Spectral Descriptors for Data Clustering and Classification

Ramandeep Kaur Grewal

A Thesis

in

The Concordia Institute

for

Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Information Systems Security) at

Concordia University

Montréal, QC, Canada

January 2016

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Ramandeep Kaur Grewal

Entitled: Spectral Descriptors for Data Clustering and Classification

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Information Systems Security)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Y. Zeng Chair

Dr. C. Alecsandru (BCEE) Examiner

Dr. J. Y. Yu (CIISE) Examiner

Dr. A. Ben Hamza Supervisor

Approved by Dr. J. Bentahar (G.P.D.)
Chair of Department or Graduate Program Director

Dr. Amir Asif
Dean of Faculty

Date ____________________

Abstract

Spectral Descriptors for Data Clustering and Classification

Ramandeep Kaur Grewal

Spectral descriptors have received much attention in recent years due in large part to their versatility as well as their ability to capture either local or global geometric information of data. While the overwhelming majority of work on spectral descriptors has concentrated primarily on image/shape retrieval and object recognition, the goal of this work is to introduce efficient algorithms for data classification and clustering in the spectral graph-theoretic setting. In addition to exploiting the dependence among the features of spectral descriptors, we perform clustering and classification on sparse codes, thereby seamlessly capturing the similarity between these features.

Unlike classification in which objects are assigned to predefined classes, clustering is different in the sense that the number (and labels) of clusters or the cluster structure are not known in advance. In this thesis, we propose a spectral graph-theoretic clustering and classification framework, called GraphFDD, which uses the Fermi density descriptor (FDD) in conjunction with graph regularized sparse coding. We also propose a unified framework for data clustering using the spectral graph wavelet descriptor, which has a strong discriminative power and good performance in capturing neighborhood information. To further enhance the effectiveness of the proposed algorithms, we not only optimize the parameters, but also determine the proper matching normalization technique.

To assess the performance of the proposed algorithms, we use several validity measures and indices, including the average clustering accuracy, normalized mutual information, confusion matrix and classification accuracy. Our experiments on different standard benchmarks not only show that the proposed approaches outperform state-of-the-art methods, but also provide attractive scalability and robustness in terms of computational efficiency.

Table of Contents

CHAPTER

CHAPTER

List of Tables

List of Figures

1 Introduction

1.1 Framework and Motivation

Recent advances in graph theory, which models pairwise relations between objects, have sparked a flurry of research activity in diverse fields with applications ranging from image processing, computer graphics and transportation networks to bioinformatics and social networks. With the extensive growth in industry, databases have become more and more complex, which has led to significant challenges in management and summary of data. The major difficulty is how to manage large and complex data sets. It is also difficult to retrieve data based on similarity and dissimilarity between features. To tackle problems related to big data, there are proven techniques in data mining such as clustering, classification and anomaly detection. Moreover, mathematical techniques are also introduced mainly dimensionality reduction methods, multidimensional scaling, sub-sampling, Principal Component Analysis (PCA) and sparse coding [1]. The main challenge in this field is to design an appropriate method that helps improve clustering and classification results. In this research, graph-theoretic frameworks are presented and described for classification and clustering using spectral descriptors. While this work focuses primarily on clustering and classification, the proposed approaches are fairly general and can be used to tackle other data mining problems.

1.1.1 Spectral Graph Theory

Graph theory is the study of graphs, which basically mathematical structures and a way of storing a set of points and lines. Graphs are used to model pairwise relations between objects and it is defined as $\mathbb{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ is the set of vertices, $\mathcal{E} = \{e_{ij}\}$ is the set of edges. The number of vertices is considered as order of graph and number of edges is known as size of graphs.Each edge $e_{ij} = [\mathbf{v}_i, \mathbf{v}_j]$ connects a pair of vertices $\{\mathbf{v}_i, \mathbf{v}_j\}$. Two distinct vertices $\mathbf{v}_i, \mathbf{v}_j \in \mathcal{V}$ are adjacent (denoted by $\mathbf{v}_i \sim \mathbf{v}_j$ or simply $i \sim j$) if they are connected by an edge, i.e. $e_{ij} \in \mathcal{E}$ [2, 3].

Spectral graph theory uses the eigenvalues and eigenvectors of matrices associated with the graph, such as the adjacency matrix, the Laplacian matrix, or the normalized Laplacian matrix, to provide information about the graph. These matrices are typically sparse, meaning that a large number of the matrix elements are zeros. We use matrices to store graphs in computers because it is an easy and efficient way [2].

Data Matrix: We may represent a data set $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ by an $m \times n$ data matrix $\mathbf{Z}$ defined as

$$\mathcal{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n) = \begin{pmatrix} z_{11} & z_{12} & \cdots & z_{1n} \\ z_{21} & z_{22} & \cdots & z_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{m1} & z_{m2} & \cdots & z_{mn} \end{pmatrix} \tag{1.1}$$

where each column $\mathbf{z}_i$ is a m-dimensional vector, called a data point, observation, or instance. In other words, $\mathbf{Z}$ is a cloud of n points in the m-dimensional space. Each entry z_{ij} of the data matrix $\mathbf{Z}$ is called a feature, variable, or attribute. Representing a data set in the form of a matrix allows computations to be performed efficiently.

Neighborhood Graph: A data set $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ with pairwise dissimilarities can be transformed into a neighborhood graph $\mathbb{G}$ by modeling the local neighborhood relationships between the data points. Each data point serves as a vertex on the neighborhood graph and connectivity between vertices is governed by the proximity of neighboring data points.

Adjacency Matrix: It is a square matrix whose elements indicates the adjacency of nodes or vertices. It contains the information regarding the representation of the graph which explains the connections between nodes and it is also known as connection matrix. The adjacency matrix $\mathbf{W} = \left(w_{ij}\right)$ of a neighborhood graph constructed from a point cloud $\mathcal{Z}$ of n data points and it is defined as

$$w_{ij} = \begin{cases} 1 & \text{if } j \in \mathcal{N}_i \\ 0 & \text{o.w.} \end{cases} \tag{1.2}$$

where $\mathcal{N}_i$ denotes the κ-neighborhood of $\mathbf{z}_i$, with κ denoting the number of neighbors [2].

Degree Matrix: It is a diagonal matrix $\mathbf{D} = (d_i)$ whose elements are given by

$$d_i = \sum_{j=1}^{n} w_{ij}, \quad i = 1, \ldots, n \tag{1.3}$$

where d_i is degree of data point z_i.

Laplacian Matrix: It is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is the degree matrix and $\mathbf{W}$ is the adjacency matrix. The Laplacian is $n \times n$ symmetric matrix of neighborhood graph. Its sum of rows and columns, is zero [2]. There are several methods for normalizing the Laplacian matrix. The details are provided in Section 3.2.1.

1.1.2 Data Mining

Data mining is the process of discovering useful information ir knowledge from data. This relatively new discipline is popular in many fields such as medical, fraud detection, science and technology [4]. Core components of data mining include:

1. Clustering
2. Classification
3. Anomaly Detection
4. Association Rule Learning
5. Visualization

1. Clustering: It is a technique used for statistical data analysis. It is also known as unsupervised learning approach which means having unlabeled data in a data-set. A collection of data objects which has similar properties is considered into same group and unrelated to other groups, is the process of clustering. It is basically based on the concept of high intra-class similarity and low

inter-class similarity. For clustering, there is no need of prior knowledge of data and its classes. This technique is very popular these days in many fields like biology, marketing, earth-quake studies, city-planning, climate and economic science. In this era, clustering is an important tool for the field of data mining which has further approaches mainly partitioning, hierarchical, density based and frequency pattern-based. There are some challenges to clustering algorithms like robustness, sensitivity, outlier detection, accuracy and scalability. So, it is still an active area in research [4–6]. The popular type of clustering is K-means clustering.

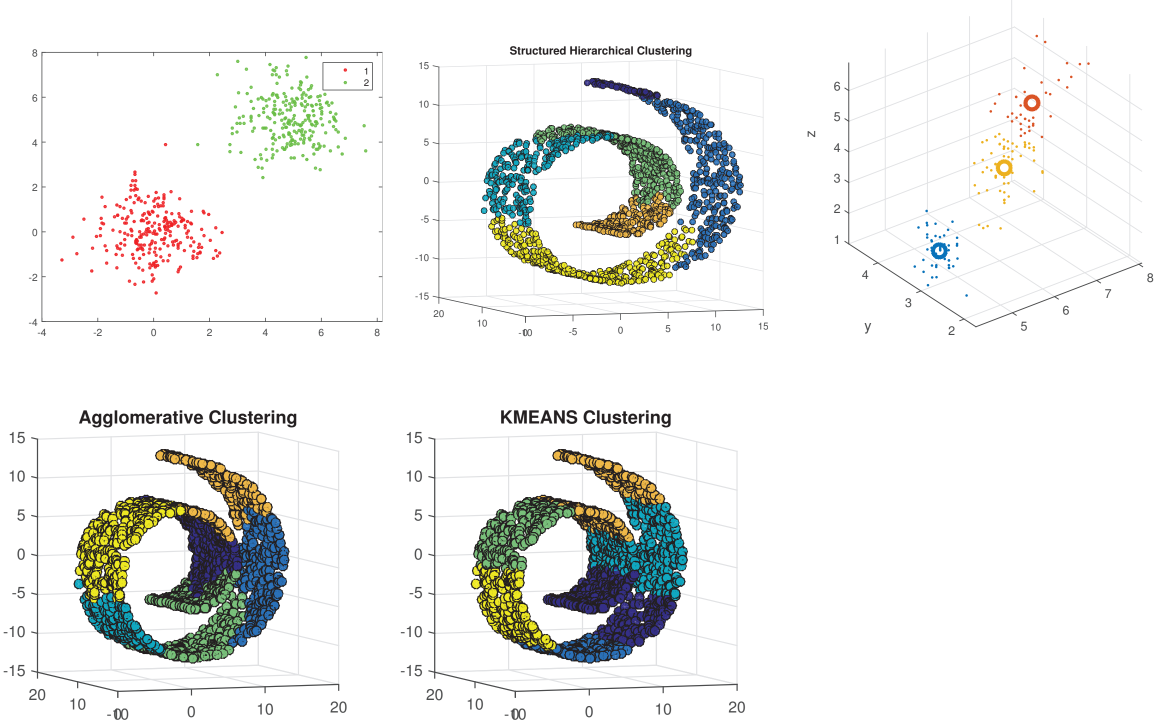


FIGURE 1.1: Examples of clustering

2. Classification: It is also known as supervised learning approach. In this method, the whole data is divided into two parts: one is for training set and another for testing set. Prior knowledge of data is required to build a classifier based on the training set of data. After learning the model with training data, then perform testing to classify the remaining data points into appropriate classes and calculate their accuracy [4].

3. Anomaly detection: It is also an important method used for the detection of anomalies or the false data which is also known as outlier detection. This method is of great significance to many

applications such as fraud detection, cancer diagnostics and virus detection [7, 8]. Outliers or anomalies are basically data points that are very different from other data points. The fundamental algorithms of clustering and anomaly detection form the basis for the vibrant field of data science. The main goal of anomaly detection algorithms is to differentiate anomalies from normal instances [5, 9].

4. Association Rule Learning: It is an interesting method for analysis of relationships between unrelated data. The idea behind this learning comes from Market Basket Analysis (MBA) which means analysis of items in the super-stores based on everyday sale. Then find the correlations and associations between the items usually brought together by the customers. For example, a customer comes to buy eggs and milk also probable to buy bread or cereal together. So, we can find the correlation with the analysis of different shopping baskets of customers. This learning is useful in promotional pricing, web data mining and also for intrusion detection. Amazon and Netflix also engage in data mining to improve the experience of shopping [6].

5. Visualization: It helps in analysis of data using visual objects. It makes data more understandable and easy to access with the help of graphs, plots, charts and tables. But for large data and data with large number of dimensions, it is very hard to visualize distances. So, there is a need of dimensionality reduction methods to reduce the number of dimensions. Multidimensional scaling (MDS) is basically used to address these issues and helps visualize data points based on their pairwise distances [10].

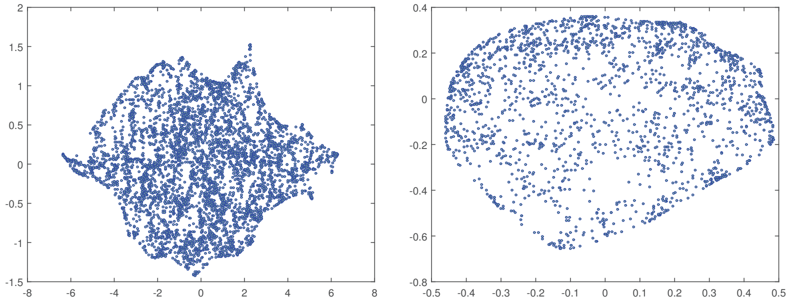


FIGURE 1.2: Spectral multidimensional scaling

In this thesis, we focus on clustering and classification methods that make use of spectral descriptors and spectral graph theory. However, to date, no comparison has been conducted in the literature, when spectral descriptors are chosen for data mining methods. Our motivation is to design efficient algorithms that would help improve the overall performance. For algorithm design, the first step is to select an appropriate spectral descriptor and the second step is to design efficient clustering and/or classification methods.

1.2 Literature Review

A great number of spectral-based approaches for clustering and classification have been proposed in recent years. But the vast majority of these algorithms have a limited performance due to several reasons: (1) There is not a proper use of spectral descriptors with the existing algorithms. (2) Investigation of proper matching normalization approaches with different methods. (3) Performance comparison in terms of parameter sensitivity. In this section, we discuss related work and recent progresses pertinent to graph theory, spectral theory, clustering, classification and dimensionality reduction methods.

1.2.1 Spectral Descriptors

In the literature, there are many research efforts that have keen interest in spectral graph theory. Many research efforts have been conducted on that uses eigenvalues and eigenvectors of matrices which are associated with the graph [11,12]. The information of graph is obtained from the spectra or eigenvalues of associated matrices. In [13], author discuss the properties of eigenfunctions. Early research work have been centered on matrices such as adjacency matrix, Laplacian matrix or normalized Lapalcian matrix based on associated graphs [14–16]. Moreover, significant efforts have been invested on random walks on graphs and eigenanalysis of Laplacian matrices [2, 17].

In recent years, the evolution of graphs is adopting new optimization techniques for supervised and unsupervised learning. On the other hand, there are many spectral descriptors have been proposed in recent researches. Many surveys have been conducted on different spectral descriptors and signatures [18–20]. Scale-Invariant Feature Transform (SIFT) was introduced as image descriptor in early years.

The recent surge of interest in the spectral analysis has resulted in a plethora of spectral shape signatures that have been successfully applied to a wide range of tasks, including object recognition and deformable shape analysis [19, 21–23], medical imaging [24] and multimedia protection [25]. Spectral descriptors are basically feature vectors which are further divided into two types: local descriptor and global descriptor. Local descriptor is applied to each point of shapes but global descriptor is used to define the whole shape or image [3]. Global point signature is referred to local descriptor, able to capture useful information of different data sets and can perform classification, segmentation and clustering operations on different data sets [22,24,26]. But this local descriptor has a problem of switching of eigenfunctions [3, 26]. After that, another signature was proposed by [27] that is wave kernel signature. The wave kernel signature is based on characterization of points in a surface with average probabilities of each particle and measure in different energy levels [26, 27]. Heat kernel signature (HKS) is also a local descriptor and helps to perform multi-scale matching between different points in a shape and based on heat diffusion in particles. HKS was proposed for segmentation and shape skeletonization [26, 28, 29]. Scale invariant heat kernel signature is logarithmically sampled using fourier transform magnitude and it is enhanced version of heat kernel signature [26,30,31]. A spectral graph wavelet framework was presented for designing of descriptors for local and global geometry of shapes. The cubic spline generating kernel is used for shape retrieval [23]. The Fermi density descriptor (FDD) has solid background of quantum theory and it is able to differentiate anomalies from normal instances in a local structure [9, 32]. Hao *et al.* [9] provides future direction to apply FDD for data clustering and classification. So, our motivation in this research is to use the FDD descriptor for data clustering and classification to achieve better performance. The details of all descriptors are provided in Chapter 2. We provide a comprehensive review of recent descriptors and compare the recent descriptors with the combination of different algorithms to achieve better understanding. An intuitive approach is to use spectral descriptors for supervised and unsupervised learning.

1.2.2 Classification and Cluster Analysis

The last decade has witnessed the growth of popularity of clustering and classification methods. Both techniques have their great impact in many applications such as pattern recognition, education, businesses and so on [6]. The key benefit to use these methods is that domain experts can be comfortable even without or less knowledge of machine learning [33]. The popular method of clustering, K-means clustering algorithm was proposed in [34]. In [35], authors had shown remarkable results with their proposed robust clustering algorithm where warping of data points is used to find the spatial information and then K-means is used for clustering. In this algorithm, noise data points are divided separately into another cluster. It works well with small data sets. In [36], a generalized maximum marginal clustering method was developed for large-scale data sets, inexpensive and able to choose the appropriate kernel functions. The idea is to divide data points into K clusters to minimize the sum of squares within-cluster. In clustering, the main desirable feature is robustness. A spectral clustering method was proposed in [37] to provide more stable results.

A survey of clustering approaches is provided in [38–40]. A recent engagement of Amazon and Netflix in data mining proves the surge of interest in all kind of businesses. Netflix problem was computation of high dimensional data that is really hard. Singular Value Decomposition (SVD) was introduced for dimensionality reduction. Moreover, the author showed that data mining and quantum mechanics have relation to each other [6].

The main problems with clustering algorithms are noise and parameter selection for scaling. A new algorithm, aggregated heat kernel was defined by the integration of time scaling parameters of heat kernel in [33]. Author has shown the combination of diffusion maps with the spectral clustering and used Laplace-Beltrami normalization approach instead of graph Laplacian normalization for manifold recovery. Previously, in [41] diffusion distance is applied to provide more robustness.

Quantum mechanics has been applied to the field of anomaly detection in data sets [32]. Authors proposed descriptor for detection of anomalies with the knowledge of local density of data points and also used five different normalization approaches. Moreover, they analyzed performance with distribution functions like Maxwell-Boltzmann Distribution, Bose-Einstein Distribution, Heat Diffusion and Gaussian Distribution and found best results with FDD [32]. In [9], same authors published their work with two descriptors, local anomaly descriptor and fermi density descriptor. They formulated FDD to achieve more stability of local density measurement with the addition of a smoothing parameter.

Classification is also a vibrant research area essential to many applications such as text categorization, intrusion detection, gene expression data analysis [42]. The classification method, bag-of-features model which is based on computation of histogram representation from visual words, used for object recognition [43]. After that many extensions of bag-of-features model have also been proposed [44–46]. The popular one is spatial pyramid matching [47]. Authors of [48], proposed another classification algorithm using naive-bayes assumption which is simple and efficient. Then, another method random forest was proposed as multi-way classifier in [49]. The main problem with these algorithms, was difficulty in classifying high dimensional data. So, dimensionality reduction techniques have been introduced in this research area.

1.2.3 Dimensionality Reduction and Sparse Coding

Most of real-world data has high dimensionality. To manage high dimensional data properly, we need to reduce the dimensions into useful or meaningful representation of data [50]. Previously, classical scaling [51] and factor analysis [51] techniques were used for dimensionality reduction. After that, simple principal component analysis was proposed in the literature [52].

Next to classical dimensionality reduction methods, sparse coding has gained huge popularity in machine learning and image processing in the last decade. Sparse representations of images make encoding process very efficient, interpretable and reduce the computational cost. Many authors have already published their work for many applications such as visual areas, image restoration, classification of signals, neuroscience, face recognition, image denoising, image classification and clustering [53–62].

Some researchers tried sparse coding with other methods such as PCA, linear spatial pyramid matching and locality-constrained linear coding [43, 63, 64]. Another approach which is based on sparse coding, with some modifications and addition of smooth operator was proposed, graph regularized sparse coding. It is also shown that the performance with graph regularized sparse coding is improved for classification and clustering in comparison of original sparse representations. The resulted representations considers geodesics of manifolds of data and used graph Laplacian as a regularization parameter in this method [65].

Inspired by the advantages of various spectral descriptors, our motivation in this research is to apply these descriptors with the combination of graph regularized sparse coding to enhance the performance for clustering and classification tasks. Moreover, motivated by the lack of a systematic comparison of different descriptors for data mining methods, this research presents a comparative study of important descriptors. Extensive experimental results has been shown with different state-of-the-art methods. Fermi density descriptor with graph regularized sparse coding improves the performance and results are shown to be more accurate and robust. In addition, we propose a spectral graph wavelet signature for data clustering. The overall results have been shown with comparison of different state-of-the-art methods on different benchmarks in terms of various evaluation measures.

1.3 Thesis Overview and Contributions

The organization of this thesis is as follows

❏ Chapter 1 contains a brief introduction of concepts, a literature review, and provides summary of relevant topics and review of functionalities that are surveyed during research.

❏ In Chapter 2, we comprehensively review different shape descriptors and analyze different clustering and classification algorithms. Then, we propose a spectral graph-theoretic clustering and classification framework, called graph Fermi density descriptor (GraphFDD), which uses the Fermi density signature in conjunction with graph regularized sparse coding. Extensive experiments and evaluations have been conducted on different data-sets for image classification and clustering. The evaluations and results are carried out in terms of accuracy and normalized mutual information measures.

❏ In Chapter 3, we present a spectral graph wavelet signature for data clustering. Extensive experimental results demonstrate the much better performance of the proposed algorithm in comparison of existing methods. The evaluations and experiments are conducted on different text datasets to demonstrate the robustness and effectiveness of the proposed algorithm.

❏ In Chapter 4, we provide the contributions and the concluding results drawn from the research work and propose several future research directions that are related to our work.

2 Sparse Coding for Clustering and Classification

This chapter presents a comprehensive review of various spectral descriptors and analysis of different algorithms used for clustering and classification techniques. Sparse coding and graph regularized sparse coding, both algorithms have gained an increasing amount of popularity and interest of researchers in recent years. Inspired by performance of popular algorithms, we investigate classification and clustering results on different image and text benchmarks. In this chapter, we present the classification and clustering algorithms using fermi density descriptor and graph regularized sparse coding. We compare different descriptors, more specifically wave kernel signature, heat kernel signature, scale invariant heat kernel signature, global point signature with our proposed work. Moreover, we investigate the performance of different baseline methods such as simple K-means, principal component analysis, sparse coding and graph regularized sparse coding algorithms with our proposed GraphFDD algorithm. Comparison of various algorithms is evaluated using accuracy and normalized mutual information measures. Extensive experiments are carried out on different benchmarks to assess the performance of different algorithms.

2.1 Introduction

Over last decade, sparse coding has been successfully applied to a variety of machine learning and image processing applications. Sparse coding is a family of unsupervised algorithms [66] that are essentially employed for learning an overcomplete set of bases, where an image or shape can be represented by a high-dimensional but sparse feature vector. The goal of sparse coding is to represent a feature vector by a linear combination of a sparse set of basis vectors. Sparse representations of images make the encoding process very efficient, interpretable and reduce the computational cost [53–55]. Other applications of sparse coding include visual areas, image restoration, classification of signals, neuroscience, face recognition, data compression, image classification and clustering [3, 56–61].

Some researchers tried sparse coding with other methods such as dimensionality reduction, PCA and so on [63]. In recent years, various coding schemes have been proposed in the literature, and have proven to be effective in a wide range of computer vision tasks. Wang *et al.* [64] introduced locality-constrained linear coding, which enforces locality instead of sparsity. In [65,67], the graph regularized sparse coding, also known as the Laplacian sparse coding, was proposed. This coding scheme takes into account the geometric structure of the data space by using a graph Laplacian regularizer in an effort to preserve the locality of the features to be encoded. Unlike sparse coding in which each feature is encoded independently, the graph regularized sparse coding encodes similar features with similar sparse codes, thereby preserving the locality information of the features to be encoded.

On the other hand, there are many spectral descriptors have been proposed in last decade which have their own properties. While spectral signatures have received much attention in nonrigid 3D shape analysis [19, 21–23], view-based methods, on the other hand, have also been successfully applied to 3D shape recognition and retrieval. The details are provided in next section. So, our effort is to apply these descriptors for data mining tasks.

More recently, FDD descriptor was proposed in [9] for anomaly detection. The descriptor has a number of attractive properties that makes it suitable for addressing other data mining problems. It is computationally efficient, robust to noise, and possesses good discriminative capabilities.

Motivated by the advantages of graph regularized sparse coding and FDD, our inspiration in this research is to apply FDD with the combination of graph regularized sparse coding to enhance the performance of classification and clustering algorithms. Moreover, a comparison between different descriptors has been shown in our research. Extensive experimental results has been shown with different state-of-the-art methods. Our proposed GraphFDD improves the performance and results are shown to be more accurate and robust. In addition, results have been shown with comparison of different state-of-the-art methods on different image and text benchmarks in terms of accuracy and normalized mutual information.

2.1.1 Contributions

The contributions in this chapter may be summarized as follows:

(i) We present a comprehensive survey of different spectral descriptors and analyze different related clustering and classification algorithms.

(ii) We propose novel clustering and classification algorithms using graph regularized sparse coding and the Fermi density descriptor.

