Department of Computing

Clustering by Pairwise Similarity

Qilin Li

This thesis is presented for the Degree of
Master of Philosophy
of
Curtin University

March 2016

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature: _____

Date: _____

Clustering by Pairwise Similarity

by

QILIN LI

Submitted to the Department of Computing
on March 10, 2016, in partial fulfillment of the
requirements for the degree of
Master of Philosophy

Abstract

An era of big data has arrived without doubt because of advancements in sensing and storage technologies and dramatic growth in digital applications. Such growth in the amount of data, as well as the variety of data, poses huge demand for the development of automatic data analysis, classification, and retrieval techniques.

Clustering is one of the most active area in the scope of data analysis. Clustering, also known as cluster analysis, is the formal study of methods and algorithms for grouping, or clustering, objects according to measured or perceived intrinsic characteristics or similarities. Clustering does not use category labels that tag objects with prior identifiers. The absence of category information distinguishes data clustering (unsupervised learning) from classification or discriminant analysis (supervised learning). This thesis focuses one of the hot areas of clustering: the image clustering. Specifically, we aim at using clustering method to deal with semantic learning and manifold learning problems in image analysis.

The goal of image clustering is to gather all images into clusters such that the images in each cluster provide essentially the same, or highly related, information. The generated clusters provide a concise summarization and visualization of the image content that can be used for different tasks, such as image retrieval. Traditionally, researchers do not care too much about what kind of information is shared in image clusters, as long as the images appear similar in each cluster. In recent years, there has been a growing interest in developing effective methods for content-based image retrieval (CBIR). It reflects the face that a more desirable retrieval system should be able to deal with semantic query. Instead of using image as a query, a more effective way from user's perspective is querying by the nature language description of the image. The challenge here is to learn high-level semantic concepts from low-level visual features, which is known as the "semantic gap" problem.

Another problem in image clustering is to define the underlying manifold structures in a image space. Image data are normally represented by a high-dimensional feature space, resulting in poor performance of Euclidean space based clustering methods. Manifold learning has been proven to significantly improve performance in many image tasks. Since the manifolds are usually of a much lower dimension than the orig-

inal space in which they lie, many manifold learning algorithms make use of dimension reduction methods. Spectral clustering is one of the most popular methods among them. It does not require estimating an explicit model of data distribution, rather a spectral analysis of the pairwise similarities needs to be conducted. Spectral clustering makes use of spectral-graph structure of an affinity matrix to project data into a much lower eigenspace. It requires robust and appropriate affinity graphs as the input in order to form clusters with desired structures. Constructing such affinity graphs is a nontrivial task due to the ambiguity and uncertainty inherent in the raw data. Most existing spectral clustering methods typically adopt Gaussian kernel as the similarity measure, and employ all available features to construct affinity matrices with the Euclidean distance, which is often not an accurate representation of the underlying data structures, especially when data is embedded in high dimensional space.

In this thesis, we first present an axiomatic fuzzy set based clustering method, namely AFS clustering, for semantic gap problem. By utilizing fuzzy membership function, we show that the semantic gap between high-level semantic concepts and low-level visual features has been bridged automatically. Moreover, unlike the conventional fuzzy based approach using the pre-defined membership function, the proposed method automatically constructs membership function derived from data itself, yielding more objective and robust representation of semantic concepts. We apply the proposed method on facial component clustering, the results are promising compared to K-means and FCM.

Next, we propose a novel unsupervised approach for manifold learning, named Axiomatic Fuzzy Set-based Spectral Clustering (AFSSC), to generate more robust affinity graphs via identifying and exploiting discriminative features to improve the performance of spectral clustering. Specifically, our model assigns a discriminative fuzzy set for each object as the new representation and derives the pairwise similarities by the membership degrees belonging to the fuzzy set. Instead of utilizing all available features blindly, the proposed model is capable of avoiding noisy features by capturing and combining subtle similarity information distributed over feature subspaces. We conduct extensive experiments to show the effectiveness of the proposed model, such as semantic clustering for facial components, UCI data clustering, handwriting digits clustering, face clustering, etc. The results have shown the superiority of our method compared to other state-of-the-art methods.

Thesis Supervisor: Ling Li
Title: Associate Professor

Thesis Supervisor: Wanquan Liu
Title: Associate Professor

Acknowledgments

I would like to thank those people who helped me in different ways during the period of the project. Without their help the thesis could not have been possible.

I want to express my sincere appreciation to my supervisors Ling Li and Wanquan Liu for their constant guidance and encouragement. Thanks for offering me such an opportunity to be a research student. It is a truly indelible and valuable experience to work with you.

I am also grateful to my schoolmates, Yi Zhang, Antoni Liang, Yan Ren, Xin Zhang, Jinglan Tian, Mingliang Xue, and Nadith for many discussions and insightful suggestions.

Special thanks to my parents for understanding and supporting my project. Also my love, Xiangmeng Tang, thanks for your warm and constant companion.

Contents

List of Figures

List of Tables

Publications

This thesis is based upon several works that have been published over the course of the author's Master Degree, listed as follows in chronological order:

- **Li, Q.**, Ren, Y., Liu, W., and Li, L. (2015). Semantic facial description via axiomatic Fuzzy Set based clustering. *In Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference on* (pp. 755-762). IEEE.

- Ren, Y., **Li, Q.**, Liu, W., and Li, L. (2016). Semantic facial descriptor extraction via Axiomatic Fuzzy Set. *Neurocomputing*, 171, 1462-1474.

- **Li, Q.**, Ren, Y., Li, L., and Liu, W. (2016). From Low-level Geometric Features to High-level Semantics: an Axiomatic Fuzzy Set Clustering Approach. *Journal of Intelligent & Fuzzy Systems.* (accepted)

Chapter 1

Introduction

1.1 Background

Advancements in sensing and storage technology and dramatic growth in applications such as Internet search, digital imaging, and video surveillance, *etc.* have created high-volume, high-dimensional data. With the growth in the amount of data, the variety of available data (text, image, and video) has also increased. E-mails, blogs, transaction data, and billions of Web pages create terabytes of new data every day. Most of the data is stored digitally in electronic media, thus providing huge potential for the development of automatic data analysis, classification, and retrieval techniques. Many of these data streams, however, are unstructured, adding to the difficulty in analyzing them [1].

The growth in both the size and diversity of data requires progress in methodologies to automatically summarize, process, and understand the data. Data analysis can be broadly classified into two major paradigms [2]: (1) *exploratory* or *descriptive*, meaning that the investigator wants to understand the latent characteristics or structure of the high-dimensional data without pre-specified models or hypotheses, and (2) *confirmatory* or *inferential*, which means the investigator wants to confirm the validity of a hypothesis/model or a set of assumptions given the available data. Many statistical techniques have been proposed to analyze the data, such as analysis of variance, linear regression, principle component analysis, discriminant analysis,

multi-dimensional scaling, and cluster analysis to name a few. A useful overview is given in [3].

In pattern recognition, data analysis is concerned with predictive modeling: given some training data, we want to predict the behavior of the unseen test data. This task is also referred to as *learning*, *mining*, and *knowledge discovery*. In fact, the terms pattern recognition, machine learning, data mining and knowledge discovery in database (KDD) are difficult to separate, as they largely overlap in their scopes. Nevertheless, machine learning and pattern recognition are the common terms for supervised learning method, whereas data mining and KDD have a large focus on unsupervised models. Supervised learning assumes that a set of training data has been provided, consisting of a set of instances that have been properly labeled with the correct output. A learning procedure then generates a model that attempts to predict correct output for testing data. Unsupervised learning, on the other hand, attempts to find inherent patterns or latent structures in the data without any training data. A hybrid setting, called semi-supervised learning, has gained a growing interest recently, which uses a combination of labeled and unlabeled data (typically a small set of labeled data combined with a large amount of unlabeled data). In the semi-supervised classification, the labels of the small portion of the training data are often diffused to the unlabeled neighbors so that the labeled and unlabeled data are sufficiently smooth with respect to the intrinsic structure [4]. Fig.1-1 illustrates the spectrum of different types of learning problems of interest in the scopes.

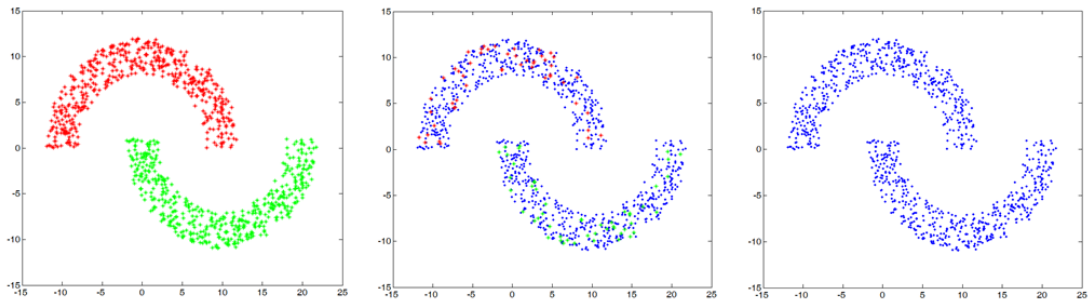

Fig. 1-1: Different learning models: red and green points are data with different labels. Blue points mean unknown labels.

Cluster analysis, known as clustering, is the formal study of methods and algorithms for grouping, or clustering, objects according to measured or perceived intrinsic characteristics or similarities. Webster [5] defines cluster analysis as "a statistical classification technique for discovering whether the individuals of a population fall into different groups by making quantitative comparisons of multiple characteristics". Clustering analysis does not use category labels that tag objects with prior identifiers. The absence of category information distinguishes data clustering (unsupervised learning) from classification or discriminant analysis (supervised learning). Cluster analysis itself is not a specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. As described by [1], an operational definition of clustering can be stated as: given a representation of N objects, find K groups based on a measure of similarity such that the similarities between objects in the same group are high while the similarities between objects in different groups are relatively low. Two significant questions here are: what is the definition of similarity? And what is the notion of a cluster? Intuitively, clusters can differ in terms of shape, size, and density. The presence of noise in the data makes the detection of the clusters even more difficult. Ideal notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem. In reality, on the other hand, a cluster is a subjective entity that is in the eye of the beholder and whose significance and interpretation requires domain knowledge. However, while humans are excellent cluster seekers in two and possibly three dimensions, we need automatic algorithms for high-dimensional data.

As a technique of knowledge exploration, clustering is prevalent in any discipline that involves analysis of multivariate data, such as biology, psychiatry, psychology, archaeology, psychiatry, geology, and marketing, *etc.* The importance and interdisciplinary nature of clustering is evident through its vast literature. It is difficult to exhaustively list the numerous scientific fields and applications that have utilized clustering techniques as well as the thousands of published algorithms. Image seg-

mentation, an important area in computer vision, can be formulated as a clustering problem [6, 7]. Documents can be clustered to generate topical hierarchies for efficient information access [8] or retrieval [9]. Clustering is also used to group customers into different types for efficient marketing [10], to group service delivery engagements for workforce management and planning [11], as well as to study genome data [12] in biology. The diversity of clustering on various fields and applications has resulted in thousands of clustering algorithms that have been published and continue to appear.

1.2 Related Work

The notion of a "cluster" cannot be precisely defined, which is one of the reasons why there are so many cluster algorithms. Different researchers employ different cluster models, and understanding their respective "cluster models" is the key to understand the differences between various algorithms. Typical cluster models include:

- **Connectivity models**: also known as hierarchical clustering, are based on the core idea of objects being more related to nearby objects than to objects farther away. For example, single-linkage clustering and complete linkage clustering.
- **Centroid models**: represent each cluster by a single mean vector (cluster center), and assign the objects to the nearest cluster center. For example, K-means clustering.
- **Distribution models**: most closely related to statistics. Clusters can then be easily defined as objects belonging most likely to the same distribution. For example, Gaussian mixture models (expectation-maximization algorithm).
- **Density models**: define clusters as connected dense regions in the data space. For example, density-based spatial clustering of applications with noise (DBSCAN).
- **Subspace models**: also known as co-clustering. Clusters are modeled with

both cluster members and relevant features. For example, sparse subspace clustering.

- **Graph models**: also known as spectral clustering, represent data points as nodes in a weighted graph. For example, normalized cut.

Although clustering methods can be broadly categorized by the above models, it is still extremely difficult to review all the published approaches. Some of the major approaches related to this work will be briefly reviewed as follows.

1.2.1 K-means clustering

K-means is the most popular and the simplest clustering algorithm. It was first published in 1956 [13]. Even though K-means was proposed over 50 years ago, it is still one of the most widely used algorithms for clustering. Ease of implementation, simplicity, efficiency, and empirical success are the main reasons for its popularity.

K-means clustering aims to partition n data points into k clusters in which each point belongs to the cluster with the minimum squared error, serving as a prototype of the cluster. Given a set of data points $(x_1, x_2, ..., x_n)$, where each point is a d-dimensional real vector, the goal is to cluster these points into a set of K $(K \leq n)$ clusters, $C = c_k, k = 1, 2, ..., K$. Let μ_k be the mean of cluster c_k. The squared error between μ_k and the points in cluster c_k is defined as:

$$J(c_k) = \sum_{x_i \in c_k} \parallel x_i - \mu_k \parallel^2 \tag{1.1}$$

The objective is to minimize the sum of the squared errors over all the K clusters,

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} \parallel x_i - \mu_k \parallel^2 \tag{1.2}$$

Minimizing this objective function is known to be an NP-hard problem (even for $K = 2$). Thus K-means, which is a greedy algorithm, can only converge to a local minimum, even though one study has shown with a large probability K-means could

converge to the global optimum when clusters are well separated [14]. K-means starts with an initial partition with K clusters and assigns patterns to clusters so as to reduce the sum of squared errors. Since the sum of squared errors always decreases with an increase in the number of clusters K ($J(C) = 0$ when $K = n$), it can be minimized only for a fixed number of clusters.

The drawbacks of K-means algorithm are evident: (1) convergency to local minima. One possible way to overcome the local minima is to run K-means with different initial partitions and choose the partition with the smallest squared error. (2) user-specified parameter K. Poor results can be yielded by an inappropriate choice of K. While no perfect mathematical criterion for the choice of K exists, a typical heuristics way is to run K-means independently for different values of K and the partition that appears to be the most meaningful to the domain expert is selected.

1.2.2 Fuzzy clustering

Fuzzy clustering, also known as soft clustering, assigns each data points to all clusters with a certain degree of membership [15]. Compared to hard (crisp) clustering, fuzzy clustering is more suitable to handle problems with vague boundaries of clusters. Moreover, the memberships may help us to discover more sophisticated relations between a given data point and the disclosed clusters [16].

The fuzzy c-means (FCM) [17] is the most popular fuzzy clustering algorithm. The FCM algorithm attempts to partition a finite collection of n data points $(x_1, x_2, ..., x_n)$ into a collection of c fuzzy clusters with respect to some give criterion. Given a finite set of data, the algorithm returns a list of c cluster centers $C = c_1, c_2, ..., c_n$ and a partition matrix $W = w_{i,j} \in [0, 1], i = 1, .., n, j = 1, ..., c$, where each element w_{ij} represents the degree to which element x_i belongs to cluster c_j. Just like the K-means clustering, the FCM aims to minimize an objective function:

$$\arg\min_{C} \sum_{i=1}^{n} \sum_{j=1}^{c} w_{ij}^{m} \parallel x_i - c_j \parallel^2 \tag{1.3}$$

where

$$w_{ij}^{m} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \tag{1.4}$$

This differs from the K-means objective function by the addition of the membership values w_{ij} and the fuzzifier $m \in \mathbb{R}$, with $m \geq 1$. The fuzzifier determines the level of fuzziness. A large m results in smaller memberships w_{ij} and hence, fuzzier clusters. At the limit $m = 1$, the memberships w_{ij} converge to 0 or 1, which implies a crisp partition.

Although superior when the boundaries among the clusters are vague and ambiguous, FCM suffers from similar drawbacks with K-means, *e.g.*, the presence of noise and outliers, the difficulty to identify the initial partitions, and the choice of k [18].

1.2.3 Graph theoretic clustering

Graph theoretic clustering, sometimes referred to as spectral clustering, represents the data points as nodes in a weighted graph, where the edges connecting the nodes are weighted by their pairwise similarity. Graph theoretic clustering are gaining increasing popularity over traditional clustering techniques, which are centered around the notion of "feature". In many real-world applications, in fact, a feasible feature-based description of objects might be difficult to obtain or inefficient for learning purposes while, on the other hand, it is often possible to obtain a measure of the similarity between objects [19]. Moreover, graph theoretic clustering methods are of significant interest since they cast cluster as pure graph theoretic problems for which solid theories and powerful algorithms have been developed. The basic idea here is to partition the nodes into two subset A and B such that the cut size, *i.e.*, the sum of the weights assigned to the edges connecting between nodes in A and B, is minimized.

Initial algorithms solved this problem using the minimum cut algorithm [20], which often results in clusters of imbalanced sizes. A cluster size (number of data points in a cluster) constraint was later adopted by the ratio cut algorithm [21]. An efficient approximate graph-cut based clustering algorithm with cluster size (volume of the clusters, or sum of edge weights within a cluster) constraint, called Normalized Cut,

was first proposed by Shi and Malik [7]. Its multi-class version was proposed by Yu and Shi [22]. Meila and Shi [23] presented a Markov Random Walk view of spectral clustering and proposed the Modified Normalized Cut algorithm that can handle an arbitrary number of clusters. Another well-known variant of spectral clustering algorithm was proposed by Ng *et al.* [24], where a new data representation is derived from the normalized eigenvectors of a kernel matrix. Laplacian Eigenmap [25] is another spectral clustering method that derives the data representation based on the eigenvectors of the graph Laplacian. Dominant sets algorithm [19] formulate the pairwise clustering problem by relating clusters to maximal cliques, which can be solved as a straightforward continuous optimization problem.

1.3 Significance and contribution

In this thesis, we propose a method based on AFS clustering for semantic learning problem in image clustering. Motivated by the effectiveness of fuzzy set for representing semantic concepts, Axiomatic Fuzzy Set (AFS) theory is adopted to bridge the semantic gap between low-level features and high-level semantic concepts. Membership functions are crucial for all fuzzy based methods. Unlike conventional fuzzy approach with pre-defined membership functions, such as triangular and Gaussian membership functions, AFS enables the membership functions to be created based on information within data, taking both fuzziness (subjective imprecision) and randomness (objective uncertainty) into account. Such superiority is significant when semantic concepts need to be defined.

Moreover, we embed the AFS framework for similarity measure by taking the fuzzy membership degree as pairwise similarity. Specifically, we first formulate a unified and generalized data distance inference framework based on AFS fuzzy theory with two innovations. Instead of using the complete feature space as a whole, the proposed model is designed to avoid indistinctive features using fuzzy membership function, yielding similarity graphs that can better express the underlying structure in data, which can significantly reduce the number features used in the clustering process.

On the other hand, the Euclidean assumption for data similarity inference is relaxed using fuzzy logic operations defined in AFS. The data distance is then smoothed by Gaussian kernel to enforce locality, resulting final similarity matrix. Motivated by the natural of mapping similarity matrix to graph (with nodes represent data points, weighted edges represent similarities), an affinity graph is constructed. Such a model is capable of capturing and combining subtle similarity information distributed over discriminative feature subspaces. We conduct standard spectral clustering method with the constructed affinity graph, and the results have shown the superiority of the proposed approach compared to other state-of-the-art methods.

1.4 Structure of the thesis

In this thesis, we propose two methods, AFS based clustering and AFS based spectral clustering, for different problems in image clustering, semantic learning and manifold learning, respectively.

In Chapter 2, we briefly introduce the AFS framework used in our model and propose a novel feature selection algorithm based AFS clustering method. Semantic facial components clustering is used as an example to show that the proposed method can automatically and objectively bridge the semantic gap between high-level semantic concepts and low-level visual features.

In Chapter 3, the AFS framework is extended for spectral clustering, and a novel algorithm named AFSSC is proposed. To this end, we formulate fuzzy membership functions as the similarity measure and the affinity graph is constructed based on the fuzzy similarity. The spectral clustering are then used to obtain the final clustering result.

