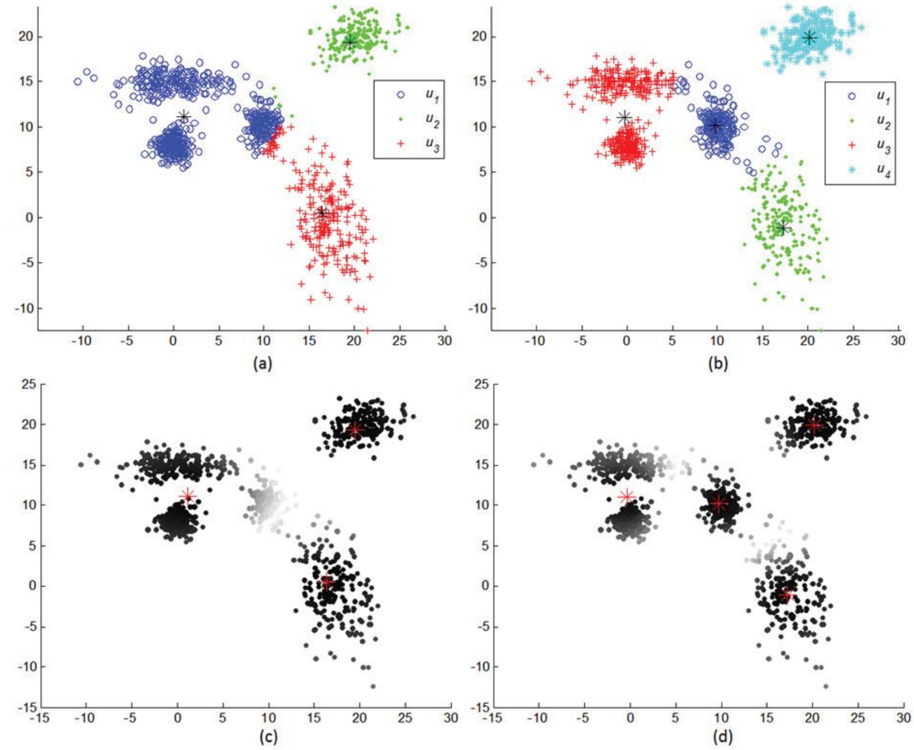

Figure 3.17: The fuzzy c-means partitions of a data set into (a) 3 and (b) 4 clusters. The weight intensities, obtained by (3.28), of the partitions in (a) and (b) are shown respectively in (c) and (d). The black and red asterisks mark the cluster centers. The partitions were obtained using $m = 2$.

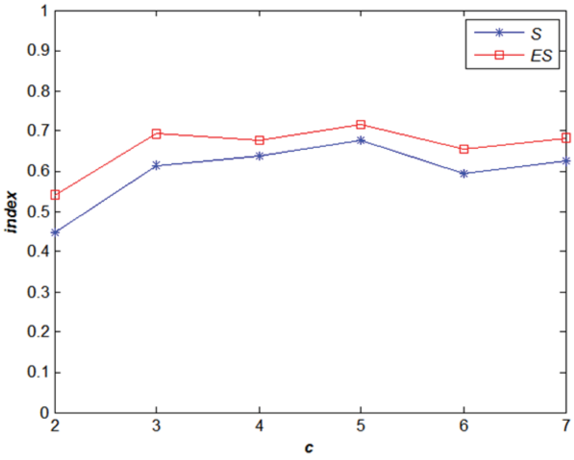

Figure 3.18: The fuzzy c-means partitions of the data set shown in Figure 3.17 into different number of clusters, represented by their number of clusters c, versus the average silhouette index S and the extended average silhouette index ES. The partitions were obtained using $m = 2$. The partitions into 3 and 4 clusters are shown in Figure 3.17.

CHAPTER 4
THE RELATIONAL FRAMEWORK

The common definition of clustering assumes data of a relational nature. More specifically, the similarities and dissimilarities between the data points are supposedly the nuts and bolts for cluster formation. The clustering results also conform to an intuitive relational perspective as will be explained soon. The preliminary distanc-based perspective on clustering which appears in [66] is formalized by the relations defined in this framework.

4.1 Distance-Based Relations

The two binary relations of similarity and dissimilarity, denoted by *SIMILAR* and *DISSIMILAR*, over a data set X are defined by means of a distance measure; therefore, they are distance-based relations. The relations *SIMILAR* and *DISSIMILAR* are subsets of the Cartesian product $X \times X$. They are characterized by the following membership functions

$$u_{SIMILAR}(x_r, x_s) = \begin{cases} 1, & d(x_r, x_s) \text{ is relatively small} \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

$$u_{DISSIMILAR}(x_r, x_s) = \begin{cases} 1, & d(x_r, x_s) \text{ is relatively large} \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

Accordingly, x_r is similar to x_s, denoted by $x_r SIMILAR x_s$, or equivalently $u_{SIMILAR}(x_r, x_s) = 1$, if and only if $d(x_r, x_s)$ is *relatively small.* These distance-based relations, however, are quite imprecise, since established, by (4.1) and (4.2), upon linguistic rather than numerical values. It is worth noting that the fuzzy linguistic model [90], the fuzzy approach for computing with words, recognizes the imprecision in linguistic values by representing them as fuzzy sets. In this model the numerical variable distance constitutes the base variable for the linguistic variable *Distance*. *Distance* assumes linguistic values, such as *relatively small* and *relatively* *large*; each becomes a fuzzy set defined on the numerical values of *distance*. The fuzzy membership functions determine the compatibility between the numerical values and the characterized linguistic values, hence called *compatibility functions*. Fuzzy sets have no sharp transition from one to other, so is the distinction between the linguistic values. Thus, the distinction of *SIMILAR* pairs from *DISSIMILAR* pairs, in $X \times X$, is imprecise. That is, if $d(x_r, x_s)$ is neither relatively small nor relatively large it is unclear if $x_r SIMILAR x_s$ or $x_r DISSIMILAR x_s$. Despite the imprecision in these relations, the graph structure, defined by the distances, is the only ground truth assumed in the problem, provided the wise selection of a relevant distance measure.

4.2 Cluster-Based Relations

In the context of a partition of a data set X into a number of clusters, two cluster-based binary relations are defined over X, namely, *intracluster* and *intercluster*, denoted by *INTRACLUSTER* and *INTERCLUSTER*. The point x_r is in intracluster relation with x_s, denoted by $x_r INTRACLUSTER x_s$, if x_r and x_s belong to the same cluster; otherwise, they are in intercluster relation, denoted by $x_r INTERCLUSTER x_s$. To account for all events that can result in a pair being in *INTRACLUSTER* or *INTERCLUSTER* relation, a number of cluster-specific relations are defined. Note that, the cluster membership values of the data points count as logical truth values; therefore, they become the operands for the logical operators used in reasoning about the defined cluster-based relations. Let $U = \{u_i\}_{i=1}^{c}$ be a partition of $X = \{x_j\}_{j=1}^{n}$ into c clusters. With respect to each cluster u_i, define the relation *INTRACLUSTER$_i$* by the following membership function

$$u_{INTRACLUSTER_i}(x_r, x_s) = \min(u_{ir}, u_{is}) \tag{4.3}$$

That is, x_r and x_s are in intracluster relation with respect to cluster u_i if and only if the conjunction of their memberships in u_i evaluates to true i.e. both belong to u_i. The conjunction is asserted by the minimum t-norm operator. Note that, there are as many intracluster relations as the number of clusters in the partition. The event of (x_r, x_s) being in an intracluster relation, regardless which cluster it is, is contained in the disjunction of the events of (x_r, x_s) being in intracluster relation over all the clusters in the partition. Thus, the membership in the general intracluster relation *INTRACLUSTER* is a disjunction of c conjunctions, i.e.

$$u_{INTRACLUSTER}(x_r, x_s) = \max_i \left(u_{INTRACLUSTER_i}(x_r, x_s)\right) \tag{4.4}$$

In (4.4), the disjunction is asserted by the maximum t-conorm operator. In contrast to intracluster relations, it requires two clusters for a pair of points to be in intercluster relation. Thus, there are as many intercluster relations as the number of 2-combinations of the c clusters. With respect to each unordered pair of clusters u_h and u_i, define $INTERCLUSTER_{h,i}$ by

$$u_{INTERCLUSTER_{h,i}}(x_r, x_s) = \max\left(\min(u_{hr}, u_{is}), \min(u_{ir}, u_{hs})\right) \tag{4.5}$$

That is, x_r and x_s are in intercluster relation with respect to u_h and u_i if and only if x_r belongs to u_h and x_s belongs to u_i, or vice-versa. The disjunction, in (4.5), accounts for the two permutations of the two points over the two clusters. In a similar fashion to *INTRACLUSTER*, the membership of (x_r, x_s) in the general intercluster relation *INTERCLUSTER* becomes a disjunction of disjunctions, one disjunction per cluster-specific intercluster relation, i.e.

$$u_{INTERCLUSTER}(x_r, x_s) = \max_{h,i} \left(u_{INTERCLUSTER_{h,i}}(x_r, x_s)\right) \tag{4.6}$$

In (4.3), (4.4), (4.5) and (4.6), the *product* t-norm and the *probabilistic sum* t-conorm, see [25], can replace the minimum t-norm and the maximum t-conorm.

4.3 Distance-Based Relations versus Cluster-Based Relations for Clustering

As opposed to the distance-based relations, the membership of a pair (x_r, x_s) in *INTRACLUSTER* or *INTERCLUSTER* is precise. The distinction of *INTRACLUSTER* pairs from *INTERCLUSTER* pairs is, therefore, a partition of the pairs, and a

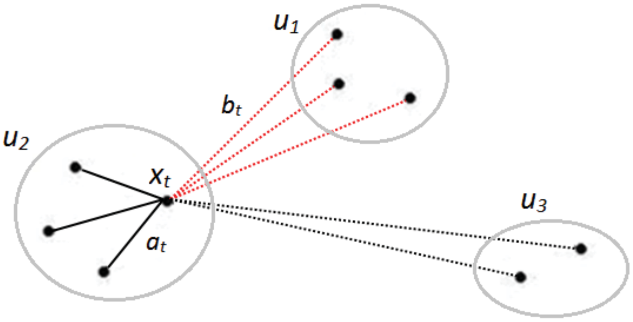

Figure 4.1: A partition of the data set is a partition of the pairwise distances: the intracluster (solid lines) and intercluster (dashed lines) distances between x_t and the data points in the context of a crisp partition into 3 clusters. Only distances associated with x_t are shown in the figure.

partition of the associated distances, into the two subsets. The fact that a partition of a data set is essentially a partition of the pairwise distances into intracluster and intercluster distances is illustrated in Figure 4.1. Since, intracluster distances are associated with points within the same cluster; they contribute to cluster compactness. By contrast, the intercluster distances contribute to cluster separation; since they are between points in different clusters. Consider the intersections below. By common sense, the larger their cardinalities are, the more compact and separated the clusters are. A better clustering is achieved by having more pairs in

- $TP = SIMILAR \cap INTRACLUSTER$
- $TN = DISSIMILAR \cap INTERCLUSTER$

In analogy with a classification task, suppose that the clustering algorithm makes predictions about the similarities and dissimilarities between the data points in terms of the $INTRACLUSTER$ and $INTERCLUSTER$ relations. In particular, allocating two points to the same cluster is the algorithm way to say that they are similar, and dissimilar if allocated to different clusters. Referring to $SIMILAR$ as the positive class and $DISSIMILAR$ as the negative class, the distance-based relations become the actual classes while the cluster-based relations become the predicted classes. It is reasonable, then, to denote the two intersections above by TP and TN, which refer to the true positives and the true negatives that constitute the true predictions. Furthermore, a confusion matrix can be used in reporting, and in the assessment of, clustering results, see Table 4.1. In practice, however, regardless how the distance-based and cluster-based relations seem intuitive for clustering, clustering algorithms operate directly on the distances between the data points rather than attempting the distance-based relations. The process of inferring the distance-based relations directly from the distances, i.e. to identify pairs of similarity and dissimilarity, is an ambiguous and complicated task, complicated by the choice of the distance measure. The task of inferring the distance-based relations over the whole set of points is the job of the clustering algorithm. The distance-based relations are inferred in terms of the cluster-based relations, by processing the distances and grouping the data points in a specified number of clusters. The changes in the cluster-based relations have the same rhythm as the cluster assignment of the points, in the chamber of the clustering process. The presence of the cluster-based relations in clustering is either implicit or explicit. In regards to k-means and fuzzy c-means, the objective functions in the pairwise distances incorporate explicitly intracluster relations; as they tie distances with intracluster membership values. More specifically, (u_{ir}, u_{is}) and $(u_{ir}, u_{is})^m$ that appear respectively in (2.6) and (2.13) in pages 12 and 18, compute the membership of (x_r, x_s) in $INTRACLUSTER_i$, under the product t-norm instead of the minimum t-norm, used in (4.3).

Table 4.1: A hypothetical use of a confusion matrix in reporting the performance of a clustering algorithm.

		Cluster-Based (Predicted Similarity)	
		INTRACLUSTER	*INTERCLUSTER*
Distance-Based	*SIMILAR*	*TP*	*FN*
(Actual Similarity)	*DISSIMILAR*	*FP*	*TN*

4.4 Cluster-Based Relations versus the Pairwise Distances for Cluster Validity

The output of the clustering algorithm, a partition, states the similarity, or dissimilarity, between a pair of points as propositions. The membership values of the pair in the cluster-based relations become the truth values of these propositions. The cluster-based relations are inferred from the pairwise distances. The distances become the operating premises in clustering, assumed to be true, to come with conclusions about point similarities and dissimilarities. To guard against incorrect inferences, the conclusions, the cluster based relations, must be verified against the premises, the distances. Incorrect inferences arise simply from, for example, an inappropriate specified number of clusters, perceived as a fallacy in the inference procedure, or false premises due to incorrect, probably noisy, data.