(iii) We systematically evaluate the proposed algorithms on several data sets and assess their performance in comparison with existing methods.

(iv) We evaluate the proposed algorithms using several metrics, including accuracy, confusion matrix and normalized mutual information.

The rest of this chapter is organized as follows. Section 2.2 briefly reviews different spectral descriptors and Section 2.3 provides some background of sparse coding. In Sections 2.4 and 2.5, we introduce our GraphFDD algorithms for data clustering and classification, and we discuss in detail their main algorithmic steps. Experimental results on various data sets and comparison with existing techniques are presented in Section 2.6.

2.2 Spectral Descriptors

Several spectral descriptors are good at capturing local features and more stable for local neighborhood density measurement, while other descriptors are usually defined on the entire shape or image and defined to capture global neighborhood density. Moreover, most point signatures can easily be aggregated to form global descriptors by integrating over the entire surface of the shape. We provide the brief overview of recently proposed spectral descriptors and their important features.

2.2.1 Wave Kernel Signature

The wave kernel signature (WKS) is well suited for characterization of points on surface of objects. It is more discriminative in nature and able to contain local and global information of each point in different shapes. WKS is more accurate, informative and robust than heat kernel signature. It helps to clearly differentiate between different scales and frequencies. WKS is based on characterization of point in a surface with average probabilities of each particle and measure in different energy levels. The wave function $\varphi(\mathbf{z}, t)$ of quantum particle for non-rigid deformation of shape is expressed as

$$\psi_E(\mathbf{z}, t) = \sum_{l=1}^{\infty} e^{i\lambda_l t} \varphi_l(\mathbf{z}) f_E(\lambda_l), \tag{2.1}$$

where time $t = 0$, E is energy of given particle and f_E is initial distribution. The approximate energy probability distribution f_E^2 with E expectation value. To measure particle at point $\mathbf{z}$, the probability is $|\psi_E(\mathbf{z}, t)|^2$. So, the average probability is defined as

$$\mathtt{P}_E(\mathbf{z}) = \lim_{T\to\infty} \frac{1}{T} \int_0^T |\psi_E(\mathbf{z}, t)|^2 = \sum_{l=1}^{\infty} \varphi_l(\mathbf{z})^2 f_E(\lambda_l)^2 \tag{2.2}$$

WKS is able to capture information from different frequencies. So, properties of different shapes are fully dependent on value of f_E^2. In WKS, time scale parameter is not considered because it is not directly interpreted in shapes. Energy parameter is used in wavelet kernel signature instead of time parameter. The n dimensional feature vector with different probabilities is represented as

$$\text{WKS}(\mathbf{z}) = \left(\mathtt{P}_{e_1}(\mathbf{z}), \mathtt{P}_{e_2}(\mathbf{z}), \cdots, \mathtt{P}_{e_n}(\mathbf{z})\right), \tag{2.3}$$

where $e_i = logE_i$ is logarithmic energy scale. WKS uses band pass filters and has more stability under non-isometric perturbations of different shapes. It can detect feature correspondence in case of noisy data [26, 27].

2.2.2 Global Point Signature

The global point signature (GPS) is based on global structure of objects. It is able to capture adequate information of different shapes and can perform classification, segmentation and clustering operations on different data sets. GPS is not used for partial matching because of its global nature. GPS on a point $\mathbf{z}$ is defined as infinite-dimensional vector and the feature map is represented as

$$\text{GPS}(\mathbf{z}) = \left(\frac{\varphi_2(\mathbf{z})}{\sqrt{\lambda_2}}, \frac{\varphi_3(\mathbf{z})}{\sqrt{\lambda_3}}, \ldots, \frac{\varphi_l(\mathbf{z})}{\sqrt{\lambda_l}}, \ldots\right) \tag{2.4}$$

GPS has many properties like it is invariant under isometric deformations of different shapes, more robust for topological changes, good power of discrimination between objects. There are some problems with GPS descriptor. The major disadvantage is switching of eigenfunctions when values of two eigenvectors are very close to each other. The other problem is lack of dealing with degenerate meshes [22, 26].

2.2.3 Heat Kernel Signature

The heat kernel signature (HKS) is a point signature that has many properties: stability under perturbation of shapes, concise, informative, arrange information about the geometry of different shapes. This signature is based on calculation of dissipation of heat that transfers from one point to another over time. This is local descriptor and helps to perform multi-scale matching between different points in a shape. The heat diffusion process is defined as

$$\Delta_{\mathbb{M}} u(\mathbf{z}, t) = -\frac{\partial u(\mathbf{z}, t)}{\partial t}, \tag{2.5}$$

where $\Delta_{\mathbb{M}}$ is the Laplace-Beltrami operator, u satisfy condition of Dirichlet boundary $u(\mathbf{z}, t) = 0$ for all $\mathbf{z} \in \partial\mathbb{M}$ and t is heat diffusion time. The heat kernel $\mathfrak{p}_t(\mathbf{z}_i, \mathbf{z}_j)$ is defined as

$$\mathfrak{p}_t(\mathbf{z}_i, \mathbf{z}_j) = \sum_{l=1}^{\infty} e^{-\lambda_l t} \varphi_l(\mathbf{z}_i) \varphi_l(\mathbf{z}_j) \tag{2.6}$$

The heat kernel $\mathfrak{p}_t(\mathbf{z}_i, \mathbf{z}_j)$ describes the transferred heat from one point $\mathbf{z}_i$ to another point $\mathbf{z}_j$ and $\mathfrak{p}_t(\mathbf{z}_i, \mathbf{z}_i)$ is the remaining amount of heat at point $\mathbf{z}_i$ after t time scale. HKS for each point $\mathbf{z}$ is defined as

$$\mathrm{HKS}(\mathbf{z}) = \left(\mathfrak{p}_{t_1}(\mathbf{z}, \mathbf{z}), \mathfrak{p}_{t_2}(\mathbf{z}, \mathbf{z}), \ldots, \mathfrak{p}_{t_n}(\mathbf{z}, \mathbf{z})\right), \tag{2.7}$$

where HKS is n-dimensional feature vector and t_1 to t_n are time scales. HKS characterizes time scale parameter and neighborhood size of $\mathbf{z}$ which gives ability of partial matching [26, 29].

2.2.4 Scale Invariant Heat Kernel Signature

The scale invariant heat kernel signature (SIHKS) is a local descriptor which is extension of heat kernel signature. It has many features like discrimination between normal and noisy data, isometric deformations and scaling. SIHKS is based on logarithmically sampled scale using fourier transform magnitude. Because of important properties in heat kernel signature, SIHKS is designed from HKS with some modifications. HKS has major disadvantage of scale sensitivity. SIHKS is designed to overcome this disadvantage using local normalization of HKS.

In SIHKS, dependency of h is removed from scale factor and new discrete function is used with logarithmic time. So, $h_\tau = h(\mathbf{z}, \alpha^\tau)$ is formed with time $t = \alpha^\tau$, where τ is a scale variable. The heat kernel of scaled shape is $\mathfrak{p}'(\tau) = a^{-2}\mathfrak{p}(\tau + 2\log_\alpha a)$ and the differential after logarithm is defined as

$$\begin{aligned} \frac{d}{d\tau}\log\mathfrak{p}'(\tau) &= \frac{d}{d\tau}(-2\log a + \log\mathfrak{p}(\tau + 2\log_\alpha a) \\ &= \frac{\frac{d}{d\tau}\mathfrak{p}(\tau + 2\log_\alpha a)}{\mathfrak{p}(\tau + 2\log_\alpha a)}. \end{aligned} \tag{2.8}$$

The other function $\tilde{\mathfrak{p}}$ is defined, where $\tilde{\mathfrak{p}'}(\tau) = \tilde{\mathfrak{p}}(\tau + 2\log_\alpha a)$ scaling function is obtained. The discrete time fourier transform is given as

$$\begin{aligned} &F\left[\tilde{\mathfrak{p}'}\right](\omega) = \tilde{H}'(\omega) = \tilde{H}(\omega)e^{-j\omega 2\log_\alpha a} \\ &|\tilde{H}'(\omega)| = |\tilde{H}(\omega)|. \end{aligned} \tag{2.9}$$

SIHKS is formed with scale invariant property which gives ability to compute descriptors on each point in different shapes. It is defined as

$$\text{SIHKS}(\mathbf{z}) = \Big(|\tilde{H}(\omega_1)|, |\tilde{H}(\omega_2)|, \ldots, |\tilde{H}(\omega_n)|\Big)\,. \tag{2.10}$$

So, SIHKS is basically another version of HKS and able to overcome the disadvantage of HKS [26, 30].

2.3 Sparse Coding

Sparse coding refers to a class of unsupervised techniques for learning sets of overcomplete bases in an effort to represent data efficiently. The objective of sparse coding is to represent an input data point (e.g., signal) as a linear combination of a small number of learned atoms or basis vectors that capture salient features or patterns in the input data. More precisely, consider a set of n data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, arranged in a $p \times n$ data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{p\times n}$, where each column vector $\mathbf{x}_i$ is a p-dimensional data point. Sparse coding allows us to represent each data point as $\mathbf{x}_i = \mathbf{V}\mathbf{u}_i$, $i = 1, \ldots, n$, where $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_k) \in \mathbb{R}^{p\times k}$ is a $p \times k$ dictionary or vocabulary matrix, and $\mathbf{u}_i$ is a sparse vector of coefficients, which is the ith column of the coefficient matrix $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_n) \in \mathbb{R}^{k\times n}$, also known as the sparse codes matrix. Sparse coding may be written as an optimization problem (also referred to as the *sparse coding* problem)

$$\min_{\mathbf{U}} \quad \sum_{i=1}^{n}\|\mathbf{x}_i - \mathbf{V}\mathbf{u}_i\|_2^2 + \lambda\sum_{i=1}^{n}\|\mathbf{u}_i\|_1. \tag{2.11}$$

We can interpret the first term of the sparse coding objective function as a reconstruction term which tries to force the algorithm to provide a good representation of $\mathbf{x}_i$ (i.e. it ensures that the sparse representations $\mathbf{V}\mathbf{u}_i$ are as close as possible to the original data points $\mathbf{x}_i$). The second term is a sparsity penalty, which forces our representation of $\mathbf{x}_i$ to be sparse. The regularization parameter λ is a positive scaling constant that determines the relative importance of these two contributions (i.e. it balances sparsity against reconstruction error).

The vocabulary matrix $\mathbf{V}$ is usually overcomplete, meaning that there are fewer rows than columns (i.e. $p < k$), and can thus capture a large number of patterns in the input data. Intuitively, an overcomplete dictionary matrix transforms a low-dimensional vector into a high-dimensional sparse vector. That is, every data point is now encoded as a sparse vector that is of higher dimensionality than the original representation. Both $\mathbf{V}$ and $\mathbf{U}$ are unknown and can found by solving the joint optimization problem

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V}} \quad & \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{V}\mathbf{u}_i\|_2^2 + \lambda \sum_{i=1}^{n} \|\mathbf{u}_i\|_1 \\ \text{s.t.} \quad & \|\mathbf{v}_r\|^2 \leq 1, \quad r = 1, \ldots, k \end{aligned} \tag{2.12}$$

which is often referred to as the *sparse modeling* problem. The constraint $\|\mathbf{v}_r\|^2 \leq 1$ is used to prevent the dictionary from having large values, which could lead to arbitrarily small values of $\|\mathbf{u}_i\|_1$. In matrix form, the sparse modeling problem may be written as

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V}} \quad & \|\mathbf{X} - \mathbf{V}\mathbf{U}\|_F^2 + \lambda \|\mathbf{U}\|_{1,1} \\ \text{s.t.} \quad & \|\mathbf{v}_r\|^2 \leq 1, \quad r = 1, \ldots, k \end{aligned} \tag{2.13}$$

where $\|\mathbf{U}\|_{1,1} = \sum_{r=1}^{k} \|\mathbf{u}^r\|_1$ with $\mathbf{u}^r$ the rth row of $\mathbf{U}$, and $\|\cdot\|_F$ is the Frobenius norm. If the $p \times k$ dictionary matrix $\mathbf{V}$ is of rank k, then $\operatorname{rank}(\mathbf{V}\mathbf{U}) = \operatorname{rank}(\mathbf{U})$.

The objective function of the sparse modeling problem (2.13) is convex in $\mathbf{U}$ (when $\mathbf{V}$ is fixed) and convex in $\mathbf{V}$ (when $\mathbf{U}$ is fixed), but not jointly convex in $(\mathbf{U}, \mathbf{V})$. So the sparse coding problem can be solved by minimizing over one variable while keeping the other one fixed. Fixing $\mathbf{V}$, the sparse modeling problem reduces to the sparse coding problem, which is an ℓ_1-regularized least squares problem (also known as the *Lasso* or *basis pursuit*), and can be solved via the feature-sign search algorithm [68]. On the other hand, fixing $\mathbf{U}$, the sparse modeling problem reduces to the ℓ_2-constrained least squares problem, which can be efficiently solved using the Lagrange dual [65, 68].

2.4 Proposed Clustering Approach

In this section, we give a detailed description of our new clustering method that makes use of fermi density descriptor [9] and graph regularized sparse coding [65]. Each point or image in the dataset is first represented by a FDD descriptor. Then, the FDD descriptors are mapped to high-dimensional sparse codes via graph regularized sparse coding. The flow chart of the proposed GraphFDD framework is depicted in Figure 3.1. The last stage of the proposed approach is to perform cluster analysis on the sparse codes using a clustering algorithm (e.g., K-means). The K-means algorithm is arguably one of the most popular and effective clustering methods. In a nutshell, K-means assigns each data point to the cluster having the nearest centroid.

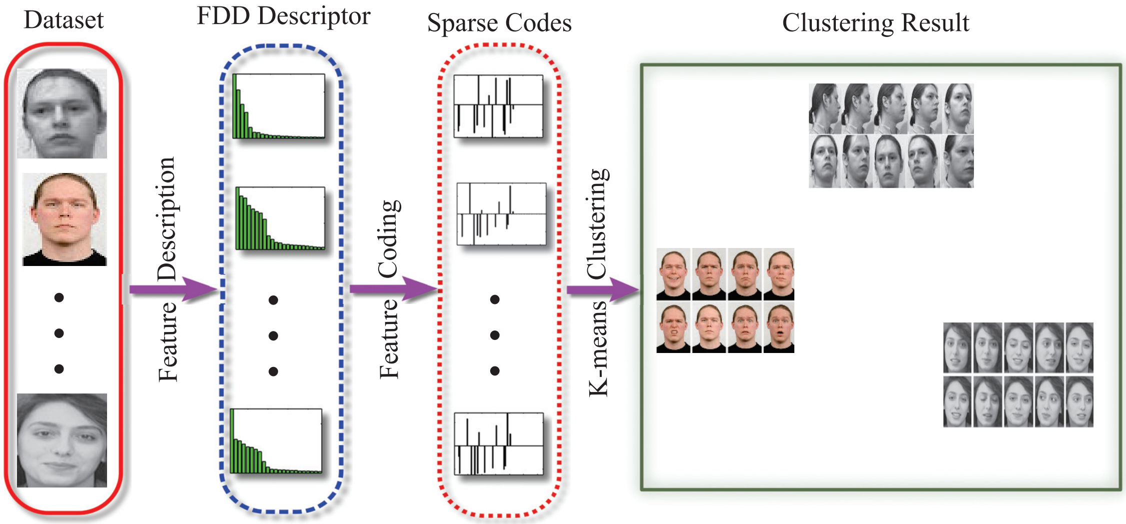


FIGURE 2.1: Flowchart of the proposed clustering approach.

Fermi Density Descriptor FDD was designed for detection of anomalies and it is based on quantum mechanics approach for the detection of objects in low density regions. This descriptor has solid background of physics and helps to differentiate anomalies from normal instances in a local structure. FDD is originated from Fermi-Dirac distribution (FD) which is defined as

$$f_{FD}(\lambda) = \frac{1}{1 + e^{(\lambda-\mu)/T}}, \tag{2.14}$$

where μ is obtained from

$$\sum_{\lambda} \frac{1}{1+e^{(\lambda-\mu)/T}} = n/2 \tag{2.15}$$

FDD is given in terms of eigenvalues and eigenfunctions :

$$\mathrm{FDD}(\mathbf{z}) = \frac{1}{C}\sum_{k=1}^{n}\left(\frac{1}{1+e^{(\lambda_k-\mu)/T}}\right)^2 \varphi_k(\mathbf{z})^2, \tag{2.16}$$

where

$$C = \sum_{k=1}^{n}\left(\frac{1}{1+e^{(\lambda_k-\mu)/T}}\right)^2 \tag{2.17}$$

For each point $\mathbf{z} \in \mathcal{Z}$, the FDD descriptor is defined as a p-dimensional feature vector $\mathbf{x} \in \mathbb{R}^p$ after eigendecomposition. Fermi-dirac distribution is used to derive value of μ [9, 32]. It should be noted that the FDD descriptor is a compact yet discriminative representation of an image, and possesses many attractive properties including stability and robustness to parameter tuning. So, basically this is a local descriptor which contains good properties like robustness to noise and with the best choice of different parameters, we found the better performance of this descriptor in conjunction with graph regularized sparse coding method. We have performed our comparison with different related descriptors in our research.

Graph Regularized Sparse Coding In contrast to dimensionality reduction methods, an overcomplete dictionary may be regarded as a linear operator that maps (or embeds) a low-dimensional dense vector into a high-dimensional sparse vector. Concretely, given n data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, we may construct a neighborhood graph $\mathbb{G}$ with n vertices, where each vertex represents a data point, as illustrated in Figure 2.2. More specifically, the κ-nearest neighbor graphs are built by connecting every pair of data point $\mathbf{x}_i$ and $\mathbf{x}_j$ by an undirected edge if $\mathbf{x}_j$ is among the κ-nearest neighbors of $\mathbf{x}_i$. The κ-neighborhood of $\mathbf{x}_i$ is denoted by $\mathcal{N}_i^{\kappa}$. Furthermore, we assume that the neighborhood graphs are symmetrically defined, i.e. $\mathbf{x}_j$ by $\mathcal{N}_i^{\kappa}$ iff $\mathbf{x}_i$ by $\mathcal{N}_j^{\kappa}$. For the sake of generality, we simply denote the neighborhood of a data point $\mathbf{x}_i$ by $\mathcal{N}_i$.

The neighborhood graph $\mathbb{G}$ is then embedded into a high-dimensional space $\mathbb{R}^k$ ($k > p$) by the coefficient matrix $\mathbf{U}$, where each column $\mathbf{u}_i$ of $\mathbf{U}$ is a k-dimensional vector that yields the embedding coordinates of the ith data point $\mathbf{x}_i$. The adjacency matrix $\mathbf{W} = (w_{ij})$ of $\mathbb{G}$ is an $n \times n$ matrix whose elements are given by

$$w_{ij} = \begin{cases} 1 & \text{if } j \in \mathcal{N}_i \\ 0 & \text{o.w.} \end{cases}$$

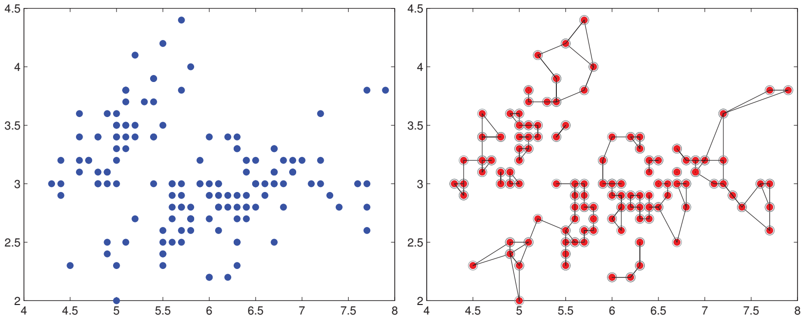


FIGURE 2.2: Point cloud of 150 points (left); κ-nearest neighborhood graph with $\kappa = 3$ (right).

where $\mathcal{N}_i$ denotes the κ-neighborhood of $\mathbf{x}_i$, and each element w_{ij} represents the similarity between a vertex pair $(\mathbf{x}_i, \mathbf{x}_j)$. The Laplacian matrix of $\mathbb{G}$ is $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D} = \operatorname{diag}(d_1, \dots, d_n)$ with $d_i = \mathbf{W1} = \sum_{j=1}^{n} w_{ij}$, $i = 1, \dots, n$. The coefficient matrix $\mathbf{U}$ may be obtained by minimizing the following objective function

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\|\mathbf{u}_i - \mathbf{u}_j\|^2 = \operatorname{tr}(\mathbf{U}\mathbf{L}\mathbf{U}^{\mathsf{T}}), \tag{2.18}$$

which measures the smoothness of the coding vectors in $\mathbf{U}$ with respect to the graph topology. That is, neighboring data points in $\mathbb{G}$ will be assigned or encoded by sparse codes that are more likely to be close, so as to reduce the differences in (2.18). The closer this value to zero, the more similar are the sparse codes at neighboring graph vertices. Incorporating (2.18) into the sparse modeling objective function yields

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V}} \quad & \|\mathbf{X} - \mathbf{V}\mathbf{U}\|_F^2 + \gamma \operatorname{tr}(\mathbf{U}\mathbf{L}\mathbf{U}^{\mathsf{T}}) + \lambda \sum_{i=1}^{n} \|\mathbf{u}_i\|_1 \\ \text{s.t.} \quad & \|\mathbf{v}_r\|^2 \leq 1, \quad r = 1, \dots, k \end{aligned} \tag{2.19}$$

which is referred to as the graph regularized sparse coding (GSC) problem [65]. Similar to sparse coding, the optimization problem (2.19) can be solved using the feature-sign search algorithm [65] to learn the sparse codes (i.e. the matrix $\mathbf{U}$). The second term (i.e. Laplacian penalty term) of the objective function of the graph regularized sparse coding problem plays a crucial role not only in explicitly taking into consideration the correlation between sparse codes, but also in preserving the locality of features to be encoded. In other words, two features or data points $\mathbf{x}_i$ and $\mathbf{x}_j$ that are close to each other in the original feature space (i.e. adjacent in $\mathbb{G}$) are encoded as sparse codes $\mathbf{u}_i$ and $\mathbf{u}_j$

that are more likely to be close to each other in the sparse codes space. Such a locality-preserving property is of paramount importance in classification and clustering tasks.

Proposed Algorithm Clustering of images or texts creates data groups or clusters, which are formed in such a way that images in the same cluster are very similar, while images in different clusters are very dissimilar. In other words, the main objective of cluster analysis is to group images or text into similar and distinct clusters.

As we mentioned before, GSC method has good discriminative power and it explicitly handles the manifold structure of data space. On the other hand, fermi density descriptor also have strong connection with physics and it has high ability to discriminate normal data from anomalies. So, our basic idea is to combine fermi density descriptor with GSC to get better performance in terms of data clustering.

The proposed algorithm consists of seven main steps. In starting three steps, we have to calculate affinity matrix, diagonal matrix and graph laplacian using normalization techniques or no normalization. Fourth step is to compute generalized eigenvectors and eigenvalues. The fifth step is to represent each data point in a dataset by the FDD descriptor, which is a normalized feature vector consisting of low-level features. More specifically, let $\mathcal{Z}$ be a dataset of n data points with m dimensions, where each data point is represented by a p-dimensional FDD descriptor $\mathbf{x}_i$, $i = 1, \ldots, n$. The n FDD descriptors can be arranged in a $p \times n$ data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{p\times n}$. The task in clustering is then to identify clusters of data points and assign each point to one of these clusters.

In the sixth step, the FDD descriptors are mapped to high-dimensional mid-level feature vectors (sparse codes) via graph regularized sparse coding, resulting in a sparse codes matrix $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_n) \in \mathbb{R}^{k\times n}$ whose columns are the k-dimensional sparse codes. In the last step, the K-means algorithm is performed on the sparse codes to partition the data matrix $\mathbf{U}$ into K mutually exclusive clusters. To assess the performance of the proposed framework, we used clustering evaluation measures and indices, which will be discussed in more detail in the next section. The main algorithmic steps of our GraphFDD approach are summarized in Algorithm 1.