In Chapter 4, the result of semantic facial component clustering results based on AFS are first presented. Semantic concepts are automatically extracted and the clustering results are improved compared to K-means and FCM. The results for AFSSC are shown next. Extensive experiments are conducted on various kinds of data, such as UCI data, USPS handwritten digits, and face data. The results have

shown the superiority of the proposed AFSSC compared to other state-of-the-art methods. The relationship of AFS and AFSSC is discussed at the end of the chapter.

Chapter 5 concludes the whole thesis. The entire project is reviewed briefly and potential future directions are discussed.

Chapter 2

AFS Clustering for Semantic Learning

2.1 Introduction

Image clustering aims at gathering all images into several clusters such that the images in single cluster provide essentially the same, or highly related information. The generated clusters provide a concise summarization and visualization of the image content that can be used for various computer vision tasks, such as image retrieval. Normally. researchers do not care too much about what kind of information is presented in image clusters, as long as the images are all similar in one cluster. In recent years, there has been a growing interest in developing effective methods for content-based image retrieval (CBIR), in which the retrieval problem can be formulated as a clustering problem [26]. The increasing demand for user interaction motivates us to build more user-friendly retrieval systems. CBIR aims to find images from an image database that closely matches a query given by users. Given a query image by users, the CBIR systems would usually extract visual features, such as color, texture *etc.* and re-rank the searched database by the similarities with the query based on the extracted features. In fact, a more desirable CBIR system from users perspective can be achieved by improving its capability to process semantic query, since in many cases the accurate query image may be not available or reliable. The semantic-based image

retrieval system allows user to input a query in terms of natural language expression [27]. For example, to query with descriptions such as "a face with large eyes and small nose" instead of using a facial image. In such cases, the user just needs to extract semantic concepts from the facial image in their mind, which is more natural and easier for human being.

Eakins *et al.* [28] mentioned three levels of queries in CBIR:

- *Level* 1: Query by primitive features such as color, texture, shape or the spatial location of image elements. Typical query is query by example, "find pictures like this".

- *Level* 2: Query of objects of given type identified by derived features, with some degree of logical inference. For example, "find a picture of a face".

- *Level* 3: Query by abstract attributes, involving a significant amount of high-level reasoning about the purpose of the objects or scenes depicted. This includes retrieval of named events, of pictures with emotional or religious significance, *etc.* For example, "find pictures of a face with large eyes and small nose".

Levels 2 and 3 together are referred to as semantic image retrieval, and the gap between Levels 1 and 2 is the well-known "semantic gap" [29]. More specifically, the discrepancy between the limited descriptive power of low-level image features and the richness of human semantics is referred to as the "semantic gap" [30]. Most of the semantic CBIR systems still focus on Level 2, and limited work has been done for Level 3. The challenge here is to find a robust and reliable method that automatically determining the semantic meaning of the images based on low level visual feature by computer. There is generally no direct connection between the low-level features of an image that the computers can detect and the high-level semantic concepts that the human would associate with the image. Therefore, to support query by high-level concepts, a CBIR system should provide full support in bridging the "semantic gap" between the numerical image features and the richness of human semantics.

In the survey of Liu *et al.* [31], the state-of-art techniques in reducing the "semantic gap" are mainly categorized into five paradigms: 1) using object ontology to define high-level concepts; 2) using machine learning tools to associate low-level features with query concepts; 3) introducing relevance feedback (RF) into retrieval loop for continuous learning of users' intention; 4) generating semantic template (ST) to support high-level image retrieval; 5) making use of both the visual content of images and the textual information obtained from the Web for WWW image retrieval.

In this work, we focus on the second paradigm in this chapter. Compared to other models such as relevance feedback, machine learning methods need less interaction with user. The models try to "learn" the semantic concepts by itself rather than depending on the feedback from user. This is critical due to the subjectivity of human perception on semantic concepts.

Machine learning methods could be categorized into supervised learning and unsupervised learning. Both of them are used to derive high-level semantic features [32, 33, 34, 35]. Supervised learning methods predict the value of an outcome measure (for example, semantic category label) based on the training set. For example, in Ref. [36], SVM is employed for image annotation. In the training stage, a binary SVM model is trained for each of the 23 selected concepts. In the testing stage, unlabeled regions are fed into all the models, and the concept from the model giving the highest positive result is associated with the region. Another widely used learning method is Bayesian classification. In [34], with binary Bayesian classifier, high-level image concepts of natural scenes are captured from low-level image feature. Bayesian network is used for indoor/outdoor image classification in [35].

Unlike supervised learning in which the presence of the outcome variable guides the learning process, unsupervised learning has no measurements of outcome. Rather, the task is to find out how the input features are organized or clustered. Image clustering is the typical unsupervised learning technique for retrieval purpose. It intends to group a set of image data in a way to maximize the similarity within clusters and minimize the similarity between different clusters. Each resulting cluster is then associated with a class label.

The traditional k-means clustering and its variations are often used for image clustering. In [37], k-means clustering is applied to low-level color features of a set of training images. The statistics measuring the variation of each cluster are then used to derive a set of mappings between the low-level features and the optimal textual characterizations (keywords) of the corresponding cluster. The mapping rules derived could be used further to index unlabeled images. Another method named "CLUE" is presented in [30] to reduce the "semantic gap" problem. Instead of retrieving the top matched images, this method attempts to retrieve semantically coherent image clusters. Given a query image, a collection of target images similar to the query image are selected as its neighbors. Based on the hypothesis that images of the same semantics tend to be clustered together, spectral clustering is then used to cluster these target images into different semantic classes. Though successful in manifold data clustering, this method cannot produce an explicit mapping function. To deal with new data points, similarities between the new points and all training data have to be measured again, which leads to a large computation cost. To tackle these problem, a locality preserving clustering (LPC) method [38] is proposed. LPC can provide an explicit mapping function so that new data can be measured directly.

Face image retrieval (FIR) is a special type of CBIR, where the goal is to find similar human faces to a query either in the form of an image or semantic descriptions. Currently only a very limited amount of work has been done on semantic-based FIR systems due to its difficulty. Among existing FIR systems, one existing approach is to use a probabilistic method based on local low-level features, such as color, texture, and high-level semantic labels to re-rank the database [39]. Another approach begins with 24 manually marked key points and associated keywords that characterize each face. Singular value decomposition is applied to create a Latent Semantic Space allowing face images to be retrieved by a semantic query [40]. However such method is not appropriate for large databases as all face images in the database need to be annotated manually. Another approach is to utilize fuzzy-based method for the semantic retrieval of face images. More specifically, it uses fuzzy sets to bridge the semantic-gap between low level visual features and high level descriptions and the

fuzzy c-means method (FCM) is used to calculate the similarities between different subjects [26]. It is a nice idea to use fuzzy sets to narrow the semantic-gap since it is easier to use fuzzy descriptions to represent high level semantic descriptions, such as "big eye", "long nose", *etc.* In all these work, the high level semantic concept descriptions are required and they are given partially or fully as the ground truth information. Such requirement hinders the applicability of these systems since it is difficult to obtain the ground truth information for semantic descriptions objectively.

In this chapter, a fuzzy theory, called Axiomatic Fuzzy Set (AFS), is employed as the machine learning model to deal with the semantic face clustering. We start with an automatic landmark detection. All facial components are detected using the landmarks and low-level features are extracted for the components. Each component is then clustered separately by the AFS clustering method and the semantic concepts are labeled to each cluster. In our model, one of the main innovations is that the pairwise similarity is not measured in the commonly used Euclidean space. Rather it is measured in the AFS fuzzy space with fuzzy membership functions. Another innovations is that the "semantic gap" is automatically bridged by AFS clustering and the semantic concepts are represented by fuzzy rules. Hence the subjectivity commonly associated with human perception is totally avoided.

2.2 AFS theory

The AFS theory was originally proposed in [41] and then extensively developed in [42, 43, 44],*etc.* AFS fuzzy sets determined by membership functions and their logic operations are algorithmically determined according to the distributions of the original data and the semantics of the fuzzy sets. The AFS framework enables the membership functions and fuzzy logic operations to be created based on information within the database, by taking both the fuzziness (subjective imprecision) and randomness (objective uncertainty) into account. Meanwhile, the membership functions and the fuzzy logic operations determined by the observed data drawn from a probability space will be consistent with those being determined by the proba-

bility distribution expressed in the probability space. The main idea behind AFS is to transform the observed data into fuzzy membership functions and implement their logic operations. Information can then be extracted from the AFS space rather than the original feature space. The research monograph [45] offers a comprehensive introduction of AFS theory and its applications.

In this chapter, we attempt to build similarity measure in the AFS space rather than the normally used Euclidean space for a better extraction of data structure. The Iris dataset from UCI data repository [46] is used as an illustrative example for part of the AFS theory used in this chapter. The Iris dataset contains 150 samples with 4 features, namely sepal length (f_1), sepal width (f_2), petal length (f_3) and petal width (f_4). Given a sample $x = (x_1, x_2, x_3, x_4)$, where x_i is the i-th feature of x. Assume we have the following fuzzy IF-THEN rules:

- **Rule** $\mathcal{R}_1$: If x_1 is *short sepal*, x_2 is *wide sepal* and x_4 is *narrow petal*, then x belongs to class1;
- **Rule** $\mathcal{R}_2$: If x_2 is *wide sepal*, x_3 is *short petal*, then x belongs to class1;
- **Rule** $\mathcal{R}_3$: If x_1 is *short sepal*, x_4 is *narrow petal*, then x belongs to class1.

Using $M = \{m_{j,k} | 1 \leq j \leq 4, 1 \leq k \leq 3\}$ to denote the set of fuzzy concepts, where $m_{j,1}, m_{j,2}, m_{j,3}$ are fuzzy terms "large", "medium" and "small", associated with feature f_j respectively, the above fuzzy rules can be re-written as:

- **Rule** $\mathcal{R}_1$: If x is "$m_{1,2}$ and $m_{2,1}$ and $m_{4,2}$", then x belongs to class1;
- **Rule** $\mathcal{R}_2$: If x is "$m_{2,1}$ and $m_{3,2}$", then x belongs to class1;
- **Rule** $\mathcal{R}_3$: If x is "$m_{1,2}$ and $m_{4,2}$", then x belongs to class1.

For each set of fuzzy terms $A \subseteq M$, $\prod_{m \in A} m$ represents a conjunction of the fuzzy terms in A. For instance, given $A_1 = \{m_{1,2}, m_{2,1}, m_{4,2}\} \subseteq M$, a fuzzy concept "$m_{1,2}$ and $m_{2,1}$ and $m_{4,2}$" with the linguistic interpretation "*short sepal* and *wide sepal* and *narrow petal*" can be represented as $\prod_{m \in A_1} m = m_{1,2} m_{2,1} m_{4,2}$. Let

$A_2 = \{m_{2,,1}, m_{3,2}\}$, $A_3 = \{m_{1,2}, m_{4,2}\} \subseteq M$, a new fuzzy set as the disjunction of $\prod_{m\in A_1} m, \prod_{m\in A_2} m, \prod_{m\in A_3} m$, i.e., "$m_{1,2}m_{2,1}m_{4,2}$ or $m_{2,1}m_{3,2}$ or $m_{1,2}m_{4,2}$", can be written as:

$$\sum_{u=1}^{3}(\prod\nolimits_{m\in A_u} m) = \prod\nolimits_{m\in A_1} m + \prod\nolimits_{m\in A_2} m + \prod\nolimits_{m\in A_3} m \tag{2.1}$$

Hence the three fuzzy rules above can be denoted as:

- **Rule** $\mathcal{R}$: If x is $\sum_{u=1}^{3}(\prod_{m\in A_u} m)$, then x belongs to class1.

The expressions in rule R can be formulated as an algebra system as follows. Let M be a set of fuzzy linguistic terms. The set EM^* is defined as:

$$EM^* = \Big\{ \sum_{i\in I}(\prod_{m\in A_i} m) | A_i \subseteq M, i \in I, I \text{ is any nonempty indexing set} \Big\} \tag{2.2}$$

Consequently, EM can be defined by EM^* associated with an equivalent relation [45]. In fact, it is proven that each fuzzy set can be uniquely decomposed as

$$\xi = \sum_{i\in I}(\prod_{m\in A_i} m), \tag{2.3}$$

where each A_i is a subset of M.

In order to establish the membership function, the following ordered relation needs to be defined. Let X be a set and M be a set of fuzzy terms on X. For $A \subseteq M$, $x \in X$, it can be written that:

$$A^{\succeq}(x) = \{y \in X \mid x \succeq_m y \ for\ any\ m \in A\} \subseteq X \tag{2.4}$$

from a linearly ordered relation "$\succeq$". For $m \in M$, "$x \succeq_m y$" implies that the degree of x belonging to m is larger than or equal to that of y. $A^{\succeq}(x)$ is the set of all elements in X whose degrees of belonging to set $\prod_{m\in A} m$ are less than or equal to that of x. $A^{\succeq}(x)$ is determined by the semantic of the fuzzy terms in A and the probability

distribution of the observed dataset. Let ν be a fuzzy term on X. A weight function can be defined as:

Definition 1 *([45])* $\rho_\nu : X \to R^+ = [0, \infty)$. *$\rho_\nu$ is called a weight function of the fuzzy term ν if ρ_ν satisfies the following conditions:*

1. $\rho_\nu(x) = 0 \Leftrightarrow x \not\succeq_m x, x \in X$;
2. $\rho_\nu(x) \geq \rho_\nu(y) \Leftrightarrow x \succeq_m y,\ x, y \in X$.

Next, the coherence membership functions can be computed as,

$$\mu_\xi(x) = \sup_{i \in I} \inf_{\gamma \in A_i} \frac{\sum_{u \in A_i^{\succeq}(x)} \rho_\gamma(u) N_u}{\sum_{u \in X} \rho_\gamma(u) N_u}, \forall x \in X, \tag{2.5}$$

where N_u is the number of samples of u and ρ is defined in Section 3.3.1.

2.3 AFS clustering

There are three fundamental components in semantic retrieval system: 1) low-level image feature extraction; 2) similarity measure; 3) "semantic gap" reduction. In this work, we use detected landmarks to extract low-level features and the feature representation of data is then mapped into the AFS fuzzy space. Instead of commonly used Euclidean space, similarity measure is built in the fuzzy space by fuzzy membership function, in which the underlying data structure is revealed properly. After that, AFS clustering is utilized to bridge the "semantic gap" and the semantic label is assigned to each class. Fig. 2-1 shows the whole framework of the approach.

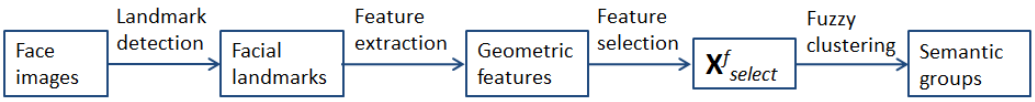

Fig. 2-1: The flow chart of the framework of our approach.

2.3.1 Low-level feature extraction

Low-level image feature is the basis of a CBIR system. They can be extracted either from the entire image or from part of the image. Representation of images at region

level is proven to be closer to human perception [47]. To perform such region-based CBIR, the first step is normally images segmentation. After that, low-level features such as color, texture, shape or spatial location can be extracted from the segmented regions. In our case, we focus on human face image, which has already been segmented naturally by facial components. Therefore the low-level features are extracted for each facial components individually.

The low-level feature extraction is based on automatically detected landmarks. The landmark detector used is proposed by Liang *et al.* [48] which is extended from Zhu and Ramanan's approach [49]. Instead of using only "lines" to represent facial components, Liang suggested to use "regions" to cover the facial components. Therefore the total landmarks for each face are increased to 130 from 68. The goal is to extract semantic concepts of facial components so that more landmarks can provide more shape information.

Based on the facial landmarks detected, we extract 5 global features, "area", "perimeter", "centroid distance", "width" and "height" as shown in Fig. 2-2. They are all defined by Euclidean distances. These five global features represent the overall size information. With these features, our aim is to categorize facial components into 3 semantic groups, "large", "medium" and "small". The experimental results with these features show that the proposed method could yield much clearer semantic groups than the traditional clustering methods, such as k-means and FCM. On the other hand, it is revealed that the shape of the facial components in the same cluster is not consistent enough. *E.g.* "a long but not wide eye" and "a wide but not long eye" could be grouped into the same cluster.

By noting that all the features are global in terms of size, we argue that global features are not enough for the consistency of shape in each cluster. We need more local features that can depict the shape of facial components. Therefor we define the "star" features for all facial components, which measures the Euclidean distances between the component's centers and every or some of the boundary points as shown in Fig.2-3. Note that the "star" feature is similar to one of the global features,

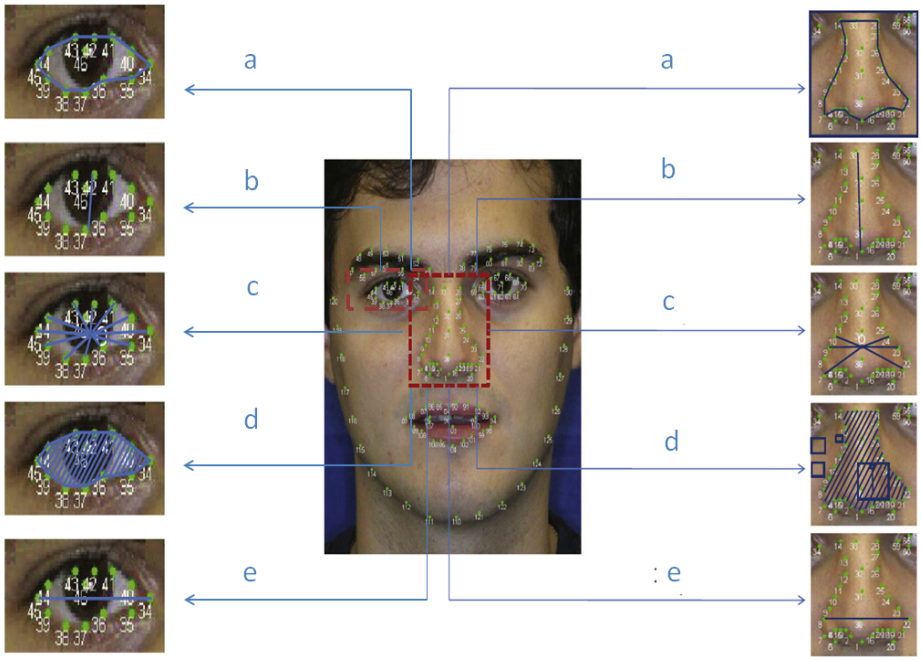

Fig. 2-2: Global features: (a) perimeter; (b) height; (c) centroid distance; (d) area; (e) width.

centroid distance. The difference is that the later is the sum of all features in the former. It can also be noticed that from Fig.2-3 that the “star" feature for nose is totally different with the centroid distance of nose. For nose, instead of using every boundary points, only some key points related to the shape of nose are chosen. With these local features, the shape in each cluster is guaranteed to be consistent, with all the global features to guide the clustering process with expectation to cluster each facial component into three classes eventually with different size. The improvement from these local features will be presented in Chapter 4.