The silhouette measure ties, implicitly, by a_t and b_t, the pairwise distances (the premises for pair similarities or dissimilarities) to the cluster-based membership values (the conclusions about pair similarities or dissimilarities). The compactness distance a_t is the average of the intracluster distances between x_t and the data points. By contrast, the separation distance b_t is an average over a subset of the intercluster distances between x_t and the data points. The correctness of the inferences, and thus the quality of the clustering results, are interpreted by how a_t and b_t relatively compare with each other. Since both quantities are computed with respect to a point x_t, it is a fine-grain assessment of clustering quality, as opposed to prototype-based measures of clustering quality, for instance using the cluster centers. Turning back to the computation of silhouettes, equation (3.20) in page 44, which computes the average distance between x_t and the points in one cluster, is inapplicable if the partition, in consideration, is fuzzy. Only by defuzzification, (3.20) becomes applicable; as it is necessary in the computation of the extended average silhouette index. Note that, fuzzy c-means is a fuzzy approach to clustering since its underlying axioms are axioms for fuzzy sets. The output of fuzzy c-means, a fuzzy partition, represents the relations between the data points and the uncertainties about these relations. Zadeh, the person who had invented fuzzy logic, claims [91]:

> *Fuzzy logic is not fuzzy. Basically, fuzzy logic is a precise logic of imprecision and approximate reasoning. More specifically, fuzzy logic may be viewed as an attempt at formalization/mechanization of two remarkable human capabilities. First, the capability to converse, reason and make rational decisions in an environment of imprecision, uncertainty, incom-*

> *pleteness of information, conflicting information, partiality of truth and partiality of possibility in short, in an environment of imperfect information.*

Therefore, defuzzification, applied to a fuzzy partition, offends fuzzy logic since it seems as a declaration of its incapability to reason about clustering quality, in an environment of uncertainty about the cluster assignment of the points. By environment, we mean the fuzzy partition. Projecting the construction of silhouettes to the cluster-based relations, the intracluster and intercluster distances discussed above, and observing that fuzzy sets and fuzzy relations are generalizations of their crisp counterparts, a natural generalization of the silhouette measure to fuzzy partitions seems plausible. Our framework, with the defined logical binary relations, supports a distance-based perspective of clustering and cluster validity. Within this framework, the next chapter explains how silhouettes could be computed directly from the fuzzy membership values.

CHAPTER 5
GENERALIZED SILHOUETTES

Two notions of silhouettes emerge within our framework by considering either the point-to-point distances or the center-to-point distances. The generlization of silhouette computation to fuzzy partitions first appeared in [66] on the basis of the general *INTRACLUSTER* or *INTERCLUSTER* relations defined respectively by (4.4) and (4.6). The generalization was then refined by [64] in the used relations to incoporate the cluster-specific relations defined by (4.3) and (4.5). Later, a new notion of silhouettes appeared in [63].

5.1 Generalized Point-Wise Silhouettes

The number of $INTRACLUSTER_i$ relations for a partition into c clusters is c relations. Whereas, the number of $INTERCLUSTER_{h,i}$ is the number of the 2-combinations of the c clusters i.e. $\binom{c}{2}$ relations. Evaluating the membership functions of $INTRACLUSTER_i$ and $INTERCLUSTER_{h,i}$, given respectively in (4.3) and (4.5) in page 56, on crisp membership values results in crisp relations. By contrast, fuzzy relations result from fuzzy membership values. The computation of a_t and b_t is reformulated in terms of these relations, either crisp or fuzzy, as the following

$$a_t = \min_i \{ \frac{\sum_{j=1,j\neq t}^{n} u_{INTRACLUSTER_i}(x_t, x_j) d(x_t, x_j)}{\sum_{j=1,j\neq t}^{n} u_{INTRACLUSTER_i}(x_t, x_j)} \mid \sum_{j=1,j\neq t}^{n} u_{INTRACLUSTER_i}(x_t, x_j) > 0 \} \tag{5.1}$$

$$b_t = \min_{h,i} \{ \frac{\sum_{j=1,j\neq t}^{n} u_{INTERCLUSTER_{h,i}}(x_t, x_j) d(x_t, x_j)}{\sum_{j=1,j\neq t}^{n} u_{INTERCLUSTER_{h,i}}(x_t, x_j)} \mid \sum_{j=1,j\neq t}^{n} u_{INTERCLUSTER_{h,i}}(x_t, x_j) > 0 \} \tag{5.2}$$

In (5.1) and (5.2), min denotes the standard numerical rather than the logical operator. The inequalities in (5.1) and (5.2) discard any relation that does not relate

x_t to any point x_j. In the virtue of the original equations (3.20), (3.21) and (3.22) in page 44, a_t and b_t are still selected among means of the distances associated with x_t. Since the compactness distance a_t is associated with intracluster distances, its computation involves $INTRACLUSTER_i$ membership values. And by contrast, the presence of $INTERCLUSTER_{h,i}$ membership values in the computation of b_t is justified. The silhouette s_t is still computed by (3.23) in terms of the resultant a_t and b_t, and so the different average silhouette measures defined by (3.25), (3.26) and (3.27). Moreover, the new equations of (5.1) and (5.2) compute the same exact a_t and b_t from any crisp partition, as obtained by the original formulas (3.20), (3.21) and (3.22), therefore reaching the same silhouettes. We refer to a_t and b_t obtained using (5.1) and (5.2), and the silhouette s_t computed on them, as the generalized compactness distance, the generalized separation distance, and the generalized silhouette, respectively. They are generalized measures because they are also computable from fuzzy membership values. The next example demonstrates a fact on how the generalized silhouettes resemble the silhouettes computed by the original formulas in the evaluation of crisp partitions.

A number of k-means partitions of a data set, also in Figure 3.2 in page 27, into $c = 2, \ldots, 7$ clusters were obtained. The partitions into $c = 2$ and $c = 3$ are shown in Figure 5.1. Denote them respectively by P2 and P3. The silhouettes were computed from the crisp partitions using the original formulas and the generalized formulas. Denote the average silhouette values obtained using the original formulas by S. Let GS and ES denote respectively the average generalized silhouette values and the extended average silhouette values. The silhouette results are shown in Figure 5.2. Before verifying the silhouettes against the generalized silhouettes, recall the argument about how the similarities and dissimilarities between the data points, implied by their pairwise distances, are in favor of P3 rather than P2. The partition P2 is shape-based or more accurately model-based, but not distance-based. A model-based clustering suits such data sets since it attempts finding the models that govern the underlying processes which generate the data observations. All the measures rank P3 as a better clustering than P2. And, all the measures reached the same exact silhouette results as shown in Figure 5.2. Trivially, the extended average values,

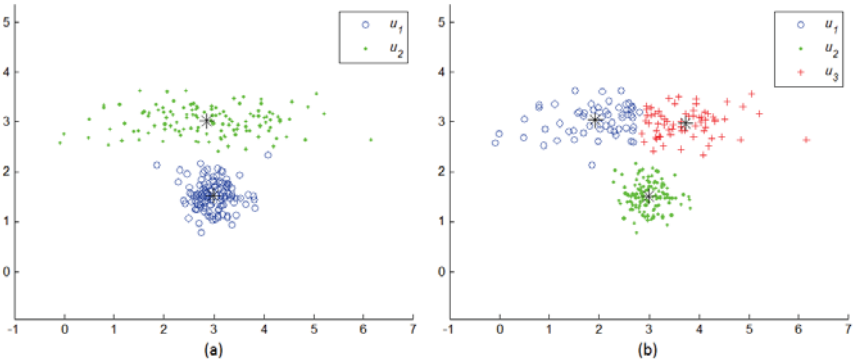

Figure 5.1: The k-means partitions of a data set into (a) 2 and (b) 3 clusters.

given by *ES*, are the same as the average silhouette values; since both incorporate the same silhouettes s_j and the extensive weighting terms reduce to $w_j = 1$. The average generalized silhouette values *GS*, on the other hand, resemble the average silhouette values by a logical reasoning about the relations between the data points, considering the events of allocating the points to the clusters. A reasoning implicitly carried by the original silhouette formulas, implemented in terms of the average distance between x_t and the data points in each cluster.

The next example, with the help of the generalized silhouettes *GS*, demonstrates the fact that Xie-Beni index *XB* employs a rough measure of separation, in contrast to its measure of compactness. In particular, the minimum center-to-center distance is used for separation while center-to-point distances are used for compactness. Any distance in our framework reflects some kind of a relationship. The separation implied by the intercluster distances between the points in the two clusters, is reduced by *XB* to one distance, being the minimum center-to-center distance. What is about the separation between the other points allocated to the remaining clusters? A number of fuzzy c-means partitions of the data set shown in Figure 5.3 into $c = 2, \ldots, 7$ were obtained using $m = 3$. Denote the partitions by P2, , and P7, respectively. The partitions were evaluated by *GS* and *XB*. The evaluation results are shown in Figure 5.4. Both measures almost agree in their evaluation of the partitions. They disagree about two particular partitions, namely, P5 and P6. Cluster u_1 in P5 is split into two clusters in P6, namely, clusters u_5 and u_6. The measure of separation in *XB* overlooks the bad separation caused by

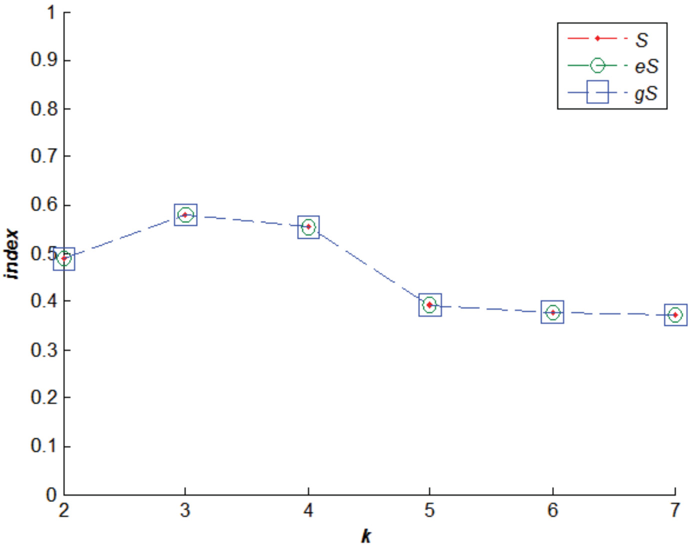

Figure 5.2: The k-means partitions of the data set shown in Figure 5.1 into different number of clusters, represented by their number of clusters k, versus the average silhouette values S, the extended average silhouette values ES and the average generalized silhouette values GS. The partitions into 2 and 3 clusters are shown in Figure 5.1.

such split. In P6, this is a bad separation because of the relatively small intercluster distances between points in u_5 and u_6. The separation of P5 is measured by $d(v_4, v_5)$ which is the minimum center-to-center distance. The separation of P6 is measured by the same distance value, but noting that u_4 and u_5 in P5 map respectively to u_1 and u_3 in P6. With respect to P6, the bad separation implied by $d(v_5, v_6)$ is ignored since $d(v_5, v_6) > d(v_1, v_3)$. By incorporating the associated intercluster distances in b_t, each generalized silhouette s_t becomes aware of the separation relative to the point x_t. Thus, GS provides a robust measurement of clustering quality, evident in its ranking of P5 relative to P6. The interested can find an experiment comparing the silhouette measure to other measures in [65].

A powerful feature of the generalized silhouettes is their ability to convey the clustering quality of the individual points in the context of either crisp or fuzzy partitions. In the consequence, it becomes possible to compare between the two approaches of clustering in regards to clustering quality, as illustrated by an example. A number of fuzzy c-means partitions of the data set shown in Figure 5.5 into

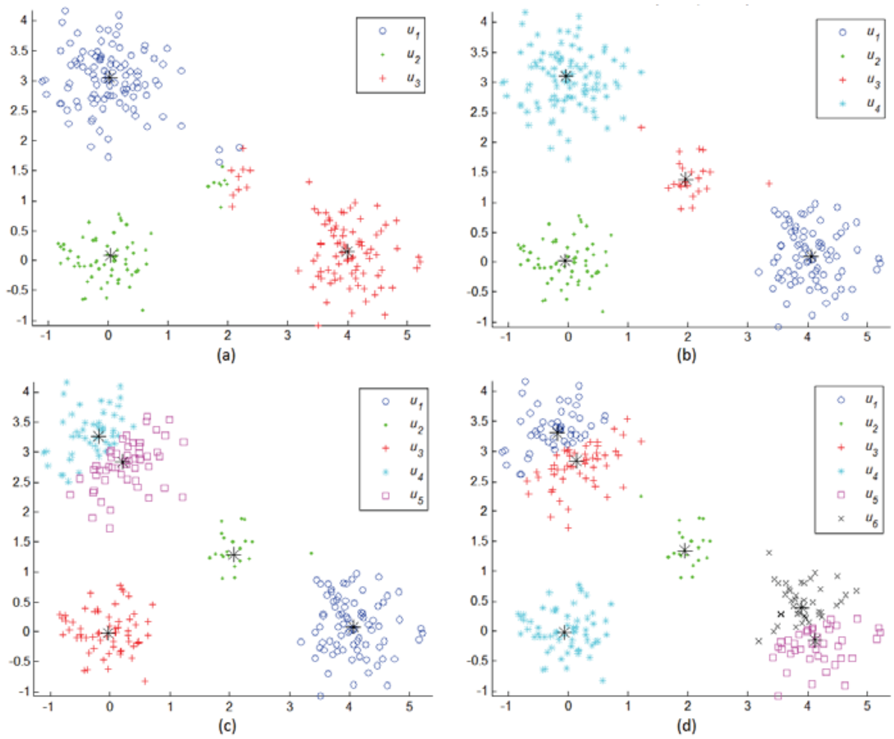

Figure 5.3: The fuzzy c-means partitions of a data set into (a) 3, (b) 4, (c) 5 and (d) 6 clusters. The partitions were obtained using $m = 3$.

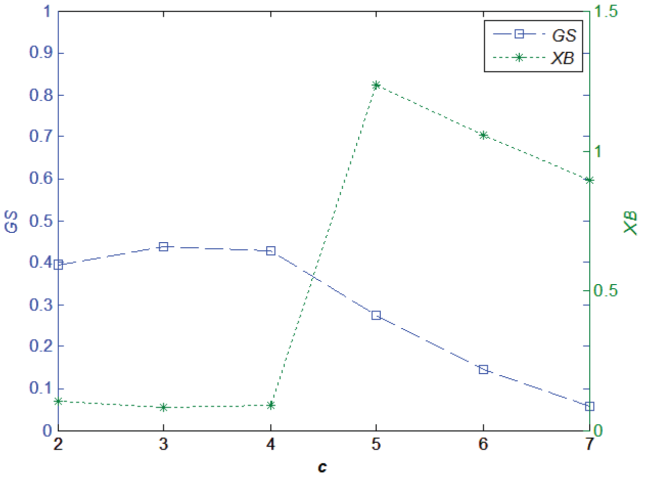

Figure 5.4: The fuzzy c-means partitions of the data set shown in Figure 5.3 into different number of clusters, represented by their number of clusters c, versus the average generalized silhouette values GS and Xie-Beni index XB. The partitions were obtained using $m = 3$. The partitions into 3, 4, 5 and 6 clusters are shown in Figure 5.3.