Algorithm 1 GraphFDD Algorithm Steps

```
Input: Dataset $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ is an $m \times n$ input data
Output: $n$-dimensional vector $\mathbf{y}$ containing cluster indices of each data point
1: Compute the $n \times n$ affinity matrix $\mathbf{W} = \big(w_{ij}\big)$, where
       $$w_{ij} = \exp\left(-\frac{||\mathbf{z}_i - \mathbf{z}_j||^2}{2\sigma^2}\right) \qquad (2.20)$$
2: Compute the diagonal matrix $\mathbf{D} = \operatorname{diag}(d_1, \ldots, d_n)$ with
       $$d_i = \mathbf{W1} = \sum_{j=1}^{n} w_{ij},\ i = 1, \ldots, n \qquad (2.21)$$
3: Compute the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$.
4: Compute the eigenvalues $\lambda_l$ and eigenvectors $\varphi_l$ of $\mathbf{L}$.
5: Compute the $p$-dimensional FDD descriptor $\mathbf{x}_i$ of each data point $\mathbf{z}_i$, $i = 1, \ldots, n$ and arrange all
   these signatures in a $p \times n$ data matrix $\mathbf{X}$.
6: Solve the graph regularized sparse coding problem (2.19) to find the $k \times n$ data matrix $\mathbf{U}$ of sparse
   codes, where $k > p$.
7: Perform K-means algorithm on $\mathbf{U}$ to find the $n$-dimensional vector $\mathbf{y}$ of cluster indices.
```

2.5 Proposed Classification Approach

In this section, we give a detailed description of our proposed classification method that makes use of fermi density descriptor [9, 32] and graph regularized sparse coding [65]. Each point or image in the dataset is first represented by a FDD. Then, the FDD signatures are mapped into high-dimensional sparse codes via graph regularized sparse coding. The flow chart of the proposed GraphFDD for classification framework is depicted in Figure 2.3. The last stage of the proposed approach is to perform classification on the sparse codes using a classification algorithm. Multiclass support vector machines (SVMs) are arguably the most popular and effective supervised learning methods used for classification. Broadly speaking, supervised learning algorithms consist of two main steps: training step and test step. In the training step, a classification model (classifier) is learned from the training data by a learning algorithm (e.g., SVMs). In the test step, the learned model is evaluated using a set of test data to predict the class labels for the classifier and hence assess the classification accuracy.

Multiclass Support Vector Machines SVMs are supervised learning models that have proven effective in solving classification problems. They are based upon the idea of maximizing the margin, i.e. maximizing the minimum distance from the separating hyperplane to the nearest example. Although

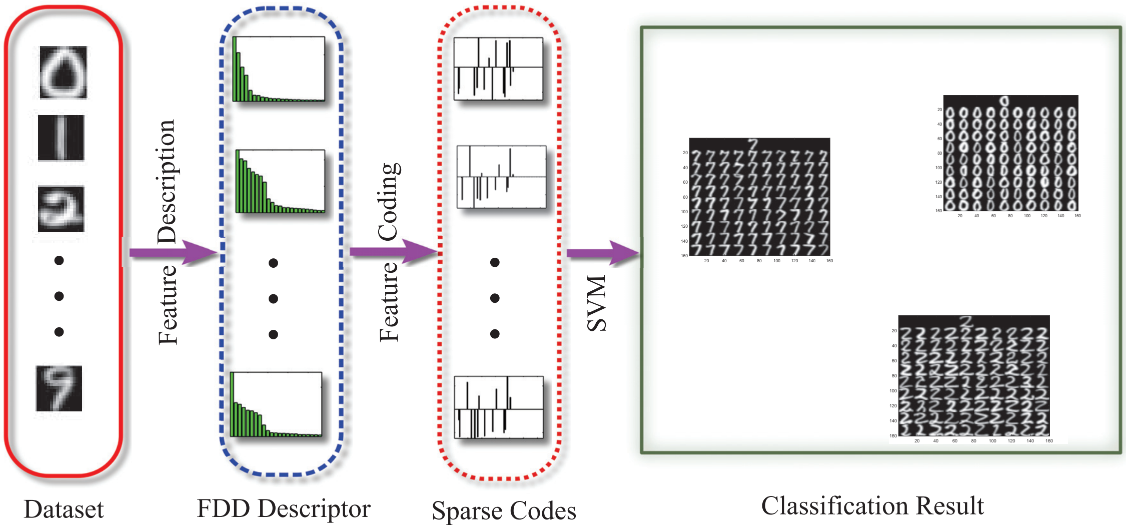


FIGURE 2.3: Flowchart of the proposed Classification framework.

SVMs were originally designed for binary classification, several extensions have been proposed in the literature to handle the multiclass classification. The idea of multiclass SVM is to decompose the multiclass problem into multiple binary classification tasks that can be solved efficiently using binary SVM classifiers. One of the simplest and most widely used coding designs for multiclass classification is the one-vs-all approach, which constructs K binary SVM classifiers such that for each binary classifier, one class is positive and the rest are negative. In other words, the one-vs-all approach requires K binary SVM classifiers, where the lth classifier is trained with positive examples belonging to class l and negative examples belonging to the remaining $K-1$ classes. A new test example is then assigned to the class with the largest value of the decision function for the binary problem of the lth class versus the rest, where $l = 1, \ldots, K$.

Proposed Algorithm Text or image classification is a supervised learning method that assigns different images or data points in a dataset to target classes. The main objective of classification is to accurately predict the target class for each data point in the dataset. Our proposed classification algorithm consists of three main steps. The first step is to represent each image or data point in a dataset by the FDD descriptor, which is a normalized feature vector consisting of low-level features. More specifically, let $\mathcal{Z}$ be a dataset of n data points or images where each data point is represented by a p-dimensional FDD $\mathbf{x}_i$, $i = 1, \ldots, n$. The n FDD descriptors can be arranged in a $p \times n$ data matrix

$\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^{p \times n}$.

In the second step, the FDD descriptors in the data matrix $\mathbf{X}$ are mapped to high-dimensional mid-level feature vectors (sparse codes) via graph regularized sparse coding, resulting in a sparse codes matrix $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n) \in \mathbb{R}^{k \times n}$ whose columns are the k-dimensional sparse codes. In the third step, a one-vs-all multiclass linear SVM classifier is performed on the sparse codes to find the best hyperplane that separates all data points of one class from those of the other classes. The task in multiclass classification is to assign a class label to each input example (sparse code). More precisely, given a training data of the form $\mathcal{U}_{\text{train}} = \{(\mathbf{u}_i, y_i)\}$, where $\mathbf{u}_i \in \mathbb{R}^k$ is the ith example (i.e. sparse code) and $y_i \in \{1, \dots, K\}$ is its ith class label, we aim at finding a learning model that contains the optimized parameters from the SVM algorithm. Then, the trained SVM model is applied to a test data $\mathcal{U}_{\text{test}}$, resulting in predicted labels $\hat{y}_i$ of new data. These predicted labels are subsequently compared to the labels of the test data to evaluate the classification accuracy of the model.

To assess the performance of the proposed classification framework, we employed two commonly-used evaluation criteria, the confusion matrix and classification accuracy, which will be discussed in more detail in the next section. The main algorithmic steps of our GraphFDD for classification approach are summarized in Algorithm 2.

Algorithm 2 GraphFDD algorithm for classification

```
Input: Dataset $\mathcal{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_n\}$ is an $m \times n$ input data
Output: $n$-dimensional vector $\hat{\mathbf{y}}$ containing predicted class labels of each data point
1: Compute the $p$-dimensional FDD descriptor $\mathbf{x}_i$ of each data point $\mathbf{z}_i$, $i = 1, \dots, n$, and arrange
   all these signatures in a $p \times n$ data matrix $\mathbf{X}$.
2: Solve the graph regularized sparse coding problem (2.19) to find the $k \times n$ data matrix $\mathbf{U}$ of sparse
   codes, where $k > p$.
3: Perform SVM on $\mathbf{U}$ to find the $n$-dimensional vector $\hat{\mathbf{y}}$ of predicted class labels.
```

2.6 Experimental Results

To access the performance of our proposed work, we conducted experiments on different benchmarks: USPS, COIL20, Ecoli, Glass, MSRC and VOC. We experimentally compare our algorithm with different spectral descriptors and related algorithms also. We performed our work for classification and clustering methods. For evaluation of results, we used evaluation measures: accuracy, confusion matrix and normalized mutual information score. We also show that the new algorithm can significantly improves accuracy and normalized mutual information score.

2.6.1 Settings

Comparing Signatures: We compared the proposed method with other spectral signatures which have been proposed in recent years, including WKS, HKS, GPS and SIHKS. It turns out that FDD with graph regularized sparse coding has high performance in comparison of other signatures.

Complexity: For implementation of our algorithms, we selected MATLAB 8.4.0 (R2014b) installed on 64 bit operating system with an Intel Core i7-4510U running at 2.60 GHz and 8 GB RAM. We have performed our experiment for classification and clustering with image and text data sets. The effectiveness of graph regularized sparse coding with FDD descriptor is validated by comparison with related algorithms. On the other hand, runtime (in seconds) of proposed algorithm is about 91.95s.

Parameter Selection: First of all, we have chosen the best parameters for all different signatures. FDD has two parameters of scaling: Gaussian scale (σ) and T. The scale σ is defined for local sensitivity and T is for environmental temperature. For HKS, we fixed the value of α, τ and time scale (t_0) parameters. Same parameters are chosen for SIHKS, α and τ. For GPS descriptor, we selected value of number of eigenvectors and eigenfunctions. We ignored the first eigen-pair because it does not contain any information. For evaluation of results with different algorithms, we set the other parameters like number of clusters, number of neighbors and number of eigenvectors for the starting computation before spectral descriptors in every algorithm. The optimal parameters settings of different algorithms in our experiments are shown in the Table 2.1. For fair comparison, we used the same parameters that have been employed in the baseline methods, and in particular the dimensions of the underlying signatures. The choice of these parameters has been found to perform well on different datasets.

WKS	GPS	HKS	SIHKS	FDD
$N = 100$ $\sigma = 0.05$	$l = 14$	$t_0 = 0.01$ $\alpha = 2$ $T = 15$ $\tau = 1/4$	$F = 193$ $T = 15$ $\tau = 1/16$ $\alpha = 2$	$\sigma = 1$ $T = 1000$

TABLE 2.1: Optimal parameter selection for spectral descriptors.

Datasets: We conduct experiments using different image datasets and text datasets for evaluation of results: USPS, COIL20, Ecoli, Glass, MSRC and VOC. USPS is a handwritten image dataset which consists 9298 images of 16 × 16 size. This dataset contains images which corresponds to different digit value. MSRC data set consists of 4324 images and VOC dataset consists 5011 images. These two datasets having similar classes. So based on those 6 semantic classes, authors of [69] combined these two datsets and constructed a MSRC vs VOC dataset which consists 1269 images for training from MSRC data set and a test set of 1530 images from VOC data set. So, we evaluated our experiments with MSRC vs VOC data set to speed up the evaluation of results. COIL20 dataset contains images of 20 objects of size 32 × 32 and every object has 72 images. Ecoli dataset which contains the class of localization site of protein. Glass dataset having the information of different types of glasses. Table 2.3 documents the statistics of different benchmark datasets.

Dataset	Type	Number of images/ instances	Features
USPS	Digit	9298	256
MSRC	Photo	1269	240
VOC	Photo	1530	240
COIL20	Photo	1440	1024
Ecoli	Text	336	8
Glass	Text	214	10

TABLE 2.2: Datasets used in experiments.

Performance Evaluation Measures: We evaluated the image classification performance using two evaluation measures: Accuracy (AC) and confusion matrix.

In practice, the available data (which has classes) $\mathcal{U}$ for classification is usually split into two disjoint subsets: the training set $\mathcal{U}_{\text{train}}$ for learning, and the test set $\mathcal{U}_{\text{test}}$ for testing. The training and test sets are usually selected by randomly sampling a set of training instances from $\mathcal{U}$ for learning and using the rest of instances for testing. The performance of a classifier is then assessed by applying it to test data with known target values and comparing the predicted values with the known values. One important way of evaluating the performance of a classifier is to compute its confusion matrix (also called contingency table), which is a $K \times K$ matrix that displays the number of correct and incorrect predictions made by the classifier compared with the actual classifications in the test set, where K is the number of classes.

Another intuitively appealing measure is the classification accuracy, which is a summary statistic that can be easily computed from the confusion matrix as the total number of correctly classified instances (i.e. diagonal elements of confusion matrix) divided by the total number of test instances. Alternatively, the accuracy of a classification model on a test set may be defined as follows

$$\text{Accuracy} = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}} = \frac{|\mathbf{u} : \mathbf{u} \in \mathcal{U}_{\text{test}} \wedge \hat{y}(\mathbf{u}) = y(\mathbf{u})|}{|\mathbf{u} : \mathbf{u} \in \mathcal{U}_{\text{test}}|}, \tag{2.22}$$

where $y(\mathbf{u})$ is the actual (true) label of $\mathbf{u}$, and $\hat{y}(\mathbf{u})$ is the label predicted by the classification algorithm. A correct classification means that the learned model predicts the same class as the original class of the test case. The error or misclassification rate is equal to one minus accuracy.

Clustering Evaluation Measure: Unlike classification, where it is easy to measure accuracy using labeled test data, for clustering we do not know what the correct clusters are, given a dataset. A commonly used evaluation method for clustering is based on ground truth, where classification datasets are used to evaluate the quality of clustering algorithms. Using such a dataset for cluster evaluation, we make the assumption that each class corresponds to a cluster. After clustering, we compare the cluster memberships with the class memberships to determine how good the clustering is. Normalized Mutual Information (NMI) is used for evaluation of clustering performance of our proposed algorithm.

On the other hand, given two sets of clusters $\mathcal{C} = \{C_r\}$ and $\mathcal{C}' = \{C'_r\}$ that are obtained from the ground truth and a clustering algorithm, the normalized mutual information (also called entropy correlation coefficient) is defined as

$$\text{NMI}(\mathcal{C}, \mathcal{C}') = \frac{2I(\mathcal{C}, \mathcal{C}')}{H(\mathcal{C}) + H(\mathcal{C}')}, \tag{2.23}$$

where $I(\mathcal{C}, \mathcal{C}')$ is the mutual information of $\mathcal{C}$ and $\mathcal{C}'$, $H(\mathcal{C})$ is the entropy of $\mathcal{C}$ and $H(\mathcal{C}')$ is the entropy of $\mathcal{C}'$. Note that NMI is a measure of similarity, ranging from 0 to 1. A value of 0 means that the two sets of clusters are independent, while a value of 1 indicates that the two sets are identical. Larger NMI values indicate better clustering solutions [3,65].

2.6.2 Classification

We evaluated graph regularized sparse coding method for image classification on the aforementioned datasets. We performed the average comparison with support vector machines (SVM), sparse coding, PCA and our proposed method. The evaluation results shows the efficiency of new method with multiclass SVM. A one-vs-all multiclass SVM is first trained on the training data to learn the model (i.e. classifier), which is subsequently used on the test data with known target values in order to predict the class labels. Figure 2.4 shows the samples of different datasets.

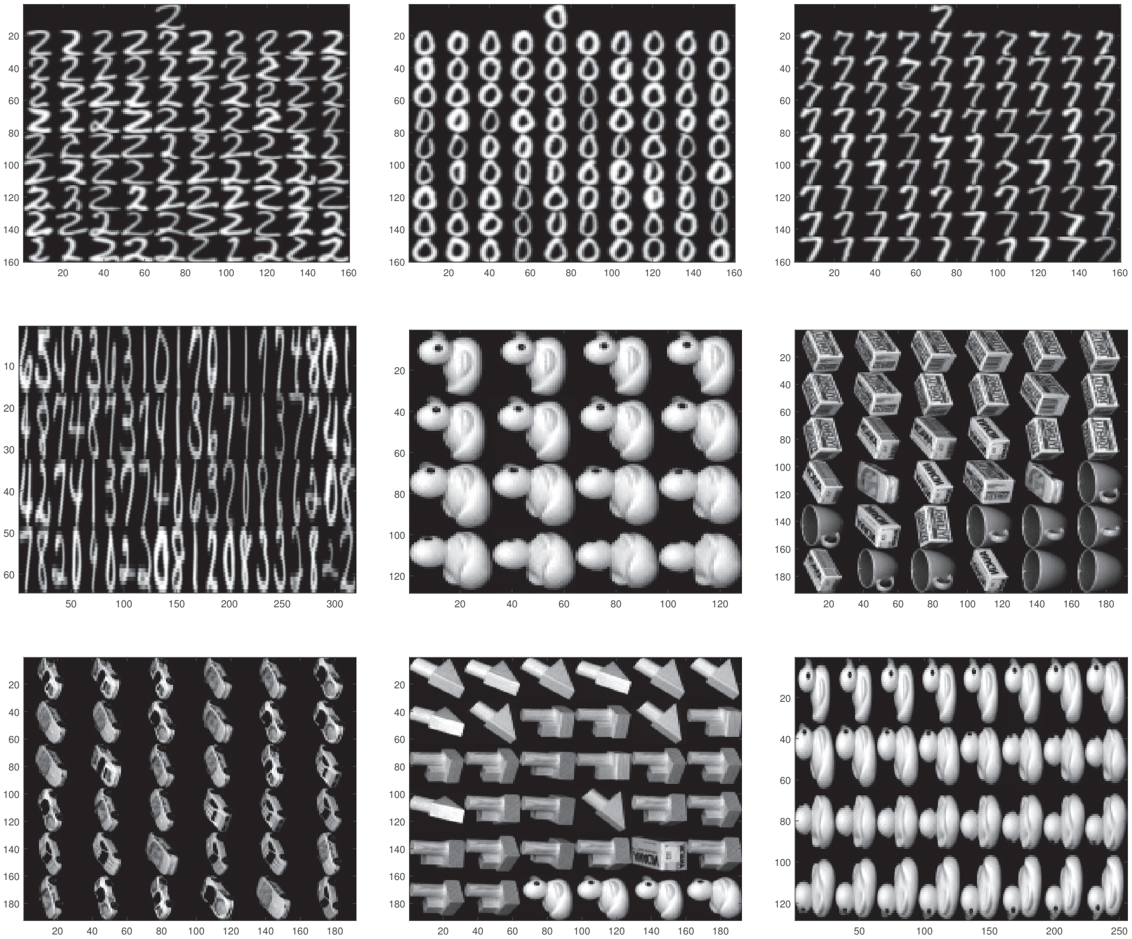


FIGURE 2.4: Samples of different datasets.

Figures 2.6 and 2.5 displays the confusion matrices for different datasets on the test data. These 10×10 confusion matrices for USPS data set and 20×20 confusion matrices for COIL20 data set, show how the predictions are made by the model. Its rows correspond to the actual (true) class of the data (i.e. the labels in the data), while its columns correspond to the predicted class (i.e. predictions made by the model). The value of each element in the confusion matrix is the number of predictions made with the class corresponding to the column for instances with the correct value as represented by the row. Thus, the diagonal elements show the number of correct classifications made for each class, and the off-diagonal elements show the errors made. As shown in figure, the proposed approach was able to more accurately classify different images in the test data as compared to other baseline methods. Such a good performance strongly suggests that GraphFDD captures well the discriminative features of the different images or data points.

For USPS dataset, we chose 6514 training images and 2784 testing images. For COIL20, we use training 1024 images and testing 420 images and for MSRC vs VOC dataset, MSRC images as training and images of VOC dataset as testing. For text datasets, we choose one-third part for testing among all data points in a dataset.

There are some important parameters that are selected for our experimental results to improve the overall performance of the algorithms. The number of basis vectors are $[32, 64, 128]$ and found that 128 shows the best results. The graph regularization parameter (γ) is selected as $[0.01, 0.02, 0.1, 0.2, 1.0, 10, 100]$ and sparsity regularization parameter (λ) is $[0.01, 0.02, 0.1, 0.2, 0.3, 0.4, 0.5]$ for experiments. Then we found the best results with 1.0 value of graph regularization parameter and 0.1 value of sparsity regularization parameter.

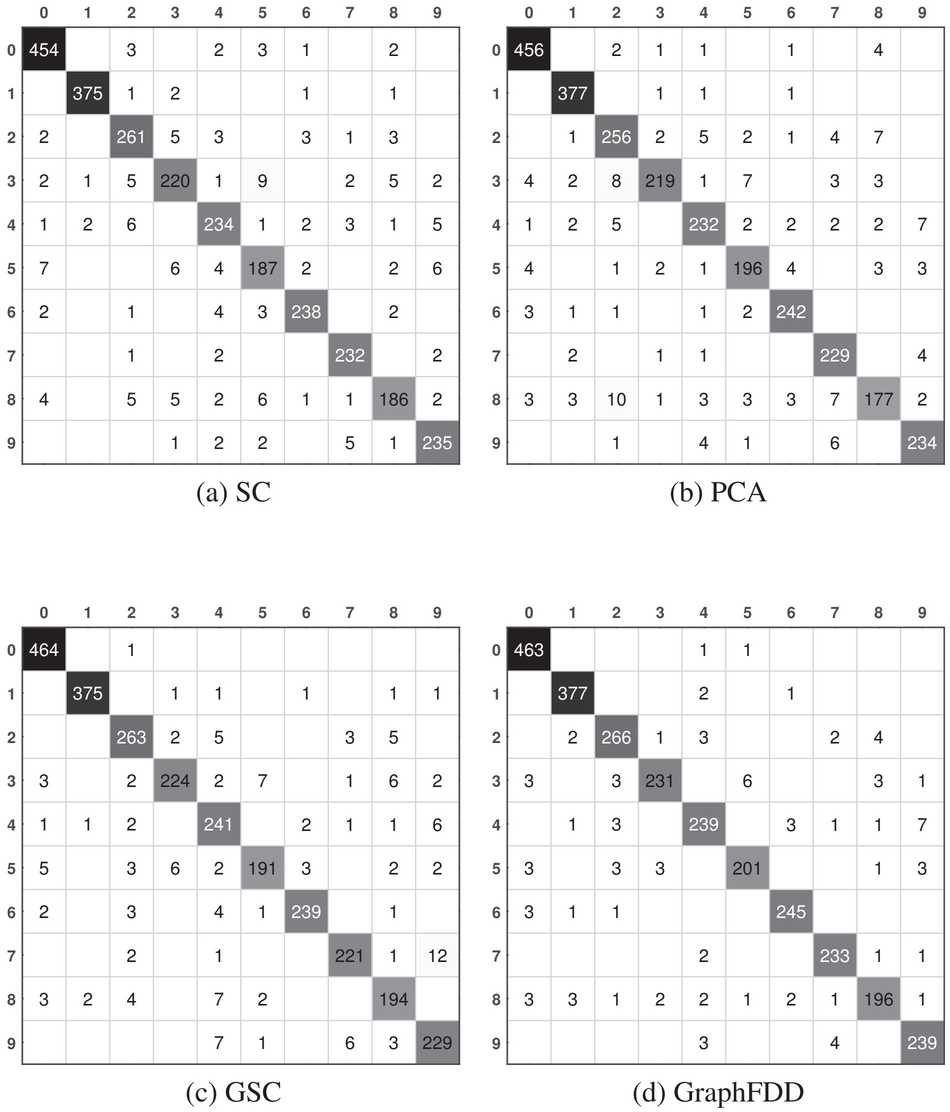
Following the common practice in classification tasks, we repeated the experimental process 10 times with different randomly selected training and test data in an effort to obtain reliable results, and the accuracy for each run was recorded. The classification accuracy results are summarized in Table 2.3, which shows the average results of the state-of-the-art classification methods and the proposed framework. As can be seen, GraphFDD achieves better performance than simple SVM, sparse coding (SC), PCA and GSC.

TABLE 2.3: Classification accuracy results on different datasets. Boldface numbers indicate the best classification performance.

	Classification Methods				
Dataset	SVM	SC	PCA	GSC	GraphFDD
USPS	94.18	89.04	90.81	94.20	**97.05**
MSRC vs VOC	42.42	41.63	41.6	42.25	**43.18**
COIL20	95.35	90.1	88.2	91.2	**96.10**
Ecoli	92.08	93.25	92.50	94.70	**95.14**
Glass	93.60	93.65	93.01	93.90	**94.05**

Moreover, the best performance is shown by the proposed method which ensures that this method has high capability of discrimination between different kind of images. With the high performance of

GraphFDD method for classification, we also decide to use this method for clustering results.



	0	1	2	3	4	5	6	7	8	9
0	454		3		2	3	1		2	
1		375	1	2			1		1	
2	2		261	5	3		3	1	3	
3	2	1	5	220	1	9		2	5	2
4	1	2	6		234	1	2	3	1	5
5	7			6	4	187	2		2	6
6	2		1		4	3	238		2	
7			1		2			232		2
8	4		5	5	2	6	1	1	186	2
9				1	2	2		5	1	235

(a) SC

	0	1	2	3	4	5	6	7	8	9
0	456		2	1	1		1		4	
1		377		1	1		1			
2		1	256	2	5	2	1	4	7	
3	4	2	8	219	1	7		3	3	
4	1	2	5		232	2	2	2	2	7
5	4		1	2	1	196	4		3	3
6	3	1	1		1	2	242			
7		2		1	1			229		4
8	3	3	10	1	3	3	3	7	177	2
9			1		4	1		6		234

(b) PCA

	0	1	2	3	4	5	6	7	8	9
0	464		1							
1		375		1	1		1		1	1
2			263	2	5			3	5	
3	3		2	224	2	7		1	6	2
4	1	1	2		241		2	1	1	6
5	5		3	6	2	191	3		2	2
6	2		3		4	1	239		1	
7			2		1			221	1	12
8	3	2	4		7	2			194	
9					7	1		6	3	229

(c) GSC

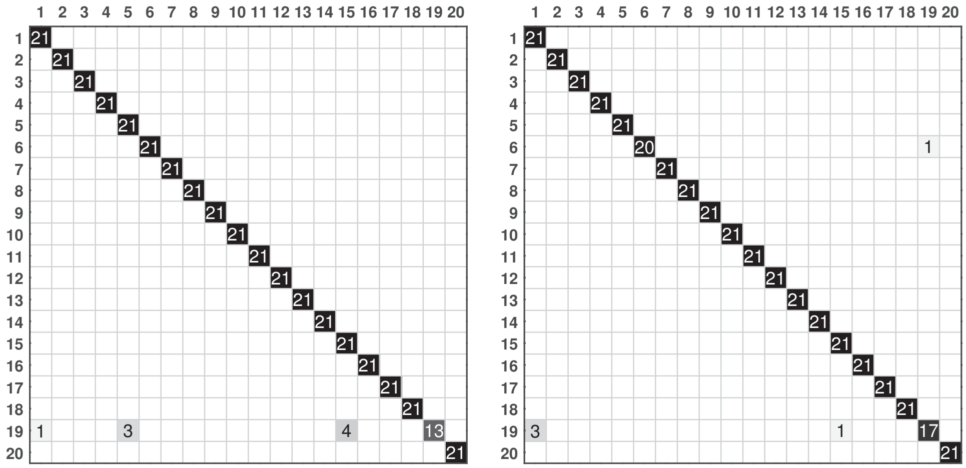
	0	1	2	3	4	5	6	7	8	9
0	463				1	1				
1		377			2		1			
2		2	266	1	3			2	4	
3	3		3	231		6			3	1
4		1	3		239		3	1	1	7
5	3		3	3		201			1	3
6	3	1	1				245			
7					2			233	1	1
8	3	3	1	2	2	1	2	1	196	1
9					3			4		239

(d) GraphFDD

FIGURE 2.5: Confusion matrices for USPS dataset.