2.3.2 Fuzzy similarity measure

Similarity measure is another critical point in a CBIR system. Most researchers employ the Minkowski-type metric to define region distance. Suppose there are two regions represented by $(x_1, x_2, ..., x_p)$, $(y_1, y_2, ..., y_p)$. The Minkowski metric is defined as:

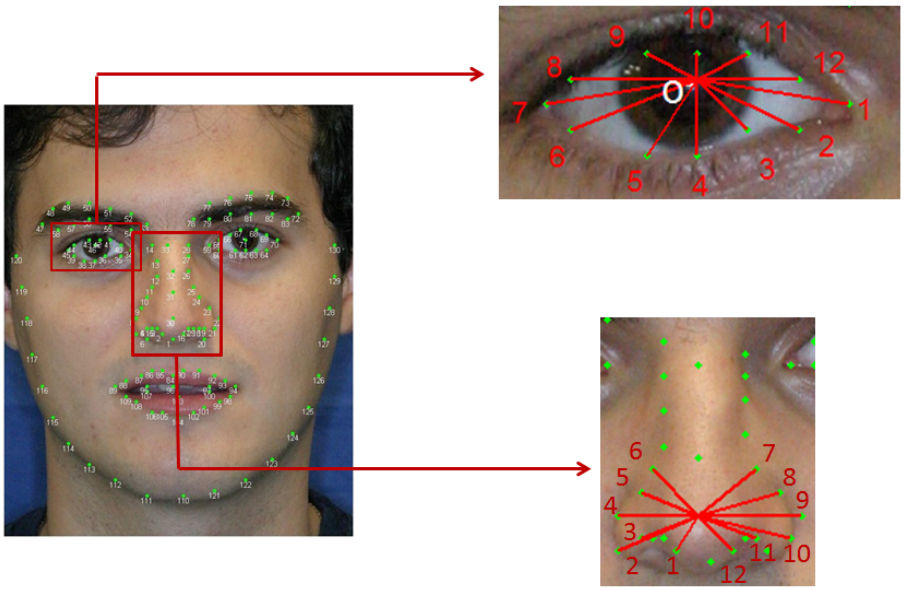

Fig. 2-3: Local features. There are 12 features for both eye and nose. They are the Euclidian distances from the center(represents by O) to the boundary points, e.g., $f_1 = d_{O1}$, $f_2 = d_{O2}$, etc.

$$d(x, y) = \left(\sum_{i=1}^{p} |x_i - y_i|^r \right)^{1/r} \tag{2.6}$$

Particularly, when $r = 2$, it is the well-known Euclidean distance (ℓ_2 norm) while it is the Manhattan distance (ℓ_1 norm) when $r = 1$.

An often-used variant version is the weighted Minkowski distance function which introduces weighting to identify important features:

$$d(x, y) = \left(\sum_{i=1}^{p} w_i |x_i - y_i|^r \right)^{1/r} \tag{2.7}$$

where w_i is the weight applied to different features.

In this work, we use fuzzy membership to measure pairwise similarity instead of the Minkowski distance. Besides, rather than manually assigning weights to important features, we come up with a novel feature ranking method to identify informative features.

After combining global and local features, it is obvious that there are some redundant features, especially for the 12 "star" features. It can be easily seen that for an

eye, f_1, f_4, f_7 and f_{10} (diagonal) provide a sufficient representation of the its "star" feature. However it is difficult to identify such a similarly useful subset for nose or other facial components. An unsupervised feature selection algorithm is hence proposed based on the fuzzy similarity in the AFS theory. For feature α and β, the similarity between them is defined as follows:

$$SI(\alpha, \beta) = \frac{\sum_{x \in X} \mu_{\alpha \wedge \beta}(x)}{\sum_{x \in X} \mu_{\alpha \vee \beta}(x)} \tag{2.8}$$

where X is the set of samples, $\mu_{\alpha \wedge \beta}(x)$ and $\mu_{\alpha \vee \beta}(x)$ are the membership functions of sample x belonging to fuzzy set $\alpha \wedge \beta$ and $\alpha \vee \beta$ respectively. The membership function is defined as Eq.2.5. This idea comes from the Jaccard similarity coefficient, which measures the similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets. Here instead of using the size of the sample sets, we are using the fuzzy membership function.

The proposed feature selection method mainly consists of two steps: (1) Vote each feature by all others based on the pairwise similarity, and find the one which is the most similar to others; (2) Depending on the selected features, iteratively add one new feature which is the most dissimilar to the existing selected features in. Intuitively, the first feature we picked contains the most information, just like principal component. Next, we search the most dissimilar one with the existing features, which is conceptually similar to the orthogonal principal component.

Assume we have n features. For each feature f_i $(1 \leq i \leq n)$, we can compute the similarity SI_{ij} with all features and rank them as f_{ij} $(1 \leq j \leq n)$ in a descending order. In fact f_{ij} is a re-ordering of $\{f_1, f_2, \cdots, f_n\}$. Let S_{ij} be the score of f_j based on the index j in the list f_{ij}, i.e.,

$$S_{ij} = j - 1 (1 \leq j \leq n) \tag{2.9}$$

where $0 \leq S_{ij} \leq n - 1$ $(S_{i1} = 0, S_{i2} = 1, ..., S_{in} = n - 1)$. It is easy to see that the more dissimilar of two features, the larger the score S_{ij}. Then we sum the scores of

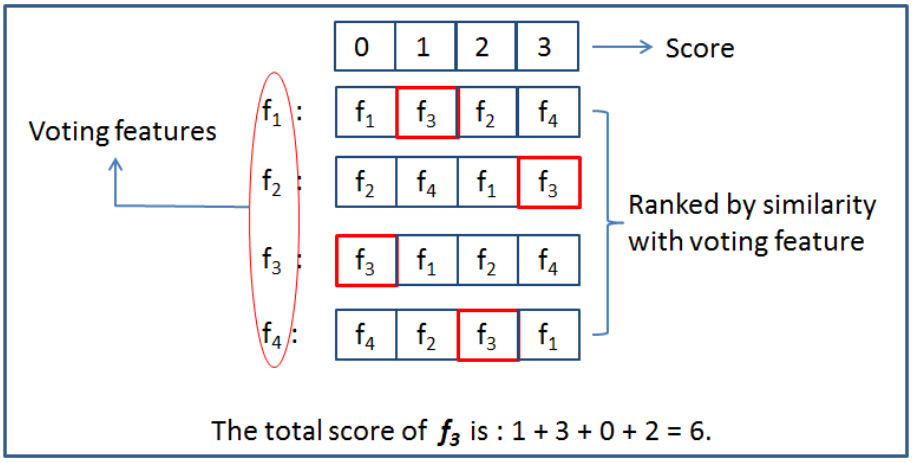

Fig. 2-4: An example of feature score voting.

each feature given by others and each feature f_j is given a total score S_j,

$$S_j = \sum_{i=1}^{n} S_{ij} \tag{2.10}$$

Note that the score is the "dissimilarity score". In this case the "base" feature is defined as the most similar one with all others, which has the lowest score. Now, let F represent the selected feature set,

$$F_1 = \{f_j | j = \arg\min_{1 \le j \le n} S_j\} \tag{2.11}$$

After we find the "base" feature, the rest is done by a greedy process. Technically, we can have all features re-scored based the selected feature set F as stated below. Let N_F be the size of F, the new score of f_j is,

$$S_j = \sum_{i=1}^{N_F} S_{ij} \tag{2.12}$$

Then our aim is to select another feature which is the most dissimilar to what we have already selected, so the new feature f_j for our selection is,

$$F_j = \{f_j | j = \arg\min_{1 \le j \le n} S_j\} \tag{2.13}$$

By repeating the above algorithm all features will be selected one by one iterative-

ly. Eventually a termination criteria is required here. One intuitive choice is that the score of newly added feature should be larger than the "average score" of the selected feature set F, i.e.,

$$S_j \geq \overline{S_F}; \tag{2.14}$$

where $\overline{S_F} = nN_Fk(0 \leq k \leq 1)$. It is easy to see that the number of selected feature is controlled by the parameter k and in fact k is a ratio of the average score to the total score. The effect of parameter k will be further analyzed in Chapter 4. Another choice for termination is by experimental validation as discussed later in Chapter 4.

To utilize AFS clustering, the first step is to transfer the data in the feature space to the AFS fuzzy space. F, the facial component set, is defined as $F = \{lefteye, righteye, nose, mouth...\}$, $f \in F$. Suppose we choose the top k features from the feature ranking in the previous step. Each feature is then divided into 3 fuzzy groups with semantic interpretation, "large", "medium", "small", represented by fuzzy terms $M^f = \{m_{i,j}^f | 1 \leq i \leq k, 1 \leq j \leq 3\}$, where $m_{i,j}^f$ is the jth fuzzy term associating with the ith feature of facial component f.

Let I_k^f represent the facial component $f \in F$ on the kth face image I_k. For example, $I_k^{f_1}$ is the right eye on the kth face image I_k. Let $\mu_m(I_k^f)$ be the membership degree of I_k^f belonging to fuzzy term m. Salient fuzzy attributes/terms are used to represent each facial component significantly, which is called fuzzy characteristic description. For such purpose, a set $B_{I_k^f}$ of fuzzy terms is defined for I_k^f which can be given as follows:

$$B_{I_k^f} = \{m \in M^f | \mu_m(I_k^f) = \mu_{\vee_{b \in M^f} b}(I_k^f)\} \tag{2.15}$$

where $\mu_{\vee_{b \in M^f} b}(I_k^f) = \max_{m \in M^f} \{\mu_m(I_k^f)\}$, i.e., the membership degrees of I_k^f belonging to fuzzy terms in $B_{I_k^f}$ should be the largest value among the membership degrees of I_k^f belonging to fuzzy terms in M^f. This set $B_{I_k^f}$ in fact contains all the best fuzzy terms to characterize this facial component f hence the fuzzy facial component characterization can be given as

Algorithm 1 Unsupervised Feature Selection based on Similarity

Input:

Feature similarity matrix $M_{n\times n}$.

Output:

Selected feature set F

Let S_{ij} represent the score of feature f_j given by feature f_i, S_j be the total score of f_j;
Sorting M on row by descending order, Scoring each feature based on ranking;
for $j := 1$ to n **do**
    $S_j = \sum_{i=1}^{n} S_{ij}$;
end for
$i := 1$;
$F_i = \arg\min_{1\leq j\leq n} S_j$;
$i \leftarrow i + 1$;
while $i \leq n$ **do**
    for $j := 1$ to n **do**
        if $j \in F_i$ **then**
            $S_j = 0$;
        else
            $S_j = \sum_{k=1}^{i} S_{kj}$;
        end if
    end for
    if $S_j \leq \overline{S_F}$ **then**
        break;
    end if
    $F_i = \arg\max_{1\leq j\leq n}\{S_j\}$;
    $i \leftarrow i + 1$;
end while

Stop.

$$\zeta_{I_k^f} = \wedge_{\beta\in B_{I_k^f}} \beta, \tag{2.16}$$

where "$\vee$" and "$\wedge$" are the fuzzy logic operations in AFS algebra defined in [45], and the semantic logic expressions of "$\vee$" and "$\wedge$" are "or" and "and". The computation details for $\mu_m(I_k^f)$, $B_{I_k^f}$ and $\zeta_{I_k^f}$ can be found in [50]. The fuzzy terms in $B_{I_k^f}$ can be regarded as the most salient characteristics of I_k^f (the facial component f of the face image I_k). Thus, the most salient characteristics are combined to describe I_k^f as Eq.(2.16).

Every sample has a fuzzy set as its description. The pairwise similarity then can be computed by the fuzzy membership degree belonging to those fuzzy sets. More specifically, the similarity of facial component I_i^f and I_j^f is defined as:

$$r_{ij} = \min\{\mu_{\zeta_{I_i} \wedge \zeta_{I_j}}(I_i), \mu_{\zeta_{I_i} \wedge \zeta_{I_j}}(I_j)\} \tag{2.17}$$

where r_{ij} $(0 \leq r_{ij} \leq 1)$ represents the similarity between I_i^f and I_j^f. The larger r_{ij} is, the more similar I_i^f and I_j^f are. Note that the self-similarity is also defined by Eq.(2.17), which means it could be less than 1.

2.3.3 Semantic gap reduction

"Semantic gap" is the most difficult problem in semantic retrieval system. In this work we use the clustering method to tackle this issue. More specifically, after the pairwise similarities are calculated, we obtain an relation matrix $R = (r_{ij})_{n \times n}$ for each facial components. The results in [43] guaranteed that there exists an integer t such that $(R^t)^2 = R^t$, i.e., fuzzy relationship matrix $Q = R^t = (q_{ij})_{n \times n}$ can yield a partition tree with equivalent classes under a given threshold α. If $q_{ij} \geq \alpha, \alpha \in [0, 1]$, face image I_i and I_j are in the same cluster under facial component f and threshold α. Note that the number of clusters in the above algorithm is determined by the threshold α. The optimal threshold α selection for the best clustering is proposed in [50] based on a validation index $\mathcal{I}_\alpha$ (Eq. (2.18)) defined as follows:

$$\mathcal{I}_\alpha = \frac{\sum\limits_{k=1,2,\ldots,n} \mu_{\zeta_{bou}}(I_k^f)}{\sum\limits_{k=1,2,\ldots,n} \mu_{\zeta_{Total}}(I_k^f)} \tag{2.18}$$

where $\zeta_{bou} = \vee_{1 \leq i,j \leq l, i \neq j}(\zeta_{\overline{C}_i} \wedge \zeta_{\overline{C}_j})$, $\zeta_{Total} = \vee_{1 \leq i \leq l} \zeta_{\overline{C}_i}$, $l \geq 2$. Fuzzy set ζ_{bou} describes the boundaries among different clusters which shows the clarity of the clusters. The smaller the degree of an object belongs to ζ_{bou}, the more clearly it is clustered, just like *compactness*. Thus the less $\mathcal{I}_\alpha$, the clearer the clustering. ζ_{Total} represents the overall characterization for all clusters, which can be treated as *separateness*. In practice, the threshold α is between the minimum and maximum values q_{ij} defined in the matrix

Q. Let $U = \{\alpha_1, \alpha_2, \ldots, \alpha_u\} = \{q_{ij} | 1 \leq i \leq n, 1 \leq j \leq n\}$ be the set of all the entries in Q and $\alpha_1 < \alpha_2 < \ldots < \alpha_u$. The best α is selected based on $\mathcal{I}_\alpha$:

$$\alpha = \arg \min_{\alpha_v \in [\alpha_1, \alpha_2, \ldots, \alpha_u]} \{\mathcal{I}_\alpha\} \tag{2.19}$$

The clusters $\overline{C}_1, \overline{C}_2, \ldots, \overline{C}_l$, which have more than one face images, can be obtained (i.e., the cluster with one single face image is discarded). In this case, we can obtain the initial clusters $\overline{C}_1, \overline{C}_2, \ldots, \overline{C}_l$ for $Q = (q_{ij})$.

Remember $\zeta_{I_k^f}$ is a characterization of I_k^f, and also we have obtained an initial clusters $\overline{C}_i$. The best $\zeta_{I_k^f}$ is then selected for constructing the semantic description of each cluster based on the initial clusters as follows.

$$\Gamma_i = \{\zeta_{I_k^f} | \frac{|\{y | y \in \overline{C}_i, \mu_{\zeta_{I_k^f}}(y) \geq \lambda\}|}{|\overline{C}_i|} \geq \omega, I_k^f \in \overline{C}_i\}, i = 1, \ldots, l \tag{2.20}$$

The elements in Γ_i are some fuzzy characteristic descriptions $\zeta_{I_k^f}$ in the ith cluster ($\overline{C}_i$). The motivation of this selection is that only some representative descriptions $\zeta_{I_k^f}$ can be used to represent the semantic descriptions of $\overline{C}_i$, and others are either not typical enough to represent its cluster or are noises. Therefore, the fuzzy descriptions of some representative face images which can represent their facial component cluster are collected in Γ_i as Eq.(2.20). The most salient characteristics in Γ_i are combined to describe the initial facial component cluster $\overline{C}_i$. Consequently, the semantic description of each facial component cluster can be defined as follows:

$$\zeta_{\overline{C}_i} = \wedge_{\gamma \in \Gamma_i} \gamma, (if\ \Gamma_i \neq \emptyset) \tag{2.21}$$

As explained previously, this set represents the salient description for each initial cluster. Obviously the universality and particularity of the semantic description of $\overline{C}_i$ can be controlled by ω and λ. The effects of the variation of parameter ω and λ are analyzed in [50], and plenty of experiments illustrate that the algorithm is not sensitive to the setting of the parameters if the parameters are selected in reasonable intervals.

The semantic description $\zeta_{\overline{C}_i}$ of each facial component cluster can be regarded as a classifier, in order to classify all the instances $I_k^f, k = 1, 2, \ldots, n$ again. This process is called the re-clustering process. Its aim is to revise the initial clusters and cluster the lost instances whose similarities r_{ij} with other instances are lower than α. Finally, I_k^f is re-clustered by measuring the membership degrees of I_k^f belonging to $\zeta_{\overline{C}_1}, \zeta_{\overline{C}_2}, \ldots, \zeta_{\overline{C}_l}$ as follows:

$$I_k^f \in C_q, if\ q = \arg\max_{1\leq i\leq l}\{\mu_{\zeta_{\overline{C}_i}}(I_k^f)\}$$

i.e., if the membership degree of I_k^f belonging to C_q is the largest among the membership degrees of I_k^f belonging to $\zeta_{\overline{C}_1}, \zeta_{\overline{C}_2}, \ldots, \zeta_{\overline{C}_l}$, $I_k^f \in C_q$.

Algorithm 2 AFS clustering

Input:

Data X in $\mathbf{R}^{n\times d}$ with selected features.

Output:

clusters $C_1, C_2, ..., C_q$ and cluster's description $\zeta_{C_1}, \zeta_{C_2}, ..., \zeta_{C_q}$.

1: Construct three fuzzy terms $m_{i,1}, m_{i,2}, m_{i,3}$ for each feature f_i, represent large f_i, small f_i and medium f_i respectively;
2: Calculate membership function $\mu_{m\in M}(x)$ for each sample;
3: Find the set of fuzzy terms B_x^ε with $B_x^\varepsilon = \{m \in M|\mu_m(x) \geq max\{\mu_m(x)\}\}$
4: Build the description ζ_x for each sample with $\zeta_x = \bigwedge_{m\in B_x^\varepsilon} m$;
5: Calculate pairwise similarity $\mathbf{S}$ based on ζ_{x_i} and ζ_{x_j}, $S_{ij} = \min\{\mu_{\zeta_{X_j}\wedge\zeta_{X_i}}(X_i), \mu_{\zeta_{X_i}\wedge\zeta_{X_j}}(X_j)\}$;
6: Find the transitive closure $\mathbf{Q} = \mathbf{S}^r$, where $(\mathbf{S}^r)^2 = \mathbf{S}^r$;
7: Find the threshold $\alpha = \arg\min_{\alpha_v\in[\alpha_1,\alpha_2,\ldots,\alpha_u]}\{\mathcal{I}_\alpha\}$, where $\mathcal{I}_\alpha = \frac{\sum_{k=1,2,\ldots,n}\mu_{\zeta_{bou}}(x_k)}{\sum_{k=1,2,\ldots,n}\mu_{\zeta_{Total}}(x_k)}$;
8: Apply α to $\mathbf{Q}$, get the initial clusters $\overline{C_1}, \overline{C_2}, ..., \overline{C_q}$;
9: Construct the description for each cluster $\zeta_{\overline{C}_i} = \wedge_{\gamma\in\Gamma_i}\gamma$, where $\Gamma_i = \{\zeta_{x_k}|\frac{|\{y|y\in\overline{C}_i,\mu_{\zeta_{x_k}}(y)\geq\lambda\}|}{|\overline{C}_i|} \geq \omega, x_k \in \overline{C}_i\}$;
10: Obtain the final cluster label by the membership of x_k belonging to cluster's description, $x_k \in C_l$, where $l = \arg\max_{1\leq i\leq q}\{\mu_{\zeta_{\overline{C}_i}}(x_k)\}$.

Stop.

2.4 Summary

In this chapter, the semantic learning for image clustering is addressed. We present a novel fuzzy clustering approach, AFS clustering, which adopts fuzzy sets for data representation and calculates the similarity information based on the fuzzy logic operations defined on these fuzzy sets. By the experiments on semantic facial components clustering, we show that the semantic gap is well bridged by fuzzy membership functions. The clustering performance of the proposed method is also improved compared to K-means and FCM. Besides, a novel feature selection algorithm based on feature similarity ranking is proposed. The experimental results also demonstrate the effectiveness of the feature selection method.

In next chapter, we will discuss another crucial problem in image clustering, manifold learning. Using the manifold structure for image-related problems is proven to significantly improve performance. To this end, we propose another clustering approach, called AFSSC, to learn the underlying manifold structures of image.

Chapter 3

AFS based Spectral Clustering for Manifold Learning

3.1 Introduction

The previous chapter presented a fuzzy method, AFS clustering, for semantic learning problem of image clustering. However another problem remains, *i.e.*, manifold learning. AFS clustering yields a multi-scale hierarchical result for better revealing the natural fuzziness of semantic descriptions. However in many cases, it is often required to achieve a hard partition of the image set. To this end, we propose another clustering approach, namely AFSSC, to group images into different clusters by exploring the underlying manifold structures embedded in high-dimensional image space.