$c = 2, \ldots, 7$ clusters were obtained using $m = 3$. The partitions were reduced to crisp partitions by means of defuzzification. The same data set was used in previous examples. It was sampled from five bivariate Gaussians, 200 points from each distribution, 1000 points in total. The average generalized silhouette values of the crisp defuzzified partitions and the fuzzy partitions are denoted respectively by S and GS. The extended average silhouette values are denoted by ES. Note that, the extended average values are computed from both the fuzzy partitions and defuzzified partitions in order to compute the silhouettes and the weights. The silhouette results are shown in Figure 5.5d. The extended average silhouette values ES are larger than the average generalized silhouette values of the crisp partitions S, because the silhouettes of small values assume a less important role in the computation of ES due to the weighting terms. Now, consider the generalized silhouettes of the crisp and fuzzy partitions given respectively by S and GS in Figure 5.5d. It immediately follows that the defuzzified crisp membership values, and so the inferred crisp relations, as opposed to their fuzzy counterparts, are more accountable for the pairwise distances in representing the relations between the data points. This is a serious observation, discussed later in the sequel.

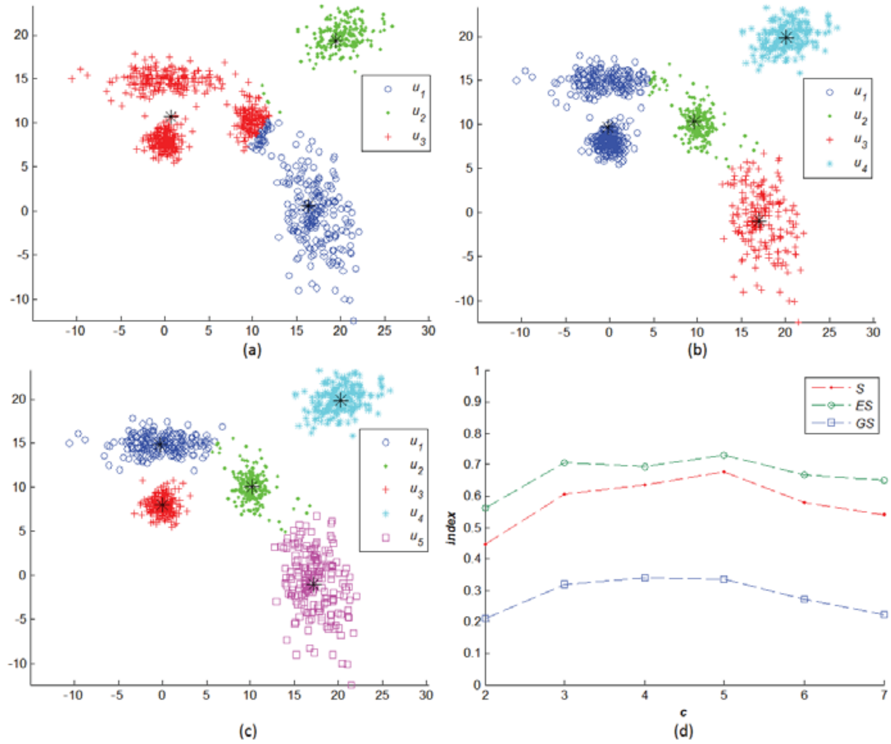

Figure 5.5: The fuzzy c-means partitions of a data set into $c = 2, \ldots, 7$ clusters, represented by their number of clusters c, versus the extended average silhouette values ES and the average generalized silhouette values GS, shown in (d) where S denotes the average generalized silhouettes of the defuzzified partitions. The partitions were obtained using $m = 3$. The partitions into 3, 4 and 5 clusters are shown respectively in (a), (b) and (c).

5.2 Center-Wise Silhouettes

The generalized silhouettes require computing the membership of each pair of points in each cluster-based relation. The computing devices are the minimum t-norm and its dual the maximum t-conorm operators that appear in (4.3) and (4.5) in page 56; as they find the membership of (x_r, x_s) in each $INTRACLUSTER_i$ and $INTERCLUSTER_{h,i}$ relation. Thus, the computational complexity of the generalized silhouettes can be expressed in the number of conjunction and disjunction operations, asserted by the logical operators.

Suppose that a data set $X = \{x_j\}_{j=1}^{n}$ is partitioned into c clusters $U = \{u_i\}_{i=1}^{c}$. Recall that, there are c intracluster relations and $\binom{c}{2}$ intercluster relations, defined with respect to U. In (4.3) in page 56, the membership of (x_r, x_s) in $INTRACLUSTER_i$

is asserted by one conjunction. By contrast, (4.5) asserts the membership of (x_r, x_s) in $\mathit{INTERCLUSTER}_{h,i}$ by a disjunction of two conjunctions. Table 5.1 summarizes the number of the logical operations necessary for computing the membership values for different number of pairs and different number of cluster-based relations. The total number of conjunctions and disjunctions, required for computing the generalized silhouettes over the whole set of data points, appear in the last two rows of Table 5.1. More specifically,

$$\begin{aligned} T(n,c) &= \text{total number of conjunctions } + \text{ total number of disjunctions} \\ &= c\binom{n}{2} + 3\binom{c}{2}\binom{n}{2} \end{aligned} \tag{5.3}$$

Table 5.1: The number of the logical operations to compute the membership of: (a) one pair in one intracluster relation, (b) one pair in one intercluster relation, (c) one pair in all intracluster relations, (d) one pair in all intercluster relations, (e) all pairs in all intracluster relations and (f) all pairs in all intercluster relations. c and n denote respectively the number of clusters and the number of data points. Cs and Ds stand respectively for conjunctions and disjunctions. Ts denotes the total number of conjunctions and disjunctions.

	$\{u\}$	**Cs**	**Ds**	**Ts**
a.	$\{u_{INTRACLUSTER_i}(x_r, x_s)\}$	1	0	1
b.	$\{u_{INTERCLUSTER_{h,i}}(x_r, x_s)\}$	2	1	3
c.	$\{u_{INTRACLUSTER_i}(x_r, x_s)\}_{1\leq i\leq c}$	c	0	c
d.	$\{u_{INTERCLUSTER_{h,i}}(x_r, x_s)\}_{1\leq h<i\leq c}$	$2\binom{c}{2}$	$\binom{c}{2}$	$3\binom{c}{2}$
e.	$\{u_{INTRACLUSTER_i}(x_r, x_s)\}_{1\leq r<s\leq n,\ 1\leq i\leq c}$	$c\binom{n}{2}$	0	$c\binom{n}{2}$
f.	$\{u_{INTERCLUSTER_{h,i}}(x_r, x_s)\}_{1\leq r<s\leq n,\ 1\leq h<i\leq c}$	$2\binom{c}{2}\binom{n}{2}$	$\binom{c}{2}\binom{n}{2}$	$3\binom{c}{2}\binom{n}{2}$

It can be seen from (5.3) that $T(n,c) \in \mathcal{O}(c^2n^2)$. As illustrated by Table 5.2, the generalized silhouettes grow very large in the number of logical operations as increasing the number of data points or the number of clusters.

Fortunately, it is possible, within our framework, to reason about clustering quality at the same level at which k-means and fuzzy c-means operate i.e. the level of the center-to-point distances. Switching to the coarse-grain of the center-to-point distances, as opposed to point-to-point distances, results in a significant gain in

Table 5.2: The number of logical operations, as computed by (5.3), for computing the general silhouettes for selected values of n and c.

c	n	$T(n,c)$
3	100	59400
5	100	173250
10	100	717750
3	1000	5994000
5	1000	17482500
10	1000	72427500

computational complexity, as will be shown soon. The cluster centers $V = \{v_i\}_{i=1}^{c}$ are supposed to be the cluster representatives; therefore, they become core points of the associated fuzzy sets, or clusters, i.e.

$$u_i(v_t) = \begin{cases} 1, & i = t \\ 0, & \text{otherwise} \end{cases} \tag{5.4}$$

Now, by treating each cluster center v_t as if it was an individual data point, (4.3) and (4.5) can compute the membership of (v_t, x_j) in the intracluster and intercluster relations. In the virtue of (5.4), (4.3) evaluates the membership of (v_t, x_j) in $INTRACLUSTER_t$ to

$$\begin{aligned} u_{INTRACLUSTER_t}(v_t, x_j) &= \min\left(u_t(v_t), u_t(x_j)\right) \\ &= \min\left(1, u_{tj}\right) \\ &= u_{tj} \end{aligned} \tag{5.5}$$

and in the remaining intracluster relations $INTRACLUSTER_i$, where $i \neq t$, to

$$\begin{aligned} u_{INTRACLUSTER_i}(v_t, x_j) &= \min\left(u_i(v_t), u_i(x_j)\right) \\ &= \min\left(0, u_{ij}\right) \\ &= 0 \end{aligned} \tag{5.6}$$

In words, v_t is in intracluster relation with the data points only with respect to its associated cluster u_t. Moreover, the membership of (v_t, x_j) in $INTRACLUSTER_t$ is the membership of x_j in u_t i.e. u_{tj}. By the inequality in (5.1) and substituting u_{tj} for $u_{INTRACLUSTER_t}(v_t, x_j)$ in (5.1), $a_t = a(v_t)$ is computed to

$$a_t = \frac{\sum_{j=1,j\neq t}^{n} u_{tj} d(v_t, x_j)}{\sum_{j=1,j\neq t}^{n} u_{tj}} \tag{5.7}$$

The inequality in (5.1) discards the intracluster relations in which v_t has no participation. Now, we turn into the intercluster relations. The fact that v_t is a core point of u_t causes (4.5) to evaluate the membership of (v_t, x_j) in $INTERCLUSTER_{t,i}$ to

$$\begin{aligned} u_{INETRCLUSTER_{t,i}}(v_t, x_j) &= \max\left(\min\left(u_t(v_t), u_i(x_j)\right), \min\left(u_i(v_t), u_t(x_j)\right)\right) \\ &= \max\left(\min\left(1, u_{ij}\right), \min\left(0, u_{tj}\right)\right) \\ &= \max\left(u_{ij}, 0\right) \\ &= u_{ij} \end{aligned} \tag{5.8}$$

and in the remaining intercluster relations $INTERCLUSTER_{h,i}$, where $h, i \neq t$, to

$$\begin{aligned} u_{INETRCLUSTER_{h,i}}(v_t, x_j) &= \max\left(\min\left(u_h(v_t), u_i(x_j)\right), \min\left(u_i(v_t), u_h(x_j)\right)\right) \\ &= \max\left(\min\left(0, u_{ij}\right), \min\left(0, u_{hj}\right)\right) \\ &= \max\left(0, 0\right) \\ &= 0 \end{aligned} \tag{5.9}$$

Among the $\binom{c}{2}$ possible intercluster relations, (v_t, x_j) has a membership only in $(c-1)$ relations in the set $\{INTERCLUSTER_{h,i} \mid \forall i, i \neq t\}$. Moreover, the membership of (v_t, x_j) in $INTERCLUSTER_{t,i}$ is the membership of x_j in u_i i.e. u_{ij}. By the inequality in (5.2) and substituting u_{ij} for $u_{INTERCLUSTER_{t,i}}(v_t, x_j)$ in the same equation, $b_t = b(v_t)$ is computed to

$$b_t = \min_{i\neq t} \left\{ \frac{\sum_{j=1,j\neq t}^{n} u_{ij} d(v_t, x_j)}{\sum_{j=1,j\neq t}^{n} u_{ij}} \right\} \tag{5.10}$$

We refer to a_t and b_t, computed by (5.7) and (5.10) , as the *center-wise compactness distance* and the *center-wise separation distance*; since they are computed in terms of the center-to-point distances relative to the cluster center v_t. The center-wise silhouette $s_t = s(v_t)$ is computed by the same silhouette equation (3.23). The overall clustering quality can be measured by the average center-wise silhouette value, taken over the centers.

No doubt, the center-wise compactness and separation distances are rough measures of compactness and separation, as opposed to their point-wise counterparts. The assessment of compactness and separation with respect to each individual point is reduced to the set of cluster centers. Despite the fact that the center-wise silhouettes carry a cost in accuracy, they pay off in computation time, in two ways. First, the number of pairs, so the number of distances of interest, is reduced from $\binom{n}{2}$, which gives the number of point-to-point distances, to $(n\ c)$, which gives the number of center-to-point distances. Second, and according to (5.5), (5.6), (5.8) and (5.9), the number of the logical operations necessary to compute the memberships in the cluster-based relations are brought down to zero; as the equations assume their values directly from the membership matrix.

To illustrate the straightforward computation of the center-wise silhouettes, as opposed to the point-wise silhouettes, consider the fuzzy c-means partition of a data set of 7 points into c=3 clusters, shown in Figure 5.6. The partition was obtained using $m = 3$. The membership matrix and the membership of the cluster centers in the three clusters, being core points, are given in Table 5.3. Suppose that we are interested in computing the center-wise silhouette $s_2 = s(v_2)$. The distances between v_2 and the data points are given in Table 5.4. The membership of each pair of (v_2, x_j) in $INTRACLUSTER_2$ is the membership of x_j in u_2. Moreover, the

Table 5.3: The membership matrix of the fuzzy c-means partition in Figure 5.6. The membership values are rounded to two decimal places in a manner that preserves the constraint (2.10b) in page 17. The membership of the cluster centers also appear in the table, being core points of their associated clusters.