2.6.3 Clustering

We conducted extensive experiments for clustering on two benchmarks, USPS and COIL20, to evaluate the performance of our proposed approach. The effectiveness of our method is validated by performing a comparison with different baseline algorithms. For data clustering, we compare the five algorithms: K-means algorithm, PCA, SC, GSC and our proposed GraphFDD algorithm. In addition,



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	21																			
2		21																		
3			21																	
4				21																
5					21															
6						21														
7							21													
8								21												
9									21											
10										21										
11											21									
12												21								
13													21							
14														21						
15															21					
16																21				
17																	21			
18																		21		
19	1				3										4				13	
20																				21

(a) SVM

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	21																			
2		21																		
3			21																	
4				21																
5					21															
6						20													1	
7							21													
8								21												
9									21											
10										21										
11											21									
12												21								
13													21							
14														21						
15															21					
16																21				
17																	21			
18																		21		
19	3														1				17	
20																				21

(b) GraphFDD

FIGURE 2.6: Confusion matrices for COIL20 dataset.

we applied FDD descriptor with PCA and SC algorithms and also perform comparison with PCA + FDD algorithm and SC + FDD algorithm.

Clustering Validation: The quality of cluster structure is usually analyzed using the silhouette plot, which is a bar plot of all the silhouette values ranked in descending order, where the length of the ith bar is equal to the ith silhouette value, as shown in Figure 2.7. The silhouette value s_i at the ith data point is given by

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}, \quad i = 1, \ldots, n \tag{2.24}$$

where a_i is the average dissimilarity of the ith data point to all other data points in the same cluster, and b_i is the average dissimilarity of the ith data point to all other data points in the neighboring cluster.

In words, the silhouette value measures how well a data point has been clustered by comparing its dissimilarity within its cluster to its dissimilarity with its nearest neighbor. A silhouette value ranges from -1 to 1. A silhouette value close to zero indicates that the ith data point has been arbitrarily clustered, i.e. it lies between two clusters. When s_i is close to -1, the ith data point is poorly clustered (i.e. its dissimilarity with other data points in its cluster is much larger than its dissimilarity with data points in the nearest cluster). A silhouette value close to 1 indicates that the data point is well-clustered (i.e. its dissimilarity with other data points in its cluster is much smaller than its dissimilarity with data point in the nearest cluster). A useful summary statistic for clustering is the so-called silhouette coefficient, which is the maximum average silhouette value across all the number of clusters. A value of silhouette coefficient in the range of 0.6 to 1 may be interpreted as an indication of a strong clustering structure.

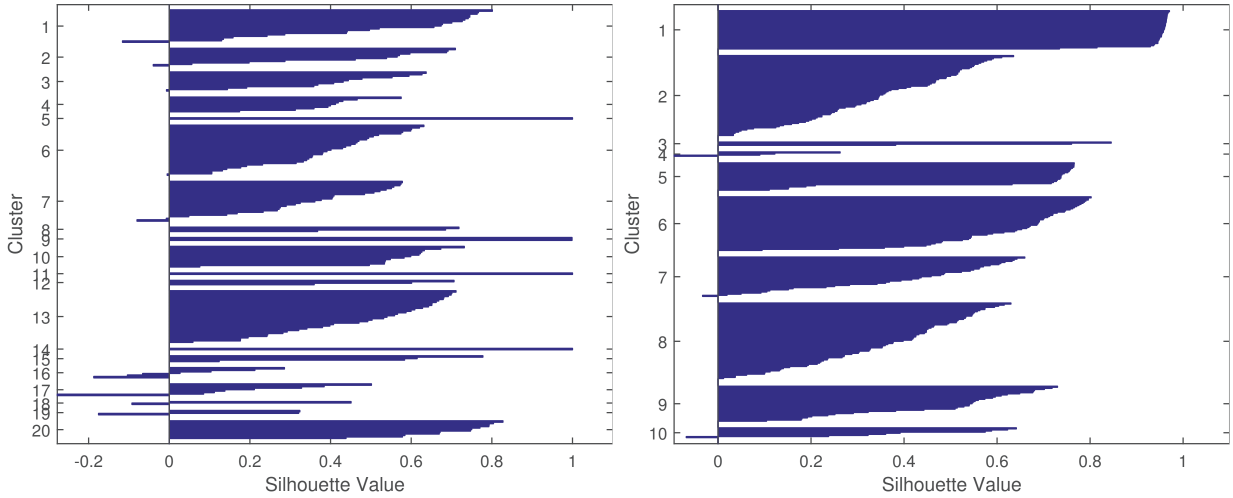
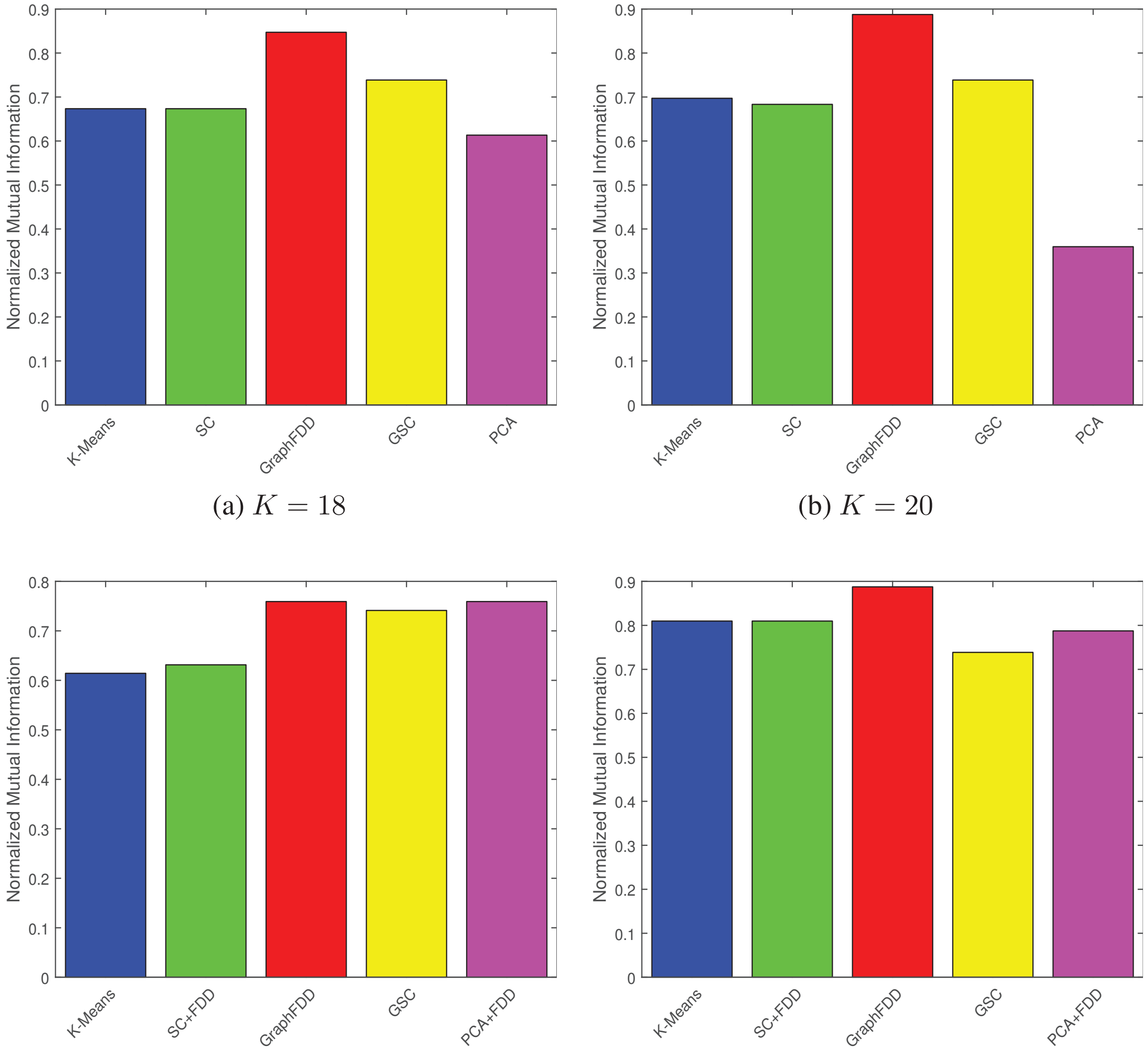


FIGURE 2.7: silhouette plot (left) for COIL20 using K-means algorithm with $K = 20$ and silhouette plot (right) for USPS using K-means algorithm with $K = 10$

2.6.4 Parameter Sensitivity

The proposed GraphFDD approach depends on four main parameters that affect its clustering performance. The first two parameters are the regularization parameter (γ) of the Laplacian penalty term and the regularization parameter (λ) of the sparsity penalty term. The third one is the number of κ-nearest neighbors, which is used to define the neighborhood graph for computing the Laplacian penalty term. The fourth parameter is the number k of basis vectors, i.e. the dimension of the sparse codes.

Moreover, we compared the proposed work with different methods. First of all, we choose the best parameters for clustering to gain further insight into performance variation between different methods. We performed our results with different number of basis vectors are $[32, 64, 128]$ and found that 128 shows the best results. The graph regularization parameter (γ) is selected as $[0.01, 0.02, 0.1, 0.2, 1.0, 10, 100]$ and sparsity regularization parameter (λ) is $[0.01, 0.02, 0.1, 0.2, 0.3, 0.4, 0.5]$ for experiments.



(a) $K = 18$

(b) $K = 20$

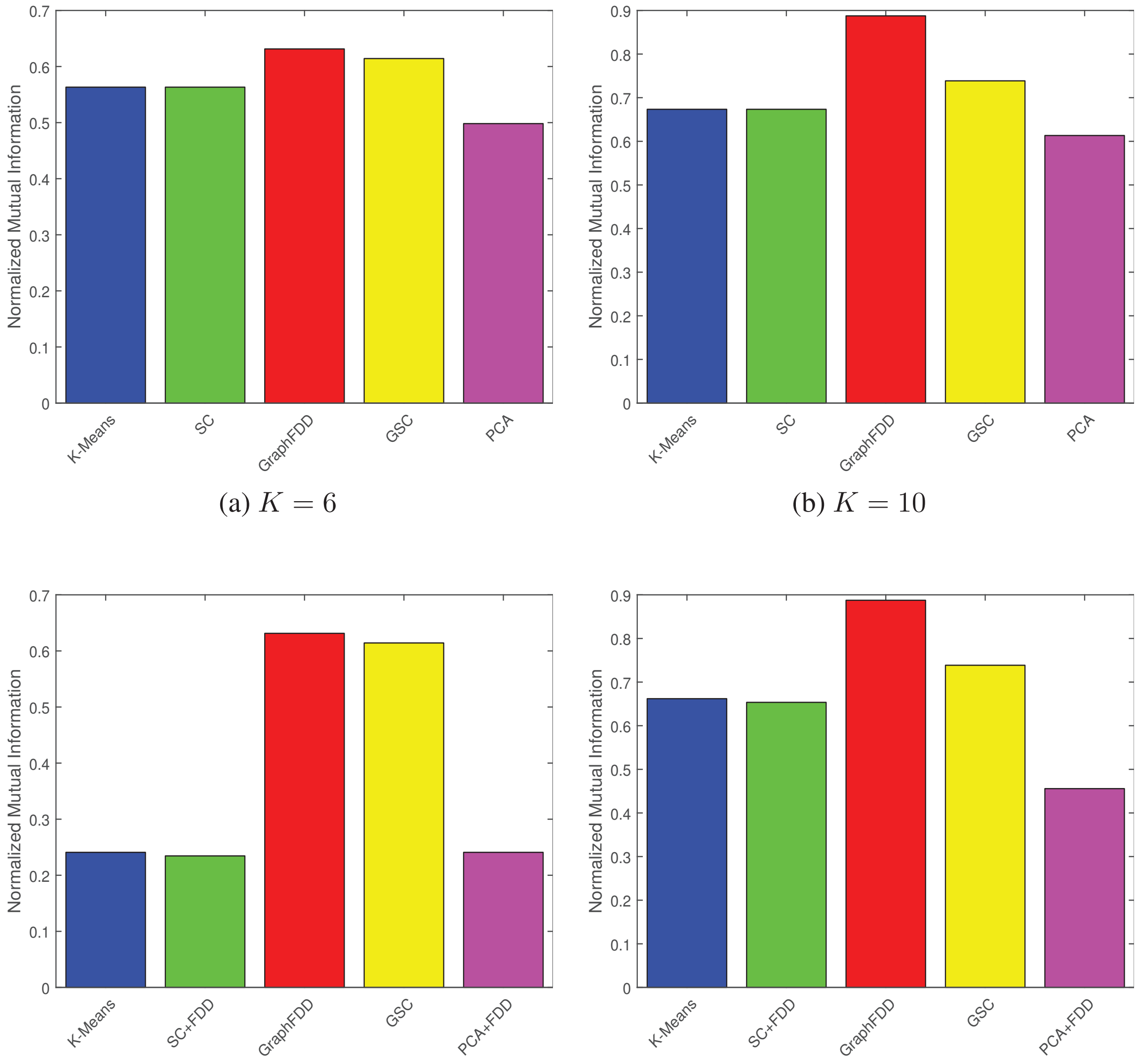
(c) $K = 18$

(d) $K = 20$

FIGURE 2.8: Clustering results with different baseline methods on COIL20 dataset.

We also choose number of clusters ($K = [7, 8, 10, 12, 15, 17, 18, 19, 20, 25, 30]$) and found best performance with $K = [20]$ which is consistent with number of classes in COIL20 dataset. In addition, we found the best results with 1.2 value of graph regularization parameter, 0.2 value of sparsity regularization parameter for different K on COIL20 dataset.

Clustering results with different number of clusters are shown in Figures 2.8 and 2.9 and found that our algorithm outperforms among other methods.



(a) $K = 6$

(b) $K = 10$

(c) $K = 6$

(d) $K = 10$

FIGURE 2.9: Clustering results with different baseline methods on USPS dataset.

For USPS image dataset, we found best performance with 1.1 value of graph regularization parameter of laplacian penalty term, 0.3 value of sparsity regularization parameter on USPS dataset. The best performance is obtained with $K = [10]$ in USPS dataset. In addition, performance of our proposed work is satisfactory for wide range of values of different parameters.

Table 2.4 lists the results of proposed algorithm with different values of parameters. As can be seen in the table, the proposed GraphFDD approach yields the best results.

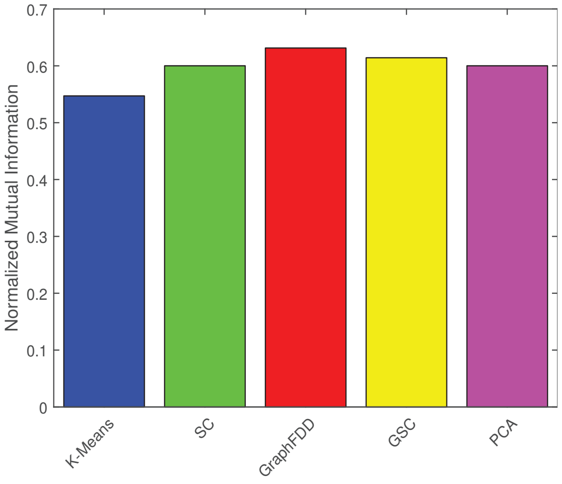
		Clustering Methods				
Dataset	Parameters	K-means	SC + FDD	PCA + FDD	GSC	GraphFDD
USPS	$\gamma = 0.01$	0.620	0.731	0.731	0.630	**0.801**
USPS	$\gamma = 0.1$	0.562	0.705	0.748	0.781	**0.821**
USPS	$\gamma = 1.1$	0.601	0.660	0.655	0.606	**0.850**
USPS	$\lambda = 0.01$	0.650	0.554	0.520	0.510	**0.757**
USPS	$\lambda = 0.1$	0.651	0.654	0.721	0.731	**0.792**
USPS	$\lambda = 0.3$	0.620	0.635	0.485	0.642	**0.883**
COIL20	$\gamma = 0.01$	0.641	0.628	0.642	0.647	**0.691**
COIL20	$\gamma = 0.1$	0.681	0.684	0.693	0.694	**0.720**
COIL20	$\gamma = 1.2$	0.680	0.804	0.632	0.857	**0.899**
COIL20	$\lambda = 0.01$	0.590	0.623	0.642	0.640	**0.680**
COIL20	$\lambda = 0.1$	0.601	0.621	0.688	0.696	**0.719**
COIL20	$\lambda = 0.2$	0.715	0.732	0.701	0.741	**0.806**

TABLE 2.4: Clustering results in terms of NMI. Boldface numbers indicate the best performance.

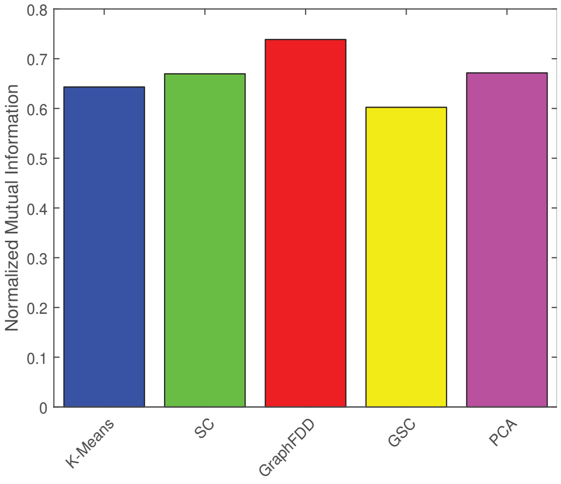
Moreover, we investigate our evaluation with different number of iterations (N). Even it affects the overall performance of proposed algorithm as well as different state-of-the-art methods. We have performed our experiment with different number of iterations $[5, 10, 15, 20, 25, 30, 45, 50, 75, 100]$. In addition, new algorithm also shows better results with more number of repeats or iterations as shown in Figure 2.10.

To systematically demonstrate the performance of GraphFDD algorithm, we tested with different values of the number of nearest neighbors. We also examine the performance of existing descriptors such as WKS, SIHKS, HKS, GPS with the proposed GraphFDD method. The comparison results with other descriptors are shown in Table 2.5.

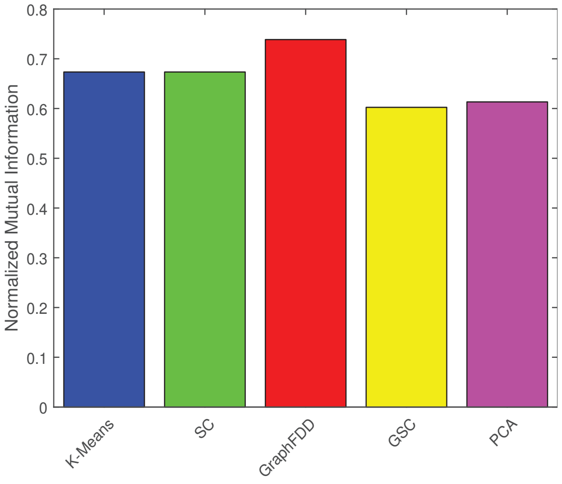
To summarize, the GraphFDD algorithm has most robust performance in terms of both evaluation measures. It can also be observed that the number of clusters also affects the results. As expected, the GPS descriptor shows the least performance and the proposed method performs the best in both datasets, indicating the consistency of the results.



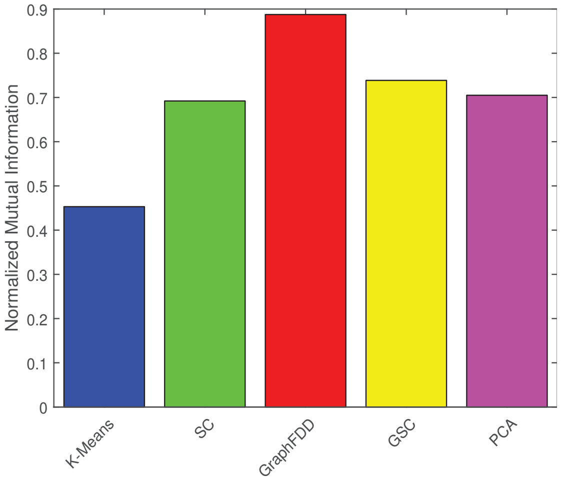
(a) COIL20, $N = 15$



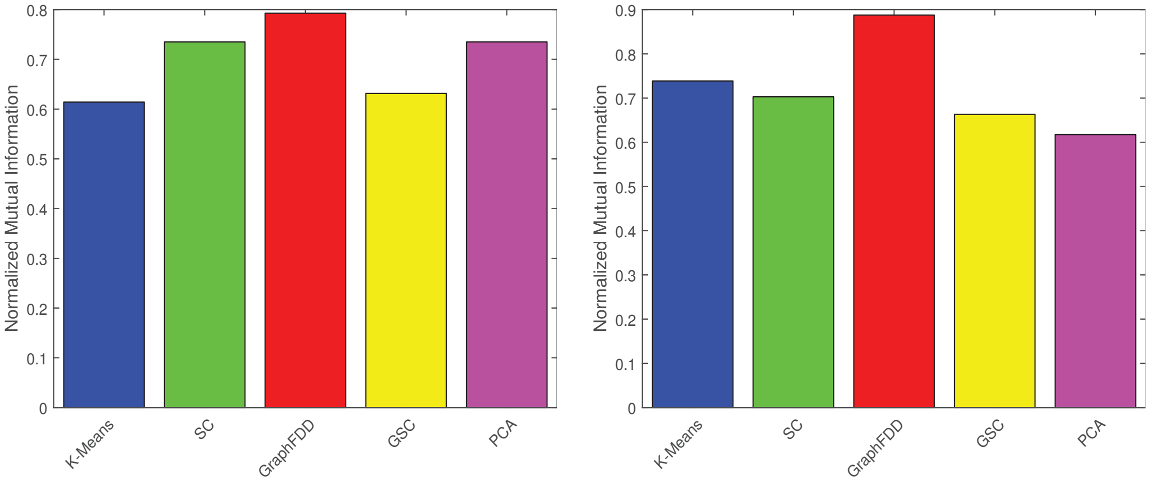
(b) COIL20, $N = 30$



(c) COIL20, $N = 45$



(d) COIL20, $N = 100$



(e) USPS, $N = 45$

(f) USPS, $N = 100$

FIGURE 2.10: Clustering results for different number of iterations.

		Clustering Methods				
K	Dataset	WKS	HKS	SIHKS	GPS	GraphFDD
2	USPS	0.402	0.483	0.402	0.354	**0.608**
4	USPS	0.485	0.503	0.338	0.285	**0.585**
8	USPS	0.382	0.526	0.464	0.355	**0.752**
10	USPS	0.506	0.624	0.552	0.421	**0.804**
12	USPS	0.426	0.482	0.490	0.295	**0.693**
16	USPS	0.518	0.395	0.368	0.291	**0.681**
20	USPS	0.680	0.659	0.600	0.340	**0.708**
2	COIL20	0.364	0.346	0.402	0.253	**0.583**
4	COIL20	0.385	0.402	0.486	0.295	**0.621**
8	COIL20	0.421	0.426	0.472	0.315	**0.638**
12	COIL20	0.411	0.484	0.500	0.350	**0.687**
16	COIL20	0.442	0.495	0.486	0.379	**0.742**
20	COIL20	0.582	0.528	0.624	0.450	**0.826**

TABLE 2.5: Comparison with other spectral descriptors, where K denotes the number of clusters. Boldface numbers indicate the best performance.

3 Spectral Graph Wavelets for Data Clustering

Feature descriptors have become an increasingly important tool in the field of data mining. Features can be extracted and subsequently used to design robust signatures for retrieval, correspondence, classification and clustering. The latter will be the focus of this chapter. More specifically, we propose a spectral approach for data clustering using spectral graph wavelet signature (SGWS) and the K-means algorithm. SGWS is a efficient descriptor that was originally designed for local as well as global geometry of 3D shapes. We evaluated the proposed approach using standard clustering evaluation measures, including normalized mutual information (NMI) and Area Under Curve (AUC). Experimental results on different benchmarks demonstrate the much better performance of our framework in comparison with other methods.