Advancements in digital camera and storage technology have created many high-resolution images. Nowadays, image data normally lie in a high-dimensional feature space. Due to the existence of noise and redundant information, however, image data is actually organized as underlying manifold structures in a much lower dimensional space, yielding poor performance of Euclidean space based approaches. Using the manifold structure instead has proven to significantly improve performance in many vision-related tasks [51, 52]. Since the manifolds are usually of a much lower dimensionality than the space in which they lie, many manifold learning algorithms make

use of dimension reduction methods, such as ISOMAP [53], LE [54], LLE [55], *etc*, to reduce the dimensionality of the feature space. Such dimensional subspaces are constructed in the hope that the Euclidean distance in the new lower dimensional subspace can better capture the geodesic distance compared with the original higher dimensional space.

The AFS framework offers us a useful tool for dimension reduction thereby revealing the latent manifold structure. However the AFS clustering method introduced in the last chapter solves similarity matrix by Transitive Closure, in which we apply different thresholds to obtain plenty of clustering results and the optimal result is picked by another evaluation criteria. Such a process is nontrivial and time-consuming, which makes it impractical for large-scale clustering problems. An alternative approach that was shown to handle similarity matrix is spectral clustering, which maps similarity into a weighted graph where the edges connecting the nodes are weighted by their pairwise similarity. The clustering is then represented as pure graph theory problems for which solid theories and powerful algorithms have been developed.

Spectral clustering normally contains two steps: constructing an affinity graph based on appropriate metric and establishing an appropriate way to "cut" the graph. Plenty of approaches exist to address the graph cut problem, such as *minimul cut* [20], *ratio cut* [21] and *normalized cut* [7], *etc*. For affinity graph construction, there are mainly three popular categories: (1) *The ε-neighborhood graph*: This kind of graph is constructed by connecting all points whose pairwise distances are smaller than a pre-set constant ε; (2) *The k-nearest neighbor graph*: Here the goal is to connect vertex v_i and v_j if v_j is among the k-nearest neighbors of v_i or v_i is among the k-nearest neighbors of v_j; (3) *The fully connected graph*: Here all vertices are connected and the edges are weighted by the positive similarities between each pair of vertices. According to [56], all three types of affinity graphs mentioned above are regularly used in spectral clustering, and there is no theoretical analysis on how the choice of the affinity graph would influence the performance of spectral clustering.

For the ε-neighborhood graph, [56] suggested to choose ε such that the resulting graph is safely connected. To determine the smallest value of ε where the graph is

connected is quite simple: one has to choose ε as the length of the longest edge in a minimal spanning tree of the fully connected graph on the data points, which can be easily solved by any minimal spanning tree algorithm. However, note that when the data contains outliers, this heuristic will choose ε so large that even the outliers are connected to the rest of the data. It also happens when the data contains several tight clusters which are very far apart from each other (multi-scale data distribution), which is quite common in real world data. In both cases, ε could be chosen too large to reflect the scale of the most important part of the data.

The k-nearest neighbor graph, on the other hand, can deal with the multi-scale problem. One can treat the k-nearest neighbor graph as a special type of the ε-neighborhood graph, where ε is a local parameter decided by the distance just to neighbors, rather than a global constant value. This is a general property of k-nearest neighbor graph which can be very useful. When working with the k-nearest neighbor graph, k should be chosen such that the resulting graph is connected, or at least has significantly fewer connected components than clusters we want to detect.

The crucial problem of constructing the fully connected graph is to define the weighting of the edges, which is the pairwise similarities. The notion of data similarity is often intimately tied to a specific metric function, typically the ℓ_2-norm (e.g. the Euclidean metric) measured with respect to the whole feature space. However, defining the pairwise similarity for effective spectral clustering is fundamentally a challenging problem [1] given complex data that are often of high dimension and heterogeneous, when no prior knowledge or supervision is available. Trusting all available features blindly for measuring pairwise similarities and constructing data graphs is susceptible to unreliable or noisy features [57], particularly so for real-world visual data, *e.g.* images and videos, where signals can be intrinsically inaccurate and unstable owing to uncontrollable sources of variations and changes in illumination, context, occlusion and background clutters, *etc.* [58]. Moreover, confining the notion of similarity to the ℓ_2-norm metric implicitly imposes unrealistic assumption on complex data structures that do not necessarily possess the Euclidean behavior [57].

In this work, we aim to deduce robust pairwise similarity so as to construct more

meaningful affinity graph, yielding performance improvement of spectral clustering. To achieve this goal, we first formulate a unified and generalized data distance inference framework based on AFS fuzzy theory [41] with two novel modifications: (1) Instead of using the complete feature space as a whole, the proposed model is designed to avoid indistinctive features using fuzzy membership function, yielding similarity graphs that can better express the underlying semantic structure in data. This will significantly reduce the number of features used in the clustering process; (2) The Euclidean assumption for data similarity inference is relaxed using fuzzy logic operations defined in AFS. The data distance is then put into the Gaussian kernel to enforce locality. It is worth mentioning that the distinctive features used to represent samples may be different from one another, *e.g.*, every sample could have its own feature subspace. Accordingly the distance measured is dependent on the pairwise feature subspace. A similar idea was presented in [59], which states that different similarities can be induced from a given sample pair if distinct propositions are taken or different questions are asked about data commonalities. In our proposed model, the assumption is that there is no optimal feature subspace which works well for all samples. Each sample pair has its own best feature subspace in terms of distance measure. In terms of AFS clustering, we propose a new method to solve the similarity matrix instead of using the transitive closure which needs additional evaluation criteria to obtain clustering result. Extensive experiments have demonstrated that the proposed method is superior compared to both the original spectral clustering and the AFS clustering, especially in high dimension due to the high capability of dimension reduction.

3.2 Related work

Large amount of work has been conducted on spectral clustering [7, 24, 60, 61, 62, 63]. Generally, existing approaches for improving spectral clustering performance can be classified into two paradigms: (1) How to improve data grouping while the affinity graph is fixed [7, 24, 62]. For example, Xiang and Gong [62] proposed to identify

informative and relevant eigenvectors of a given data affinity matrix; (2) How to construct appropriate affinity graphs so as to improve the clustering results with standard spectral clustering algorithms [60, 64, 19, 52, 65, 66]. For example, Chang and Yeung [65] proposed to use path-based similarity to construct robust affinity graph. We concentrate on the second paradigm in this work since the AFS is more related to similarity measure rather than data grouping.

Many approaches have been proposed for improving the robustness of affinity graphs adapting to the local data structures [7, 64, 67]. Particular attention has been focused on learning an adaptive scaling factor σ for the Gaussian kernel $\exp(-\frac{dist^2(x_i,x_j)}{\sigma^2})$, when computing the similarity between samples x_i and x_j. For example, Zelnik-Manor and Perona [60] proposed a local scale similarity measure by adjusting the scaling factor as follows:

$$A_{ij} = \exp\Big(-\frac{dist^2(x_i, x_j)}{\sigma_i \sigma_j}\Big) \tag{3.1}$$

where σ_i is the distance between point x_i and its k-th nearest neighbor. Yang [68] proposed a similar local scaling factor, which is the mean distance of the k nearest neighbors rather than just considering the k-th neighbor. These methods, however, are still susceptible to the presence of noisy and irrelevant features [57], as well as the choice of k.

To deal with this issue, Pavan and Pelillo [19] proposed a dominant sets algorithm for forming tight neighborhoods by selecting the maximal cliques (or maximizing average pairwise affinity), with the hope of constructing graphs with fewer false affinity edges between samples. Given an affinity matrix $\mathbf{A}$, and a probabilistic indicator vector x, dominant sets can be extracted by maximizing a quadratic objective function:

$$\max \quad f(x) = x^T \mathbf{A} x \quad s.t. \quad x \in \triangle \tag{3.2}$$

where,

$$\triangle = \{x \in \mathbb{R}^n \mid x \geq 0 \ \wedge \ 1^T x = 1\} \tag{3.3}$$

The above optimization problem can be solved by the so-called replicator dynamics, an iterative procedure, which is guaranteed to converge at optimal locations. However, Premachandran and Kakarala [52] argued that the above process can easily stuck at a local optima. Even if the dynamics converges at global maxima, there is no guarantee that the node whose neighborhood are actually being searched is even part of the maxima clique. This problem is actually quite serious when the objective is to learn a local neighborhood on the manifold. Another kNN based graph generation method was proposed in [52] where the consensus information from multiple kNN is used for discarding noisy edges and identifying strong local neighborhoods. The performance of the method turns out to be questionable, especially when a large quantity of potential noisy edges exist in the given kNN due to the unreliable input data, leading to possibly inconsistent neighbor votes from multiple kNNs.

More recently, a random forest based approach was proposed in [57]. This method exploits similarity information from the tree hierarchy, leading to a non-linear way of affinity graph construction. More specifically, all data go through the tree structure node by node via binary split function until the termination criterion is satisfied. Intuitively, a sample pair is considered more similar if they travel more nodes together. Assuming the travel paths of a sample pair (x_i, x_j) are P_i and P_j on tree t, their similarity on this tree is:

$$a_{i,j}^t = \frac{overlap(P_i, P_j)}{\max(P_i, P_j) - 1} \tag{3.4}$$

which is the length of the overlapping path of P_i and P_j divided by the maximum path between P_i and P_j (except the root node since all paths start from it). With the property of random forest, they combine multiple decision trees to generate the final smooth affinity as:

$$A_{i,j} = \frac{1}{T} \sum_{t=1}^{T} a_{i,j}^t \tag{3.5}$$

On the other hand, with the random forest framework, the model is capable of removing noisy features. The same idea is proposed in different ways in our approach.

Instead of blindly trusting all available variables, our proposed graph inference method exploits discriminative features for measuring more appropriate pairwise similarities. The affinity matrix created is thus more robust against the noisy real-world data.

AFS theory based clustering has been attempted in [43, 50, 69]. Instead of using the popular Euclidean metric, AFS clustering approaches capture the underlying data structure through fuzzy membership functions, and the distances between samples are represented by the membership degree. Furthermore, by extracting the description of samples, the methods are able to establish discriminative feature subspaces for distance measure, which provides a way to deal with commonly existed noise in real-world data. However, in the original AFS clustering, the similarity matrix $S = (s_{ij})_{N\times N}$ does not necessarily satisfy the fuzzy transitive condition $s_{ij} \geq \vee_k(s_{ik} \wedge s_{jk})$, where $\vee$, $\wedge$ stand for *max* and *min*, respectively. Usually an object is considered similar to another if and only if the degree of similarity between them is greater than or equal to a predefined threshold α. Therefore, the transitive condition states that, for any three objects i, j and k, if object i is similar to object k ($s_{ik} \geq \alpha$) and object k is similar to object j ($s_{kj} \geq \alpha$), object i is similar to object j ($s_{ij} \geq \alpha$) as well. Since the transitive condition is indispensable for clustering, the matrix can always be transformed into its Transitive Closure (denoted by $TC(S) = (t_{ij})_{N\times N}$). $TC(S)$ is defined as a minimal symmetric and transitive matrix. Usually, $TC(S)$ is obtained by searching for an integer k such that $(S^k)^2 = S^k$. With a given α, objects can now be partitioned into different clusters. The problem here is, each threshold α leads to a particular clustering result therefor an evaluation criteria is necessary to obtain a crisp result. It is nontrivial to build such criteria especially in fuzzy clustering. Furthermore, the similarity matrix may not be reflexive (*e.g.*, $s_{ii} = 1$ does not always hold), which means some samples cannot be clustered with certain α (when $s_{ii} < \alpha$). Therefore, a re-clustering process is needed for the original AFS clustering [69] (*e.g.*, to pick up samples which are not clustered in the previous clustering process). The above processes are nontrivial and time-consuming.

The problems mentioned above motivate us to use the AFS theory to construct the affinity matrix, and build affinity graph based on the affinity matrix by applying

the pairwise similarities as the weighting of the edges in the graph. Standard graph cut algorithm can then be used to solve the graph problem. We name this approach as AFSSC.

3.3 AFS based spectral clustering

3.3.1 Data representation by fuzzy sets

Given a set of data points X in R^n, and a feature set $F = \{f_1, f_2, ..., f_l\}$, a set of fuzzy terms $M = \{m_{i,j} | 1 \leq i \leq l, 1 \leq j \leq k_i\}$ can be defined, where $m_{i,1}, m_{i,2}, \ldots, m_{i,l}$ are fuzzy terms associated with the feature f_i in F. Usually we set $k = 3$ meaning that each feature f_i is associated with 3 fuzzy terms $m_{i,1}, m_{i,2}, m_{i,3}$, representing 3 different semantic concepts (*E.g.*,"large", "medium", and "small" respectively). However if a certain feature f_i is a Boolean parameter, k_i is set as 2 and only two fuzzy terms $m_{i,1}, m_{i,2}$ are defined for this feature. The feature dimension is hence expanded from ℓ to $k\ell$. In this expanded feature space, which is more specific and discriminative, it is easier to identify the feature subspaces as the description for each sample.

With such explicit feature space, it is unavoidable that noisy features exist for either the pairwise distance or similarity. As mentioned before, our assumption is that there is no optimal feature subspace which works well for all samples. Each sample pair holds its own best feature subspace in terms of distance measure. Hence, different subspace needs to be found for each sample, in which the sample is the most discriminative in it. The fuzzy membership function as given in Eq.2.5 is utilized for such a purpose. With $m_{i,1}, m_{i,2}, m_{i,3}$ as the fuzzy terms defined as "large", "medium" and "small" respectively on the feature f_i of data X with N samples, their weight functions can be defined following the Definition 1. $\forall x \in X$,

$$\rho_{m_{i,1}}(x) = \frac{f_i(x) - h_{i,2}}{h_{i,1} - h_{i,2}} \tag{3.6}$$

$$\rho_{m_{i,2}}(x) = \frac{h_{i,4} - |f_i(x) - h_{i,3}|}{h_{i,4} - h_{i,5}} \tag{3.7}$$

$$\rho_{m_{i,3}}(x) = \frac{h_{i,1} - f_i(x)}{h_{i,1} - h_{i,2}} \tag{3.8}$$

where $h_{i,1} = max\{f_i(x_k)\}$, $h_{i,2} = min\{f_i(x_k)\}$, $h_{i,3} = \frac{1}{N}\sum_{k=1}^{N} f_i(x_k)$,$h_{i,4} = max\{|f_i(x_k) - h_{i,3}|\}$, $h_{i,5} = min\{|f_i(x_k) - h_{i,3}|\}$. The weight functions are actually measures on the original feature space for semantic concepts "large", "medium" and "small". More specifically, with these weight functions the original feature space can be transformed to the AFS measure space, where semantic concepts are defined.

Next, in the new measure space, for each sample x, we find a salient fuzzy subset $\zeta_x = \prod_{m \in M} m$, such that ζ_x, rather than the entire fuzzy set, is good enough to represent x. This set is called "the description of x". Here the fuzzy membership function is used as the measure of goodness of features (fuzzy terms). If the membership of x_k belonging to $m_{i,j}$ is larger than a certain threshold, it means $m_{i,j}$ is good enough to distinguish x_k from the others. It can be easily seen that the so-called "description of x" is basically a feature subspace. Instead of using the complete feature space as a whole, the proposed method is designed to avoid indistinctive features using the fuzzy membership function, yielding sample representation that can better express the underlying semantic structure in data. A set B_x^ε can be defined using the "good enough" fuzzy terms:

$$B_x^\varepsilon = \{m \in M | \mu_m(x) \geq max\{\mu_m(x)\} - \varepsilon\} \tag{3.9}$$

where ε is a small positive number representing error threshold, which is set empirically. B_x^ε is the set of all possible fuzzy terms that can represent x very well. The description of x can be given as:

$$\zeta_x = \bigwedge_{m \in B_x^\varepsilon} m \tag{3.10}$$

where $\bigwedge$ is the fuzzy conjunction logic operations in AFS algebra(refer to Eq.2.1). By doing so, all desirable fuzzy terms are conjuncted together for the sample representation. For instance, if m_1 is "a tall man", m_2 is "an old man", then the conjunction

$m_1 \bigwedge m_2$, denoted by $m_1 m_2$, is "a tall old man".

3.3.2 Distance measure by AFS theory

The procedure above allows us to represent samples with different fuzzy terms in the discovered discriminative fuzzy subspaces. These subspaces are selected by fuzzy membership function with the indistinctive or noise features eliminated. Therefore they are considered to be able to improve the intra-similarities and reduce the inter-similarities. Furthermore, the Euclidean assumption is relaxed for data distance inference by the fuzzy membership and logic operations defined in AFS. More specifically, we use the membership degree of one sample belonging to another's description represented by the fuzzy set as the distance metric. For two samples X_i and X_j, the distance between them is defined as:

$$D_{ij} = 1 - \min \ \{\overline{\mu}_{\zeta_{X_j}}(X_i),\ \overline{\mu}_{\zeta_{X_i}}(X_j)\} \tag{3.11}$$

$$\overline{\mu}_{\zeta_{X_i}}(X_j) = \left\{ m_k \in \zeta_{X_i} | \frac{\sum_{k=1}^{N} \mu_{m_k}(X_j)}{N} \right\} \tag{3.12}$$

where $\mu_{m_k}(X_j)$ $(0 \leq \mu_{m_k}(X_j) \leq 1)$ is the membership of X_j belong to the fuzzy term m_k, as defined in Eq.2.5. Note that m_k represents each fuzzy term belonging to ζ_{X_i} which is the description of x_i. Clearly, $\overline{\mu}_{\zeta_{X_i}}(X_j)$ represents the mean membership degree of x_j belonging to the description of x_i. Instead of using the entire feature space blindly, this distance measure only considers the distinctive features shared by the pair of samples, yielding pairwise distances that better express the real structure in data by reducing useless or noisy features.

3.3.3 Spectral clustering

The widely used method to construct affinity graph for spectral clustering is the Gaussian kernel, which is also employed in the proposed method. However, as shown in Section 3.2, in order to construct a better affinity graph, the pairwise similarity

needs to be adaptive to the neighborhood information. Inspired by the k-nearest neighbor distances in [60, 68], we use a local adaptive kernel size. The new affinity is constructed by:

$$A_{ij} = \exp\Big(- \frac{D_{ij}^2}{\sigma \beta_{ij}} \Big) \tag{3.13}$$

$$\beta_{ij} = mean(KNN(X_i)) \cdot mean(KNN(X_j)) \tag{3.14}$$

where σ is the original Gaussian kernel and $KNN(X_i)$ are the distances between sample X_i and its k-nearest neighbors. It can be seen that our kernel is kind of combining the original kernel size σ^2 with the self-tuning kernel $\sigma_i \sigma_j$. This is based on the assumption that an appropriate kernel size needs not only the neighbor information to enforce locality, but also the global scaling parameter to control how rapidly the similarity falls off with distance.

Comparing to the widely used Gaussian kernel similarity function, there are three innovations in our affinity graphs construction:

- Instead of using the complete feature space, the proposed model is designed to avoid indistinctive features using fuzzy membership function, yielding similarity graphs that can better express the underlying semantic structure in data;
- The Euclidean assumption is relaxed for data similarity inference by fuzzy logic operations defined in AFS, so the metric is not ℓ_2-norm anymore;
- The distances are still put into the Gaussian kernel, but with a local kernel size adapted to the k-nearest neighbors.

After the affinity graph is constructed, the most widely used spectral clustering from [24], Ng-Jodan-Weiss (NJW) algorithm, is employed. The summary of our algorithm can be found in Algorithm3.

Algorithm 3 Spectral Clustering based on Axiomatic Fuzzy Set Similarity

Input:

Data set X in $\mathcal{R}^{n\times l}$, number of clusters k.