U	x_1	x_2	x_3	x_4	x_5	x_6	x_7	v_1	v_2	v_3
u_1	0.05	0.07	0.09	0.11	0.2	0.87	0.87	1	0	0
u_2	0.06	0.09	0.11	0.84	0.73	0.1	0.1	0	1	0
u_3	0.8	0.84	0.8	0.05	0.07	0.03	0.03	0	0	1

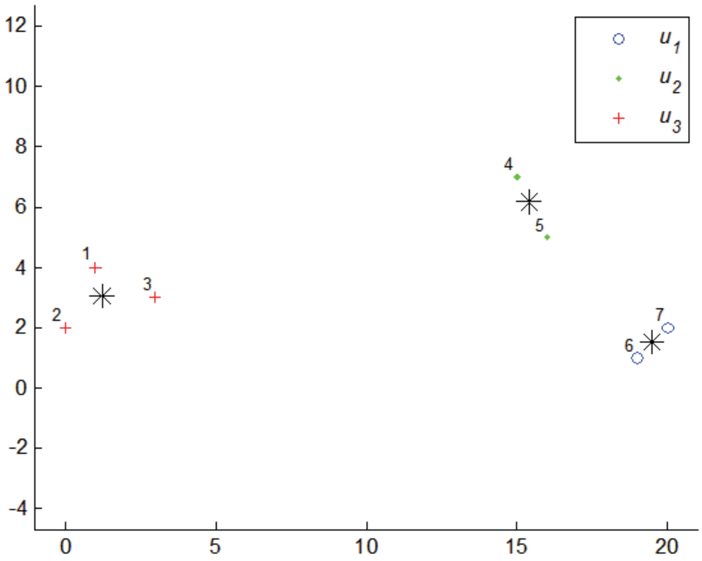

Figure 5.6: The fuzzy c-means partition of a data set into c=3 cluster, obtained using $m = 3$. Cluster centers are shown in black asterisks.

membership of each pair of (v_2, x_j) in $INTERCLUSTER_{2,1}$ is the membership of x_j in u_1. And, the membership of each pair of (v_2, x_j) in $INTERCLUSTER_{2,3}$ is the membership of x_j in u_3. Table 5.5 gives the memberships of the pairs (v_2, x_j), over each data point x_j, in the cluster-based relations, being directly obtained from U. The same pairs have no membership in the remaining cluster-based relations; thus, the relations are discarded from the computation of $a_2 = a(v_2)$ and $b_2 = b(v_2)$. Equation (5.7) computes $a_2 = a(v_2)$ as a weighted mean of the distances in Table 5.4 using the membership values of cluster u_2 as weights in place of $INTRACLUSTER_2$ membership values. Similarly, (5.10) computes $b_2 = b(v_2)$ as the minimum of the two weighted means obtained using u_1 and u_3 memberships values as weights in place of $INTERCLUSTER_{2,1}$ and $INTERCLUSTER_{2,3}$ membership values, respectively. Based on the values given in the tables, $a_2 = 3.3015$, $b_2 = 6.3179$ and $s_2 = 0.4774$. Another example puts the average center-wise silhouette index to test against the average generalized point-wise silhouette index, and other measures.

Consider the data set shown in Figure 5.7. A number of fuzzy c-means partitions of the data set into $c = 2, \ldots, 9$ clusters were obtained using $m = 3$. The partitions were reduced to crisp partitions by means of defuzzification. Denote the fuzzy partitions by P2, P3, ..., and P9, respectively. P2, P3, ..., P7 are shown

Table 5.4: The center-to-point distances between v_2 and the data points.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$d(v_2, x_j)$	14.54	15.94	12.78	0.9	1.34	6.32	6.23

Table 5.5: The cluster-based membership values relevant to the center-wise silhouette $s_2 = s(v_2)$. The values were directly fetched from the membership matrix U given in Table 5.3.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$u_{INTRACLUSTER_2}(v_2, x_j)$	0.06	0.09	0.11	0.84	0.73	0.1	0.1
$u_{INTERCLUSTER_{2,1}}(v_2, x_j)$	0.05	0.07	0.09	0.11	0.2	0.87	0.87
$u_{INTERCLUSTER_{2,3}}(v_2, x_j)$	0.8	0.84	0.8	0.05	0.07	0.03	0.03

in Figure 5.7. In Figure 5.8, the average generalized point-wise silhouette values of the fuzzy partitions are denoted by *GS*. The average generalized point-wise silhouette values of the defuzzified partitions are denoted by *S*. The average center-wise silhouette values are denoted by *VS*. The values of Xie-Beni index are denoted by *XB*. The partitions are ranked according to each evaluation in Table 5.6. As seen in the table, the average center-wise silhouette index, with a less accurate evaluation, almost agrees in its ranking with the average generalized point-wise silhouette index. But, its ranking resembles the one obtained by the Xie-Beni index. The accuracy of each measure is claimed on the basis of the incorporated distances, becoming rough measures as we move from point-to-point distances, to center-to-point distances and finally to center-to-center distances. If one can bear with their rough, still reliable, evaluation, center-wise silhouettes are computed on larger data sets in no time, as opposed to the point-wise silhouettes. Besides being a simple benchmark for the center-wise silhouettes, this example also provides a good context for comparing the clustering results of the fuzzy partitions and their defuzzified partitions. Toward this end, consider the average silhouette values of the crisp and fuzzy partitions given respectively by *S* and *GS* in Figure 5.8a. Again, it seems that the defuzzified crisp partitions are accountable for the pairwise distances in representing the pairwise similarities and dissimilarities, to a greater extent than the fuzzy partitions. Moreover, the generalized silhouettes suggest that the defuzzification of P6 is the best partition among all others, crisp or fuzzy. But, P6 is not the best partition among the fuzzy partitions; it is P2. The clustering given by P2, and its defuzzification, shown in Figure 5.7a, seems satisfactory, considering the good separation between the two constituting clusters. On the other hand, P6, shown in Figure 5.7e, is undoubtedly better in compactness but not in separation, especially considering the separation between u_4 and u_5. P2 has a better separation than the defuzzification of P6. It is interesting to observe that the crisp membership values are in favor of the clustering with less separation, therefore, more overlapping, more than the fuzzy membership values. This raises doubts if the fuzzy membership values are better in representing the data relations in the context of overlapping clusters i.e. under uncertainty, than the crisp membership values. Keep in mind that the clustering results in this example were generated by the fuzzy c-means algorithm.

Table 5.6: The fuzzy c-means partitions of the data set in Figure 5.7 into $c = 2, \ldots, 9$ clusters sorted according to their ranking by a number of measures, namely, the average generalized point-wise silhouette values *GS*, the average center-wise silhouette values *VS* and Xie-Beni index *XB*. *S* is the ranking of the defuzzified partitions by the average generalized point-wise silhouette values. Pc denotes the partition into c clusters.

S	P6	P5	P2	P7	P9	P8	P3	P4
GS	P2	P6	P5	P3	P7	P4	P8	P9
VS	P2	P6	P3	P5	P4	P7	P8	P9
XB	P2	P6	P3	P5	P4	P7	P8	P9

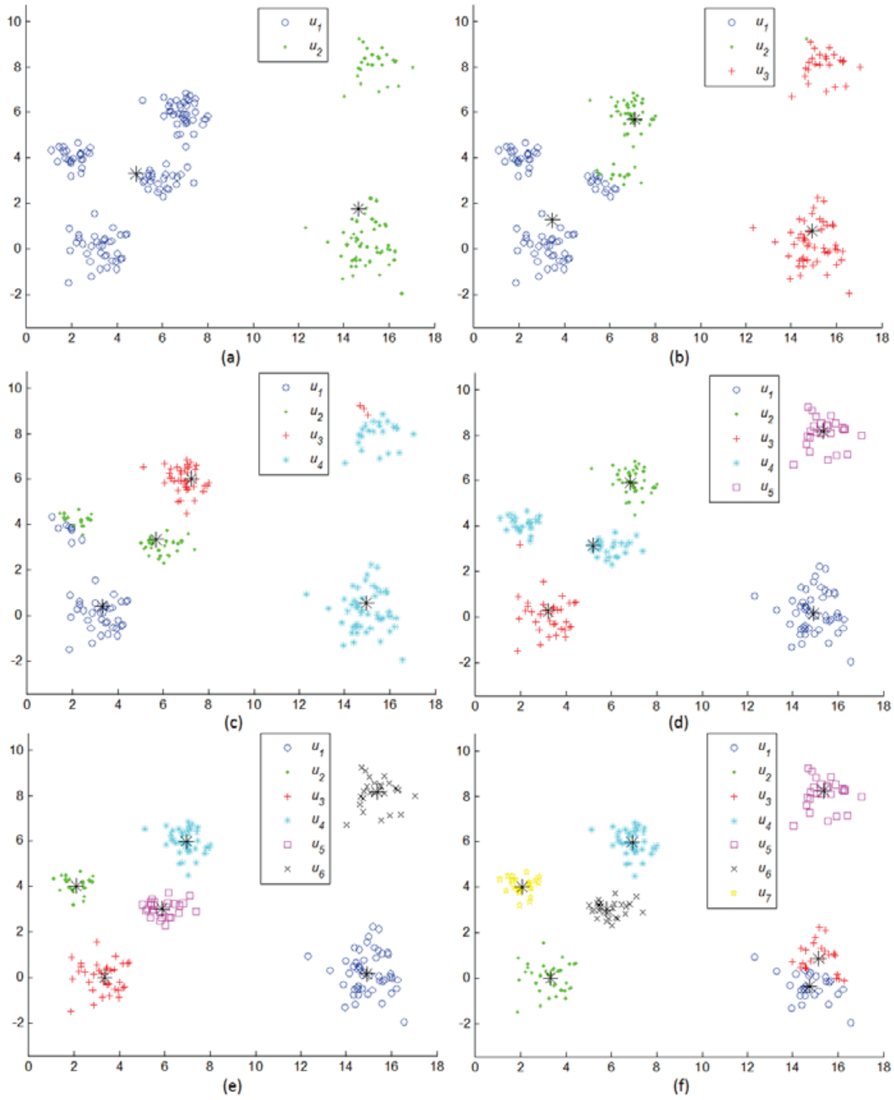

Figure 5.7: The fuzzy c-means partitions of a data set into (a) 2, (b) 3, (c) 4, (d) 5, (e) 6 and (f) 7 clusters. The partitions were obtained using $m = 3$. Note that, these plots were generated on the fuzzy partitions by means of defuzzification.

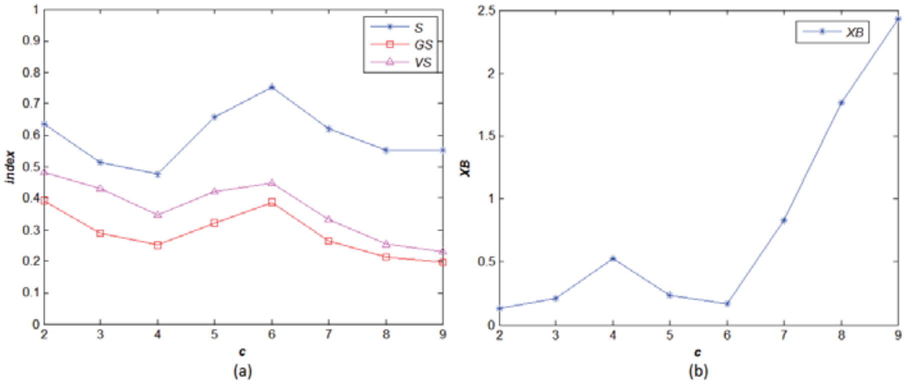

Figure 5.8: The fuzzy c-means partitions of the data set in Figure 5.7 into $c = 2, \ldots, 9$ clusters, represented by their number of clusters c, versus (a) the average generalized silhouette values GS, the average center-wise silhouette values VS and (b) Xie-Beni index XB. S denotes the average generalized silhouette values of the defuzzified partitions. The partitions into 2, 4, 6 and 7 clusters are shown in Figure 5.7. The partitions were obtained using $m = 3$.

CHAPTER 6
THE NON-CONVEX FUZZY SETS BY FUZZY c-MEANS

It seems, by comparing the average silhouette values of each fuzzy partition and its defuzzification in Figure 5.5d and Figure 5.8a, that defuzzification incurs a relative improvement in clustering quality. One might ask if this is the case for any fuzzy partition, generated by fuzzy c-means. We have attempted, by fixing the number of clusters and varying the value of the fuzzifier i.e. the parameter m, to generate a fuzzy partition that might have larger silhouettes than its defuzzification. A number of fuzzy c-means partitions of the data set in Figure 6.1 into $c = 3$ clusters were obtained using $m = 1.1, 1.5, 1.75, 2, 3, 5, 10$ and 13. The average generalized silhouette values of the fuzzy and defuzzified partitions, denoted respectively by GS and S, are shown in Figure 6.2. The fuzzy partitions, all of them, reduce to the same defuzzified partition, therefore the average silhouette value S does not change over all of the defuzzified partitions. Recall that as m increases, becoming too large, the cluster centers converge to the grand mean of the whole data set, causing the membership values to approach $\frac{1}{c}$. This limiting property of fuzzy c-means is apparent in Figure 6.1; the cluster centers of the fuzzier partition in Figure 6.1b are within smaller proximity, in contrast to the partition in Figure 6.1a. Before discussing the evaluation results any further, in chapter two we pointed out that $m = 3$ should be the best choice for the fuzzifier, noting the following:

i. When $m < 3$, the algorithm becomes more concerned with the cluster whose center is closest to the point, to the extent of unequivocally assigning the point to this cluster. Therefore, there is a partial loss in the information conveyed by the distances.

ii. The worse could happen when $m > 3$; since the algorithm becomes less and less concerned, hence fuzzier, about all the distances between the point and the cluster centers, to the extent of equivocally assigning the point to all clusters. It is a complete loss in the information conveyed by the distances.

iii. We have anticipated that the relations between the data points should be best represented by the membership values, in accordance with their distances, if obtained using $m = 3$. Other values of the fuzzifier should result in degradation in clustering quality.

Projecting the silhouette results in Figure 6.2 onto the observations above:

I. As expected, the degradation in clustering quality for being fuzzier, $m > 3$, is evident in the decreasing average silhouette values, given by *GS*, as m increases. Moreover, it seems that the overhead in fuzziness introduced into the membership values by selecting $m > 3$ is worse than the loss in the representation of the relations between the data points due to defuzzification. This is evident from comparing *S* and *GS* for $m > 3$.

II. For $m < 3$, the average silhouette value increases as m decreases. This seems counterintuitive since it suggests that as crispier the membership values become as better the clustering is. However, there is a partial loss in the information about the data relations conveyed by the distances, especially for those points in overlapping areas.

III. For $m = 3$, the silhouette results of the fuzzy partition and its defuzzification also seem counterintuitive. In accordance with the associated distances, the crisp partition is doing a better job in representing the relations between the data points. Despite the fact that it does not account for all of the relations between the data points.