3.1 Introduction

Clustering, also known as unsupervised learning, is performed on the basis of proximity or dissimilarity measure. The dissimilarity measure is basically used to quantify the degree of 'closeness' of two data points. The smaller the dissimilarity within a cluster is, the better the clustering results are. There is no need of prior knowledge of data and/or its classes. There are, however, some challenges in clustering algorithms like robustness, sensitivity, outlier detection, accuracy and scalability [5].

The spectral graph wavelet framework was presented for designing of descriptors for local and global geometry of shapes. The cubic spline generating kernel is used for shape retrieval. The SGWS signature is multi-resolution, compact, highly discriminative, and parameter-insensitive in nature [23]. The main problems with the vast majority of clustering algorithms are parameter selection for scaling and noise. To overcome these problems, Hao *et al.* [33]. They combined diffusion maps with spectral clustering and used a Laplace-Beltrami normalization approach instead of graph Laplacian normalization for manifold recovery. On the other hand, quantum mechanics have been applied in the field of anomaly detection in data sets [32] using the Fermi density descriptor.

Inspired by the effectiveness of SGWS in 3D shape retrieval, we propose a data clustering approach based on the SGWS descriptor and the K-means algorithm.

3.1.1 Contributions

The main contributions in this chapter may be summarized as follows:

(i) We present a novel clustering algorithm based on the SGWS descriptor.

(ii) We analyze different related clustering algorithms on different high and low dimensional data sets.

(iii) We perform a comprehensive experimental comparison using various evaluation measures, including AUC score, mean silhouette plot, Calinski-Harabasz method and NMI metric.

The rest of this chapter is organized as follows. In Section 3.2, we provide some background of clustering algorithms and our motivation for this research. Then, we briefly describe different normalization methods. In Section 3.3, we propose our algorithm for data clustering using the SGWS descriptor. Section 3.4 provides an explanation of different data sets and various evaluation measures that are used in experimental results. Extensive experiments on different data sets and comparison with other algorithms are provided in Section 3.5.

3.2 Background and Motivation

There are numerous clustering algorithms proposed in the field of data mining for solving of challenges faced in big data. But every algorithm has its own advantages and disadvantages. Our motivation in this research is to utilize the advantages of existing algorithms and try to lessen the disadvantages of previously implemented algorithms. Next, we provide the brief overview of various normalization methods.

3.2.1 Normalization Methods

There are some normalization methods which have their own impact on clustering results. Normalization is basically a preprocessing task to employ before applying clustering on data sets. It helps equalize or scale the weight of different kinds of attributes. Some attributes may have more variations than others. For example, savings in a bank and age of the customers. Age attribute has less variation than the saving attribute. Age attribute ranges between 30 to 70 but savings in an account can range from 100 dollar to thousands [32, 70].

We used different normalization methods in our work in an effort to further improve the clustering results, as listed below:

1. No-normalization (NN): It is independent on diagonal elements of affinity matrix. It is without the normalization of attributes, directly applied

$$\mathbf{L}_{\mathrm{NN}} = \mathbf{D} - \mathbf{W}, \tag{3.1}$$

where $\mathbf{L}_{\mathrm{NN}}$ is the Laplacian matrix without normalization, $\mathbf{D}$ is the degree matrix and $\mathbf{W}$ is affinity matrix or similarity matrix.

2. Random Walk Normalization (RW): Random walk matrix is a transition matrix which is basically used for computation of long term probability.

$$\mathbf{L}_{\mathrm{RW}} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}, \tag{3.2}$$

where $\mathbf{D}^{-1}$ is the inverse degree matrix having values reciprocal of values in the degree matrix.

3. Symmetric Normalization (SM): Symmetric matrix has stable locality

$$\mathbf{L}_{\mathrm{SM}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}. \tag{3.3}$$

4. Fokker-Plank Normalization (FP): FP and Laplace-Beltrami normalization are basically used for non-uniform entries. FP is defined as

$$\mathbf{L}_{\mathrm{FP}} = \mathbf{I} - \mathbf{D}^{-1}\widetilde{\mathbf{W}}_1, \tag{3.4}$$

where $\widetilde{\mathbf{W}}_1 = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$.

5. Laplace-Beltrami Normalization (LB): This is also a popular kind of normalization which is useful in multidimensional datasets having non-uniform entries.

$$\mathbf{L}_{\text{LB}} = \mathbf{I} - \mathbf{D}^{-1}\widetilde{\mathbf{W}}_2, \tag{3.5}$$

where $\widetilde{\mathbf{W}}_2 = \mathbf{D}^{-1}\mathbf{W}\mathbf{D}^{-1}$.

These different variants of graph Laplacians have their own properties and impact on the clustering results [32, 70].

3.2.2 Spectral Clustering

Spectral clustering is one of the most popular methods for data clustering. This is primarily due to the fact that spectral clustering methods have significant effect in capturing the manifold structure. But they are sensitive to noise and do not provide promising results for high dimensional data sets.

Algorithm 3 Spectral clustering algorithm

```
Input: $\mathcal{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_n\}$ is an $m \times n$ input data and $k$ is number of clusters
Output: $n$-dimensional vector $\mathbf{y}$ containing cluster indices of each data point
Compute the $n \times n$ affinity matrix $\mathbf{W} = \big(w_{ij}\big)$, where
    $w_{ij} = \exp\left(-\frac{||\mathbf{z}_i - \mathbf{z}_j||^2}{2\sigma^2}\right)$    (3.6)
Compute the diagonal matrix $\mathbf{D} = \operatorname{diag}(d_1, \dots, d_n)$ with
    $d_i = \mathbf{W}\mathbf{1} = \sum_{j=1}^{n} w_{ij},\ i = 1, \dots, n$    (3.7)
Compute the graph laplacian using $\mathbf{L}_{\text{NN}}$, $\mathbf{L}_{\text{RW}}$ or $\mathbf{L}_{\text{SM}}$.
Find the first $k$ eigenvectors of graph laplacian and form the eigenvector matrix $\mathbf{E}$ by stacking eigenvectors in columns.
Re-normalize the rows of matrix $\mathbf{E}$.
Perform K-means on the normalized $\mathbf{E}$.
```

The other main problem with spectral clustering is the selection of the scaling parameter. Moreover, few outliers affect the overall performance of spectral clustering [33, 70, 71].

3.2.3 Aggregated Heat Kernel

The strong base of this method is the important properties of the heat kernel. The aggregated heat kernel method is a slight modification of the traditional heat kernel that uses the time scale parameter. This time scale parameter embedded into the heat kernel and integrates the total time from 0 to ∞. The aggregated heat kernel matrix $\mathbf{H} = (h_{ij})$ is defined as

$$h_{ij} = \int_{t=0}^{\infty} h_t(i,j)dt = \sum_{l=1}^{n} \frac{1}{\lambda_l} \varphi_{il}\varphi_{jl}, \tag{3.8}$$

where $h_t(i,j)$ is the heat kernel. For large t, the heat kernel covers the large area around a given data point and for small value of t, coverage of neighborhood of a given data point is also small. So, the value of t affects the results from local to global structure.

Algorithm 4 Aggregated heat kernel algorithm

```
Input: $\mathcal{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_n\}$ is an $m \times n$ input data, $k$ is number of clusters and $\gamma$ is smoothing parameter
Output: $n$-dimensional vector $\mathbf{y}$ containing cluster indices of each data point
1: Compute the $n \times n$ affinity matrix $\mathbf{W} = (w_{ij})$, where
       $$w_{ij} = \exp\left( - \frac{||\mathbf{z}_i - \mathbf{z}_j||^2}{2\sigma^2} \right) \tag{3.9}$$
2: Compute the diagonal matrix $\mathbf{D} = \operatorname{diag}(d_1, \dots, d_n)$ with
       $$d_i = \mathbf{W1} = \sum_{j=1}^{n} w_{ij},\ i = 1, \dots, n \tag{3.10}$$
3: Compute the graph laplacian using $\mathbf{L}_{\mathrm{FP}}$, $\mathbf{L}_{\mathrm{RW}}$ or $\mathbf{L}_{\mathrm{LB}}$.
4: Compute eigenvalues $\lambda_l$ and eigenvectors $\varphi_l$ of $\mathbf{L}$.
5: Construct $\mathbf{H} = (h_{ij})$ matrix, where
       $$h_{ij} = \sum_{l=2}^{n} \frac{1}{r + \lambda_l} \varphi_{il}\varphi_{jl} \tag{3.11}$$
6: Find the first $k$ eigenvectors of $\mathbf{H}$ using normalization method.
7: Re-normalize the rows of eigenvector matrix $\mathbf{E}$.
8: Perform K-means on the normalized $\mathbf{E}$.
```

The heat kernel has its own weaknesses like the best selection of time scale parameter t and sensitivity to noise [33]. We have already discussed FDD in the last chapter in which FDD was applied to anomaly detection [9, 32]. Moreover, we applied FDD to data clustering in our research to perform a comparison with our proposed clustering algorithm. To improve the robustness and accuracy, we propose a new method for data clustering.

3.3 Proposed Approach

In this section, we give a detailed description of our clustering method that makes use of SGWS descriptor. Each data point in the dataset is first represented by a SGWS descriptor. The flow chart of the proposed framework for data clustering is depicted in Figure 3.1. The last stage of the proposed approach is to perform cluster analysis on the signatures using a clustering algorithm (e.g., K-means). The K-means algorithm is arguably one of the most popular and effective clustering methods. In a nutshell, K-means assigns each data point to the cluster having the nearest centroid.

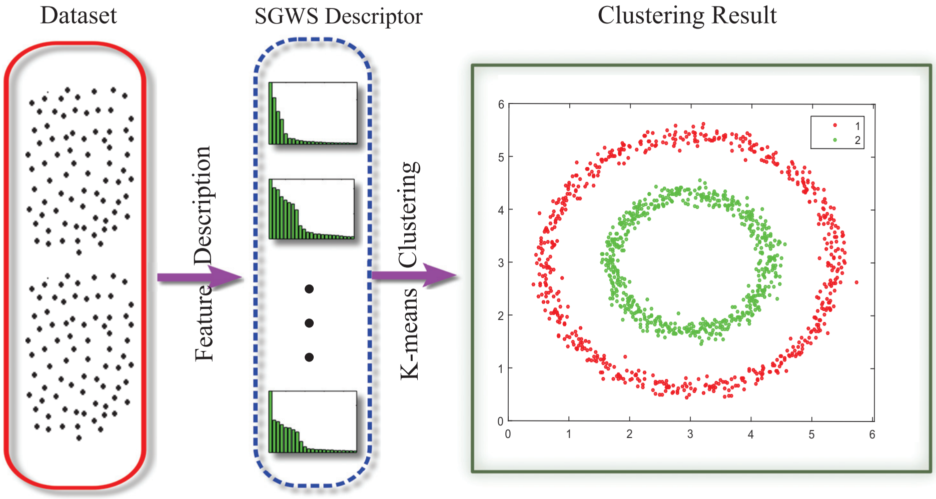


FIGURE 3.1: Flowchart of the proposed approach.

3.3.1 Spectral Graph Wavelet Signature

SGWS was designed for 3D shape retrieval and also for extracting useful geometric features from shapes. Our goal is to employ SGWS in data clustering using different benchmarks. The idea of multi-resolution SGWS comes from the spectral graph wavelet transform (SGWT), which is based on two functions: scaling function and wavelet function [23, 72].

For the wavelet function, the spectral graph coefficients of a function f are given by:

$$W_f(t, \mathbf{z}_j) = \langle \psi_{t,\mathbf{z}_j}, f \rangle = \sum_{\ell=1}^{n} g(t\lambda_\ell)\hat{f}(\ell)\varphi_\ell(\mathbf{z}_j), \tag{3.12}$$

where g be a kernel function, $\psi_{t,\mathbf{z}_j}$ is a version of mother wavelet function at t scale and at $\mathbf{z}_j$ vertex. The spectral graph wavelet is defined as

$$\psi_{t,\mathbf{z}_j}(\mathbf{z}_i) = \sum_{\ell=1}^{n} g(t\lambda_\ell)\varphi_\ell{}^*(\mathbf{z}_j)\varphi_\ell(\mathbf{z}_i), \tag{3.13}$$

and $g(t\lambda_\ell)$ can modulate the graph wavelet $\psi_{t,\mathbf{z}_j}$ but only for λ_ℓ and value of $\lambda_{\max}$ is important to implement practically. The $\lambda_{\max}$ is the upper bound of eigenvalues. For scaling function, the coefficients are

$$S_f(\mathbf{z}_j) = \langle \omega_{\mathbf{z}_j}, f \rangle = \sum_{\ell=1}^{n} h(\lambda_\ell)\hat{f}(\ell)\varphi_\ell(\mathbf{z}_j), \tag{3.14}$$

where $\omega_{\mathbf{z}_j}$ is scaling function. It is given by

$$\omega_{\mathbf{z}_j}(\mathbf{z}_i) = \sum_{\ell=1}^{n} h(\lambda_\ell)\varphi_\ell{}^*(\mathbf{z}_j)\varphi_\ell(\mathbf{z}_i), \tag{3.15}$$

which is used for modulation of wavelets. The kernel g should satisfy $g(0) = 0$ and $\lim_{x\to\infty} g(x) = 0$ as a band-pass filter and the function h should satisfy $h(0) > 0$ and $h(x) \to 0$ as $x \to \infty$ to act as a low-pass filter. The spectral graph wavelet signature formed from the combination of this scaling and graph wavelet where level of resolution (r) helps to modulate whole spectrum. For each point $\mathbf{z}$, SGWS signature is defined as a p-dimensional feature vector after eigendecomposition.

$$\mathcal{S}_R(\mathbf{z}) = \{\mathbf{s}_r(\mathbf{z}) \mid r = 1, \ldots, R\}, \tag{3.16}$$

where R is resolution parameter, r is level of resolution and $\mathbf{s}_r(\mathbf{z})$ is signature which is defined as

$$\mathbf{s}_r(\mathbf{z}) = \{W_{\delta_\mathbf{z}}(t_b, \mathbf{z}) \mid b = 1, \ldots, r\} \cup \{S_{\delta_\mathbf{z}}(\mathbf{z})\}. \tag{3.17}$$

The coefficients in spectral graph wavelet signature are defined as

$$W_{\delta_\mathbf{z}}(t_b, \mathbf{z}) = \sum_{\ell=1}^{n} g(t\lambda_\ell)\varphi_\ell^2(\mathbf{z}) \qquad \text{and} \qquad S_{\delta_\mathbf{z}}(\mathbf{z}) = \sum_{\ell=1}^{n} h(\lambda_\ell)\varphi_\ell^2(\mathbf{z}). \tag{3.18}$$

Moreover, the cubic spline wavelet and scaling function kernels has been chosen from the following equations:

$$g(x) = \begin{cases} x^2 & \text{if } x < 1 \\ -5 + 11x - 6x^2 + x^3 & \text{if } 1 \leq x \leq 2 \\ 4x^{-2} & \text{if } x > 2 \end{cases} \tag{3.19}$$

and

$$h(x) = \gamma \exp\left(-\left(\frac{x}{0.6\lambda_{\min}}\right)^4\right), \tag{3.20}$$

respectively, where $\lambda_{\min} = \lambda_{\max}/20$, γ is set such that $h(0)$ has the same value as the maximum value of g. Both maximum and minimum scales are set to $t_1 = 2/\lambda_{\min}$ and $t_r = 2/\lambda_{\max}$.

SGWS is having a pyramid structure at different resolution levels and the signature consists only two elements at resolution level $r = 1$. One is scaling coefficient $S_{\delta_{\mathbf{z}}}(\mathbf{z})$ and other is wavelet coefficient $W_{\delta_{\mathbf{z}}}(t_1, \mathbf{z})$. On second level, wavelet coefficients of spectral graph are increased means there are two elements for wavelet function, $W_{\delta_{\mathbf{z}}}(t_1, \mathbf{z})$ and $W_{\delta_{\mathbf{z}}}(t_2, \mathbf{z})$. The signature $\mathbf{s}_r(\mathbf{z})$ consists of three elements. Hence, if the resolution is set to $R = 2$, then the signature $\mathcal{S}_R(\mathbf{z})$ is composed of a total of 5 elements. So, basically SGWS is having a resolution level $r = 1$ to R as shown in Figure 3.2 [23, 72].

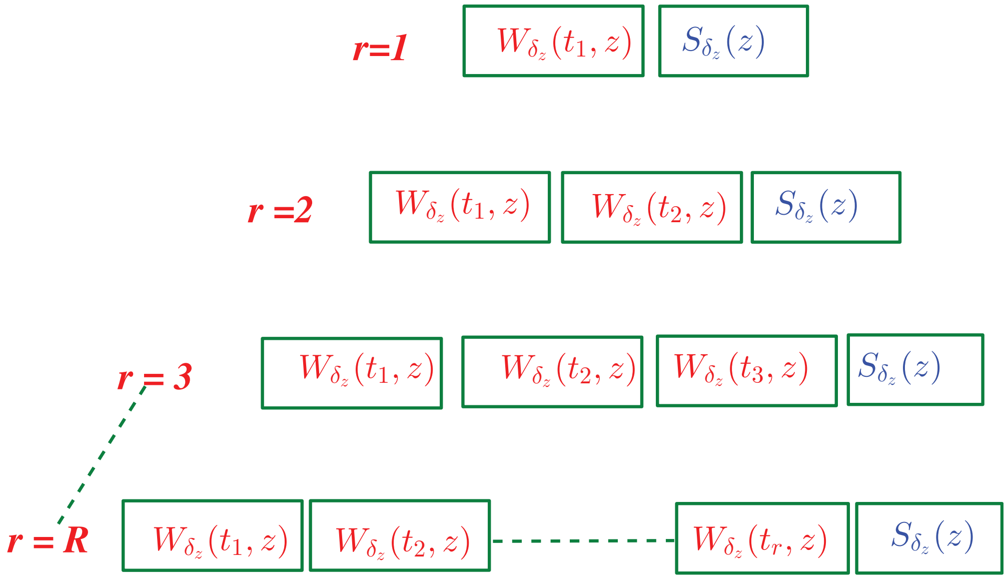


FIGURE 3.2: SGWS at different levels of resolutions.

3.3.2 Proposed Clustering Framework

The SGWS has several interesting properties. First, it has high efficiency of capturing information even with less number of observations. Second, it is of band-pass nature and useful to analyze macro structures which are difficult to capture from different shapes. This descriptor has high discriminative power, which is helpful in comparison different classes of data in a benchmark. Subsequent to investigate these special properties of SGWS, it is introduced as a robust algorithm for clustering of data with different normalization techniques.

The proposed algorithm consists of six main steps. In starting three steps, we calculate affinity matrix, diagonal matrix and graph laplacian using one of normalization technique ($\mathbf{L}_{\text{NN}}$, $\mathbf{L}_{\text{RW}}$, $\mathbf{L}_{\text{SM}}$, $\mathbf{L}_{\text{FP}}$ or $\mathbf{L}_{\text{LB}}$). Next step is to compute generalized eigenvectors and eigenvalues. The fifth step is to represent each data point in a dataset by the SGWS signature, which is a normalized feature vector. More specifically, let $\mathcal{Z}$ be a dataset of n data points with m dimensions, where each data point is represented by a p-dimensional SGWS descriptor $\mathbf{x}_i$, $i = 1, \ldots, n$. The n SGWS descriptors can be arranged in a $p \times n$ data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{p \times n}$. The task in clustering is then to identify clusters of data points and assign each point to one of these clusters.

In the last step, the K-means algorithm is performed on the data matrix $\mathbf{X}$ into K mutually exclusive clusters. To assess the performance of the proposed framework, we used several clustering evaluation measures and indices, which will be discussed in more detail in the next section. The main algorithmic steps of our SGWS approach for data clustering are summarized in Algorithm 5.

This algorithm works very well for the formation of clusters and gives promising results for clustering of data on different benchmarks: Ecoli, Glass, Pima and Yeast. It is required to carefully set the parameters for the greatest outcomes. This descriptor is closely related to HKS and AHK but for data clustering, it yields better performance than both of these descriptors.

Relation to HKS and AHK : Heat Kernel function is based on heat diffusion theory and having properties like symmetric in nature, multi-scale and is highly related to markov chain. In a given time scale, it explains the transferred heat from the source. It is low pass filter and is not able to provide multiresolution on HKS [26, 29]. AHK has also powerful features of heat kernel. It is symmetric and multi-scale but still multiresolution strategy is not possible in AHK [33]. So, SGWS has robust performance and high stability as compared to HKS and AHK .

Algorithm 5 Proposed Algorithm: SGWS Clustering

```
Input: Dataset $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ is an $m \times n$ input data, $k$ is number of clusters, $\sigma$ is scaling parameter
Output: $n$-dimensional vector $\mathbf{y}$ containing cluster indices of each data point
Compute the $n \times n$ affinity matrix $\mathbf{W} = \big(w_{ij}\big)$, where
    $$w_{ij} = \exp\left( - \frac{||\mathbf{z}_i - \mathbf{z}_j||^2}{2\sigma^2} \right) \qquad (3.21)$$
Compute the diagonal matrix $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_n)$ with
    $$d_i = \mathbf{W1} = \sum_{j=1}^{n} w_{ij},\ i = 1, \ldots, n \qquad (3.22)$$
Compute the graph laplacian $\mathbf{L}$ using one of normalization method ($\mathbf{L}_{\text{NN}}$, $\mathbf{L}_{\text{RW}}$, $\mathbf{L}_{\text{SM}}$, $\mathbf{L}_{\text{FP}}$ or $\mathbf{L}_{\text{LB}}$), where
    • $\mathbf{L}_{\text{NN}} = \mathbf{D} - \mathbf{W}$,
    • $\mathbf{L}_{\text{RW}} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$,
    • $\mathbf{L}_{\text{SM}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$,
    • $\mathbf{L}_{\text{FP}} = \mathbf{I} - \mathbf{D}^{-1}\widetilde{\mathbf{W}}_1$ where $\widetilde{\mathbf{W}}_1 = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$,
    • $\mathbf{L}_{\text{LB}} = \mathbf{I} - \mathbf{D}^{-1}\widetilde{\mathbf{W}}_2$ where $\widetilde{\mathbf{W}}_2 = \mathbf{D}^{-1}\mathbf{W}\mathbf{D}^{-1}$.
Compute first $l$ eigenvectors $\varphi_l$ and corresponding eigenvalues $\lambda_l$ of $\mathbf{L}$.
Compute the $p$-dimensional SGWS signature $\mathbf{x}_i$ of each data point using resolution level $r$ and arrange all these signatures in a $p \times n$ data matrix $\mathbf{X}$..
Perform K-means algorithm on $\mathbf{X}$ to find the $n$-dimensional vector $\mathbf{y}$ of cluster indices.
```

3.4 Evaluation Measures

For evaluation of results, we used different measures in our experiments, including NMI, AUC, Calinskiharabasz method and Mean Silhouette Value (MSV).

3.4.1 Data sets

The datasets used for evaluating our approach as listed in Table 3.1: Ecoli dataset which contains the class of localization site of protein. Glass dataset having the information of different types of glasses. Pima dataset is basically pima indian diabetes data set having all female patients with more than 21 years. Yeast data set also based on prediction of cellular localization sites of proteins.

Dataset	Number of instances	Number of attributes
Ecoli	336	8
Glass	214	10
Yeast	1484	8
Pima	768	8

TABLE 3.1: Different datasets used in evaluation.

3.4.2 Mean Silhouette Value

As shown in Figure 3.3, the maximum average silhouette value across all the number of clusters in different datasets is in the range of 0.68 to 1, which may be interpreted as an indication of a strong clustering structure [73].

3.4.3 Calinski-Harabasz Criterion

Two of the key tasks in cluster analysis are to decide on the appropriate number of clusters and on how to notify a good cluster from a bad one. The appropriate number of clusters can be estimated using the Calinski-Harabasz criterion, as depicted in Figure 3.4, which shows that the optimal number of clusters on different data sets using K-means clustering algorithm. This method is also known as Variance Ratio Criterion (VRC) which means cluster evaluation is performed with the calculation of variances within the cluster and between the clusters [3, 74].

3.4.4 Area Under Curve

The area under curve (AUC) is a criterion used for comparison of different clustering algorithms. The Receiver Operating Characterstics (ROC) helps visualize the results and shows the trade-offs between the True Positive Rate (TPR) and False Positive Rate (FPR). The True Positive Rate (TPR), also known as sensitivity or recall, is defined as

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{3.23}$$

and the False Positive Rate (FPR), also known as fall-out, is defined as

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \tag{3.24}$$

where TP = True Positive, FN = False Negative, TN = True Negative and FP = False Positive.