Output:

Class labels y_i

1: Construct the fuzzy terms $m_{i,j}$ for each feature f_i;
2: Calculate the membership function $\mu_{m\in M}(x)$ with Eq.2.5 for each sample;
3: Find the set of fuzzy terms B_x^ε with $B_x^\varepsilon = \{m \in M | \mu_m(x) \geq max\{\mu_m(x)\} - \varepsilon\}$
4: Build the description ζ_x for each sample with $\zeta_x = \bigwedge_{m\in B_x^\varepsilon} m$;
5: Calculate the pairwise distance $\mathbf{D}$ based on their descriptions using $D_{ij} = 1 - \min\{\overline{\mu}_{\zeta_{X_j}}(X_i),\ \overline{\mu}_{\zeta_{X_i}}(X_j)\}$;
6: Construct the affinity matrix $\mathbf{A}$ with the Gaussian kernel by $A_{ij} = \exp(-\frac{D_{ij}^2}{\sigma\beta_{ij}})$ and $\beta_{ij} = mean(KNN(X_i)) \cdot mean(KNN(X_j))$;
7: Define $\mathbf{Q}$ to be the diagonal matrix whose (i,i) elements are the sum of $A's$ i-th row, and construct the Laplacian matrix $\mathbf{L} = \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{Q}^{-1/2}$;
8: Find the k largest eigenvectors of $\mathbf{L}$ (chosen to be orthogonal to each other in the case of repeated eigenvalues), $e_1, e_2, ..., e_k$, and form the matrix $E = [e_1 e_2 ... e_k] \in \mathcal{R}^{n\times k}$ by stacking the eigenvectors in columns;
9: Form matrix $\mathbf{Y}$ from $\mathbf{E}$ by re-normalizing each of $\mathbf{E}$'s rows to have unit length (i.e. $Y_{ij} = E_{ij}/(\sum_j E_{ij}^2)^{1/2}$);
10: Treating each row of $\mathbf{Y}$ as a point $\mathcal{R}_k$, cluster them into k clusters via k-means;
11: Finally, assign the original point x_i to cluster j if and only if row i of matrix $\mathbf{Y}$ was assigned to cluster j.

Stop.

3.4 Summary

In this chapter, we proposed a novel clustering method, AFSSC, to address manifold learning in image clustering. To this end, we first formulate AFS as a unified and generalized data distance inference framework with two innovations: (1) Instead of using the complete feature space as a whole, the proposed model is designed to avoid indistinctive features by removing the features with low membership degree, yielding similarity graphs that can better express the underlying structure of data; (2) The Euclidean assumption for data distance inference is relaxed by fuzzy logic operations and fuzzy sets. Since the context information is proven to be significantly useful for pairwise similarity, a novel neighborhood adaptive Gaussian kernel is then proposed to map the pairwise distance to similarity as well as enforce the locality. Standard spectral clustering method is finally used to obtain the result. We demonstrate the

experiments on various of data, such as UCI real-world data, USPS handwritten digits, face data, *etc*. The proposed method AFSSC is compared to K-means, FCM, AFS, and other state-of-the-art methods. The qualitative comparison of the affinity graphs shows that AFSSC can better reveal the pairwise similarity by increasing the intra-class similarity and decreasing the inter-class similarity. The improvement of the clustering performance based on clustering error and normalized mutual information also prove the superiority of AFSSC compared to other state-of-the-art methods.

Chapter 4

Experimental Results and Analysis

In chapter 2 and 3, we present two clustering methods, AFS clustering and AFSSC. The AFS clustering is designed for semantic learning problem. By representing semantic concepts as fuzzy sets, the semantic gap is bridged automatically and objectively using the AFS clustering. Next AFS is formulated as a similarity measure and combine with spectral clustering to generate a novel clustering approach, named AFSSC, in order to learn the manifold structure for image data. AFSSC is fundamentally based on the AFS clustering method, especially on the distance/similarity part. Nevertheless, AFSSC is proposed to reveal the underlying manifold structure. Instead of a multi-scale hierarchical clustering, the goal of AFSSC is an optimal hard partition of data. To this end, the similarity measure is improved by a neighborhood adaptive Gaussian kernel to better reveal pairwise relationship. The measured similarity matrix is mapped onto a weighted graph so that the clustering problem can be solved by an optimized graph cut method.

The two proposed methods have been applied to various types of datasets in extensive experiments to evaluate their performances. Firstly AFS clustering is applied for semantic facial component clustering. Both the extracted semantic concepts and clustering results are evaluated for detailed comparisons. Next, the AFSSC is applied on UCI data, USPS handwritten digits, and face data. The results have shown the improvement of clustering performance compared to AFS clustering and other state-of-the-art methods.

4.1 Experiments for AFS clustering

The proposed AFS clustering method is applied for facial components clustering on Multi-PIE and AR databases. Specifically, 249 frontal faces from 249 subjects in Session one of Multi-PIE and 134 frontal faces from 134 subjects in Session one of AR database are used for this experiment. Since the face sizes vary in scale, the images were scaled so that the Euclidean distance between the pupil centers in every image tallies with the average distance of the database. The results are illustrated in two aspects. The first one is the semantic extraction, where the relationship between semantic concepts and low-level features are demonstrated. Each facial component is clustered into three groups, "large", "medium" and "small". "Medium" is not usually a useful semantic concept in many applications. It is not so useful to describe a person with "medium eye" or "medium nose". Therefore we focus on two salient clusters, "large" and "small". For these features, we show that the clustering result by the proposed approach is better than those obtained by K-means and FCM in terms of area of facial components and defined clustering indexes.

4.1.1 Eye clustering

We first evaluate the semantic concepts represented by the low-level features. The "semantic gap" is well bridged by the relations between them. We start with the proposed feature selection, and the global feature, centroid distance, plus several local features are selected as the low level features for these two databases, as shown in Fig.4-1.

As shown in Fig.4-1, in addition to the centroid distance which is selected as the global feature, three local features are also selected to represent the shape information. This is very important, because the global feature can guide the clustering process while the local features can help to ensure the shape consistency. In our system, these features are automatically selected by a feature selection algorithm, for both Multi-PIE and AR data sets. To illustrate the advantage of the feature selection and eye clustering, the relationships between semantic concepts and the low-level features

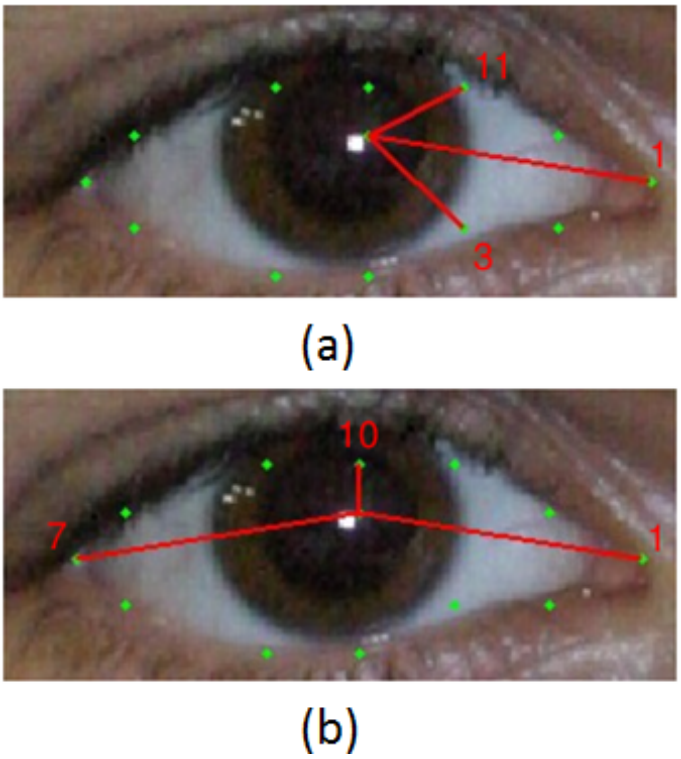

Fig. 4-1: Selected "local" Features of (a) Multi-PIE (b) and AR.

are shown as below after clustering.

- **Multi-PIE Large Eyes Cluster**: $\zeta_{C_l} = m_{15,1}^{f_1} m_{11,1}^{f_1} m_{1,1}^{f_1}$, with the semantic rules: "*The eyes in this cluster have large* f_{15}*, large* f_{11} *and large* f_1";

- **Multi-PIE Small Eyes Cluster**: $\zeta_{C_s} = m_{15,2}^{f_1} m_{11,2}^{f_1} m_{3,2}^{f_1}$, with the semantic rules: "*The eyes in this cluster have small* f_{15}*, small* f_{11} *and small* f_3".

- **AR Large Eyes Cluster**: $\zeta_{C_l} = m_{15,1}^{f_1} m_{10,1}^{f_1} m_{7,1}^{f_1}$, with the semantic rules: "*The eyes in this cluster have large* f_{15}*, large* f_{10} *and large* f_7";

- **AR Small Eyes Cluster**: $\zeta_{C_s} = m_{15,2}^{f_1} m_{10,2}^{f_1} m_{7,2}^{f_1}$, with the semantic rules: "*The eyes in this cluster have small* f_{15}*, small* f_{10} *and small* f_7".

From the description above, it is easy to see that the characteristic combinations contain not only the global feature $m_{15}^{f_1}$ (centroid distance) but also the local features which are related to the height and width of the eye. Such features can make the

clustering better and more consistent. Fig.4-2 shows the improvement of our method in terms of shape by including local features. Some long and narrow eyes that would be inappropriately clustered as "large" with only the global features, are correctly clustered as "medium" with the additional local features.

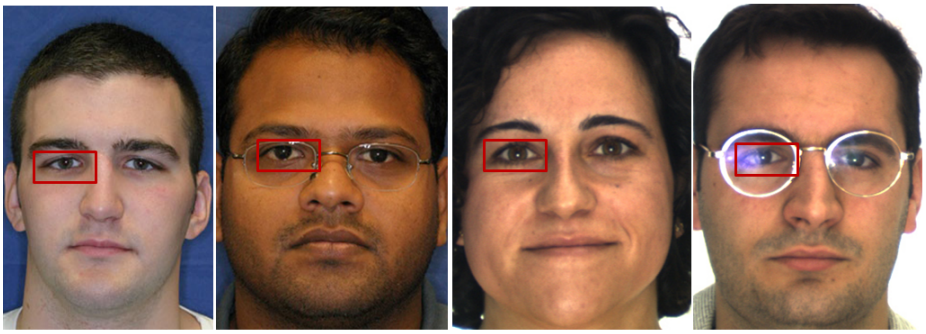

Fig. 4-2: Examples of eyes clustered as "large" by using only the global features and clustered as "medium" by using both the global and local features.

In order to show the quality of the clustering visually, Fig.4-3 shows four examples of facial image with large eye in "large eyes" cluster and four in "small eyes" cluster. The comparison result is very encouraging in the sense that the eyes in the "large eyes" cluster are distinctly larger than those in the "small eyes" cluster, and also the eye shapes are more consistently within the clusters.

Next, we evaluate the clustering performance of the proposed AFS clustering method. In the available data sets there is no semantic ground truth to evaluate the accuracy of our results. As we are actually clustering the "size" of the facial components, the "area" measurement is chosen to be the validity criteria, which is the closest to human perception (i.e., a larger eye is the one with a larger area). The area criteria is also adopted in [70]. Three parameters representing class separability are defined as follows for evaluating the clustering performance,

- The average area of small eyes class $\overline{A_S}$. $\overline{A_S}$ is defined as the mean area of all subjects in the "small" cluster; a lower value means the eyes in the "small" cluster are smaller the eye of an average person;

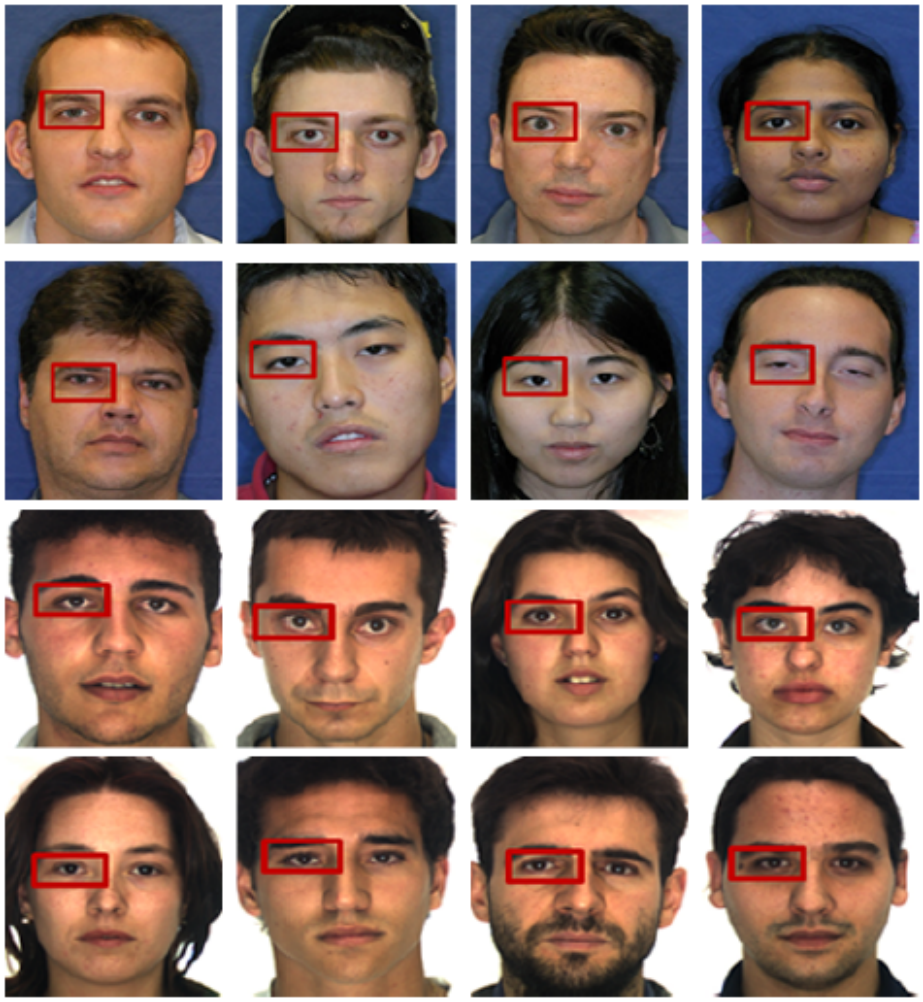

Fig. 4-3: Examples of large eyes and small eyes. From top row to bottom row, they are Multi-PIE large, Multi-PIE small, AR large and AR small respectively.

Table 4.1: Comparison of Clustering Algorithms for Eye Clustering

Data set	Method	Evaluation Criteria		
		$\overline{A_S}$	$\overline{A_L}$	CS
Multi-PIE	K-means	273	419	0.456
	FCM	344	427	0.358
	Proposed(global)	293	455	0.363
	Proposed(global+local)	**177**	**478**	**0.760**
AR	K-means	523	586	0.417
	FCM	523	569	0.349
	Proposed(global)	468	623	0.613
	Proposed(global+local)	**447**	**635**	**0.730**

- The average area of large eyes class $\overline{A_L}$. $\overline{A_L}$ is the mean area of the "large" cluster; a higher value means the eyes in the "large" cluster are larger than that of an average person;

- The class separability CS. CS is defined as the quotient of the minimum inter-class distance and the maximum intra-class distance.

$$CS = \frac{\min\left\{D_{mn} | D_{mn} = \sum_{j=1}^{M} \sqrt{(\overline{X}_{m,j} - \overline{X}_{n,j})^2}\right\}}{\max\left\{D_{pq} | D_{pq} = \sum_{j=1}^{M} \sqrt{(X_{p,j} - X_{q,j})^2}\right\}} \tag{4.1}$$

  where $X_{p,j}$ denotes feature value for p along the jth image, $\overline{X_m}$ is the center/mean of the m cluster, k and M are numbers of clusters and features.

Our proposed method is compared with the conventional clustering algorithms, K-means and FCM on the face image data sets Multi-PIE and AR. Due to their sensitivity to the initial conditions, K-means and FCM are each performed 100 times to obtain the average result. The proposed method is applied with global features only, as well as global and local features combination.

From Table 4.1, it can be seen that compared to K-means and FCM, our method performs much better for all three indices in both the Multi-PIE and AR databases.

Our method can generate much clearer and more reliable clusters in terms of the size and shape of the eye. In addition, K-means and FCM can only produce the partition clusters, but not semantic descriptions which are useful features for semantic FIR systems. Furthermore, it can be seen that the combination of global and local features performs significantly better in the Multi-PIE data set and is slightly superior in the AR data set.

4.1.2 Nose clustering

Fig.4-4 shows us the features selected by our automatic feature selection algorithm for nose clustering. Two global features, centroid distance and height, with two local features from "star" are selected for the AR database, while in the Multi-PIE database, centroid distance and four local features are selected.

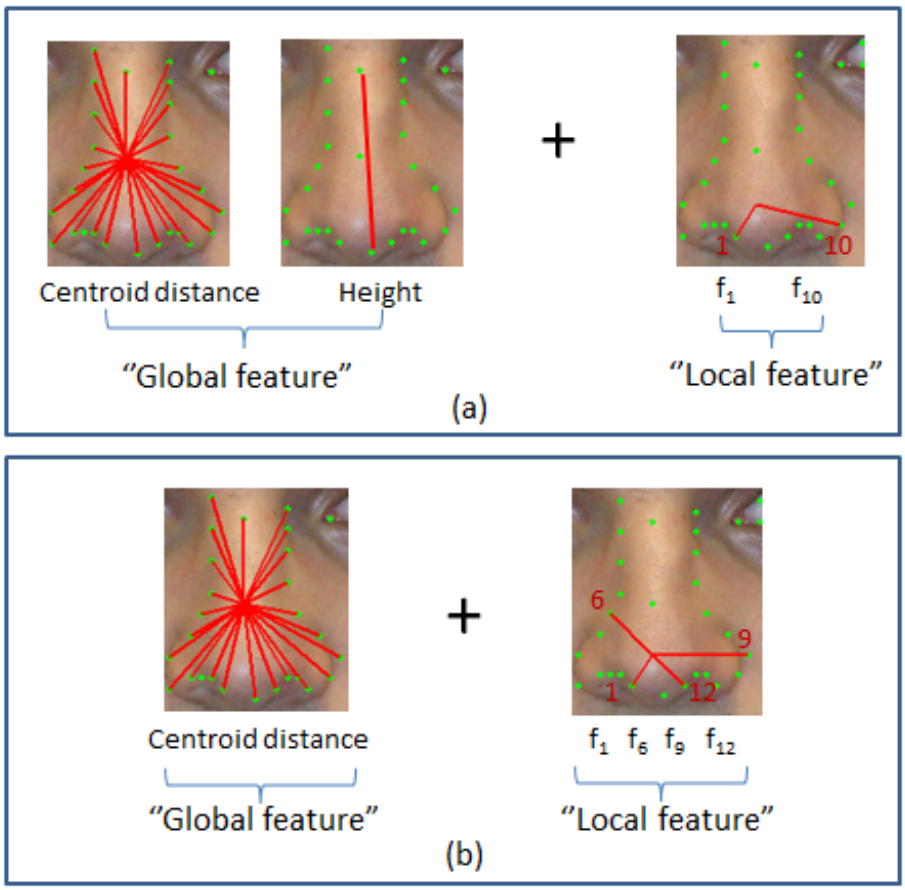

Fig. 4-4: Selected features for nose clustering. (a) AR; (b) Multi-PIE.

To prove the superiority of these feature combinations, the clustering description is listed below. Note that the clustering description is the representation of low-level features, and meanwhile in terms of high-level semantics, we can easily know which one is "large nose" while the other is "small nose". Hence semantic concepts are expressed by low-level features and the semantic gap is naturally bridged.

- **Multi-PIE Large Noses Cluster**: $\zeta_{C_l} = m_{15,1}^{f_2} m_{14,1}^{f_2} m_{1,1}^{f_2} m_{10,1}^{f_2}$, with the semantic rules: "*The eyes in this cluster have large* f_{15}*, large* f_{14}*, large* f_1 and large f_{10}";

- **Multi-PIE Small Noses Cluster**: $\zeta_{C_s} = m_{15,2}^{f_2} m_{14,2}^{f_2} m_{1,2}^{f_2} m_{10,2}^{f_2}$, with the semantic rules: "*The eyes in this cluster have small* f_{15}*, small* f_{14}*, small* f_1 and small f_{10}";

- **AR Large Noses Cluster**: $\zeta_{C_l} = m_{15,1}^{f_2} m_{1,1}^{f_2} m_{9,1}^{f_2} m_{12,1}^{f_2}$, with the semantic rules: "*The eyes in this cluster have large* f_{15}*, large* f_1*, large* f_9 *and large* f_{12}";

- **AR Small Noses Cluster**: $\zeta_{C_s} = m_{15,2}^{f_2} m_{6,2}^{f_2} m_{9,2}^{f_2}$, with the semantic rules: "*The eyes in this cluster have small* f_{15}*, small* f_6 *and small* f_9".