The unreasonable silhouette results for m3 encouraged us to investigate the fuzzy sets generated by fuzzy c-means, in order to find if they are proper for the overall clustering task. According to Dubois [26], there are three basic information-driven tasks addressed by means of fuzzy sets, namely, classification and data analysis, decision-making problems, and approximate reasoning. In such application-oriented tasks, the characterizing membership functions are no longer abstract set-theoretic notions rather they are related to some measurements of distance, frequency and cost. Accordingly, these tasks exploit three semantics of the membership values. The semantic relevant to clustering is the ‘degree of similarity’

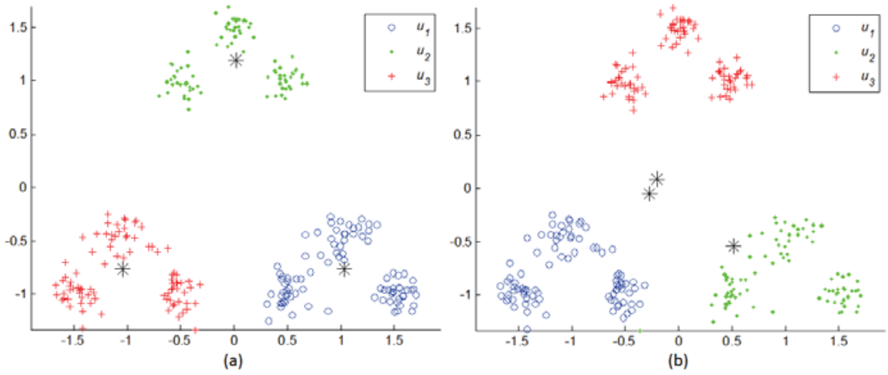

Figure 6.1: The fuzzy c-means partitions of a data set into 3 clusters using (a) $m = 3$ and (b) $m = 13$. The cluster centers are shown in black asterisks.

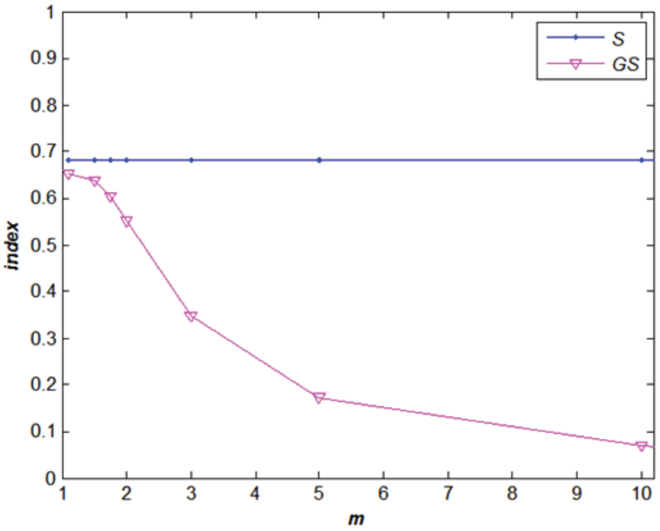

Figure 6.2: The fuzzy c-means partitions of the data set in Figure 6.1 into $c = 3$ clusters, obtained using $m = 1.1, 1.5, 1.75, 2, 3, 5, 10$ and 13, each represented by the used m value, versus the average generalized silhouette index GS. S denotes the average generalized silhouette values computed on the defuzzified partitions.

where each membership value $u_i(x_t)$ is interpreted as the degree of proximity of x_t to the prototype element of the fuzzy set u_i i.e. the associated cluster center v_i. This, prototype-based and distance-based, interpretation suits k-means and fuzzy c-means; since they are prototype-based clustering algorithms, with the objective of minimizing the distances between the cluster prototypes and the member points.

Any convex fuzzy set[9], with a membership function whose peak is at the center of the fuzzy set, should convey such semantic. In practice, triangular, trapezoidal and Gaussian membership functions typically characterize convex fuzzy sets. By means of a simple example, it can be shown that fuzzy c-means generates non-convex fuzzy sets that oppose the distance-based interpretation of the membership values as degrees of similarity. In order to visualize the membership functions that characterize the fuzzy clusters generated by fuzzy c-means, the one-dimensional data set in Figure 6.3 was partitioned into $c = 4$ clusters using $m = 3$. The membership values of the fuzzy partition and its defuzzification are plotted against the data points in Figure 6.4. Furthermore, the membership values that correspond to each fuzzy cluster are plotted separately in Figure 6.5. From the figures, it seems that fuzzy c-means has almost constructed triangular membership functions. With respect to each cluster, points farther away from the cluster center are assigned higher membership values than some of the closer points. In particular, such abnormal membership values are assigned to those farther points that lie beyond the centers of the neighbor clusters. Thus, farther points are claimed by the membership values to be more similar to the cluster members than the closer points. Clearly, such interpretation of the membership values as degrees of similarity is not distance-based. The large distances between these farther points and other points in the cluster, become intracluster distances with relatively high grades of membership. This definitely affects the overall compactness of the fuzzy partition. As part of the silhouette measure, the relatively bad compactness is measured by the large compactness distances computed from the large intracluster distances. Defuzzification reconstructs the membership functions in a manner that eliminates those large distances from the set of the intracluster distances. Accordingly, there is an improvement in the overall compactness, measured by the smaller compactness distances. The increasing behavior of the average silhouette value as m decreases from 3 in Figure 6.2, is justified by the fact that the fuzzy c-means partitions are becoming crispier.

The problem of interpreting the fuzzy membership values as degrees of similarity is better explained in the context of some linguistic variable, for example the

[9] A fuzzy set A is said to be convex, see [88], if its membership function u_A satisfies the inequality $u_A(\tau x_1 + (1-\tau)x_2) \geq \min(u_A(x_1), u_A(x_2))$ for every pair of points (x_1, x_2) where $\tau \in [0, 1]$.

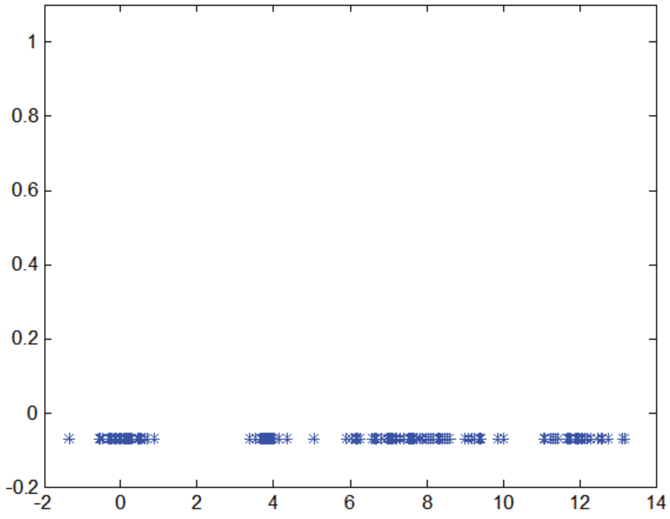

Figure 6.3: A data set of 180 points that was sampled from 4 univariate Gaussians.

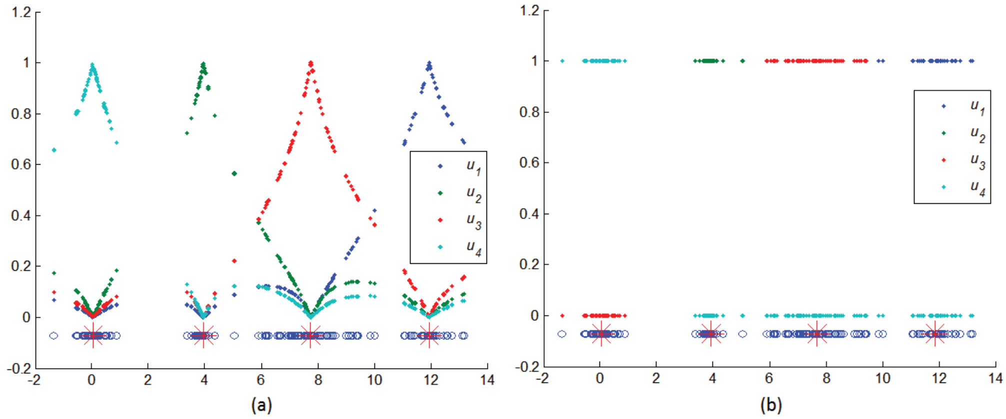

Figure 6.4: The membership values of (a) the fuzzy partition of the data set in Figure 6.3 into 3 clusters and (b) its defuzzification. The centers of the fuzzy and crisp clusters are shown in red asterisks.

variable *Age*. Recall how *relatively small* and *relatively large* become linguistic values of the linguistic variable *Distance*, each represented by a fuzzy set in the fuzzy linguistic model. Assume that the data points in Figure 6.5a, probably multiplied by 10, correspond to measurements of the numerical variable *age*, given in years. Suppose u_1 represents the linguistic value *old*, then according to the membership values, a person who is 60 years old, is claimed to be older than another person who is 80 years old. The problematic interpretation of the non-convex fuzzy clusters was also reported by Liao et al. in [49] and Goktepe et al. [33]. In the former study, fuzzy

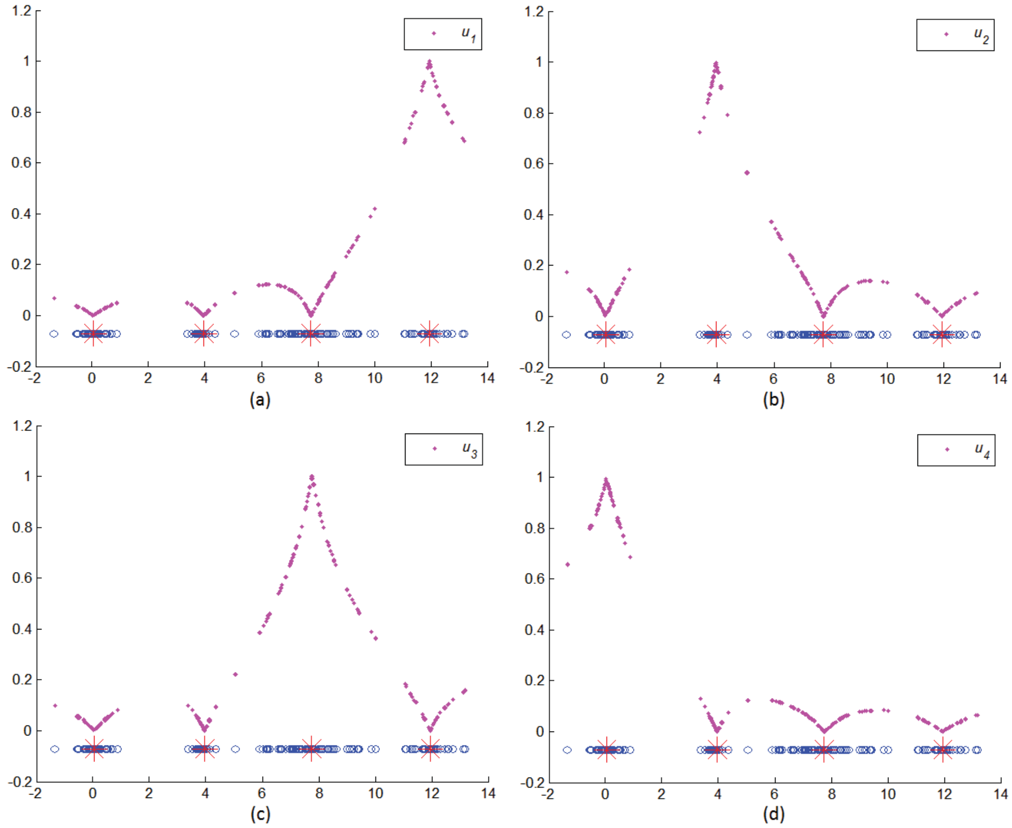

Figure 6.5: The same membership values in Figure 6.4a separated into 4 plots, one per fuzzy cluster.

c-means was used for generating fuzzy term sets on three-feature data that was extracted from radiographic data. A fuzzy term, here, is no different from a linguistic value. Fuzzy *c*-means was applied to each feature separately, therefore generating a term set for each feature. After pointing out the problem with interpreting the membership values, they proposed a variant of fuzzy *c*-means. However, it is applicable only to one-dimensional data sets for two reasons. First, it keeps adjusting the centers of the two extreme terms to the minimum and maximum values. Second, and upon convergence, the algorithm simply redistributes the non-convex membership values, among the surrounding terms; thus it requires finding the left and right term centers relative to the data point. In the latter study, by Goktepe et al., they observed the abnormal, or using their exact words (unusual, non-convex and subnormal), behavior of the membership functions generated by fuzzy *c*-means in the application of clustering soil samples. Rather than attributing the problem to fuzzy c-means, they explained the strange results by the uncertainty in soil parameters and the presence of extreme data points.

Toward the end of generating a fuzzy partition that might have larger silhouettes than its defuzzification, we devised a naïve clustering algorithm to cluster the data set in Figure 6.3, rather than using the variant of fuzzy c-means mentioned above. The naïve algorithm is given in Algorithm 6.1. The algorithm, first, generates the cluster centers using k-means. Each fuzzy membership value u_{ij} is computed by a univariate Gaussian function of the distance between the cluster center v_i and the data point x_j. The Gaussian function uses a zero mean, and for its variance parameter it uses a distance of some specified percentile rank, being computed on the center-to-point distances relevant to the cluster center. Two fuzzy partitions of the data set in Figure 6.3 into three clusters were obtained by fuzzy c-means and the nave algorithm, using respectively $m = 3$ and $= 25$. Let PN1 and PN2 denote respectively the fuzzy partition, by the naïve algorithm, and its defuzzification, shown in Figure 6.6. Let PF1 and PF2 denote respectively the fuzzy c-means partition and its defuzzification, shown in Figure 6.7. The average

Algorithm 6.1: NAÏVE FUZZY CLUSTERING

```
Parameters: $c$ (number of clusters)
            $\alpha$ (a percentage which specifies a percentile distance value)
Input     : $X$ (set of $n$ $p$-dimensional vectors)
Output    : U (the membership matrix a partition of $c$ clusters)
            $V$ (the associated set of cluster centers)
$V \leftarrow V_c$                    // $V_c$:  $c$ cluster centers as returned by $k$-means

for $i \leftarrow 1$ to $c$ do
    for $j \leftarrow 1$ to $n$ do
        compute $d_{ij} = d(v_i, x_j))$
    $\sigma_i^2 \leftarrow$ the distance at the $\alpha^{th}$ percentile among $\{d_{ij}\}_{j=1}^{n}$;
for $j \leftarrow 1$ to $n$ do
    for $i \leftarrow 1$ to $c$ do
        $u_{ij} \leftarrow e^{\frac{-(d_{ij}^2)}{\sigma_i^2}}$
return U, $V$
```

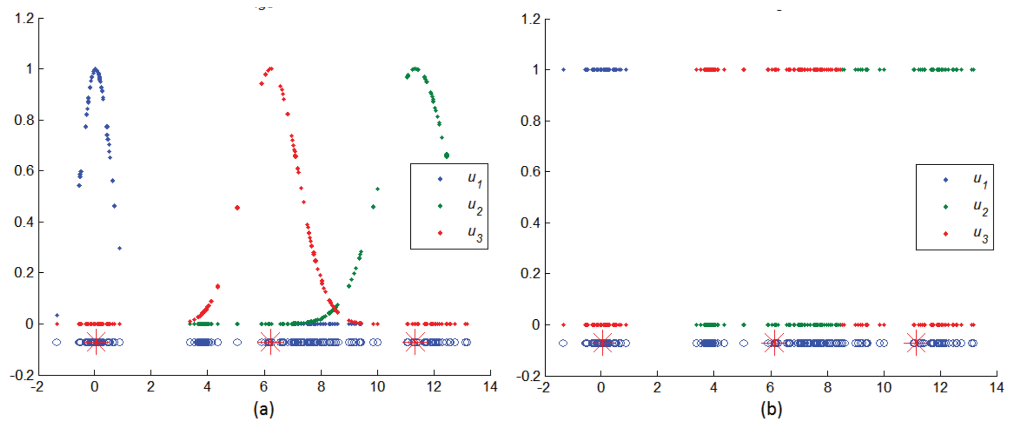

Figure 6.6: The membership values of (a) the fuzzy partition of the data set in Figure 6.3 into 3 clusters obtained by the naïve algorithm, Algorithm 6.1, using $\alpha = 25$, and (b) its defuzzification. The centers of the fuzzy and crisp clusters are shown in red asterisks.