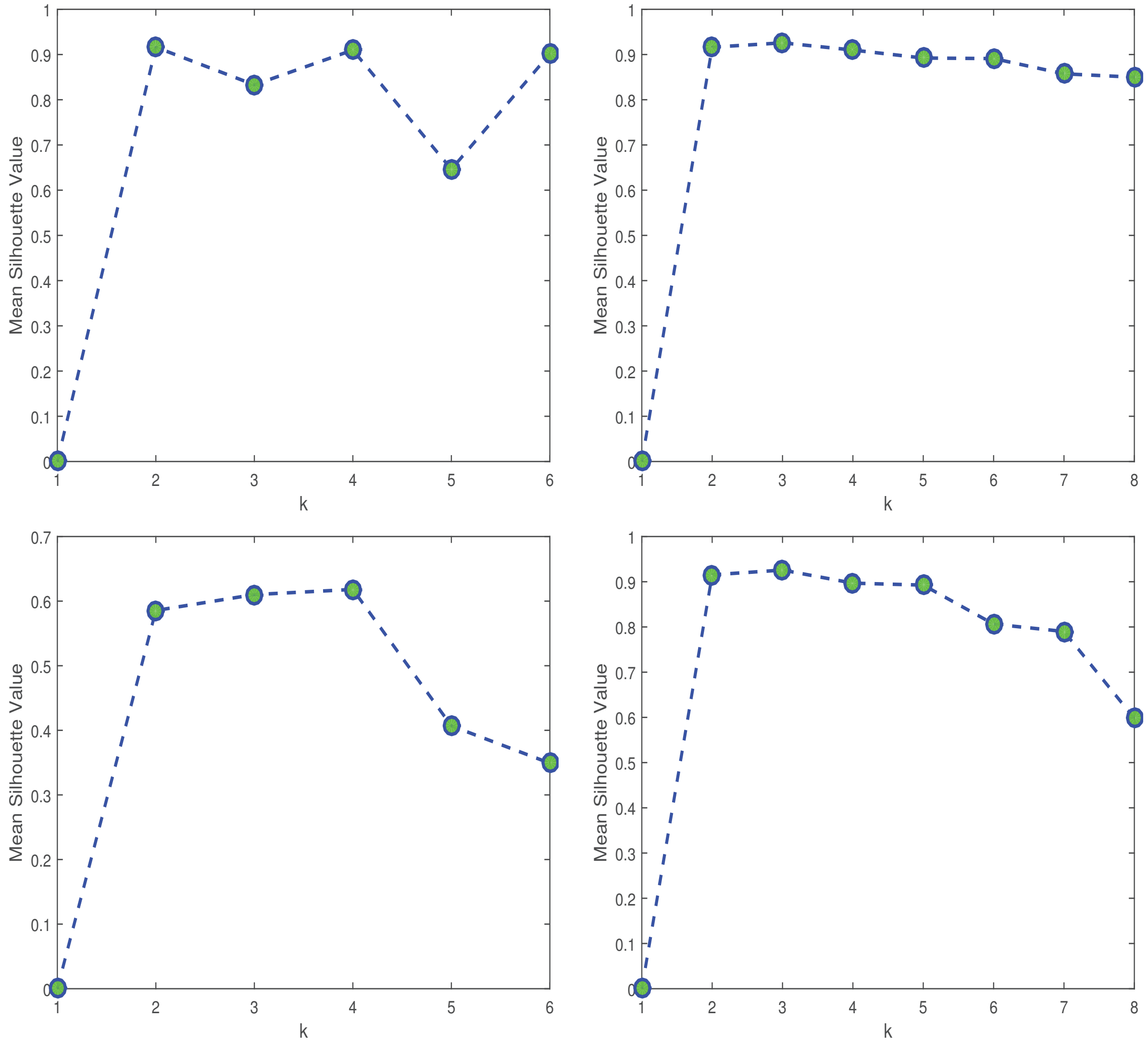


FIGURE 3.3: Mean silhouette value to find the optimal k.

Therefore, AUC considers all possible threshold values and is better for classification and clustering results [75–77].

3.5 Experimental Results

With the outcomes of extensive experiments we compare different descriptors on four benchmarks: Ecoli, Glass, Pima and Yeast. To assess the performance of our proposed algorithm, we conduct extensive experiments for data clustering using NMI and AUC evaluation measures. Results also

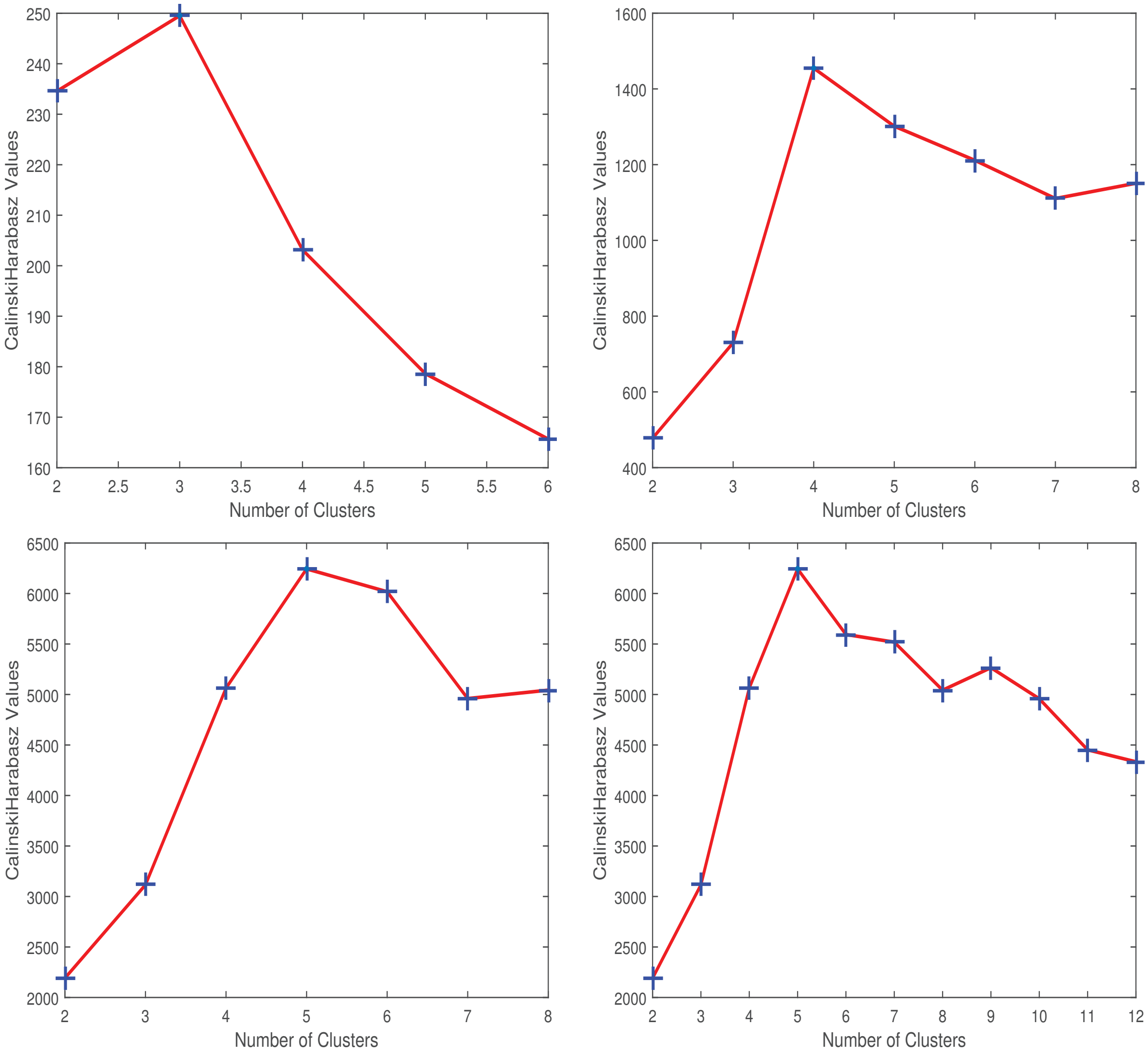
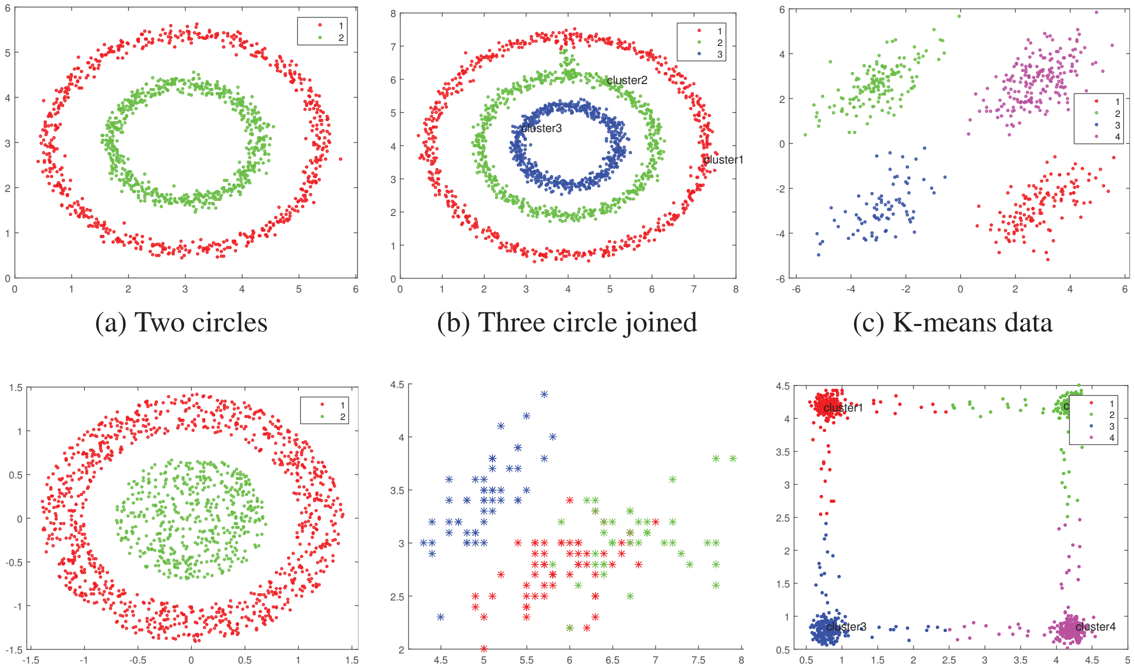


FIGURE 3.4: Calinski-Harabasz results.

show that the proposed algorithm can significantly increase the clustering performance in comparison of related spectral descriptors. Moreover, we start our experiment with small datasets: twocircles, threecirclesjoined, fourclouds, fisheriris etc., then we applied the important findings on large datasets: Ecoli, Glass, Pima and Yeast. In our experimental results, we closely follow the theoretical original works and select the parameters for the optimal performance of our proposed algorithm.



(a) Two circles (b) Three circle joined (c) K-means data

(d) K-means online data (e) Fisheriris (f) Four cloud

FIGURE 3.5: Clustering results on small datasets.

3.5.1 Settings

Comparing Signatures and Methods: There are some classical signatures, including WKS, HKS, GPS, SIHKS and FDD. We have selected these signatures because of their high popularity in recent years. Moreover, we have performed our comparison with popular clustering methods such as spectral clustering and AHK.

Complexity: For implementation of our algorithms, we selected MATLAB 8.4.0 (R2014b) installed on 64 bit operating system with an Intel Core i7-4510U running at 2.60 GHz and 8 GB RAM. Initially, we have performed our experiment on small data sets like fisheriris, two circles, three circles joined etc. then on large data sets. Experiments on large data sets with different evaluation measures are listed in Section 3.4. Even for large data sets, proposed algorithm has shown significant results with effectiveness and efficiency (which is about 30 sec) of the algorithm.

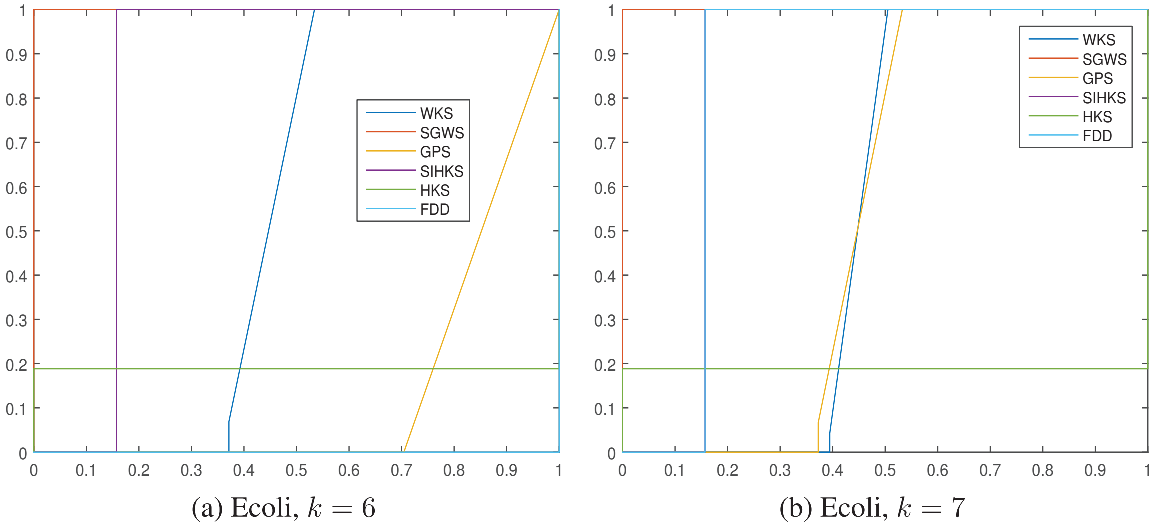
Parameter Selection: For fair comparison, we have chosen the best parameters for aforementioned signatures. For evaluation of results, we need to set the other parameters like number of clusters (k), number of neighbors (k_n) and number of eigenvectors (l) for the starting computation before spectral descriptors in every algorithm. The optimal parameters settings of different algorithms in our experiments are shown in the Table 3.2.

WKS	GPS	HKS	SIHKS	FDD	SGWS	AHK
$N = 100$ $\sigma = 0.05$	$l = 14$	$t_0 = 0.01$ $\alpha = 2$ $T = 15$ $\tau = 1/4$	$F = 191$ $T = 15$ $\tau = 1/16$ $\alpha = 2$	$\sigma = 1$ $T = 1000$	$l = 10$ $k_n = 15$	$\gamma = 0.02$ $l = 10$

TABLE 3.2: Optimal parameter selection for different descriptors.

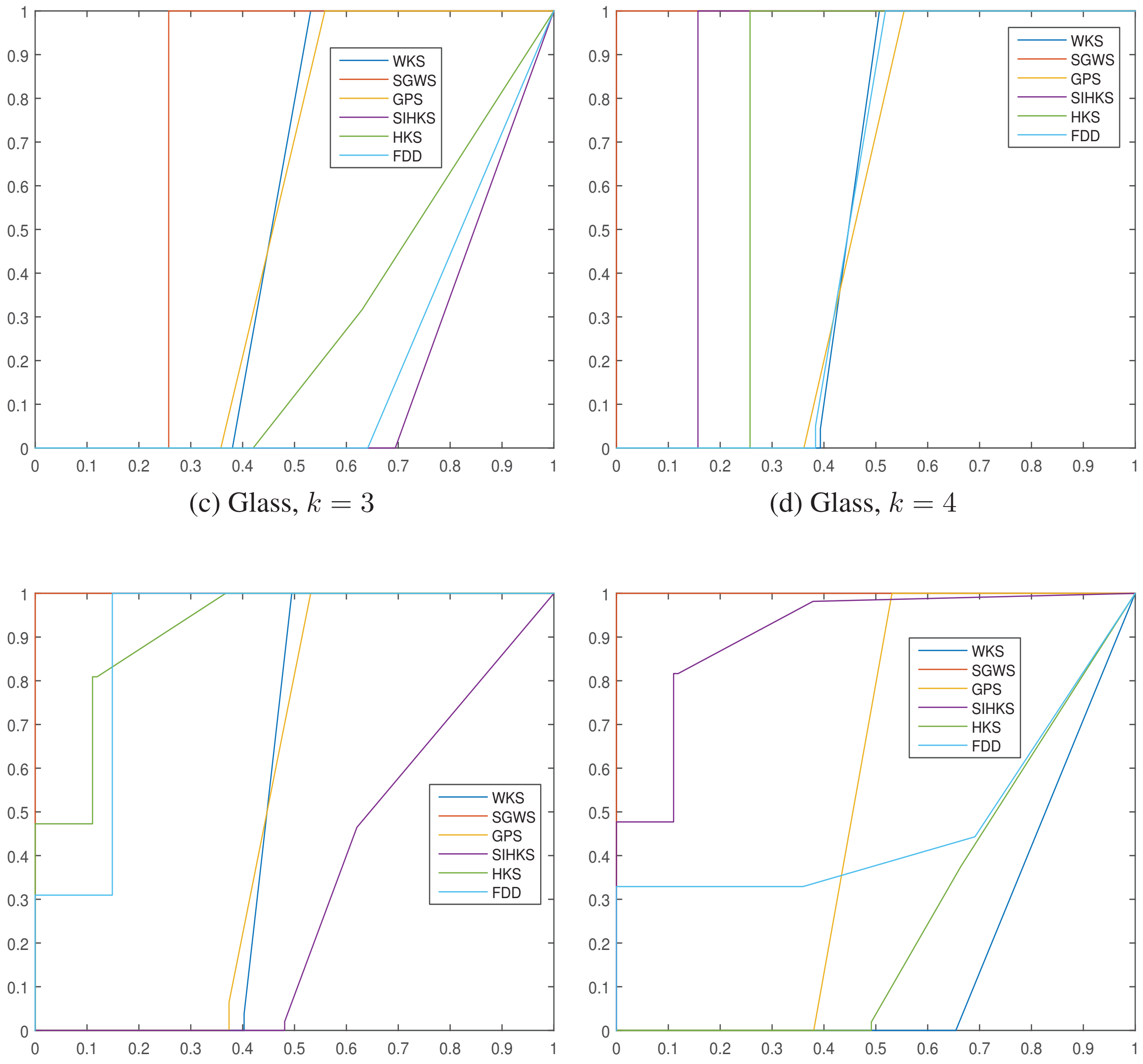
3.5.2 Comparison of average performance

We evaluate our proposed SGWS algorithm for data clustering as well as other popular descriptors WKS, HKS, SIHKS, GPS, AHK and FDD with the different data sets. To gain further insight into performance variation between different descriptors, the AUC and NMI results are shown for different parameters. For our new algorithm, ecoli data set shows best result for $k = [6, 7]$, glass data set for $k = [3, 4]$, yeast for $k = [5, 8]$ and pima for $k = [5, 6]$. Figure 3.6 and Figure 3.7 explains the best AUC and NMI results of each method corresponding to different number of clusters (k) in different benchmarks.



(a) Ecoli, $k = 6$

(b) Ecoli, $k = 7$



(c) Glass, $k = 3$ (d) Glass, $k = 4$

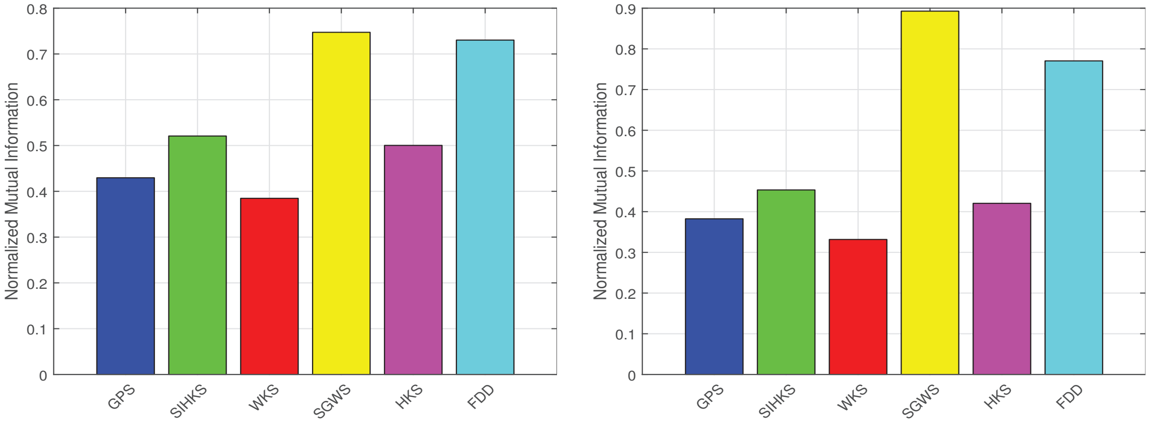
(e) Yeast, k=8 (f) Pima, k=5

FIGURE 3.6: (a)-(b) AUC value for Ecoli dataset for k = 6 and 7. (c)-(d) AUC value for Glass dataset for k = 3 and 4. (e) AUC value for Yeast dataset for k = 8. (f) AUC value for Pima dataset for k = 5.

In Table 3.3 and Table 3.4, we document the close results of our new algorithm in a theoretical view and found that our proposed algorithm has higher AUC and NMI results. Even, we found that other algorithms has scaling parameters like AHK and FDD, these algorithms shows almost opposite results with change in value of scaling parameters. But our proposed work does not have this kind of dependency of scaling parameters which affects overall performance of the algorithm. WKS shows the least performance with normalized mutual information measure. FDD has good performance but

	Spectral Descriptors						
Dataset	WKS	GPS	HKS	SIHKS	FDD	SGWS	AHK
Ecoli, k=6	0.432	0.235	0.232	0.720	0.153	**0.889**	0.509
Ecoli, k=7	0.462	0.457	0.254	0.123	0.892	**0.998**	0.552
Glass, k=3	0.465	0.535	0.307	0.257	0.253	**0.853**	0.452
Glass, k=4	0.432	0.505	0.642	0.749	0.653	**0.983**	0.524
Yeast, k=8	0.443	0.503	0.763	0.355	0.837	**0.975**	0.637
Pima, k=5	0.220	0.524	0.267	0.841	0.447	**0.867**	0.556

TABLE 3.3: Average results with different algorithms.



(a) Ecoli, k = 6

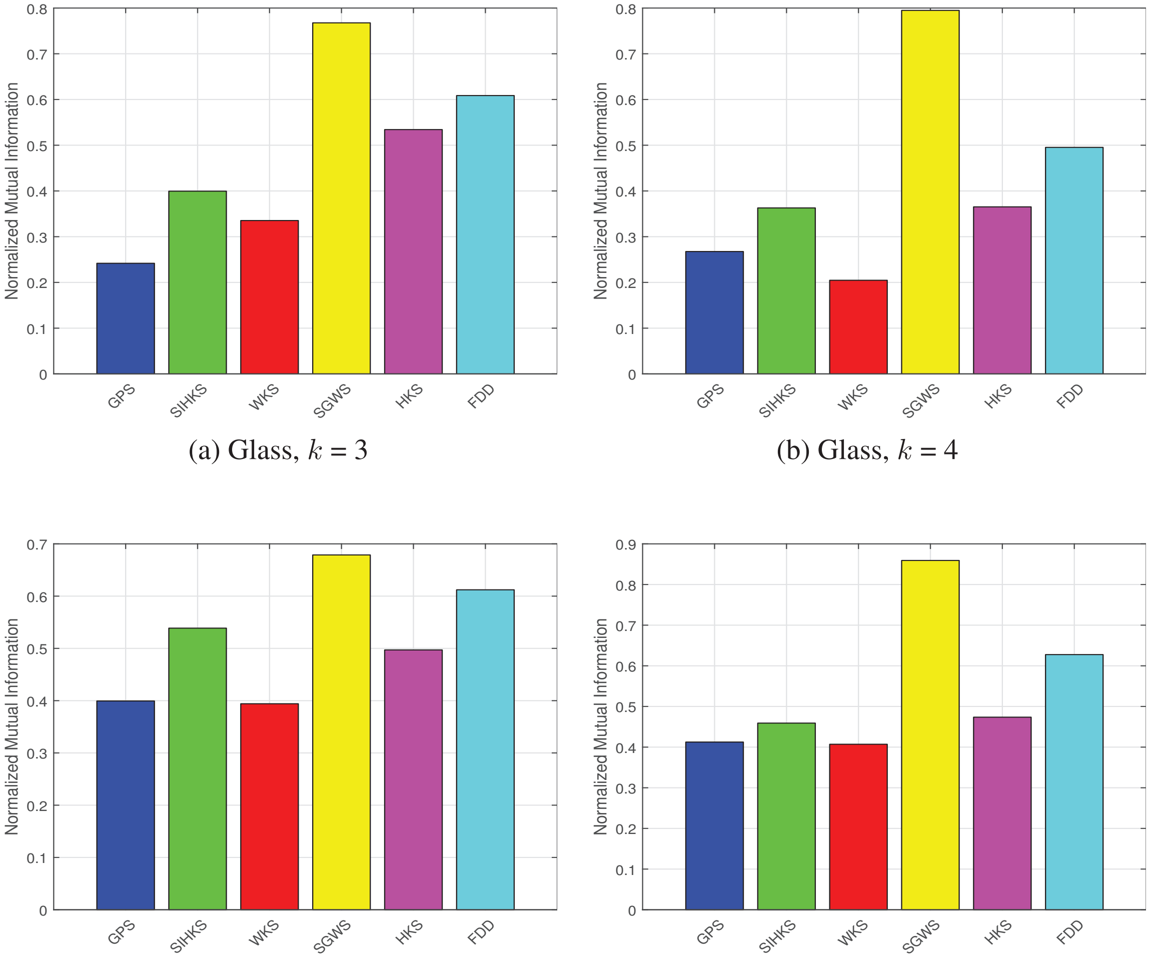
(b) Ecoli, k = 7

FIGURE 3.7: (a)-(b) Normalized Mutual Information on Ecoli dataset.