From the clustering description, it is clear that not only the global features are used, but also several local features. In this case, different noses are distinctively grouped into different clusters depends on both their sizes and shapes. In order to show the quality of the clustering visually, Fig.4-5 shows four examples of facial image in large nose cluster and four in small noses cluster. The comparison result is very encouraging in that the noses in the "large noses" cluster are distinctly larger than those in the "small noses" cluster, and also the shape is kept more consistently.

To show the improvement of the added local features, Fig.4-6 shows some examples of noses that are inappropriately clustered as "large" when using only the global features such as "area". When combing with local features, they are correctly clustered as "medium" as they are not large enough in terms of the local features used in our method.

Fig. 4-5: Examples of large noses and small noses. From top row to bottom row, they are AR small, AR large, Multi-PIE small and Multi-PIE large respectively.

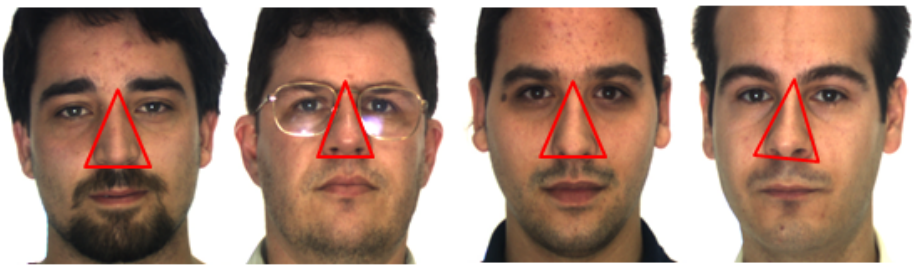

Fig. 4-6: Examples of noses clustered as "large" by using only the global features but clustered as "medium" by using the global and local features.

Table 4.2 presents the performance comparison of different algorithms for nose clustering. Similar metrics as used for eye clustering are used here as well. It can be clearly seen that K-means and FCM produce similar results while the proposed method with only the global features performs much better on the Multi-PIE but performs similarly on the AR database. Compared to the traditional approaches K-means and FCM, or the proposed method with only the global features, the proposed method using both the global and local features is superior on both the Multi-PIE and AR datasets. It proves the superiority of the proposed method as well as the advantage of the added local features, which is evident in all three evaluation criteria. The mean area of the "small nose" group is the smallest while the mean area in the "large nose" group is much larger than those produced by the other methods. Moreover, the class separability demonstrates that the proposed method produces a much clearer clustering result.

Table 4.2: Comparison of Clustering Algorithms for Nose Clustering

Data set	Method	Evaluation Criteria		
		$\overline{A_S}$	$\overline{A_L}$	CS
Multi-PIE	K-means	1679	2143	0.494
	FCM	1718	2131	0.406
	Proposed(global)	1578	2214	0.818
	Proposed(global+local)	**1487**	**2264**	**0.840**
AR	K-means	2564	3169	0.579
	FCM	2530	3162	0.583
	Proposed(global)	2506	3168	0.561
	Proposed(global+local)	**2480**	**3582**	**0.832**

4.1.3 Parameter analysis

In the proposed algorithm, the size of the reduced feature subset and hence the scale of details of the data representation is controlled by the parameter k (refer Eq.(2.14)). Figure 4-7 illustrates such an effect on two data sets, Multi-PIE and AR. For certain

range of k, it is observed that there is no change in the reduced subset, i.e., no reduction in dimension occurs. However, as expected, the size of the reduced subset decreases overall with the increment of k. In this way, the representation of data at different degrees of details is controlled by the choice of k. This characteristics is useful in many areas where multi-scale representation of the data is necessary. Note that the said property may not always be possessed by other algorithms where the input is usually the desired size of the reduced feature set. The reason is that changing the size of the reduced set may not necessarily result in any change in the level of details. In contrast, for the proposed algorithm, k acts as a scale parameter which controls the degree of details in a more direct manner, making it easier for the user to have a good control on the final clustering outcome in relation to the level of details.

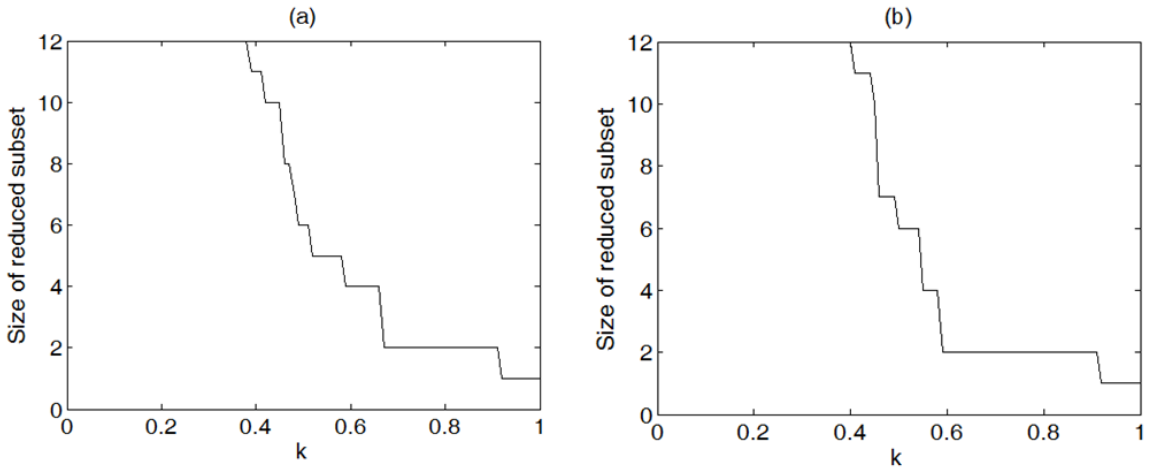

Fig. 4-7: Effect of parameter k. (a) Multi-PIE eye; (b) AR eye

Some experiments are conducted to show the change of clustering performance along with the number of selected features. Two validation indices, $V1$ and $V2$ are used, which are related to the indices used in the previous experiments. $V1$ is defined as the difference of $\overline{A_L}$ and $\overline{A_S}$. $V2$ is the class separability CS. They are normalized to $[0, 1]$ for clear observation.

Fig.4-8 shows that the clustering performance is sensitive with the change of number of features. In both the Multi-PIE and AR data set, four features provides the best clustering performance. With more than four features, the performance would

actually decrease. It also demonstrates the significance of our feature selection algorithm that can help to reduce computation cost as well as to find the best scale of details of data representation.

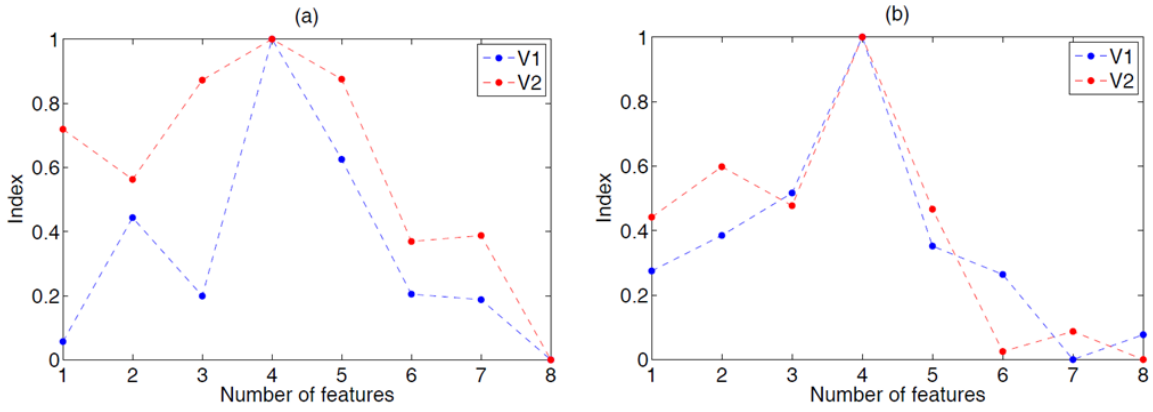

Fig. 4-8: Clustering performance with different number of features. (a) Multi-PIE eye; (b) AR eye

4.2 Experiments for AFS based spectral clustering

4.2.1 Experimental Settings

The proposed method AFSSC method and the widely used NJW spectral clustering (SC) [24], self-tuning spectral clustering (STSC) [60], and AFS clustering (AFS) [69] methods are applied on the same data sets from UCI, USPS handwritten digits and CMU-PIE, Yale face images. In the experiment, for SC, the value of σ is obtained by searching and the one which gives the best result is prcked, as suggested in [24]. With STSC and AFSSC, M varies from 1 to 100 (including 7 as suggested in [60]) and the one that gives the best result is chosen. The values of ϵ and σ are set by searching over the ranges of 0.1 to 0.3 and 0.1 to 1 respectively, and those giving the best results are used [24].

4.2.2 Performance Measure

To evaluate the performance of the proposed algorithm, we compare the clustering results with other clustering methods using two performance metrics: *Clustering Error* and *Normalized Mutual Information*.

Clustering Error [71] is widely used to evaluate clustering performance. For a clustering result, the permutation mapping function $map(\cdot)$ needs to be built that maps the generated cluster index to a true class label. The clustering error (CE) can then be computed as:

$$CE = 1 - \frac{\sum_{i=1}^{n} \delta(y_i, map(c_i))}{n} \tag{4.2}$$

where y_i and c_i are the true class label and the obtained cluster index of x_i respectively, $\delta(x, y)$ is the delta function that equals 1 if $x = y$ and equals 0 otherwise. The clustering error is defined as the minimum error among all possible permutation mappings. This optimal matching can be found with the Hungarian algorithm [72], which is devised for obtaining the maximum weighted matching of a bipartite graph.

Normalized Mutual Information (NMI) is another widely used metric for measuring clustering performance. Vinh et al. [73] reported that some popular metrics do not facilitate informative clustering comparisons because they either do not have a predetermined range or do not have a constant baseline value. For those metrics, a poor clustering could yield a very high performance index, especially when there are many clusters. However, according to Wu et al. [74], when clustering performances are hard to distinguish, the normalized variation of mutual information, i.e. NMI, could still work very well. For fair comparison, we employ NMI as another metric for comparing clustering performance. For two random variables X and Y, the NMI is defined as [75]:

$$NMI(X, Y) = \frac{I(X, Y)}{\sqrt{H(X)H(Y)}} \tag{4.3}$$

where $I(X, Y)$ is the mutual information between X and Y, while $H(X)$ and $H(Y)$ are the entropies of X and Y respectively. Clearly $NMI(X, X) = 1$, which is the

maximum possible value of NMI. Given a clustering result, NMI in Eq.4.3 is estimated as [75]:

$$NMI = \frac{\sum_{l=1}^{c}\sum_{h=1}^{c} n_{l,h}\log(\frac{nn_{l,h}}{n_l\hat{n}_h})}{\sqrt{(\sum_{l=1}^{c} n_l \log\frac{n_l}{n})(\sum_{h=1}^{c}\hat{n}_h\log\frac{\hat{n}_h}{n})}} \tag{4.4}$$

where n_l denotes the number of data contained in the cluster $C_l(1 \leq l \leq c)$,$\hat{n}_h$ is the number of data belonging to the h-th class$(1 \leq h \leq c)$, and $n_{l,h}$ represents the number of data that are in the intersection between the cluster C_l and the h-th class. The larger the NMI, the better the performance.

4.2.3 Experiments on UCI Datasets

We applied our algorithm and the other methods on 9 benchmark data sets from UCI data repository [46]. Details of the data sets are listed in Table 4.3.

Table 4.3: Description of the UCI datasets.

Data set	instances	dimensions	classes
heart	270	13	2
hepatitis	155	19	2
sonar	208	60	2
wobc	699	9	2
wdbc	569	30	2
iris	150	4	3
wine	178	13	3
protein	552	77	8
libras	360	90	15

Experimental results based on Clustering Error (CE) are presented in Table 4.4. It clearly shows that the proposed AFSSC outperforms SC, STSC and AFS clustering methods. Compared to the AFS, our method improves over 10% on several data sets, such as heart, hepatitis, and wdbc. It proves that utilizing spectral theory to partition similarity matrix is much better than transitive closure theory used previously. Considering the validation loop for transitive closure method, our approach is also relatively faster and easier to implement. It is also observed that AFS clustering is

easy to fail in multi-cluster cases, such as protein and libras. The reason is that the original AFS selects the best partition of data based on the boundaries of each cluster. As the number of clusters increases, so does the difficulty of finding the boundaries. Compared to the Euclidean distance based methods, SC and STSC, the proposed algorithm also shows the superiority. Slightly worse results are observed on iris and wine data sets compared to STSC. Since there are only 150 and 178 samples in these two data sets respectively, the differences are actually just 1 or 2 samples, while AFSSC achieves 6%, 11%, and 7% improvements on heart, hepatitis, and sonar, which equals to 16, 17 and 14 respectively, compared to STSC. It also can be observed from Table 4.4 that removing the Gaussian kernel, AFSSC achieves results comparable to SC and STSC. However, adding kernel consistently improves the performance of AFSSC. Thus, kernel will always be included with our proposed AFSSC and is used for all the other experiments.

Table 4.4: Comparison of Clustering Error (%) on UCI data sets.

Method	heart	hepatitis	sonar	wobc	wdbc	iris	wine	protein	libras	Average
SC	19.6	29.7	42.3	3.4	9.5	10.0	2.8	53.7	53.4	24.9
STSC	21.1	38.7	42.8	3.3	7.2	**7.3**	**2.8**	56.2	53.4	25.9
AFS	29.6	44.1	37.2	2.7	18.1	9.6	3.4	64.6	62.7	30.2
AFSSC(without kernel)	17.4	32.3	38.5	3.6	10.4	11.3	5.1	54.3	54.2	25.2
AFSSC	**15.2**	**27.1**	**35.6**	**2.7**	**6.0**	8.8	3.4	**51.1**	**52.3**	**22.5**

The clustering performances measured in Normalized Mutual Information (NMI) are shown in Table 4.5. The proposed AFSSC outperforms the other methods by over 5%. It can be seen from the result that STSC is not very robust. It performed extremely poor on both the hepatitis and sonar datasets in terms of NMI. Since neighborhood information is used both in STSC and AFSSC, we can conclude that adopting fuzzy membership with distinctive feature subspace as the distance measure is more robust in terms of affinity graph construction, compared to using the Euclidean distance based on the entire feature space.

Table 4.5: Comparison of NMI (%) on UCI data sets.

Method	heart	hepatitis	sonar	wobc	wdbc	iris	wine	protein	libras	Average
SC	28.5	14.5	7.5	77.1	63.3	77.8	89.3	54.4	63.7	52.9
STSC	25.8	4.8	1.6	80.0	61.4	79.2	**89.3**	46.2	64.9	50.4
AFS	17.6	3.2	17.2	73.8	60.3	78.5	85.5	35.6	38.1	45.5
AFSSC(without kernel)	33.1	8.9	3.9	76.4	52.2	72.7	81.6	57.1	65.2	50.1
AFSSC	**38.1**	**15.8**	**19.0**	**81.7**	**68.9**	**81.4**	86.9	**62.1**	**68.0**	**58.0**

4.2.4 Experiments on USPS Datasets

Next the spectral clustering algorithms are applied on handwritten digits from the widely used USPS database [76]. The dataset contains numeric data obtained from the scanning of handwritten digits from envelopes by the U.S. Postal Service. The original scanned digits are binary and of different sizes and orientations. The images we used have been size-normalized, resulting in 16×16 grayscale images with 256 dimensions. Fig.4-9 shows some samples in this dataset. It contains 7291 training instances and 2007 test instances (9298 in total). Digit subsets {0,8}, {4,9}, {0,5,8}, {3, 5, 8}, {1,2,3,4} and {0,2,4,6,7} are used to test our algorithm. The details of these subsets are shown in Table.4.6. Experiments are conducted based on each subset separately and CE and NMI are used to measure the performance. Note that AFS is not applied on those datasets, since that when running AFS on these data sets, the 8GB RAM in our Windows 7 desktop is not big enough. We believe this shows one of the feasibility limitations of the AFS algorithm. On the other hand, even if the AFS algorithm could be successfully run for these datasets, we believe it would still perform poorly. The reason is that given a similarity matrix, AFS attempts to obtain clustering results by applying different threshold on the matrix. Each threshold results in one clustering result and a clustering validity index is given to find the best result among them. With large amounts of samples, there are lots of entries in the similarity matrix. It is highly unlikely that the best result can be obtained this way since constructing a reliable clustering validity index is not a trivial task.

Fig.4-10 shows that, in terms of Clustering Error (CE), AFSSC outperforms STSC and SC for all cases. In the more challenging cases {3,5,8} and {0,2,4,6,7}, AFSSC

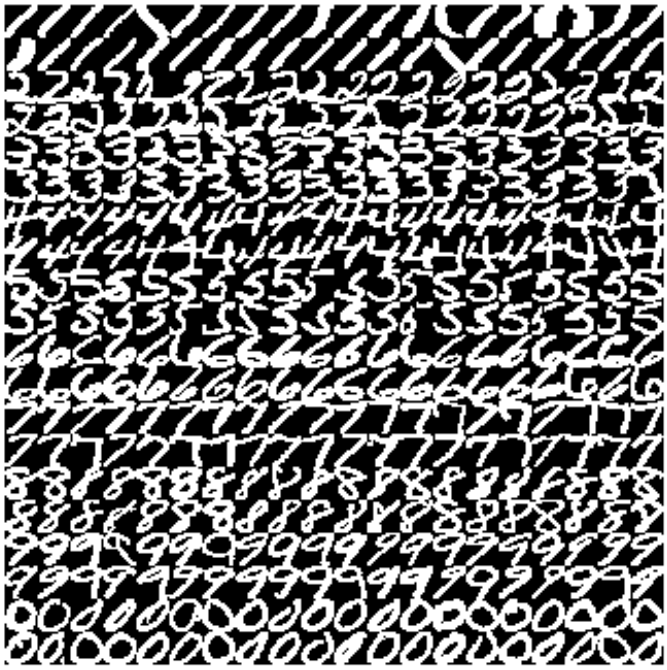

Fig. 4-9: Samples of USPS dataset.

Table 4.6: Description of USPS datasets.

Data set	instances	dimensions	classes
{0,8}	2261	256	2
{4,9}	1673	256	2
{0,5,8}	2977	256	3
{3,5,8}	2248	256	3
{1,2,3,4}	3874	256	4
{0,2,4,6,7}	4960	256	5

performs more than 10% better compared to SC. STSC performs poorly in most cases. Our method performs 20% better compared to STSC.

The results on the USPS datasets in terms of NMI are presented in Fig.4-11. It can be seen that our method outperforms STSC on all datasets. Compared to SC, AFSSC also shows its superiority, especially for the challenging case {3,5,8} and multi-classes cases {1,2,3,4},{0,2,4,6,7}.