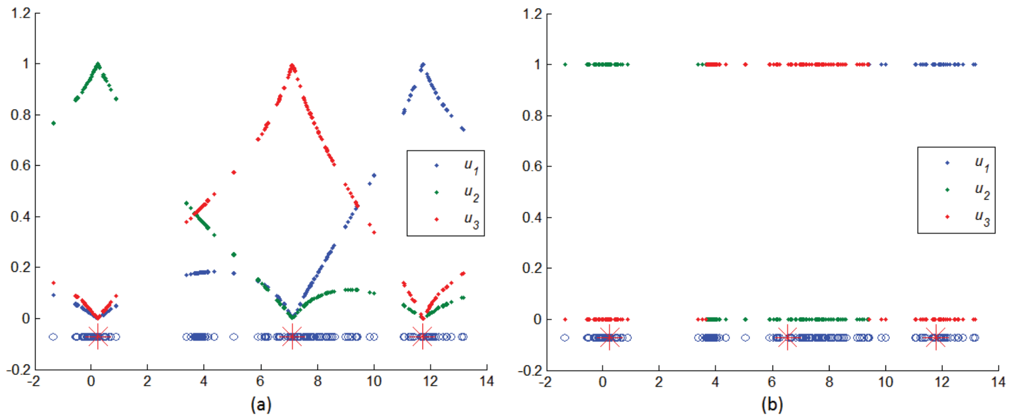

Figure 6.7: The membership values of (a) the fuzzy partition of the data set in Figure 6.3 into 3 clusters obtained by fuzzy c-means using $m = 3$, and (b) its defuzzification. The centers of the fuzzy and crisp clusters are shown in red asterisks.

silhouette values of PN1, PN2, PF1 and PF2 are respectively 0.677, 0.633, 0.456 and 0.633. Hereby, the fuzzy membership values generated by the naïve algorithm are better in representing the similarities and dissimilarities between the data points, in accordance with the associated pairwise distances, than their defuzzified counterparts. In contrast, defuzzifying the partition obtained by fuzzy c-means results in a better overall clustering quality.

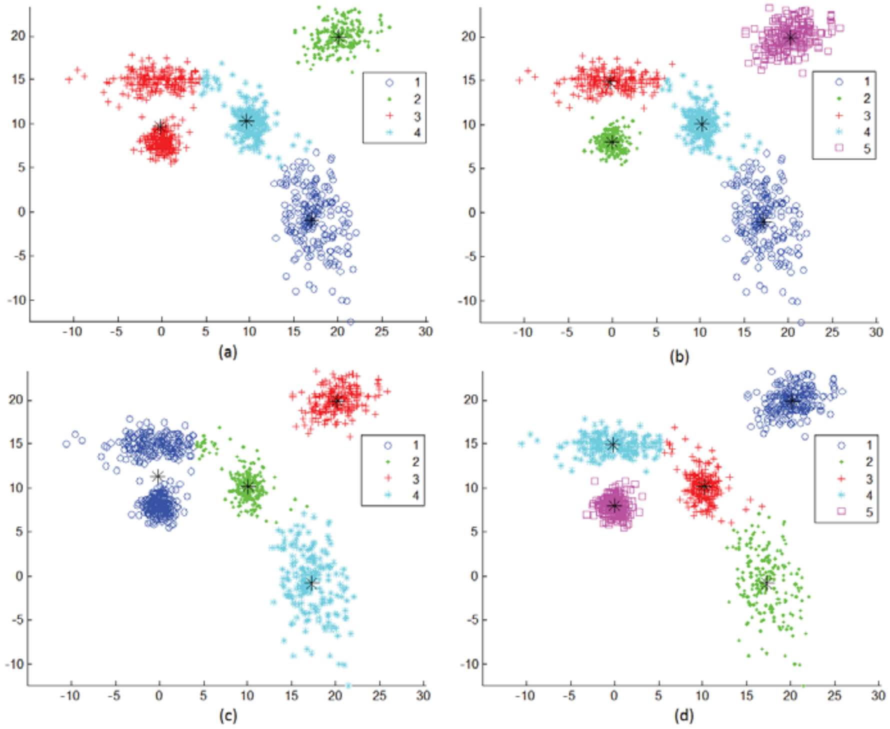

Figure 6.8: The fuzzy c-means partitions of a data set into (a) 4 and (b) 5 clusters, obtained using $m = 3$. Another two fuzzy partitions into (c) 4, and (d) 5 clusters by the naïve clustering algorithm, Algorithm 6.1, obtained using $\alpha = 25$.

To confirm further that the fuzzy partitions, generated by the naïve clustering algorithm, could be better than their defuzzified partitions, a number of partitions of the two-dimensional data set in Figure 6.8 were obtained by the naïve algorithm and fuzzy c-means, using respectively $\alpha = 25$ and $m = 3$. Let PN and Pn denote respectively the partitions by the naïve algorithm and the associated defuzzified partitions. Similarly, let PF and Pf denote respectively the fuzzy c-means partitions and the associated defuzzified partitions. From the silhouette results in Figure 6.9a, defuzzifying the partitions generated by fuzzy c-means still improves the clustering results. Conversely, and for this particular data set, defuzzification results in no improvement in the clustering results by the naïve algorithm, rather degradation is observed; see Figure 6.9b.

Fuzzy c-means is mathematically a sound algorithm and this chapter is by no means demoting its use in clustering. However, it is worthwhile noting that the non-convex fuzzy clusters generated by the algorithm are less convenient for the

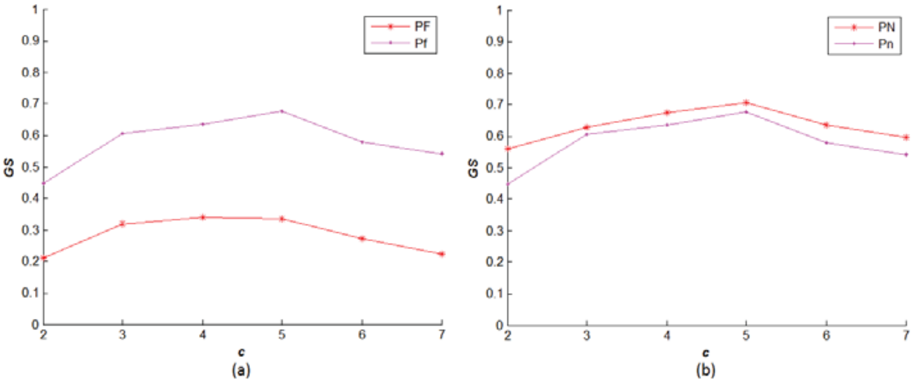

Figure 6.9: The fuzzy c-means partitions obtained using $m = 3$, denoted by PF, of the data set in Figure 6.8 obtained using $m = 3$, and the associated defuzzified partitions, denoted by Pf, versus (a) the average generalized silhouette value GS. The fuzzy partitions obtained by the naïve algorithm, Algorithm 6.1, using $\alpha = 25$, denoted by PN, of the same data set, and the associated defuzzified partitions, denoted by Pn, versus (b) the average generalized silhouette values GS.

overall clustering task than their defuzzified counterparts. This was illustrated in a number of examples throughout this thesis, by comparing the average silhouette values of the fuzzy and defuzzified partitions. This suggests the use of k-means; as it generates the same crisp partitions as the defuzzified fuzzy c-means partitions. The generation of the non-convex fuzzy clusters is due to the strive for completeness in the membership values. Completeness is in the sense of each data point having a total membership of unity in the fuzzy partition, as imposed by 2.10b. Fortunately, it is possible to generate fuzzy partitions where defuzzification becomes unnecessary for improving the clustering results; therefore, the partitions might be better than the k-means partitions into the same number of clusters. This was demonstrated by the naïve clustering algorithm. The success of the algorithm is due to the fact that it does not require the total membership of each individual point to be unity, and it employs a membership function that characterizes convex fuzzy sets. However, it is still an ad hoc algorithm and sometimes defuzzification improves its results.

CHAPTER 7
THE SILHOUETTE MEASURE IN FEATURE SELECTION FOR CLASSIFICATION

Classification, although treated in a supervised setting rather than unsupervised, share something in common with clustering; as they deal with data points that belong to groups. A group of data points is known as a *class* in classification, whereas the group is called a *cluster* in clustering. In classification, the points which constitute a training data set are associated with class labels. The task becomes learning a classifier on the training data, or more specifically, on the feature representation and the labels of the data points. The success of the classification task is determined by the extent to which the classifier generalizes to stray data points whose class is unknown; as it predicts the class. If it generalizes well then the classifier becomes trustworthy in predicting the class of any given data point, whether the point was seen or unseen during the training of the classifier. Since the classifier, in some way or another with more or less success, models the distinctive aspects of the classes, classification is an abstraction operation in the same sense that was pointed out earlier by Bellman, Kalaba, and Zadeh. In particular, support vector machines (SVM) classification is a large-margin method concerned with hyperplanes that separate the classes, consider [8] for a tutorial on SVMs. Such class separation is inherent in the structure of the training data set. A structure is in the sense of the relations between the data points, embodied in a feature space, therefore, no different in principle from the graph structure mentioned earlier in the context of clustering. If the classes are not well separated by the feature representation of the data points, the case of the structure being irrelevant to the classes, then a kernel can reestablishes the relations in a manner that complies with the class labels.

Related to the above discussion are the concepts of the *target classifier* and the *generalization error* of the learned classifiers [56]. The target classifier perfectly predicts the class of any data point. The generalization error is defined in terms of the *approximation* and *estimation errors*. The approximation error is attributed to the *model complexity* of the classifier i.e. the model's learning capacity. This capacity is limited by the finite number of model parameters. The model, by the different combinations of parameter values, defines a family of concrete classifiers. The approximation error is the prediction error made by the concrete classifier that best approximates the target classifier. In contrast, the difference in predictions between this best approximating classifier and any learned classifier becomes the estimation error. This error is attributed to *sample complexity* i.e. to the size of the training data set necessary to ensure small estimation error. Thus, what can be learned is limited by the model, by its finite number of parameters. And, estimating the parameter values that correspond to the best learnable classifier is limited by the data, by its finite size. As more complex the model is, as better the chance of learning the perfect target classifier. But, as more complex it gets as more data is needed to estimate the best combination of parameter values. The increase in data size incurs considerable training time. The trade-off between the model complexity and sample complexity is similar in sense to the dilemmatic *bias-variance* trade-off [32]. A model of low complexity may misrepresent the distinctive aspects of the classes where the classification results biased toward the model aspects. On the other hand, if there are no enough data to estimate the model parameters, *over-fitting* occurs where the prediction error has a high variance.

How does this relate to SVM? Simply by noting that the equation of the separating hyperplane grows in parameters as the data set grows in features. For a finite training data set, the *kernel trick* [96] has the potential of reducing the approximation error. The kernel implicitly, and nonlinearly, maps the data into a higher dimensional feature space i.e. an increase in the number of parameters, perceived as an increase in model complexity. Alternatively, feature selection has the potential of reducing the estimation error by selecting relevant features from among all possible features; which results in a lower dimensional feature space (subspace). Consider [46] for a discussion of two methods of feature selection, namely *filters* and *wrappers*, and a number of selection criteria. The relevance among the features and classes is established upon *correlation*, *information*, *consistency*, or *distance* measures [22]. But since SVM classification exploits the separation between the classes, it seems reasonable to use the silhouette measure, as a measure of compactness and separation, in selecting the features. To test such potential of the silhouette measure, we used the data from the KDD Cup 2008 challenge.

7.1 Data Set

KDD Cup 2008 is a challenge that focused on the early detection of breast cancer from X-ray images of the breast. A detailed description of the challenge data set is found in [62]. The data consists of various region of interest (ROI) identified in each of the left and right breast medical images, each candidate ROI being described by 117 features. No information was publicly disclosed about the nature of the features and what they identify. A separate file provides the class labels which identify each candidate ROI image as *malignant* or *benign*. ROIs associated with normal individuals were presumed to be benign. In addition to the class labels, the file also has Patient ID, ROI coordinates, and other features. However, the winning team found that Patient ID is a predictive feature which leaks information about the target classes [61]. Patient ID is clearly unrealistic feature for diagnostic prediction. Except for the labels, none of the additional features given in the file were used in our experiment. In total, there are 12787 candidate ROIs used in the experiment, 78 of which are malignant. The data set was sampled from 102,294 candidate ROIs from the whole KDD Cup 2008 data set in a manner that preserved the class ratio.

7.2 Error Measures

The performance of the trained classifier, whether SVM-based or not, is assessed by the error rate of its prediction. With respect to error measurement, it is important to choose error measures that are *class-wise*, in the sense of measuring the overall performance with respect to the target classes rather than the whole dataset in an indistinctive manner. This is a particularly important issue in the case of imbalanced data set [76–78]. By way of example, in a binary classification problem based on a set in which 95% of the points in one class, a classification rule assigning each data point to the major class commits a small 5% error rate, in spite of completely missing the minor class. Two measures for assessing a classifier performance can be extracted from the confusion matrix associated to it, and shown in Table 7.1. The *true positive* rate (*TP*), and *false positive* rate (*FP*), or (1 − *specificity*), can be computed for each class, reducing an m-class problem, if necessary, to m binary classification problems, denoting one particular class as the positive class and the remaining $(m-1)$ classes collectively as the negative class. *TP* and *FP* are defined in terms of the entries in the confusion matrix as shown in equations (7.1) and (7.2). The classifier performance is assessed by computing the average *TP* and *FP* over all classes.