	Spectral Descriptors						
Dataset	WKS	GPS	SIHKS	HKS	FDD	SGWS	AHK
Ecoli, k=6	0.393	0.434	0.520	0.521	0.733	**0.827**	0.690
Ecoli, k=7	0.330	0.450	0.454	0.523	0.788	**0.898**	0.728
Glass, k=3	0.351	0.335	0.561	0.492	0.653	**0.808**	0.633
Glass, k=4	0.211	0.305	0.422	0.489	0.643	**0.829**	0.714
Yeast, k=5	0.410	0.403	0.501	0.480	0.737	**0.781**	0.687
Yeast, k=8	0.421	0.392	0.516	0.492	0.629	**0.805**	0.637
Pima, k=5	0.390	0.384	0.487	0.481	0.630	**0.898**	0.690

TABLE 3.4: Average results using NMI on different data sets for different k.

it does not have stable results. It has fluctuating results with change in data set and with different parameters. SIHKS has drastic change in result with pima, $k = 5$ (AUC 0.841).



(a) Glass, k = 3

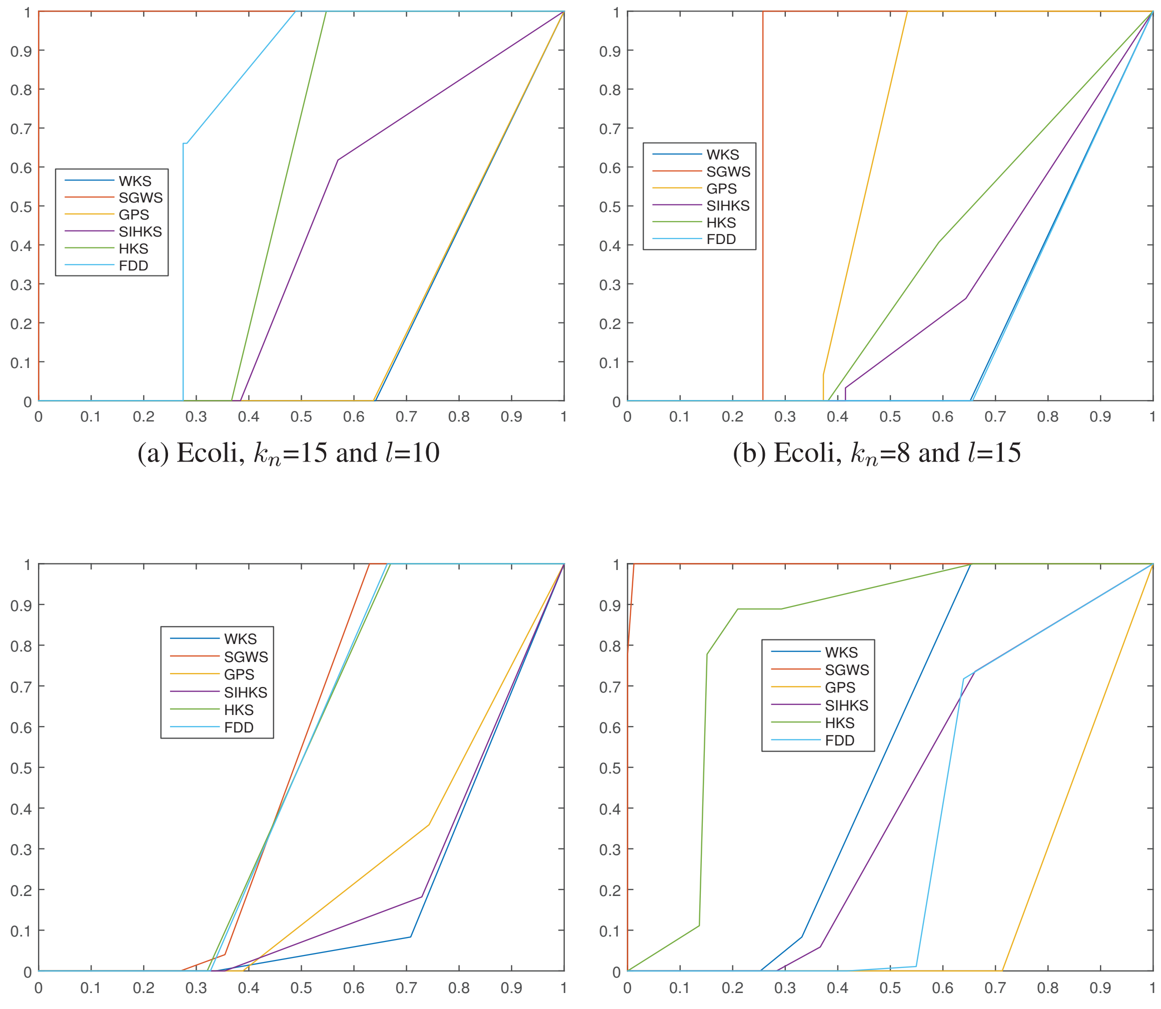
(b) Glass, k = 4

(c) Yeast, k = 8

(d) Pima, k = 5

FIGURE 3.8: (a)-(b) Normalized Mutual Information on Glass dataset. (c) Normalized Mutual Information on Yeast dataset. (d) Normalized Mutual Information on Pima dataset.

The parameter determination helps to find the appropriate results and proper discrimination of data from different classes in a data set. The other crucial parameters are number of nearest neighbors (k_n) and number of eigenvectors (l). To study the influence of k_n and l, we changed value of k_n from 10 to 100 and number of eigenvectors from 10 to 50. Experimentally, we find that the better performance is obtained when value of k_n is between $[5, 7, 10, 15, 17]$ for different data sets and even with different number of clusters k and eigenvectors l. However, k_n = 15 and l = 10 yields best performance with the proposed algorithm on ecoli dataset (AUC .993 and NMI .798) and FDD gives the second highest results but WKS shows least performance in terms of AUC (.146) with these parameters. With increase in number of l parameters, all other algorithms except SGWS show their worst performance. For k_n = 8 and l = 15, WKS has approximately doubled performance but FDD works worst with this value. Even, with these parameter values, we achieved good performance with glass data set either. On yeast data set, we changed values of k_n = 10 and l = 10 and for pima data set, k_n = 8 and l = 12 to get better performance of different algorithms. For glass and pima data sets, GPS gives bad performance but WKS works fine. Therefore, Figures 3.10 and 3.9 shows best results of our method using AUC score. Our proposed algorithm is very hands-on and effective on many data sets.



(a) Ecoli, k_n=15 and l=10

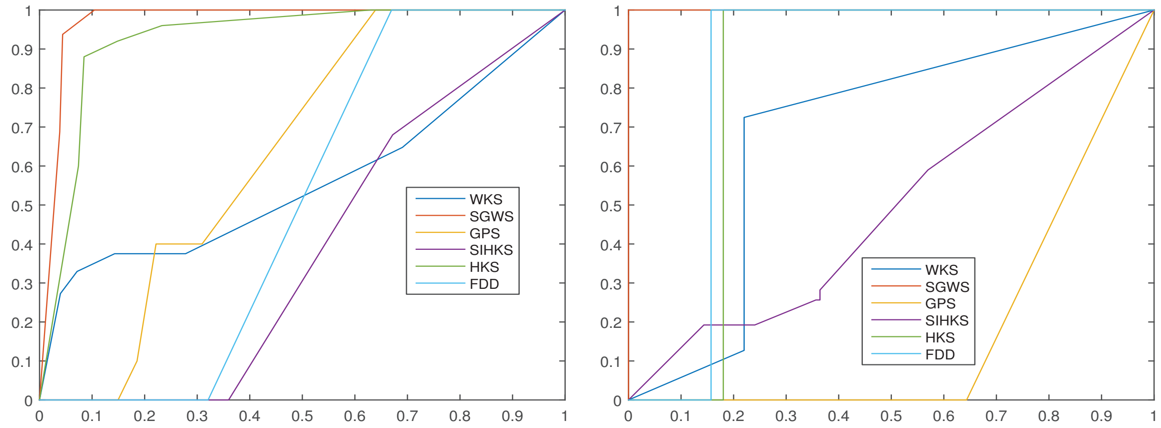
(b) Ecoli, k_n=8 and l=15

(c) Glass, k_n=15 and l=10

(d) Glass, k_n=8 and l=15

FIGURE 3.9: (a)-(b)AUC value for Ecoli dataset for different k_n and l. (c)-(d)AUC value for Glass dataset for different k_n and l.

Table 3.5 and Table 3.6 documents the results in terms of AUC and NMI using different number



(a) Yeast, k_n=10 and l=10

(b) Pima, k_n=8 and l=12

FIGURE 3.10: (a) AUC value for Yeast dataset for different k_n and l. (b) AUC value for Pima dataset for different k_n and l.

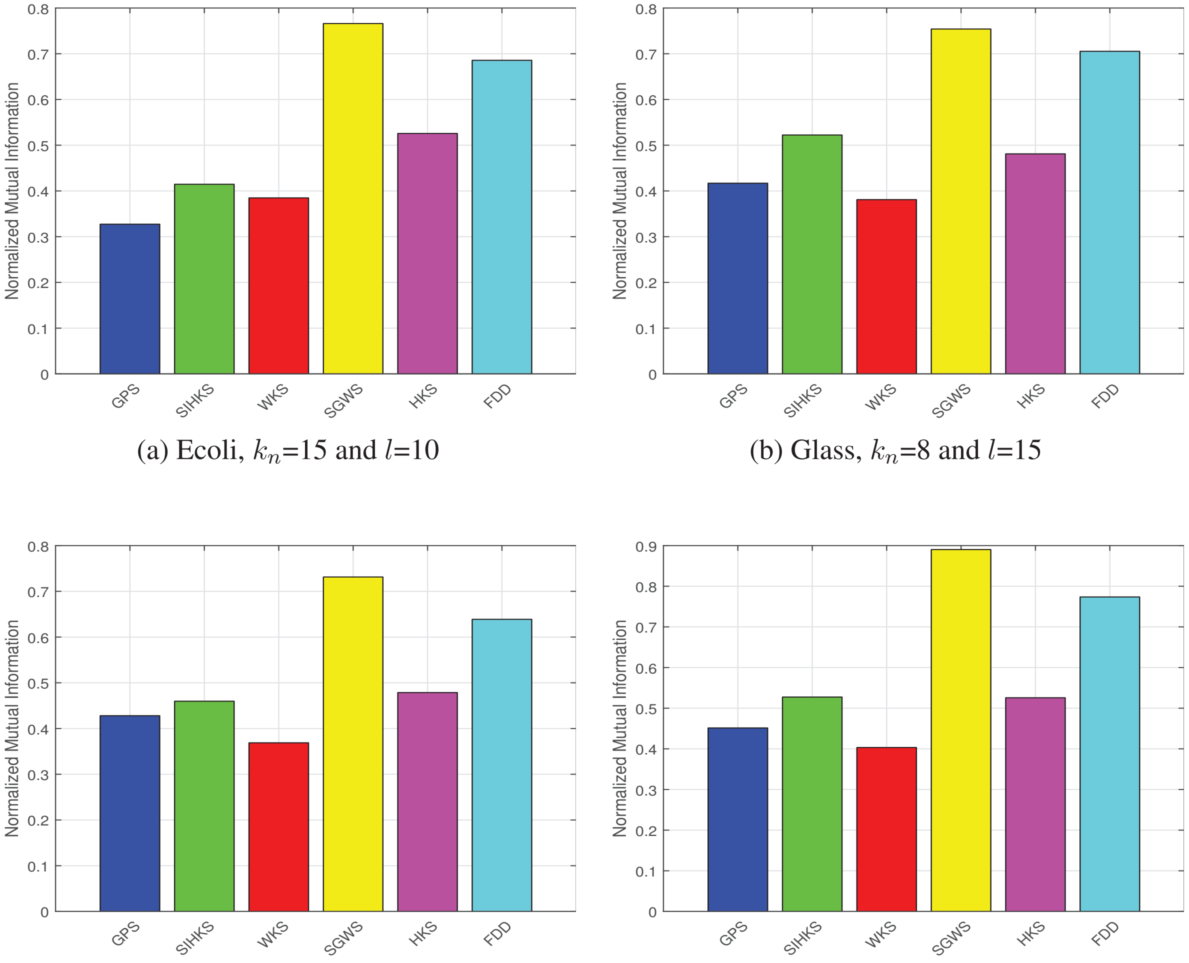
of eigenvectors and number of neighbors. Figure 3.11 shows best results of our proposed method in terms of NMI.

	Spectral Descriptors						
Dataset	WKS	GPS	HKS	SIHKS	FDD	SGWS	AHK
Ecoli, k_n=15, l=10	0.146	0.264	0.531	0.410	0.840	**0.993**	0.520
Ecoli, k_n=8, l=15	0.275	0.553	0.353	0.323	0.256	**0.801**	0.350
Glass, k_n=15, l=10	0.485	0.236	0.507	0.294	0.470	**0.753**	0.454
Glass, k_n=8, l=15	0.342	0.244	0.846	0.398	0.373	**0.970**	0.729
Yeast, k_n=10, l=10	0.581	0.505	0.868	0.367	0.510	**0.913**	0.534
Pima, k_n=8, l=12	0.590	0.253	0.760	0.410	0.860	**0.977**	0.650

TABLE 3.5: Average results with k_n and l parameters.

	Spectral Descriptors						
Dataset	WKS	GPS	HKS	SIHKS	FDD	SGWS	AHK
Ecoli, k_n=15, l=10	0.381	0.340	0.543	0.436	0.743	**0.798**	0.660
Ecoli, k_n=8, l=15	0.350	0.425	0.498	0.569	0.756	**0.760**	0.380
Glass, k_n=15, l=10	0.523	0.435	0.431	0.537	0.686	**0.709**	0.390
Glass, k_n=8, l=15	0.334	0.443	0.447	0.588	0.735	**0.780**	0.624
Yeast, k_n=10, l=10	0.380	0.439	0.482	0.493	0.684	**0.830**	0.338
Pima, k_n=8, l=12	0.432	0.489	0.557	0.531	0.790	**0.910**	0.590

TABLE 3.6: Average results with k_n and l parameters using NMI.



(a) Ecoli, k_n=15 and l=10

(b) Glass, k_n=8 and l=15

(c) Yeast, k_n=10 and l=10

(d) Pima, k_n=8 and l=12

FIGURE 3.11: (a) Normalized Mutual Information on Ecoli dataset for different l and k_n. (b) Normalized Mutual Information on Glass dataset for different l and k_n. (c) Normalized Mutual Information on Yeast dataset for different l and k_n. (d) Normalized Mutual Information on Pima dataset for different l and k_n.

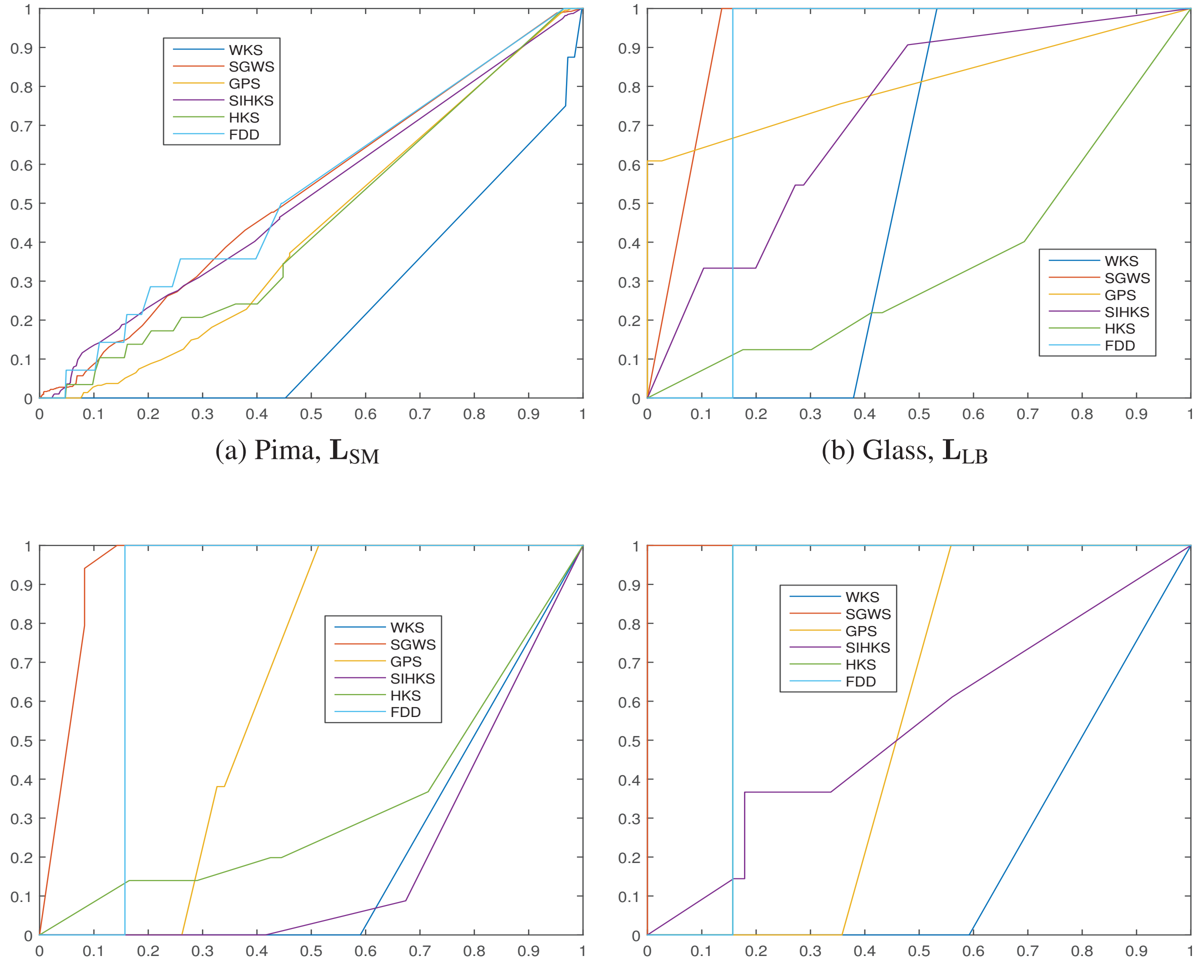
To systematically demonstrate the performance with different number of neighbors, we found that the evaluation results are not so good with number of corresponding neighbors more than 20.

To guarantee not to favour any method, we tested with different parameters for each algorithm in our experiments and shows the finest results with every method. SIHKS and HKS performs almost same in terms of NMI evaluation measure because of their same multiscale nature. We found that WKS shows least performance because it does not represent large scale dimensions well. GPS also have average results because of its global nature but it does not have the best performance as our proposed algorithm.

To summarize, SGWS algorithm has most robust performance in terms of both evaluation measures and simple parameters selection. Next, we examine the behaviour of our algorithm with different normalization methods. We have already introduced our selection of laplacians but now we analyze the purpose of different laplacians on different datasets. Experimentally, we demonstrate the results using AUC and NMI with different normalization techniques.

To examine the influence of different normalization techniques, we set the other parameters with their best results for each data set and the proposed algorithm outperforms in comparison of state-of-the-art methods. The smaller datasets are usually more sensitive to scaling parameters. That is the reason, we chose small datasets also for our experimental evaluations.

Compared to other approaches, the SGWS algorithm shows best stability in random walk normalization and no normalization, as demonstrated in both NMI and AUC score. This property of proposed algorithm is derived from wavelet and scaling function of SGWS. FDD also shows high quality performance due to inherent properties from quantum mechanics. We test with different laplacians $\mathbf{L}_{\text{NN}}, \mathbf{L}_{\text{RW}}, \mathbf{L}_{\text{SM}}, \mathbf{L}_{\text{FP}}$ and $\mathbf{L}_{\text{LB}}$ to preserve the density differences between data points. SGWS algorithm has the most robust performance on parameter-tuning which is extremely important for clustering the results and reliable analysis of different data sets. Ecoli and yeast data set shows the best performance with laplace beltrami and no normalization but glass and pima data sets has better performance with random walk normalization with other parameters tuning. WKS shows the least performance with different laplacians also which is consistent with the results with other parameters k, l and k_n. HKS shows best results with laplace beltrami on pima data set but it gives least results with other data sets. Surprisingly, GPS gives good results with glass data set using $\mathbf{L}_{\text{LB}}$. FDD works well on ecoli data set with $\mathbf{L}_{\text{RW}}$ and second highest with $\mathbf{L}_{\text{NN}}$ also but gives worst performance with other laplacians such as $\mathbf{L}_{\text{NN}}$ on yeast dataset.



(a) Pima, $\mathbf{L}_{\mathrm{SM}}$ (b) Glass, $\mathbf{L}_{\mathrm{LB}}$

(c) Ecoli, $\mathbf{L}_{\mathrm{RW}}$ (d) Ecoli, $\mathbf{L}_{\mathrm{NN}}$

FIGURE 3.12: Comparison of algorithms with different Normalization techniques.

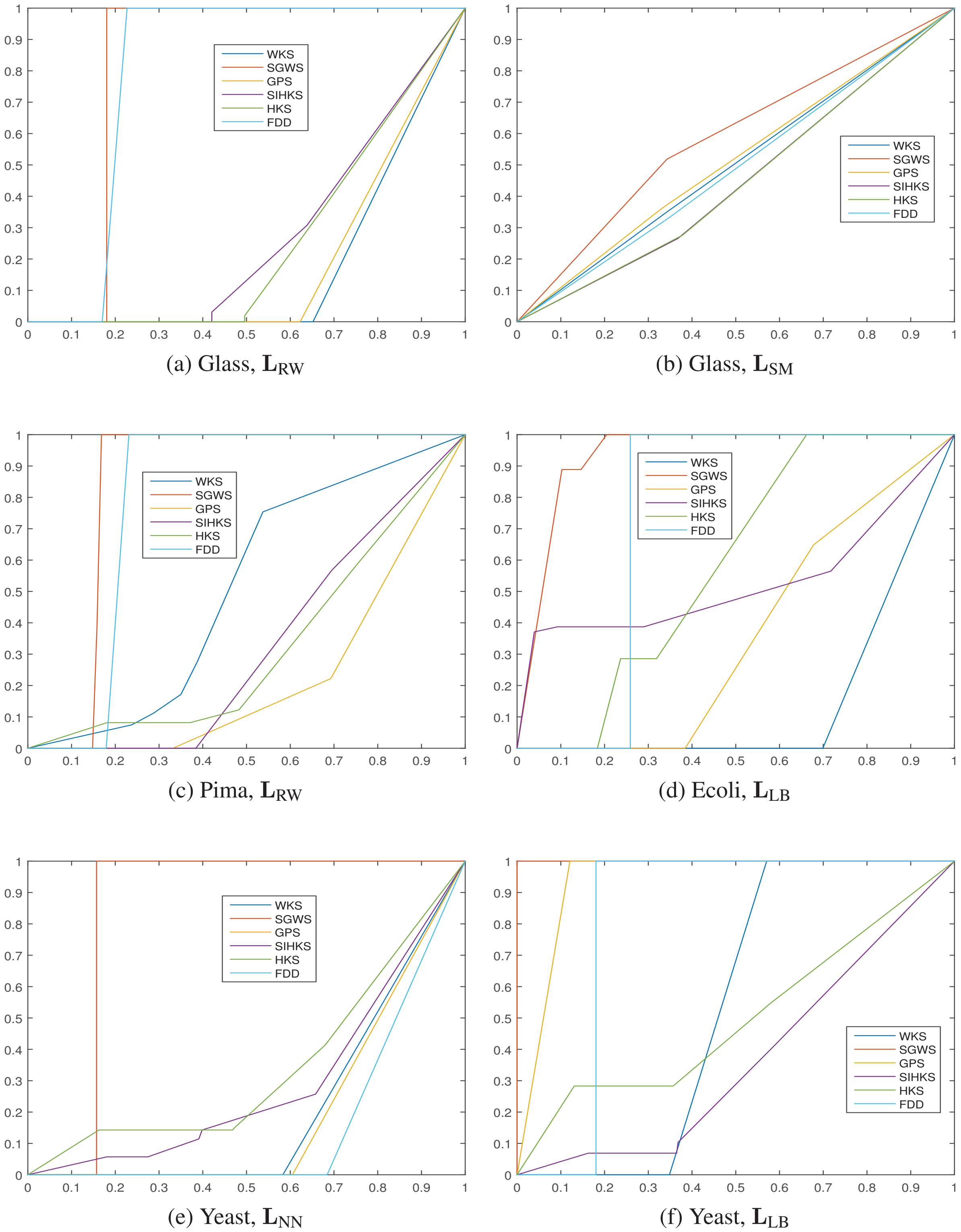
Hence, different Laplacians $\mathbf{L}_{\mathrm{NN}}, \mathbf{L}_{\mathrm{RW}}, \mathbf{L}_{\mathrm{SM}}, \mathbf{L}_{\mathrm{FP}}$ and $\mathbf{L}_{\mathrm{LB}}$ play an essential role in the design of our proposed algorithm. Figure 3.15 and Figure 3.14 show that our algorithm yields the best results using normalized mutual information measure with different normalized Laplacians, as summarized in Table 3.7 and Table 3.8. Moreover, comparison of the proposed algorithm with various other state-of-the-art methods such as Bose-Einstein (BE) distribution, Maxwell-Boltzmann (MB) distribution and spectral clustering (SC) are shown in Figure 3.16.