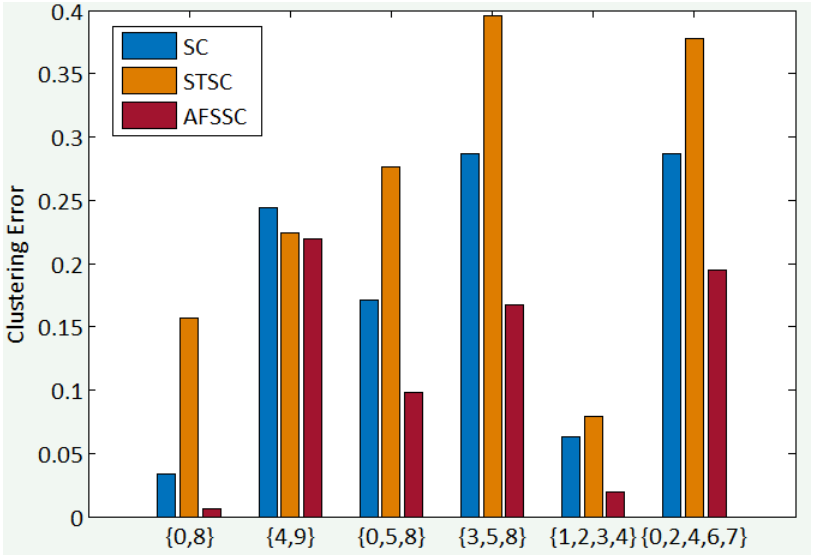

Fig. 4-10: Performance comparison based on CE for the USPS datasets.

4.2.5 Experiments on Face Data

Lastly the spectral clustering algorithms are applied on face images from the CMU-PIE [77] and Yale [78] databases. From the CMU-PIE database which consists of 41368 images of 68 people, 10 different people are randomly chosen, each with 20 images of near frontal poses and various expressions and lighting conditions, as shown in Fig.4-12. The whole Yale database is used in our experiment which consists of 165 grayscale images of 15 individuals, each with 11 images of near frontal poses and various expression, lighting conditions, and configurations, as shown in Fig.4-13. All the face images are normalized and cropped into 32×32 resolution, and the raw pixel values are then employed as the representation of the faces. Such kind of representation is greatly affected by the large variations in illumination, facial expression and head pose. The proposed algorithm is able to remove less-informative or noisy features and capture subtle information distributed over discriminative feature subspaces.

In this experiment, we compare the proposed method to the baseline methods, SC and STSC, and we also compare to another state-of-art method, ClustRF-Strct [57]. ClustRF-Strct (hereby called RFSC) is a structure aware affinity inference model based on clustering random forest. The model takes into account hierarchical structures of the whole tree, *i.e.* a tree path from the root until leaf nodes traversed by

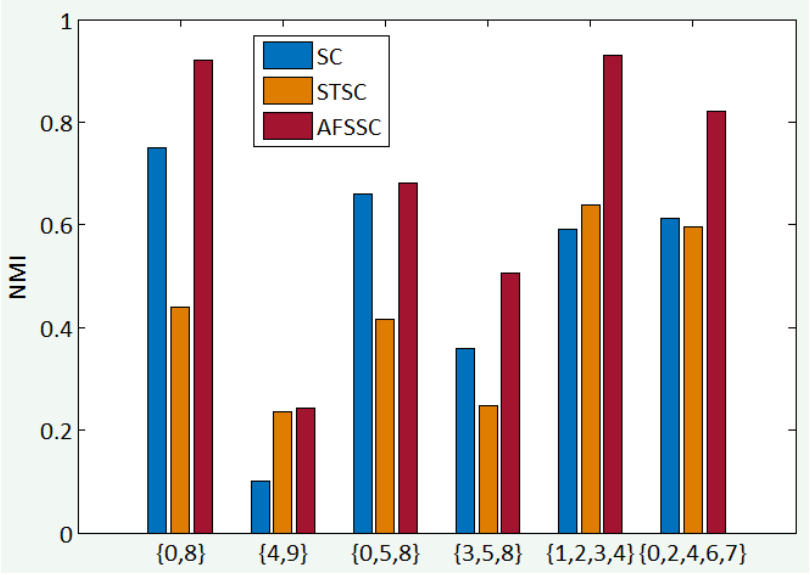

Fig. 4-11: Performance comparison based on NMI for the USPS datasets.

Fig. 4-12: CMU-PIE: each row corresponds to one person.

data samples. The assumption is that a sample pair is considered more similar if they travel more tree nodes together. To implement RFSC, the same setting in [57] is used. The number of trees in the clustering forest is set to 1000 since the larger the forest size, the more stable the results are. The random feature subset is set to $\sqrt{d}$ where d is the feature dimensionality of the input data.

We first examine the data affinity graphs constructed by SC, STSC, RFSC and AFSSC, which could qualitatively reflect how effective an affinity graph construction method is. Fig. 4-14 depicts some examples of affinity matrices generated by all these methods.

It can be observed that the proposed AFSSC produces affinity matrices with more distinct block structure and less false edges compared with the other methods. This suggests the superiority of the proposed method in learning the underlying semantic structures in data, potentially leading to more compact and separable clusters.

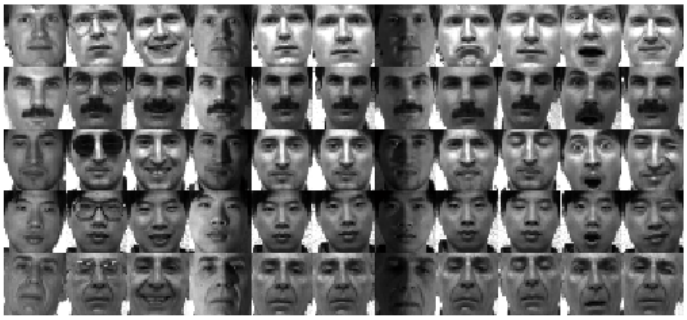

Fig. 4-13: Yale: each row corresponds to one person.

Specifically, SC produces less false edges but the block structure is not clear, which means the inter-similarities (low) are well estimated while the intra-similarities (high) are not produced properly. In contrast, STSC produces clear block structure but also a lot of noisy edges, that means the inter and intra similarities are both high, which is not correct. RFSC produces a similar graph with STSC, clear block but a lot of false edges. In Fig. 4-14, it can be clearly seen that AFSSC produces much less false edges in the inter-class graph, while the intra-class graph is comparable to the one from RFSC.

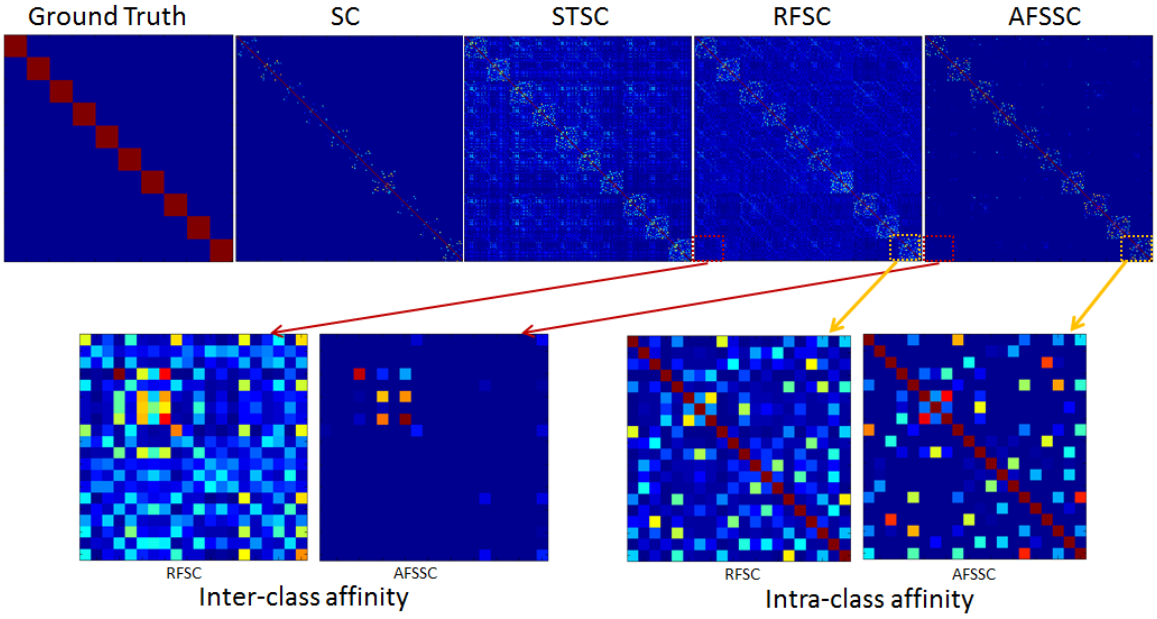

Fig. 4-14: Qualitative comparison of the affinity graphs of CMU-PIE generated by different methods. The second row shows the closed-up view of the comparison between RSFC and AFSSC. Better viewed by color and Zoom-In.

We now evaluate the clustering performance using the affinity graphs constructed above. Note that, instead of constructing the fully connected affinity graph as in the previous experiments, we utilize the similarity matrix to construct knn affinity graph, *e.g.*, given similarity matrix W, we set $W_{ij} = W_{ij}$ if $j \in knn(i)$, otherwise $W_{ij} = 0$. By doing so, we show that the proposed method can construct proper affinity graph as well as select robust neighborhood. Such a local constraint is also proven to be more effective than the fully connected affinity graph, especially when there exists a large quantity of noisy edges [52, 57].

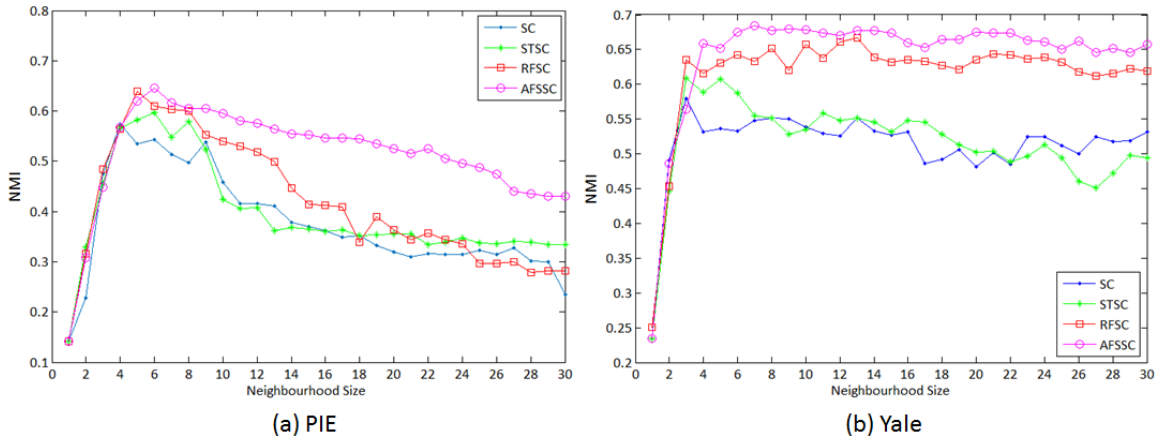

Fig. 4-15: NMI curve: comparison of the proposed method against existing methods on the spectral clustering performance given different scales of neighborhood k.

It is observed from Fig.4-15 that the proposed AFSSC outperforms the existing spectral clustering methods for different settings of neighborhood size k. Since the CMU-PIE and Yale data sets have 20 and 11 objects per class respectively, it comes as no surprise that all methods achieve the best results with a relatively small k ($4 \leq k \leq 10$). As k increases, so do the chances of adding in noisy edges, especially for face images, which are intrinsically ambiguous due to the large variations in illumination and facial expression, as demonstrated in Fig.4-12 and Fig.4-13. The proposed AFSSC produces not only the best but also the most robust result. It is observed on the CMU-PIE data set that after the best results are achieved, the performances of SC, STSC and RFSC reduce dramatically as k increases while the proposed AFSSC still performs reasonably well. The robustness is quite useful in totally unsupervised scenarios when there is no pre-knowledge about how many objects exist in each class, or in some cases

the numbers of objects are different for each class. It is desirable for the clustering algorithm to be robust given a reasonable range of k.

The Euclidean-distance-based models, SC and STSC, produce poor results in all cases. They are more likely to be affected by the large quantity of potentially noisy edges in the given kNN when dealing with unreliable input data. Their extracted features are therefore unreliable. Employing such features to construct affinity matrices with the Euclidean distance, it is understandable that SC and STSC perform poorly. RFSC attempts to constructs affinity graph using a non-linear adaptive tree hierarchical model as the neighborhood information embedded in the tree structure automatically. This approach is proven problematic as well, particularly when a large quantity of noisy features exist. The main drawback of the tree-structure similarity is that the first several nodes have too much effect on the whole process. A pair of samples are considered to be not similar at all if they are split at the root node, no matter what happens in the other parts of the tree. Since the tree nodes are split by a single feature selected from a random feature subspace, there is a high possibility that a noisy feature is selected at the very beginning, leading to inappropriate selection of neighborhoods. Unlike the other methods, the proposed AFSSC could mitigate the noisy features due to its capability to capture and aggregate subtle data proximity distributed over discriminative feature subspaces, thereby leading to more reliable and robust neighborhood construction.

4.2.6 Analysis of Feature Selection

One of the crucial advantages of the proposed AFSSC algorithm, is the capability of selecting discriminative features. As it is shown in the experiments, AFSSC performs much better than the other methods for digit images and face images, in which a large quantity of redundant and/or noisy features exist. In this section, several experiments are conducted to show the effectiveness of the proposed AFSSC for unsupervised feature selection. The following four unsupervised feature selection algorithms are compared against the case where all features are indiscriminately used:

- **Max-Variance**, which selects features of maximum variances are selected in order to obtain the best expressive power;
- **Laplacian Score** [79], which selects features that are consistent with the Gaussian Laplacian matrix to best preserve the local manifold structure;
- **MCFS** [80], which selects features based on spectral analysis and ℓ_1-norm regularization to best preserve the multi-cluster structure of the data;
- **The proposed AFSSC**, which selects feature subset distinguishingly for every single sample to best preserve discrimination.

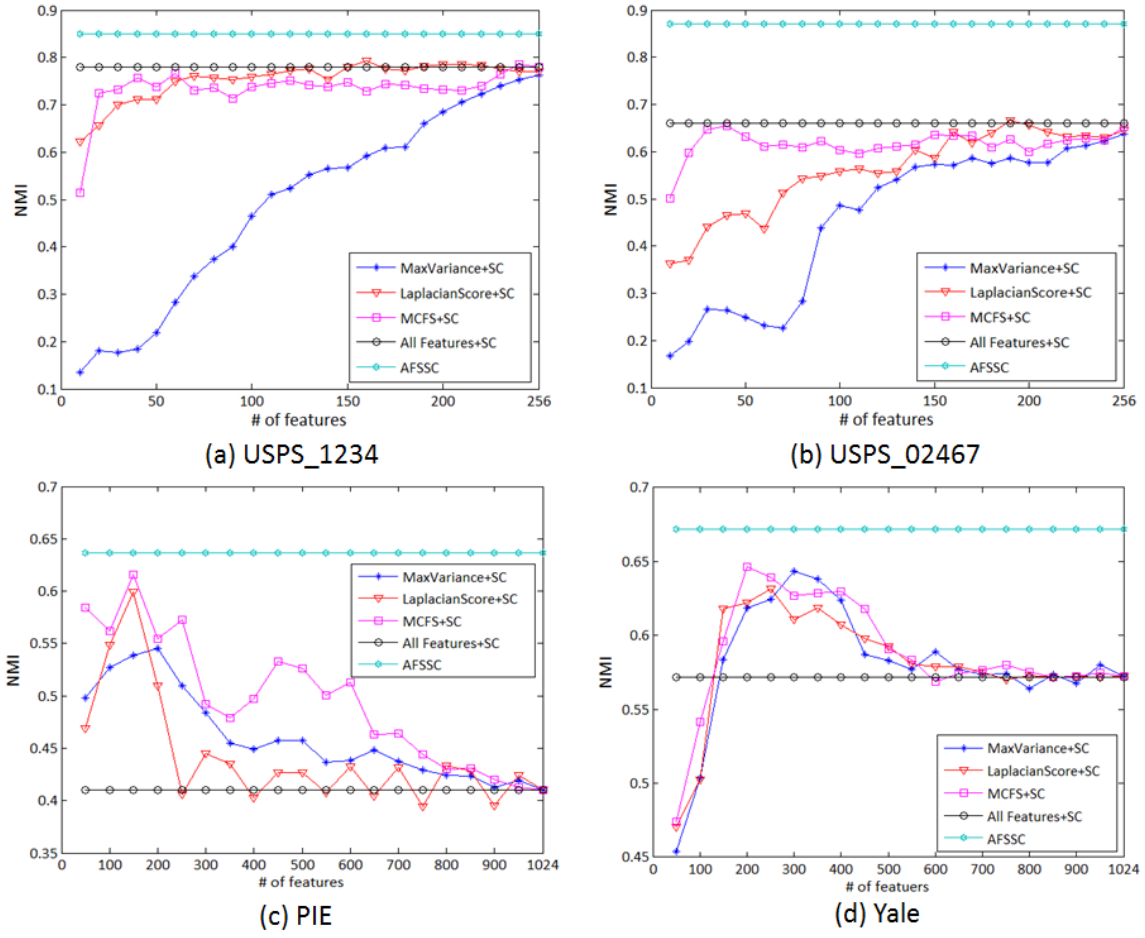

Fig. 4-16: Comparison of unsupervised feature selection methods on the spectral clustering performance given different scales of dimension.

Note that the proposed AFSSC integrates feature selection and data representation into a joint framework. By selecting a distinguishable feature subset for every

single sample, the discriminative information distributed among samples is best preserved. Since we select optimum feature subset separately for each single sample, the selected features are different between samples. As a result, every individual feature could be used for some samples, which explains why the performance of the proposed method is consistent in Fig.4-16.

From Fig.4-16, it can be seen that the proposed AFSSC method outperforms all other unsupervised feature selection methods on all datasets. For USPS handwritten digits, it is quite interesting to note that the other feature selection methods cannot improve the spectral clustering performance based on all features, while AFSSC improves around 10% on USPS_1234 and 20% on USPS_02467. MCFS and LapalacianScore performs reasonable well with limited dimensions, while MaxVariance does not work properly in this case.

For face image clustering, it comes as no surprise that spectral clustering utilizing all features performs the worst due to the large quantities of noisy features. Still, the proposed AFSSC outperforms other feature selection methods even for their best cases. MaxVariance works quite well in this case which is consistent with the effectiveness of PCA on face images. Note that PCA shares the same principle of maximizing variance, but it involves feature transformation and obtains a set of transformed features rather than a subset of the original features.

Based on their performances, these four unsupervised feature selection methods can be ranked in the order of AFSSC, MCFS, LaplacianScore, and MaxVariance. Both AFSSC and MCFS select features by considering feature subsets, while LaplacianScore and MaxVariance select features by features individually. It is proven by our experiments that unsupervised feature selection method considering features individually cannot produce an optimal feature subset as they neglect the possible correlation between different features, which is also claimed by the authors of MCFS [80]. The advantage of our method compared to MCFS is that instead of selecting the same feature subset for all samples, we propose to combine feature selection and data representation into a joint framework, and select different feature subset optimally for each single sample to best preserve their differences. Our assumption is that

the optimal feature subsets are different for different samples. By selecting the most discriminative feature subset for each sample, it is more likely that the samples from the same clusters will produce the same feature subset as their representation while those samples from different clusters will be distributed in different feature subspace, yielding improved performance of spectral clustering.

4.2.7 Parameter Analysis

It is well-known that the kernel size σ plays a significant role in spectral clustering. Unfortunately, SC is quite sensitive to this parameter and it is nontrivial to set it appropriately. In our method, the kernel size is adapted to the local structure modeled by M-nearest neighbors. As showed in Eq.3.13, both the global (σ) and the local structure (M) are taken into account in the proposed approach. The following experiments demonstrate that the proposed AFSSC is more superior in terms of parameter robustness.

The wdbc from UCI machine learning repository, digit set {1,2,3,4} of USPS, and CMU-PIE, Yale databases of face images are used in this experiment. First of all, the performance of AFSSC is tested with σ changing from {0.1,0.2,...,1} and M from {5,10,...,50}. As shown in Fig.4-17, the performance of the proposed AFSSC is stable with different values of σ and M on all data sets. AFSSC achieves consistently good performance with σ varying from 0.1 to 1. With respect to M, better performances are observed with relatively small value on USPS-{1234} and CMU-PIE.