Table 7.1: The confusion matrix. The positive and negative classes are confused for each other in predicting the class of $(b+c)$ points.

		Predicted Class	
		positive	*negative*
True Class	*positive*	a	b
	negative	c	d

$$TP = \frac{a}{a+b} \tag{7.1}$$

$$FP = \frac{c}{c+d} \tag{7.2}$$

7.3 Cross Validation

Rather than being concerned with a particular concrete classifier as we compute *TP* and *FP*, cross validation goes beyond the classifier to validate the underlying model, to find if it produces classifiers that generalize to an independent data set. In other words, it finds if the parameters can be estimated reliably from available data which results in over-fitting free classification. It is a *bias-variance estimator* that repeats training and testing classifiers on data. In each repetition, the training sample is different from the test sample. The average performance of the classifiers over the repetitions estimates the bias and variance of the model (approximation and estimation errors). In *stratified* k-fold cross validation, the data set is partitioned into k folds with each fold having the same class ratio as the whole data set. In our experiment, we use 7-fold cross validation.

7.4 Center-Wise Silhouette-Based Feature Selection

The center-wise silhouette-based filter is the best-first search algorithm whose pseudocode is given in Algorithm 7.1. The search process starts with the full set of features and maintains a list of alternative subsets (search branches) for later consideration, denoted by *OPEN*. Feature subsets with maximum average center-wise silhouette values are picked up for expansion from *OPEN*. By means of backward elimination, new alternative feature subsets are generated from the picked subset, and added to *OPEN*. The algorithm updates its records of the best feature subset, denoted by *BEST*, in the process. The parameter ϵ controls the update of *BEST*; as it skips those subsets that result in an insignificant increase in the average silhouette value. The list *OPEN* may be implemented as a priority queue where feature subsets are ordered according to the average center-wise silhouette value. The search is terminated once the list of alternative subsets *OPEN* is empty or no update has occurred in the last M iterations. Point-wise silhouettes are more accurate in evaluating the feature subsets but the center-wise silhouettes definitely facilitate much faster search.

Algorithm 7.1: SILHOUETTE-BASED FILTER for feature selection

```
**Parameters**: $M$ (maximum number of iterations with no change on
                $BEST$)
                $\epsilon$ (minimum improvement required to update $BEST$)
**Input**       : $X$ (set of $n$ $p$-dimensional vectors)
                U (the indicator matrix of the partition of $X$ into *malignant*
and *benign*)
                $f(X, \mathrm{U}, w)$ (returns the average center-wise silhouette value
over the features in $w$)
**Output**      : $BEST$ (the features found to result in the largest value of $f$)
$BEST \leftarrow$ the whole set of features
$OPEN \leftarrow \{BEST\}$
$CLOSED \leftarrow \{\}$
$m \leftarrow 0$
while $m < M$ and $OPEN \neq \{\}$ do
    $q \leftarrow \mathrm{argmax}_{w \in OPEN} f(X, \mathrm{U}, w)$
    $OPEN \leftarrow OPEN \setminus \{q\}$
    if $f(X, \mathrm{U}, q) - \epsilon > f(X, \mathrm{U}, BEST)$ then
        $BEST \leftarrow q$
        $m \leftarrow 0$
    else
        $m \leftarrow m + 1$
    generate each possible subset $\ddot{q}$ from $q$ by eliminating a single feature
    for each child subset $\ddot{q}$ do
        if $\ddot{q} \notin OPEN$ and $\ddot{q} \notin CLOSED$ then
            $OPEN \leftarrow OPEN \cup \{\ddot{q}\}$
    $CLOSED \leftarrow CLOSED \cup \{q\}$
return BEST
```

7.5 Results

The data set of 12787 candidate ROIs was partitioned into $k = 7$ folds. Thus, there were 7 iterations as part of 7-fold cross validation. At each iteration:

- One fold was used for:

    i. Feature selection.

    ii. Training two SVM classifiers: one considering the whole set of features and another using the features selected by the filter in Algorithm 7.1.

- The remaining 6 folds were used for testing both classifiers.

During the whole cross validation process, each fold was used exactly once as a training fold. The SVM library under Matlab Statistics toolbox was used for training the linear SVM classifiers and using them in predictions. The cross validation results are given in Table 7.2. Each row in the table gives the detailed results specific to one of the cross validation iterations; as follows:

- The feature subset that was selected by Algorithm 7.1; for example, the subset $\{1, 12\}$ in the first row selected in the first iteration. Note that $\{1, \ldots, 117\}$ is the whole set of features.

- The *TP* and *FP* rates of the classifier trained on the whole set of features over the training and test samples.

- The *TP* and *FP* rates of the classifier trained on the selected features over the training and test samples.

- The number of malignant and benign points in the training and test samples.

The last row summarizes the results by taking the average of the entries in the table over the different iteration stages, accumulated from the previous rows. But rather than showing a feature subset, it shows the average number of selected features.

According to the results in the last row of Table 7.2, the classifiers that were trained on the whole set of features suffer from the problem of over-fitting the training samples i.e. high variance. Obviously the number of malignant points is too small, 11.14 points on average, in the training samples compared to the number of features, the 117 features. With respect to the training samples, the average *TP* and *FP* over the two classes (averages of averages) are respectively

$$\begin{aligned}\bar{TP}_{\text{all,train}} &= \frac{1+1}{2} = 1\\ \bar{FP}_{\text{all,train}} &= \frac{0+0}{2} = 0\end{aligned}$$

Whereas, with respect to the test samples, they are

$$\begin{aligned}\bar{TP}_{\text{all,test}} &= \frac{0.26+0.99}{2} = 0.63\\ \bar{FP}_{\text{all,test}} &= \frac{0.01+0.74}{2} = 0.38\end{aligned}$$

Take the difference between both rates, as an overall measure, over the training and test samples as follows

$$\begin{aligned}D_{\text{all,train}} &= \bar{TP}_{\text{all,train}} - \bar{FP}_{\text{all,train}}\\ &= 1-0\\ &= 1\\ D_{\text{all,test}} &= \bar{TP}_{\text{all,test}} - \bar{FP}_{\text{all,test}}\\ &= 0.63-0.38\\ &= 0.25\end{aligned}$$

Over-fitting the training samples is clear from comparing $D_{\text{all,train}}$ and $D_{\text{all,test}}$. The change in the overall measure is 0.75, hence classifiers that were trained on the whole set of features did not generalize well to the test samples. Now consider the results of those classifiers that were trained on the selected feature subsets. Using again the last row in Table 7.2, compute

$$\begin{aligned}\bar{TP}_{\text{feat,train}} &= \frac{0.86+0.82}{2} = 0.84\\ \bar{FP}_{\text{feat,train}} &= \frac{0.18+0.14}{2} = 0.16\\ \bar{TP}_{\text{feat,test}} &= \frac{0.65+0.82}{2} = 0.74\\ \bar{FP}_{\text{feat,test}} &= \frac{0.18+0.35}{2} = 0.27\end{aligned}$$

Taking the difference between TP and FP with respect to the training and test samples gives

$$\begin{aligned}
D_{\text{feat,train}} &= \bar{TP}_{\text{feat,train}} - \bar{FP}_{\text{feat,train}} \\
&= 0.84 - 0.16 \\
&= 0.68 \\
D_{\text{feat,test}} &= \bar{TP}_{\text{feat,test}} - \bar{FP}_{\text{feat,test}} \\
&= 0.74 - 0.27 \\
&= 0.47
\end{aligned}$$

The change in the overall measure is only 0.21. Thus it shows that the classifiers trained on the selected features were more stable in their predictions than the ones obtained on the whole set of features. Moreover, the use of the selected features has boosted the prediction rate of malignant cases from 0.26 to 0.65, a critical improvement to the detection of cancer in early stages. Reducing the number of features has reduced SVM model complexity, which resulted in a better estimation of the distinctive aspects of the classes, especially in regards to the minority malignant class. The experiment is an evident of the potential of the silhouette measure in improving the classification results for highly imbalanced data, by reducing the effect of over-fitting the training data.

Table 7.2: Detailed and average results of 7-fold cross validation, comparing the performance of linear SVM classifiers trained on the whole set of features versus feature subsets obtained with the silhouette-based filter in Algorithm refalg:7-1, $\epsilon = 0.02$ and $M = 35$. In each iteration stage, one fold was used for training, and six folds were used for testing.

Iteration	**Features**	**One Training Fold**						**Six Test Folds**					
		Malignant	**TP**	**FB**	**Benign**	**TP**	**FP**	**Malignant**	**TP**	**FP**	**Benign**	**TP**	**FP**
1	$\{ 1, \ldots, 117 \}$	12	**1**	**0**	1816	**1**	**0**	66	0.32	**0.01**	10893	**0.99**	0.68
	$\{1, 12\}$	points	0.92	0.16	points	**0.84**	0.08	points	**0.59**	0.17	points	0.83	**0.41**
2	$\{ 1, \ldots, 117 \}$	11	**1**	**0**	1815	**1**	**0**	67	0.19	**0.01**	10894	**0.99**	0.81
	$\{ 1, 20, 12 \}$	points	1	0.21	points	0.79	0.00	points	**0.58**	0.21	points	0.79	**0.42**
3	$\{ 1, \ldots, 117 \}$	11	**1**	**0**	1816	**1**	**0**	67	0.37	**0.01**	10893	**0.99**	0.63
	$\{ 3, 4 \}$	points	0.91	0.23	points	0.77	0.09	points	**0.78**	0.24	points	0.76	**0.22**
4	$\{ 1, \ldots, 117 \}$	11	**1**	**0**	1815	**1**	**0**	67	0.19	**0.01**	10894	**0.99**	0.81
	$\{ 1, 5, 6 \}$	points	0.82	0.12	points	0.88	0.18	points	**0.67**	0.12	points	0.88	**0.33**
5	$\{ 1, \ldots, 117 \}$	11	**1**	**0**	1816	**1**	**0**	67	0.21	**0.01**	10893	**0.99**	0.66
	$\{ 3, 4 \}$	points	0.91	0.14	points	0.86	0.09	points	**0.75**	0.14	points	0.86	**0.25**
6	$\{ 1, \ldots, 117 \}$	11	**1**	**0**	1816	**1**	**0**	67	0.34	**0.01**	10893	**0.99**	0.66
	$\{ 3, 4 \}$	points	0.82	0.2	points	0.8	0.18	points	**0.76**	0.2	points	0.8	**0.24**
7	$\{ 1, \ldots, 117 \}$	11	**1**	**0**	1815	**1**	**0**	67	0.16	**0.01**	10894	**0.99**	0.84
	$\{ 1, 2, 59 \}$	points	0.64	0.17	points	0.83	0.36	points	**0.39**	0.16	points	0.84	**0.61**
AVG	$\{ 1, \ldots, 117 \}$	11.14	**1**	**0**	1815.57	**1**	**0**	66.86	0.26	**0.01**	10893.43	**0.99**	0.74
	2.43 features	points	0.85	0.18	points	0.82	0.14	points	**0.65**	0.18	points	0.82	**0.35**

CHAPTER 8
CONCLUDING REMARKS AND FUTURE WORK

Clustering is ubiquitous in applications because it is efficient in dealing with data complexity. To be more specific, clustering is supposed to reduce the similarity relations between the data points in the form of clusters, as intuitively stated by the common definition of clustering. In other words, clustering translates the distance-based similarity relations into the readily comprehended cluster-based similarity relations. For example, two documents in the same cluster, similar in this sense, are also similar in the sense of being specific to the same topic. Dissimilarity may be interpreted by an example in a similar fashion. In regards to similarity, the animal classification of Celestial Emporium of Benevolent Knowledge seems a 'bizarre' classification because each constituting class (cluster) entails its own, probably abnormal, notion of similarity. Similarly, a 'non-bizarre' partition into arbitrary-shape clusters also entails different notions of similarity. Certainly, it seems more natural to cluster the animals according to, for example, appearance, activity, or both, as one unified criterion; such clustering appears in [59]. Clearly, the notion of similarity, so the distance measure, is at the core of clustering. A meaningful partition into clusters that represent, for example, the subjects of a collection of news articles, as in business or sports, requires choosing a relevant distance measure; so becomes the graph structure. It is more probable for a clustering which operates on the fine-grain pairwise distances to ensure the goal fulfillment of the task with respect to each individual point. However, due to uncertainty in inferring the pairwise relations from the input pairwise distances, k-means and fuzzy c-means operate on the coarse-grain center-to-point distances. It is worthwhile noting that the potential use of more complex relations in clustering is investigated in [1, 95]. Rather than the simple dyadic (pairwise) relations, triadic, tetradic or higher relations are considered; as they define hypergraph structures on the data set.

In this thesis, a unifying relational perspective to the problems of clustering and cluster validity is delivered by the developed framework. The perspective is from a number of explicitly defined distance-based and cluster-based pairwise relations that entail similarity. Two notions of similarity arise from the common definition of clustering: with one being an input to the task, measured by a distance measure. Whereas, the other is in the context of clusters; two points are similar if they belong to the same cluster, otherwise dissimilar. These two notions of similarity are formalized by the distance-based and cluster-based relations, defined as part of the relational framework. One can reason with the relations about the two problems of clustering and cluster validity, in conjunction with the pairwise distances. By such reasoning, the silhouette measure was successfully generalized from crisp to fuzzy partitions. The generalization is as natural as the generalization of crisp sets and ordinary logic by their fuzzy counterparts. The results by any clustering algorithm whether crisp, fuzzy, or even probabilistic can now be evaluated by the measure, using, if necessary, proper logical devices. Accordingly, the measure becomes a reference to select between any partitions. It identifies the partition which best represents the pairwise relations between the data points in accordance with their pairwise distances. Such powerful feature of the silhouette measure exposed a problem with fuzzy c-means; as it generates some counterintuitive membership values, counterintuitive in regards to the distances measure. Such results are due to the non-convex fuzzy clusters always generated by the algorithm. The associated troublesome membership values are avoided by selecting $m \approx 1$ or remedied by defuzzification. In situations where the computational resources are of a major concern, the center-wise silhouettes are at our disposal. The notion of center-wise silhouettes, as opposed to point-wise, is new; as it has emerged within our framework. The results obtained by the experiment in chapter seven demonstrated a fact about the tasks of clustering and classification; as they have a couple of things in common. On one hand, they deal with groups of the data points i.e. clusters or classes. On the other hand, they exploit the relations between the data points to accomplish the goal of the task. As was shown in the same experiment, the task of highly imbalanced classification can significantly benefit from a silhouette-based feature selection.