Next the performances of SC and AFSSC with reference to σ values are tested. Since there is no M in SC, we fix $M = 5$ in AFSSC on all these data sets for a fair comparison. The results are shown in Fig.4-18. It is clear seen that the proposed AFSSC is far superior to SC in terms of parameter robustness, as well as clustering performance. For example, for the wdbc data set, AFSSC achieves very good results with σ varying from 0.2 to 1, while SC only performs well when σ is between 0.7 and 1; for the Yale data set, AFSSC achieves much better performances for all σ. Even though there is a additional parameter M, AFSSC can still outperform SC with a fixed M value.

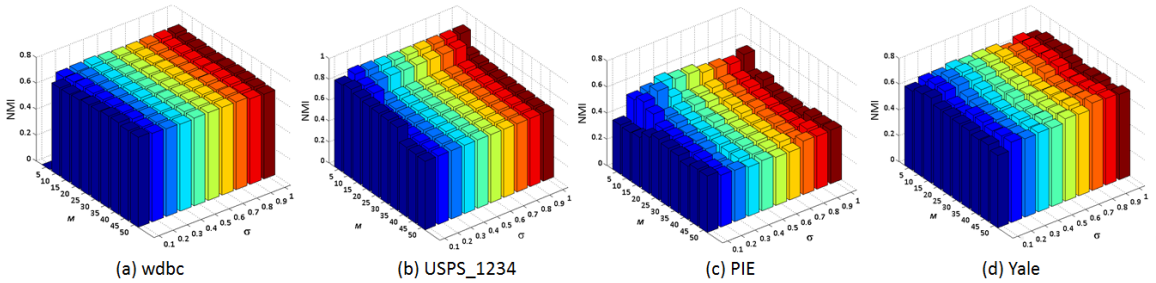

Fig. 4-17: Clustering performance of the proposed AFSSC with different M and σ.

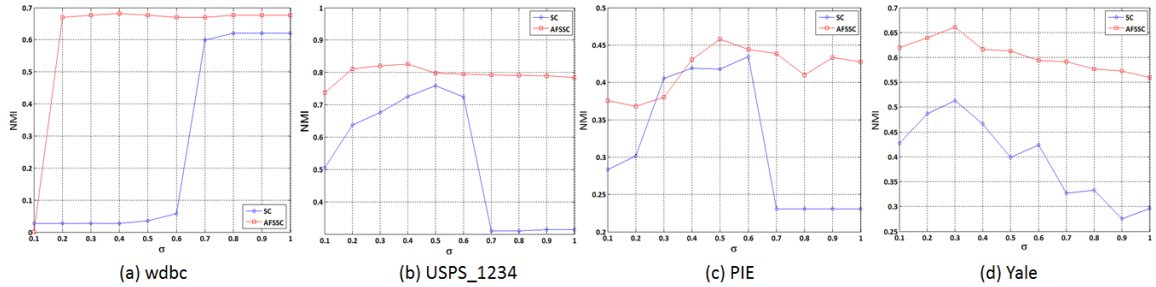

Fig. 4-18: Comparison of clustering performance of SC and AFSSC with different σ. We fix $M = 5$ for AFSSC on all data sets.

4.3 Comparison of AFS clustering and AFSSC

We proposed two clustering methods, AFS clustering and AFSSC, for different purposes, semantic learning and manifold learning, respectively. AFS clustering and AFSSC share the similar clustering framework. They both focus on deducing proper pairwise similarity measure and clustering based on the measured similarity. However, since the motivations are different, AFS clustering aims at learning semantic expressions which cannot be precisely defined, by grouping similar objects into one semantic cluster. Considering the native fuzziness of natural semantics, it is hard and impractical to represent semantic concept by a stationary and crisp cluster. Therefore, AFS clustering adopts transitive closure to update the measured similarity matrix so that the convergent similarity matrix offers us a multi-scale hierarchical clustering result by applying different threshold on the matrix. Such a multi-scale representation of semantic concepts provides a flexible way to adapt the natural fuzziness.

AFSSC is fundamentally based on the AFS clustering method, especially on the distance/similarity part. Nevertheless, AFSSC is proposed to reveal the underlying

manifold structure. Instead of a multi-scale hierarchical clustering, the goal of AFSSC is an optimal hard partition of data. To this end, the similarity measure is improved by a neighborhood adaptive Gaussian kernel to better reveal pairwise relationship. The measured similarity matrix is mapped onto a weighted graph so that the clustering problem can be solved by an optimized graph cut method. The experimental results show that, considering just hard partition of the data, AFSSC is more efficient and effective than the AFS clustering.

Chapter 5

Conclusions

Advancements in sensing and storage technology and dramatic growth in web applications have created many high-volume, high-dimensional data, especially for images. Most of images are stored digitally in electronic media, thus providing huge potential for the development of automatic image analysis, classification, and retrieval techniques. Many of these images, however, are unstructured, adding to the difficulty in analyzing them. Organizing images into sensible groups then arises naturally for better managing, understanding, and processing the image data. It is, therefore, not surprising to see the continued popularity of image clustering.

Conventional image clustering aims at gathering all images into several clusters such that the images in a single cluster provide essentially the same information. The question about what the information is, however, is not cared too much. Due to the growing interest in developing effective methods for content-based image retrieval (CBIR), which is closely related to image clustering, a more desirable way to group images from user's perspective is semantic clustering. In such settings, if we can cast image data into semantic groups, the user can query by semantic concepts, such as those used in the natural language, rather than an essential image. The challenge here is that it is quite difficult to map the low-level visual features to high-level semantics. Such semantic gap is the crucial obstacle for semantic image analysis.

Another problem for image clustering lies in the high-dimensional feature space of image data. Such representation is not effective especially when there are a lot of

redundant and noisy features. Manifold learning offers a solution for the problem. Using the underlying manifold structure has proven to significantly improve performance in many image-related tasks. With the assumption that images lie in a much lower manifold space, many manifold learning algorithms make use of dimension reduction methods to reduce the dimensionality of the feature space. Such dimension reduction techniques map the features onto a lower dimensional subspace in hope that the Euclidean distance in this new lower subspace can better capture geodesic distance than in the original higher feature space.

Spectral clustering is such a method that makes use of Laplacian EigenMaps for dimension reduction and K-means for the final clustering. The performance of spectral clustering heavily relies on the goodness of the data affinity graph as it defines an approximation to the pairwise distances between data samples. In most contemporary techniques, the data affinity graph, *e.g* a kNN graph, is constructed from a pairwise similarity matrix measure between samples. The notion of data similarity is often intimately tied to a specific metric function, typically the ℓ_2-norm (or the Euclidean metric) which is measured considering the whole feature space. Trusting all available feature blindly is susceptible to unreliable and noisy features, particularly so for image data where signals can be intrinsically inaccurate and unstable owing to uncontrollable sources of variation, changes in illumination, context, occlusion and background clutters. Moreover, confining the notion of similarity to the ℓ_2-norm metric implicitly imposes unrealistic assumption on complex data structures that do not necessarily possess the Euclidean behavior.

5.1 AFS clustering

In this thesis, we propose an Axiomatic Fuzzy Set (AFS) based clustering method for semantic learning problem. To this end, we focus on the semantic facial components clustering. The approach starts with an automatic landmark detection and all the facial components are represented by a bunch of landmarks. Low-level features are then extracted based on the geometry information derived from the landmarks.

The challenge is mapping the low-level features to the high-level semantic representations. Most of the previous approaches either manually assign semantic concepts to the corresponding features, or manually label some training data and learn the connection by machine learning techniques. Such manual settings bring in the subjective of human perception, thus leading to inappropriate analysis subsequently. The proposed approach automatically learns the fuzzy membership functions and represents the data by fuzzy sets. Pairwise similarity is then defined between the learned fuzzy sets. Given the similarity matrix, the proposed method iteratively updates the matrix to its transitive closure form so that the later matrix can be used for data partition. Since the data have already been represented by fuzzy sets which possess semantics naturally, after we group data into different clusters, each cluster can automatically obtain semantic representation based on the data inside it. We conduct experiments for semantic facial components clustering on both the Multi-PIE and AR datasets. The evaluation for semantic learning shows that the semantic gap is bridged automatically without any manually setting, and objectively by data-driven membership function construction. On the other hand, we compared the clustering performance and the results have shown significant improvement compared to the traditional K-means and FCM clustering methods.

5.2 AFS based spectral clustering

Next, we further extend AFS clustering to AFSSC for another crucial problem, manifold learning, in image clustering. To this end, AFS is first formulated as a generalized distance measurement with two innovations: (1) Rather than using the entire feature space to calculate the distance, the proposed model is designed to avoid indistinctive features by removing those features with low membership degree, yielding affinity graphs that can better express the underlying structure of data; (2) Instead of using traditional ℓ_2-norm distance, the AFS based fuzzy membership function is adopted as distance. Since the context information is proven to be significantly useful for pairwise similarity, a novel neighborhood adaptive Gaussian kernel is then proposed

to map the pairwise distance to similarity as well as enforce the locality. Standard spectral clustering method is finally used to obtain the result. Extensive experiments have been conducted on various of data, such as UCI real-world data, USPS handwritten digits, face data, *etc.* The proposed method, AFSSC, is compared to K-means, FCM, AFS clustering, and other state-of-the-art methods. The qualitative comparison of the affinity graphs shows that AFSSC can better reveal the pairwise similarity by increasing the intra-class similarity as well as decreasing the inter-class similarity. The improvement of the clustering performance based on clustering error and normalized mutual information also prove the superiority of AFSSC compared to other state-of-the-art methods.

5.3 Future work

Organizing data into sensible groupings arises naturally in many scientific fields, resulting in the continued popularity of data clustering. While numerous clustering algorithms have been published and new ones continue to appear, there is no single clustering algorithm that has been shown to dominate other algorithms across all application domains. With the emergence of new applications, it has become increasingly clear that the task of seeking the best clustering principle might indeed be futile. A clustering method that solves one problem may not be able to solve another. Clustering is in the eye of the beholder so that data clustering must involve the user or application needs. Thus, the future work on clustering should focus on achieving a tighter integration between clustering algorithms and the application needs.

On the other hand, as mentioned earlier, an era of big data has arrived so that the challenging task of large-scale clustering is another promising direction. Considering that clustering is inherently difficult and supervised classification is impracticable for large-scale problem, it makes more sense to develop semi-supervised clustering techniques, making use of side information.

Bibliography

[1] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[2] J. W. Tukey, "Exploratory data analysis," *Addison-Wesley*, 1977.

[3] B. G. Tabachnick and L. S. Fidell, *Using Multivariate Statistics (5th Edition)*. Needham Heights, MA, USA: Allyn & Bacon, Inc., 2006.

[4] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Advances in neural information processing systems*, vol. 16, no. 16, pp. 321–328, 2004.

[5] M. Webster, "Cluster analysis," 2008. [Online]. Available: http://http://www.merriam-webster.com/

[6] H. Frigui and R. Krishnapuram, "A robust competitive clustering algorithm with applications in computer vision," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 5, pp. 450–465, 1999.

[7] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888–905, 2000.

[8] M. Sahami, "Using machine learning to improve information access," Ph.D. dissertation, stanford university, 1998.

[9] S. K. Bhatia and J. S. Deogun, "Conceptual clustering in information retrieval," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 28, no. 3, pp. 427–436, 1998.

[10] G. Punj and D. W. Stewart, "Cluster analysis in marketing research: review and suggestions for application," *Journal of marketing research*, pp. 134–148, 1983.

[11] J. Hu, B. K. Ray, and M. Singh, "Statistical methods for automated generation of service engagement staffing plans," *IBM Journal of Research and Development*, vol. 51, no. 3.4, pp. 281–293, 2007.

[12] P. Baldi and G. W. Hatfield, *DNA microarrays and gene expression: from experiments to data analysis and modeling*. Cambridge university press, 2002.

[13] H. Steinhaus, "Sur la division des corp materiels en parties," *Bull. Acad. Polon. Sci*, vol. 1, pp. 801–804, 1956.

[14] M. Meilă, "The uniqueness of a good optimum for k-means," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 625–632.

[15] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[16] R. Xu, D. Wunsch *et al.*, "Survey of clustering algorithms," *Neural Networks, IEEE Transactions on*, vol. 16, no. 3, pp. 645–678, 2005.

[17] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013.

[18] E. G. Mansoori, "Frbc: a fuzzy rule-based clustering algorithm," *Fuzzy Systems, IEEE Transactions on*, vol. 19, no. 5, pp. 960–971, 2011.

[19] M. Pavan and M. Pelillo, "Dominant sets and pairwise clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 1, pp. 167–172, 2007.

[20] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 11, pp. 1101–1113, 1993.

[21] L. Hagen and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering," *Computer-aided design of integrated circuits and systems, IEEE Transactions on*, vol. 11, no. 9, pp. 1074–1085, 1992.

[22] S. X. Yu and J. Shi, "Multiclass spectral clustering," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 313–319.

[23] M. Meila and J. Shi, "A random walks view of spectral segmentation," 2001.

[24] A. Y. Ng, M. I. Jordan, Y. Weiss *et al.*, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[25] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering." in *NIPS*, vol. 14, 2001, pp. 585–591.

[26] P. Conilione and D. Wang, "Fuzzy approach for semantic face image retrieval," *The Computer Journal*, p. bxs041, 2012.

[27] N. Rasiwasia, P. J. Moreno, and N. Vasconcelos, "Bridging the gap: Query by semantic example," *Multimedia, IEEE Transactions on*, vol. 9, no. 5, pp. 923–938, 2007.

[28] J. P. Eakins and M. E. Graham, "Content-based image retrieval, a report to the jisc technology applications programme," 1999.

[29] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 12, pp. 1349–1380, 2000.

[30] Y. Chen, J. Z. Wang, and R. Krovetz, "An unsupervised learning approach to content-based image retrieval," in *Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on*, vol. 1. IEEE, 2003, pp. 197–200.

[31] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognition*, vol. 40, no. 1, pp. 262–282, 2007.

[32] I. K. Sethi, I. L. Coman, and D. Stan, "Mining association rules between low-level image features and high-level concepts," in *Aerospace/Defense Sensing, Simulation, and Controls*. International Society for Optics and Photonics, 2001, pp. 279–290.

[33] C. Town and D. Sinclair, "Content based image retrieval using semantic visual categories," *TR2000-14, AT&T Labs Cambridge*, 2000.

[34] A. Vailaya, M. A. Figueiredo, A. K. Jain, and H.-J. Zhang, "Image classification for content-based indexing," *Image Processing, IEEE Transactions on*, vol. 10, no. 1, pp. 117–130, 2001.

[35] J. Luo and A. Savakis, "Indoor vs outdoor classification of consumer photographs using low-level and semantic features," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 2. IEEE, 2001, pp. 745–748.

[36] R. Shi, H. Feng, T.-S. Chua, and C.-H. Lee, "An adaptive image content representation and segmentation approach to automatic image annotation," in *Image and Video Retrieval*. Springer, 2004, pp. 545–554.

[37] D. Stan and I. K. Sethi, "Mapping low-level image features to semantic concepts," in *Proceedings of the SPIE*, 2001, pp. 172–179.

[38] X. Zheng, D. Cai, X. He, W.-Y. Ma, and X. Lin, "Locality preserving clustering for image database," in *Proceedings of the 12th annual ACM international conference on Multimedia*. ACM, 2004, pp. 885–891.

[39] K. Sridharan, S. Nayak, S. Chikkerur, and V. Govindaraju, "A probabilistic approach to semantic face retrieval system," in *Audio-and video-based biometric person authentication*. Springer, 2005, pp. 977–986.

[40] H. Ito and H. Koshimizu, "Face image retrieval and annotation based on two latent semantic spaces in fiars," in *Multimedia, 2006. ISM'06. Eighth IEEE International Symposium on*. IEEE, 2006, pp. 831–836.

[41] X. Liu, "The fuzzy theory based on AFS algebras and AFS structure," *Journal of Mathematical Analysis and Applications*, vol. 217, no. 2, pp. 459–478, 1998.

[42] X. Liu, W. Pedrycz, and Q. Zhang, "Axiomatics fuzzy sets logic," in *Fuzzy Systems, 2003. FUZZ'03. The 12th IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 55–60.

[43] X. Liu, W. Wang, and T. Chai, "The fuzzy clustering analysis based on AFS theory," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 35, no. 5, pp. 1013–1027, 2005.

[44] X. Liu, W. Pedrycz, T. Chai, and M. Song, "The development of fuzzy rough sets with the use of structures and algebras of axiomatic fuzzy sets," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 3, pp. 443–462, 2009.

[45] X. Liu and W. Pedrycz, *Axiomatic fuzzy set theory and its applications*. Springer, 2009.

[46] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[47] F. Jing, M. Li, L. Zhang, H.-J. Zhang, and B. Zhang, "Learning in region-based image retrieval," in *Image and Video Retrieval*. Springer, 2003, pp. 206–215.

[48] A. Liang, W. Liu, L. Li, M. R. Farid, and V. Le, "Accurate facial landmarks detection for frontal faces with extended tree-structured models," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 538–543.

[49] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2879–2886.

[50] X. Liu and Y. Ren, "Novel artificial intelligent techniques via AFS theory: Feature selection, concept categorization and characteristic description," *Applied Soft Computing*, vol. 10, no. 3, pp. 793–805, 2010.

[51] O. Tuzel, F. Porikli, and P. Meer, "Human detection via classification on riemannian manifolds," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[52] V. Premachandran and R. Kakarala, "Consensus of k-nns for robust neighborhood selection on graph-based manifolds," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 1594–1601.

[53] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[54] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[55] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[56] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[57] X. Zhu, C. C. Loy, and S. Gong, "Constructing robust affinity graphs for spectral clustering," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1450–1457.

[58] S. Gong, C. C. Loy, and T. Xiang, "Security and surveillance," in *Visual Analysis of Humans*. Springer, 2011, pp. 455–472.

[59] D. Lin, "An information-theoretic definition of similarity." in *ICML*, vol. 98, 1998, pp. 296–304.

[60] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in neural information processing systems*, 2004, pp. 1601–1608.

[61] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nystrom method," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 2, pp. 214–225, 2004.

[62] T. Xiang and S. Gong, "Spectral clustering with eigenvector selection," *Pattern Recognition*, vol. 41, no. 3, pp. 1012–1029, 2008.

[63] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, "Affinity aggregation for spectral clustering," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 773–780.

[64] J. Wang, S.-F. Chang, X. Zhou, and S. T. Wong, "Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels," in *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*. IEEE, 2008, pp. 1–8.

[65] H. Chang and D.-Y. Yeung, "Robust path-based spectral clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 191–203, 2008.

[66] U. Ozertem, D. Erdogmus, and R. Jenssen, "Mean shift spectral clustering," *Pattern Recognition*, vol. 41, no. 6, pp. 1924–1938, 2008.

[67] K. Taşdemir, B. Yalçin, and I. Yildirim, "Approximate spectral clustering with utilized similarity information using geodesic based hybrid distance measures," *Pattern Recognition*, vol. 48, no. 4, pp. 1461–1473, 2015.

[68] X. Yang, X. Bai, L. J. Latecki, and Z. Tu, "Improving shape retrieval by learning graph transduction," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 788–801.

[69] X. Liu, X. Wang, and W. Pedrycz, "Fuzzy clustering with semantic interpretation," *Applied Soft Computing*, vol. 26, pp. 21–30, 2015.

[70] Y. Ren, Q. Li, W. Liu, and L. Li, "Semantic facial descriptor extraction via axiomatic fuzzy set," *Neurocomputing*, 2015.

[71] M. Wu and B. Schölkopf, "A local learning approach for clustering," in *Advances in neural information processing systems*, 2006, pp. 1529–1536.

[72] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.

[73] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: is a correction for chance necessary?" in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 1073–1080.

[74] J. Wu, H. Xiong, and J. Chen, "Adapting the right measures for k-means clustering," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 877–886.

[75] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *The Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2003.

[76] J. J. Hull, "A database for handwritten text recognition research," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 5, pp. 550–554, 1994.

[77] S. Baker and M. Bsat, "The cmu pose, illumination, and expression database," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 12, p. 1615, 2003.

[78] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 6, pp. 643–660, 2001.

[79] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Advances in neural information processing systems*, 2005, pp. 507–514.

[80] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2010, pp. 333–342.

Every reasonable effort has been made to acknowledgement the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.