We highlight the following problems for future research:

1. To mathematically reformulate the fuzzy c-means problem but without enforcing a total membership of unity; this permits the identification of outliers. The success of such attempt is determined by whether the reformulated problem is tractable by convex programming.

2. Or alternatively, to devise a sophisticated clustering algorithm that generates convex fuzzy clusters, possibly by improving Algorithm 6.1.

3. To investigate further the clustering results within the framework by constructing plots of the pairwise distances versus the membership values of the cluster-based relations, as means of illustrating those properties, with respect to the distance measure and the partition, addressed in Blum's framework.

4. To attempt a silhouette-based heuristic clustering.

5. To investigate whether highly dimensional classification tasks can benefit from a silhouette-based feature selection. Noting that, a highly imbalanced classification task is probably a highly dimensional classification task with respect to the minority classes.

6. To publicly deliver an efficient library for the computation of the silhouette measure, either point-wise or center-wise, that scales well to large data sets.

7. To see if it is convenient to approach the problem of graph clustering within the framework, adapting the framework if necessary.

REFERENCES

[1] Sameer Agarwal, Jongwoo Lim, Lihi Zelnik-Manor, Pietro Perona, David Kriegman, and Serge Belongie. Beyond pairwise clustering. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 838–845. IEEE, 2005.

[2] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. Np-hardness of euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–248, 2009.

[3] anonymous. A short fuzzy logic tutorial. Technical report, Updated 4/8/2010), Available at: http://www.cs.bilkent.edu.tr/ bulbul/depth/fuzzy.pdf, 2010.

[4] Geoffrey H Ball and David J Hall. Isodata, a novel method of data analysis and pattern classification. Technical report, DTIC Document, 1965.

[5] Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning distance functions using equivalence relations. In *ICML*, volume 3, pages 11–18, 2003.

[6] Richard Bellman, Robert Kalaba, and L Zadeh. Abstraction and pattern classification. *Journal of Mathematical Analysis and Applications*, 13(1):1–7, 1966.

[7] Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. Clustering gene expression patterns. *Journal of computational biology*, 6(3-4):281–297, 1999.

[8] Asa Ben-Hur, Cheng Soon Ong, Sören Sonnenburg, Bernhard Schölkopf, and Gunnar Rätsch. Support vector machines and kernels for computational biology. *PLoS computational biology*, 4(10):e1000173, 2008.

[9] Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer, 2006.

[10] James C Bezdek. *Pattern recognition with fuzzy objective function algorithms.* Kluwer Academic Publishers, 1981.

[11] James C Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy¡ i¿ c¡/i¿-means clustering algorithm. *Computers & Geosciences*, 10(2):191–203, 1984.

[12] James C Bezdek, Richard J Hathaway, Michael J Sabin, and William T Tucker. Convergence theory for fuzzy c-means: counterexamples and repairs. *Systems, Man and Cybernetics, IEEE Transactions on*, 17(5):873–877, 1987.

[13] James Christian Bezdek. Fuzzy mathematics in pattern classification. 1973.

[14] James C Bezdek. Cluster validity with fuzzy sets. 1973.

[15] Avrim Blum. Thoughts on clustering. In *NIPS Workshop on Clustering Theory*, 2009.

[16] David Bollier and Charles M Firestone. *The promise and peril of big data.* Aspen Institute, Communications and Society Program Washington, DC, USA, 2010.

[17] Jorge Luis Borges. *Other Inquisitions, 1937-1952.* University of Texas Press, 1964.

[18] Brad Brown, Michael Chui, and James Manyika. Are you ready for the era of big data? *McKinsey Quarterly*, 4:24–35, 2011.

[19] Ricardo JGB Campello and Eduardo R Hruschka. A fuzzy extension of the silhouette width criterion for cluster analysis. *Fuzzy Sets and Systems*, 157(21):2858–2875, 2006.

[20] Ronald A Cole, Joseph Mariani, Hans Uszkoreit, Annie Zaenen, Giovanni Varile, Antonio Zampolli, Ron Cole, and Victor Zue. Survey of the state of the art in human language technology. 1995.

[21] Sanjoy Dasgupta and Yoav Freund. Random projection trees for vector quantization. *Information Theory, IEEE Transactions on*, 55(7):3229–3242, 2009.

[22] Manoranjan Dash and Huan Liu. Feature selection for classification. *Intelligent data analysis*, 1(3):131–156, 1997.

[23] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[24] Luca Donetti and Miguel A Munoz. Detecting network communities: a new systematic and efficient algorithm. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(10):P10012, 2004.

[25] Didier Dubois and Henri Prade. A class of fuzzy measures based on triangular norms a general framework for the combination of uncertain information. *International Journal Of General System*, 8(1):43–61, 1982.

[26] Didier Dubois and Henri Prade. The three semantics of fuzzy sets. *Fuzzy sets and systems*, 90(2):141–150, 1997.

[27] JC Dunn. Indices of partition fuzziness and the detection of clusters in large data sets. *Fuzzy Automata and Decision Processes, Elsevier, New York*, 1977.

[28] Joseph C Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. 1973.

[29] Liat Ein-Dor, Or Zuk, and Eytan Domany. Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proceedings of the National Academy of Sciences*, 103(15):5923–5928, 2006.

[30] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996.

[31] Chris Fraley and Adrian E Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

[32] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

[33] AB Goktepe, S Altun, and A Sezer. Soil clustering by fuzzy c-means algorithm. *Advances in Engineering Software*, 36(10):691–698, 2005.

[34] John C Gower and GJS Ross. Minimum spanning trees and single linkage cluster analysis. *Applied statistics*, pages 54–64, 1969.

[35] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. *Information systems*, 25(5):345–366, 2000.

[36] Isabelle Guyon, Ulrike Von Luxburg, and Robert C Williamson. Clustering: Science or art. In *NIPS 2009 Workshop on Clustering Theory*, 2009.

[37] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145, 2001.

[38] David A Hanauer, Daniel R Rhodes, and Arul M Chinnaiyan. Exploring clinical associations using -omics based enrichment analyses. *PLoS One*, 4(4):e5203, 2009.

[39] Pierre Hansen and Brigitte Jaumard. Cluster analysis and mathematical programming. *Mathematical programming*, 79(1-3):191–215, 1997.

[40] Richard J Hathaway, John W Davenport, and James C Bezdek. Relational duals of the¡ i¿ c¡/i¿-means clustering algorithms. *Pattern recognition*, 22(2):205–212, 1989.

[41] Lawrence Hubert. Monotone invariant clustering procedures. *Psychometrika*, 38(1):47–62, 1973.

[42] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[43] Thomas Jefferson. Thomas jefferson on politics and government. *University of Virginia. Available online at etext. virginia. edu/jefferson/quotations/jeffcont. htm. Accessed in May 2oo5*, 2007.

[44] Anupam Joshi and Raghu Krishnapuram. Robust fuzzy clustering methods to support web mining. In *Proc. Workshop in Data Mining and knowledge Discovery, SIGMOD*, pages 15–1. Citeseer, 1998.

[45] Jon Kleinberg. An impossibility theorem for clustering. *Advances in neural information processing systems*, pages 463–470, 2003.

[46] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[47] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research*, 5:27–72, 2004.

[48] Douglas Laney. 3-d data management: Controlling data volume, velocity and variety. *META Group Research Note, February*, 6, 2001.

[49] T. Warren Liao, Aivars K Celmins, and Robert J Hammell II. A fuzzy c-means variant for the generation of fuzzy term sets. *Fuzzy Sets and Systems*, 135(2):241–257, 2003.

[50] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[51] Stuart Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[52] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA, 1967.

[53] Sam Madden. From databases to big data. *Internet Computing, IEEE*, 16(3):4–6, 2012.

[54] Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan. The planar k-means problem is np-hard. In *WALCOM: Algorithms and Computation*, pages 274–285. Springer, 2009.

[55] Maged Michael, Jose E Moreira, Doron Shiloach, and Robert W Wisniewski. Scale-up x scale-out: A case study using nutch/lucene. In *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*, pages 1–8. IEEE, 2007.

[56] Partha Niyogi and Federico Girosi. On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions. *Neural Computation*, 8(4):819–842, 1996.

[57] Malay K Pakhira, Sanghamitra Bandyopadhyay, and Ujjwal Maulik. Validity index for crisp and fuzzy clusters. *Pattern recognition*, 37(3):487–501, 2004.

[58] Nikhil R Pal and James C Bezdek. On cluster validity for the fuzzy c-means model. *Fuzzy Systems, IEEE Transactions on*, 3(3):370–379, 1995.

[59] Elias Pampalk, Werner Goebl, and Gerhard Widmer. Visualizing changes in the structure of data for exploratory feature selection. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 157–166. ACM, 2003.

[60] Tan Pang-Ning, Michael Steinbach, Vipin Kumar, et al. Introduction to data mining. In *Library of Congress*, 2006.

[61] Claudia Perlich, Prem Melville, Yan Liu, Grzegorz Świrszcz, Richard Lawrence, and Saharon Rosset. Breast cancer identification: Kdd cup winner's report. *ACM SIGKDD Explorations Newsletter*, 10(2):39–42, 2008.

[62] R Bharat Rao, Oksana Yakhnenko, and Balaji Krishnapuram. Kdd cup 2008 and the workshop on mining medical data. *ACM SIGKDD Explorations Newsletter*, 10(2):34–38, 2008.

[63] Mohammad Rawashdeh and Anca Ralescu. Center-wise intra-inter silhouettes. In *Scalable Uncertainty Management*, pages 406–419. Springer, 2012.

[64] Mohammad Rawashdeh and Anca Ralescu. Crisp and fuzzy cluster validity: Generalized intra-inter silhouette index. In *Fuzzy Information Processing Society (NAFIPS), 2012 Annual Meeting of the North American*, pages 1–6. IEEE, 2012.

[65] Mohammad Rawashdeh and Anca Ralescu. Fuzzy cluster validity with generalized silhouettes. In *Proceedings of the 23rd Annual Midwest Artificial Intelligence and Cognitive Science Conference, Cincinnati, Ohio, April*, pages 11–18. Citeseer, 2012.

[66] Mohammad Rawashdeh and Anca Ralescu. A pairwise distance view of cluster validity. In *Advances on Computational Intelligence*, pages 561–570. Springer, 2012.

[67] Lior Rokach. A survey of clustering algorithms. In *Data mining and knowledge discovery handbook*, pages 269–298. Springer, 2010.

[68] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[69] Enrique H Ruspini. A new approach to clustering. *Information and control*, 15(1):22–32, 1969.

[70] Philip Russom. Big data analytics. *TDWI Best Practices Report, Fourth Quarter*, 2011.

[71] Bernhard Scholkopf. The kernel trick for distances. *Advances in neural information processing systems*, pages 301–307, 2001.

[72] Shokri Z Selim and Mohamed A Ismail. K-means-type algorithms: a generalized convergence theorem and characterization of local optimality. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (1):81–87, 1984.

[73] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[74] Amit Singhal. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

[75] E Trauwaert. On the meaning of dunn's partition coefficient for fuzzy clusters. *Fuzzy Sets and Systems*, 25(2):217–242, 1988.

[76] Sofia Visa and Anca Ralescu. Learning imbalanced and overlapping classes using fuzzy sets. In *Proceedings of the ICML*, volume 3, 2003.

[77] Sofia Visa and Anca Ralescu. The effect of imbalanced data class distribution on fuzzy classifiers-experimental study. In *Fuzzy Systems, 2005. FUZZ'05. The 14th IEEE International Conference on*, pages 749–754. IEEE, 2005.

[78] Sofia Visa and Anca Ralescu. Issues in mining imbalanced data sets-a review paper. In *Proceedings of the sixteen midwest artificial intelligence and cognitive science conference*, pages 67–73. sn, 2005.

[79] Fei-Yue Wang. A big-data perspective on ai: Newton, merton, and analytics intelligence. *Intelligent Systems, IEEE*, 27(5):2–4, 2012.

[80] Sharon Weinberger. Spotting the hot zones: Now we can monitor epidemics hour by hour. wired magazine. Technical report, Updated 6/23/2008), Available at: http://www.wired.com/science/discoveries/magazine/16-07/pb_hotzones, 2008.

[81] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

[82] Xuanli Lisa Xie and Gerardo Beni. A validity measure for fuzzy clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(8):841–847, 1991.

[83] Eric P Xing, Michael I Jordan, Stuart Russell, and Andrew Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 505–512, 2002.

[84] Rui Xu, Donald Wunsch, et al. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.

[85] Miin-Shen Yang, Yu-Jen Hu, Karen Chia-Ren Lin, and Charles Chia-Lee Lin. Segmentation techniques for tissue differentiation in mri of ophthalmology using fuzzy clustering algorithms. *Magnetic Resonance Imaging*, 20(2):173–179, 2002.

[86] Yiming Yang and Jan O Pedersen. A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420, 1997.

[87] Jian Yu, Qiansheng Cheng, and Houkuan Huang. Analysis of the weighting exponent in the fcm. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(1):634–639, 2004.

[88] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[89] Lotfi A Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *Systems, Man and Cybernetics, IEEE Transactions on*, (1):28–44, 1973.

[90] Lotfi A Zadeh. The concept of a linguistic variable and its application to approximate reasoningi. *Information sciences*, 8(3):199–249, 1975.

[91] Lotfi A Zadeh. Is there a need for fuzzy logic? *Information Sciences*, 178(13):2751–2779, 2008.

[92] Osmar R Zaïane, Andrew Foss, Chi-Hoon Lee, and Weinan Wang. On data clustering analysis: Scalability, constraints, and validation. In *Advances in Knowledge Discovery and Data Mining*, pages 28–39. Springer, 2002.

[93] Daoqiang Zhang, Songcan Chen, and Zhi-Hua Zhou. Learning the kernel parameters in kernel minimum distance classifier. *Pattern Recognition*, 39(1):133–135, 2006.

[94] Ying Zhao and George Karypis. Criterion functions for document clustering: Experiments and analysis. *Machine Learning*, 2001.

[95] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2006.

[96] Ji Zhu, Saharon Rosset, Trevor Hastie, and Robert Tibshirani. 1-norm support vector machines. In *NIPS*, 2003.