For some data sets and sensitive parameter tuning, FDD also works well but our method can maintain similar performance due largely to its insensitivity to the scaling parameters. The major advantages of our proposed algorithm are stability and robustness. The other important property is reliability

TABLE 3.7: Average results with different normalization techniques using AUC. Boldface numbers indicate the best clustering performance.

		Spectral descriptors						
Dataset	Normalization method	WKS	GPS	HKS	SIHKS	FDD	SGWS	AHK
Ecoli	$\mathbf{L}_{\mathrm{NN}}$	0.329	0.504	0.101	0.554	0.802	**0.959**	0.253
Ecoli	$\mathbf{L}_{\mathrm{LB}}$	0.334	0.423	0.524	0.566	0.760	**0.947**	0.450
Ecoli	$\mathbf{L}_{\mathrm{RW}}$	0.318	0.524	0.456	0.280	0.802	**0.903**	0.555
Ecoli	$\mathbf{L}_{\mathrm{SM}}$	0.284	0.386	0.436	0.301	0.810	**0.890**	0.205
Ecoli	$\mathbf{L}_{\mathrm{FP}}$	0.276	0.360	0.458	0.250	0.630	**0.767**	0.440
Glass	$\mathbf{L}_{\mathrm{NN}}$	0.442	0.452	0.040	0.402	**0.651**	0.650	0.3524
Glass	$\mathbf{L}_{\mathrm{LB}}$	0.496	0.806	0.421	0.714	0.820	**0.905**	0.420
Glass	$\mathbf{L}_{\mathrm{RW}}$	0.319	0.320	0.411	0.423	0.832	**0.841**	0.467
Glass	$\mathbf{L}_{\mathrm{SM}}$	0.583	0.525	0.452	0.519	0.525	**0.654**	0.346
Glass	$\mathbf{L}_{\mathrm{FP}}$	0.374	0.320	0.324	0.340	0.530	**0.587**	0.283
Yeast	$\mathbf{L}_{\mathrm{NN}}$	0.270	0.418	0.440	0.489	0.456	**0.602**	0.235
Yeast	$\mathbf{L}_{\mathrm{LB}}$	0.513	0.320	0.450	0.670	0.451	**0.671**	0.227
Yeast	$\mathbf{L}_{\mathrm{RW}}$	0.267	0.280	0.262	0.671	**0.702**	**0.702**	0.350
Yeast	$\mathbf{L}_{\mathrm{SM}}$	0.356	0.405	0.355	0.551	0.701	**0.808**	0.152
Yeast	$\mathbf{L}_{\mathrm{FP}}$	0.435	0.503	0.325	0.255	0.501	**0.606**	0.504
Pima	$\mathbf{L}_{\mathrm{NN}}$	0.350	0.339	0.387	0.396	0.701	**0.721**	0.402
Pima	$\mathbf{L}_{\mathrm{LB}}$	0.461	0.421	0.442	0.350	0.650	**0.758**	0.421
Pima	$\mathbf{L}_{\mathrm{RW}}$	0.550	0.321	0.367	0.391	0.760	**0.821**	0.331
Pima	$\mathbf{L}_{\mathrm{SM}}$	0.422	0.490	0.488	0.510	0.526	**0.523**	0.436
Pima	$\mathbf{L}_{\mathrm{FP}}$	0.363	0.320	0.352	0.340	0.452	**0.608**	0.360

in data analysis, and also simplicity in the sense that no prior knowledge is needed to use our algorithm for clustering. In comparison with other approaches, extensive experimental results and evaluations have demonstrated stable and robust performance of the proposed SGWS algorithm.



(a) Glass, $\mathbf{L}_{\mathrm{RW}}$

(b) Glass, $\mathbf{L}_{\mathrm{SM}}$

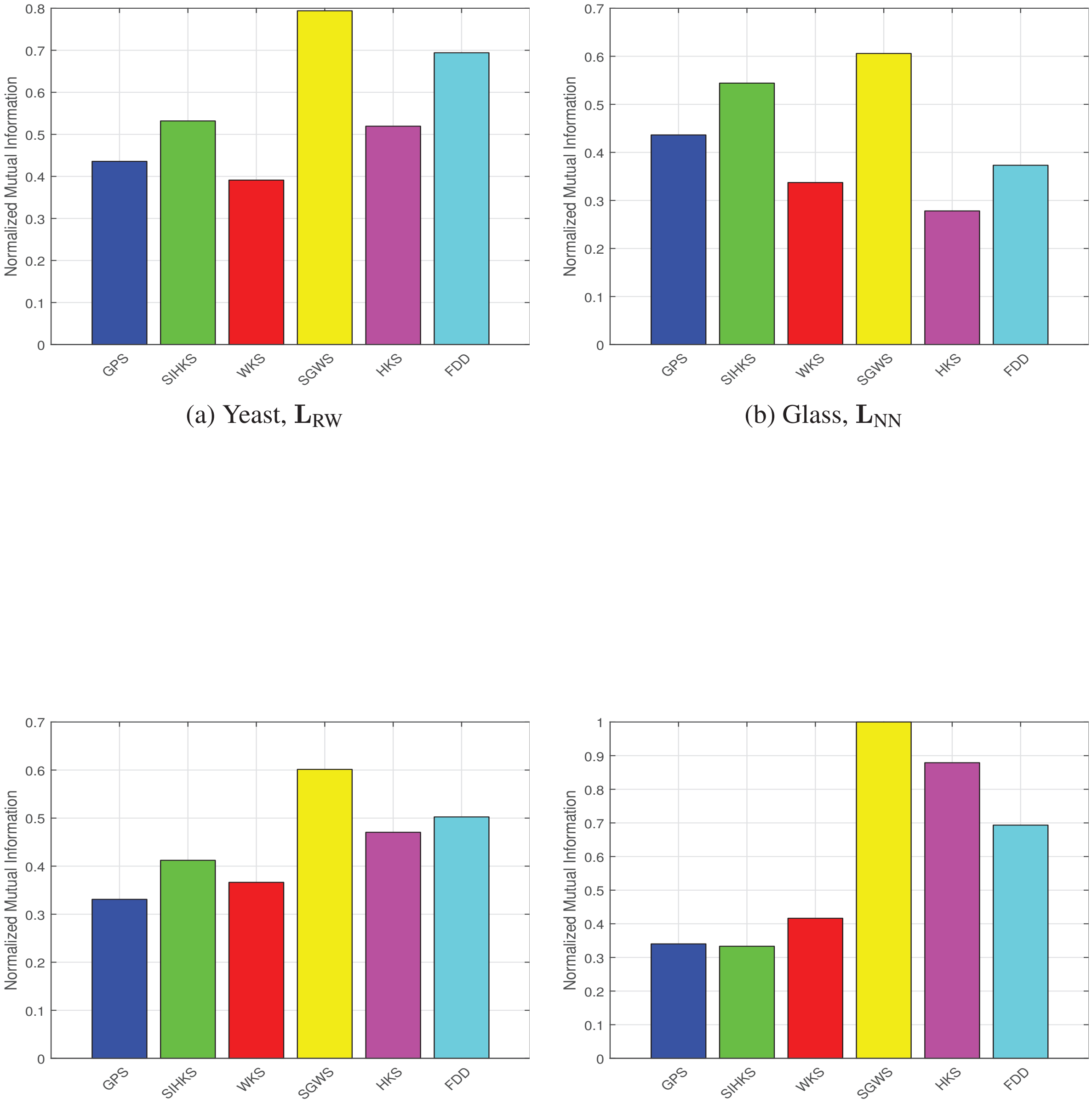
(c) Pima, $\mathbf{L}_{\mathrm{RW}}$

(d) Ecoli, $\mathbf{L}_{\mathrm{LB}}$

(e) Yeast, $\mathbf{L}_{\mathrm{NN}}$

(f) Yeast, $\mathbf{L}_{\mathrm{LB}}$

FIGURE 3.13: ROC with different Normalization techniques.



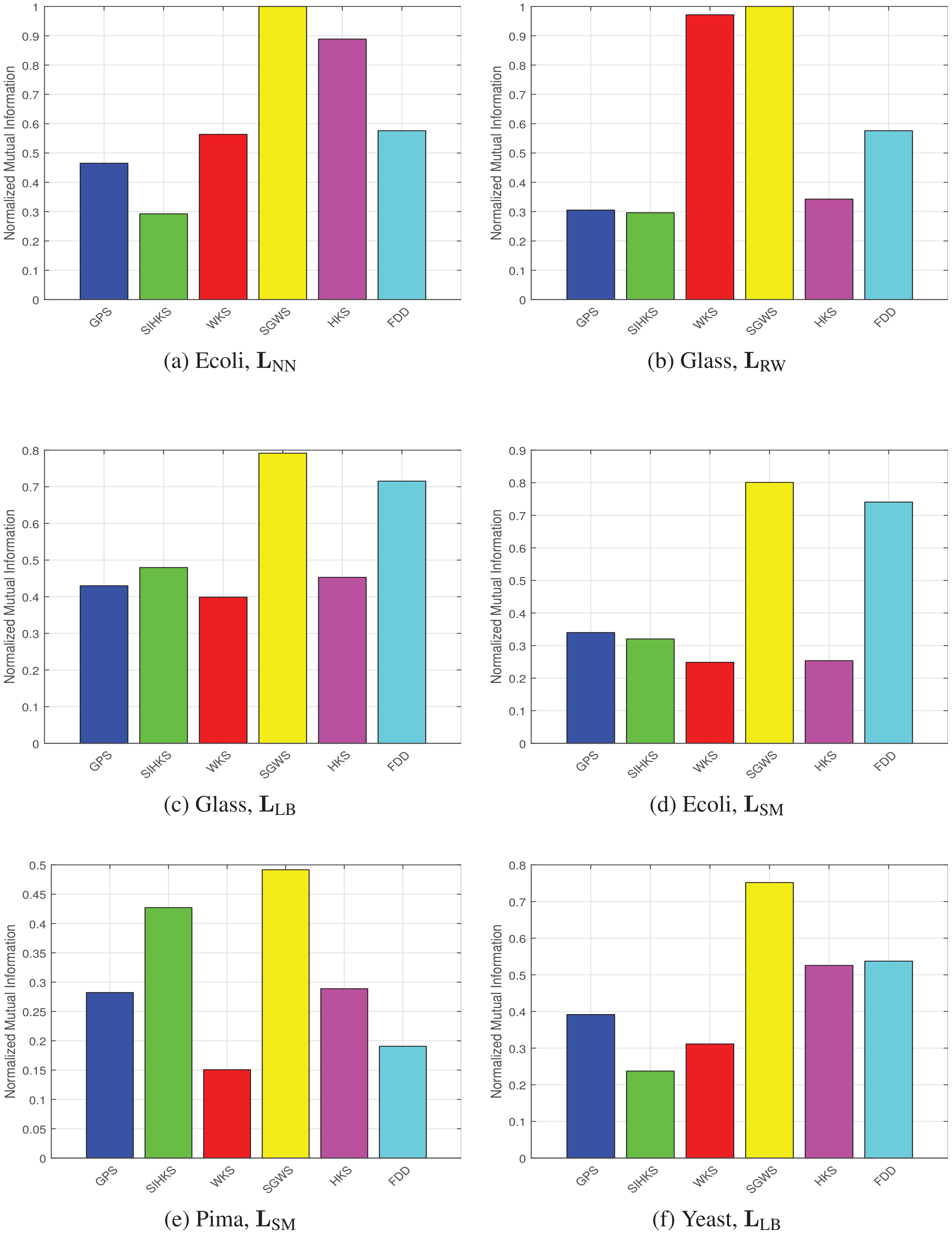
(a) Yeast, $\mathbf{L}_{\mathrm{RW}}$

(b) Glass, $\mathbf{L}_{\mathrm{NN}}$

(c) Ecoli, $\mathbf{L}_{\mathrm{RW}}$

(d) Pima, $\mathbf{L}_{\mathrm{RW}}$

FIGURE 3.14: Comparison of algorithms with different Normalization techniques using NMI.



(a) Ecoli, $\mathbf{L}_{\mathrm{NN}}$

(b) Glass, $\mathbf{L}_{\mathrm{RW}}$

(c) Glass, $\mathbf{L}_{\mathrm{LB}}$

(d) Ecoli, $\mathbf{L}_{\mathrm{SM}}$

(e) Pima, $\mathbf{L}_{\mathrm{SM}}$

(f) Yeast, $\mathbf{L}_{\mathrm{LB}}$

FIGURE 3.15: NMI for different datasets with different normalization techniques.

TABLE 3.8: Average results with different normalization techniques using NMI. Boldface numbers indicate the best clustering performance.

		Spectral descriptors						
Dataset	Normalization method	GPS	WKS	HKS	SIHKS	FDD	SGWS	AHK
Ecoli	$\mathbf{L}_{\text{NN}}$	0.485	0.580	0.882	0.300	0.621	**0.989**	0.450
Ecoli	$\mathbf{L}_{\text{LB}}$	0.449	0.620	0.708	0.350	0.724	**0.740**	0.380
Ecoli	$\mathbf{L}_{\text{RW}}$	0.350	0.381	0.486	0.427	0.501	**0.628**	0.459
Ecoli	$\mathbf{L}_{\text{SM}}$	0.355	0.260	0.273	0.328	0.760	**0.800**	0.340
Ecoli	$\mathbf{L}_{\text{FP}}$	0.281	0.275	0.340	0.524	0.654	**0.698**	0.440
Glass	$\mathbf{L}_{\text{NN}}$	0.431	0.350	0.281	0.522	0.388	**0.621**	0.351
Glass	$\mathbf{L}_{\text{LB}}$	0.452	0.400	0.467	0.500	0.715	**0.820**	0.329
Glass	$\mathbf{L}_{\text{RW}}$	0.310	0.921	0.355	0.302	0.600	**0.975**	0.651
Glass	$\mathbf{L}_{\text{SM}}$	0.380	0.341	0.491	0.521	**0.704**	**0.704**	0.500
Glass	$\mathbf{L}_{\text{FP}}$	0.283	0.297	0.485	0.502	0.510	**0.640**	0.441
Yeast	$\mathbf{L}_{\text{NN}}$	0.401	0.412	0.460	0.480	0.710	**0.768**	0.431
Yeast	$\mathbf{L}_{\text{LB}}$	0.410	0.320	0.562	0.255	0.522	**0.750**	0.451
Yeast	$\mathbf{L}_{\text{RW}}$	0.430	0.460	0.522	0.520	0.700	**0.818**	0.421
Yeast	$\mathbf{L}_{\text{SM}}$	0.380	0.418	0.534	0.571	0.511	**0.852**	0.705
Yeast	$\mathbf{L}_{\text{FP}}$	0.451	0.400	0.300	0.553	0.618	**0.770**	0.450
Pima	$\mathbf{L}_{\text{NN}}$	0.364	0.479	0.541	0.580	0.690	**0.872**	0.386
Pima	$\mathbf{L}_{\text{LB}}$	0.221	0.450	0.520	0.620	0.705	**0.760**	0.426
Pima	$\mathbf{L}_{\text{RW}}$	0.356	0.410	0.887	0.350	0.710	**0.990**	0.316
Pima	$\mathbf{L}_{\text{SM}}$	0.300	0.146	0.380	0.461	0.200	**0.550**	0.187
Pima	$\mathbf{L}_{\text{FP}}$	0.296	0.421	0.340	0.480	0.708	**0.759**	0.250

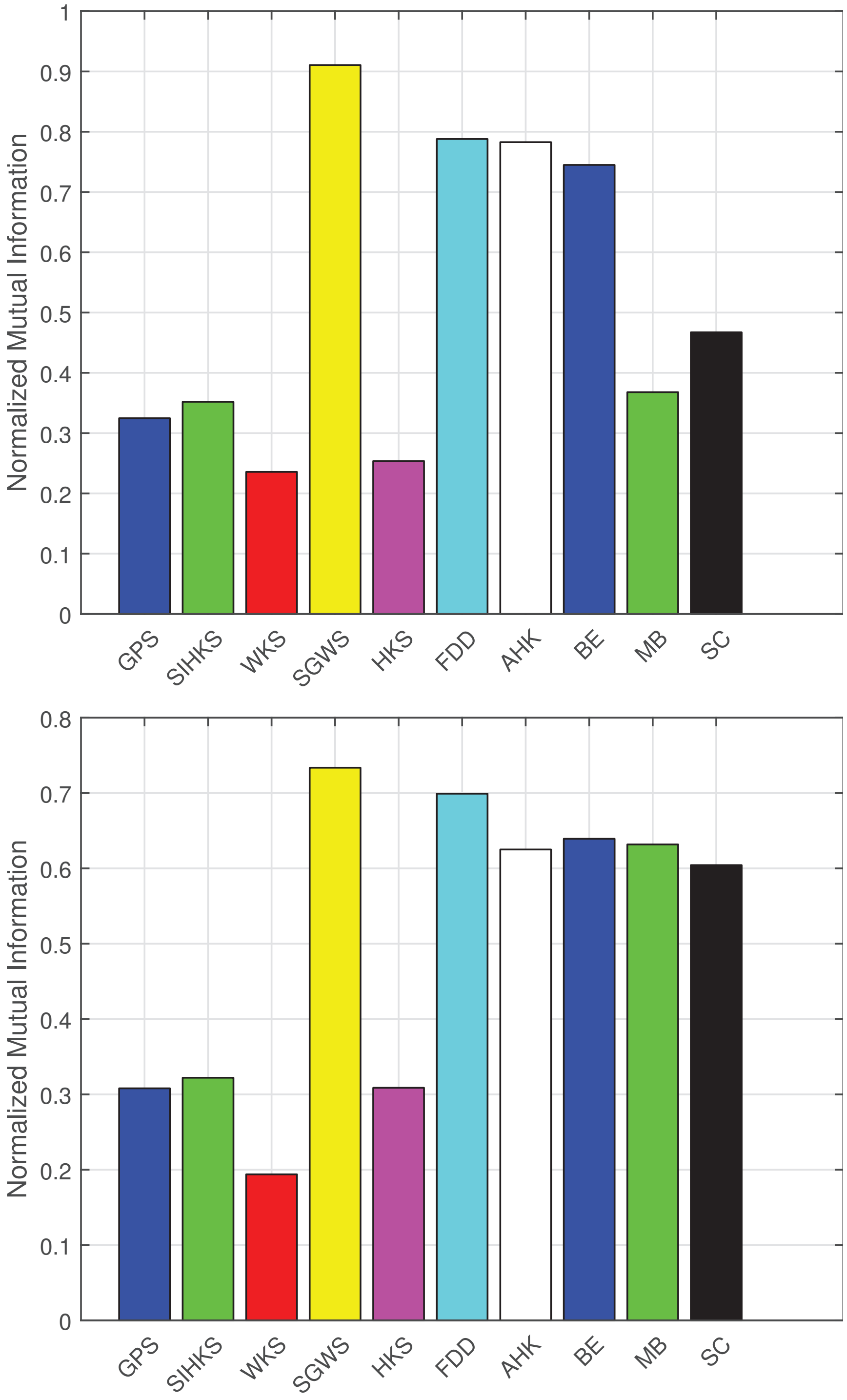


FIGURE 3.16: Comparison of different algorithms and SGWS.

4 Conclusions and Future Work

This thesis has presented novel clustering and classification algorithms for data clustering and classification using spectral descriptors. Extensive experimental results and evaluations have demonstrated the stability, robustness and high performance of our proposed approaches compared the existing methods. Section 4.1 provides the contributions and the concluding results drawn from the research work. In Section 4.2, suggestions for future research directions are discussed.

4.1 Contributions of the Thesis

4.1.1 Sparse Coding for Data Clustering and Classification

We have presented a comprehensive survey of different shape descriptors and analyze different clustering and classification algorithms. It turn out that FDD works very well with the combination of graph regularized sparse coding algorithm. We have introduced our work for image clustering and classification. To further enhance the effectiveness of our proposed GraphFDD algorithm, we have explored the best parameters such as number of clusters, nearest neighbors, graph regularization and sparsity regularization parameters. The proposed work outperforms state-of-the-art methods in terms of accuracy, confusion matrix and NMI measures. Extensive experiments and evaluations have demonstrated on five benchmarks for image and data classification and on two benchmarks for image clustering.

4.1.2 Spectral Graph Wavelets for Data Clustering

We have developed a new clustering algorithm with strong robustness and remarkable performance. This algorithm is based on the SGWS descriptor, which is multi-resolution, discriminative and compact in nature. Extensive experimental results have demonstrated the significant performance of the proposed algorithm in comparison with other popular spectral descriptors.

4.2 Future Research Directions

Several interesting research directions, motivated by this thesis, are discussed below:

4.2.1 Classification and Anomaly Detection

In the current form, SGWS descriptor is proposed for data clustering and it has been shown the best performance on various text benchmarks. The main advantage of our proposed algorithm is that our work is generic and can be extended on other problems such as classification and anomaly detection that are modeled on graphs. In the future, we plan to apply the proposed approach to classification and anomaly detection. In addition, the proposed GraphFDD also yields good performance for clustering and classification but much more extensive experiments are still required to validate this conjecture. Another future direction is to investigate GraphFDD for anomaly detection.

4.2.2 Sparse Coding

We intend to explore SGWS with sparse coding techniques in our future work. Our potential research direction is to explore our work with sparse coding and graph regularized sparse coding. Moreover, we can apply this signature for supervised learning with dimensionality reduction techniques. An important avenue of future research lies in investigating the use of the proposed approaches in tackling other data analysis problems efficiently and in the most flexible manner. Even, we will try to extend our work with other real world applications and with other image or text benchmarks.

4.2.3 Semi-supervised Learning

In the current work, we focus on supervised and unsupervised methods. Immediate future work will be concentrated on semi-supervised learning, which is a new type of learning based on labeled and unlabeled examples. Additionally, designing semi-supervised algorithms using appropriate signatures for many real-time applications is a promising future work direction that we plan to explore.

References

[1] P. J. Wolfe, “Making sense of big data,” in *Proceedings of National Academy of Sciences*, pp. 18031–18032, 2013.

[2] H. Krim and A. Ben Hamza, *Geometric Methods in Signal and Image Analysis*. Cambridge University Press, 2015.

[3] A. Ben Hamza, “Graph regularized sparse coding for 3D shape clustering,” *Knowledge-Based Systems*, vol. 92, pp. 92–103, 2016.

[4] M. Zaki and W. Meira, *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press, 2014.

[5] B. Liu, *Web Data Mining*. springer, 1998.

[6] M. Weinstein, “Strange bedfellows: Quantum mechanics and data mining,” *Nuclear Physics B (Proc. Suppl.)*, vol. 199, no. 1, pp. 74–84, 2010.

[7] X. Zhu, X. Wu, and C. Zhang, “Vague one-class learning for data streams,” in *In Proceedings of the 9th International Conference on Data Mining*, pp. 657–666, 2009.

[8] B. Pogorelc and M. Gams, “Discovery of gait anomalies from motion sensor data,” in *In Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence*, pp. 331–336, 2010.

[9] H. Huang, H. Qin, S. Yoo, and D. Yu, “Physics-based anomaly detection defined on manifold space,” *ACM Transactions on Knowledge Discovery from Data*, vol. 9, no. 2, pp. 14:1–14:39, 2014.

[10] Y. Aflaloa and R. Kimmel, "Spectral multidimensional scaling," in *Proceedings of National Academy of Sciences of the United States of America*, pp. 18052–18057, 2013.

[11] F. Chung, "Spectral graph theory," in *Reginonal Conference Series in Mathematics, vol. 92, American Mathematical Society*, 1997.

[12] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 98, pp. 298–305, 1973.

[13] K. Uhlenbeck, "Generic properties of eigenfunctions," *American Journal of Mathematics*, vol. 98, no. 4, pp. 1059–1078, 1976.

[14] J. Bondy and U. Murty, *Graph Theory with Applications*. North Holland, 1976.

[15] D. West, *Introduction to Graph Theory*. Prentice Hall, 2000.

[16] A. Brouwer and W. Haemers, *Spectra of graphs*. Springer, 2011.

[17] F. Fouss, A. Pirotte, J. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 355 – 369, 2007.

[18] H. Zhang, O. Kaick, and R. Dyer, "Spectral mesh processing," *Computer Graphics Forum*, vol. 29, no. 6, pp. 1865–1894, 2010.

[19] A. M. Bronstein, "Spectral descriptors for deformable shapes," in *Computer Vision and Pattern Recognition*, pp. 1–29, 2011.

[20] I. Kokkinos, M. M. Bronstein, and A. Yuille, "Dense scale-invariant descriptors for images and surfaces," Tech. Rep. INRIA RR-7914, Project Teams GALEN, 2012.

[21] M. Reuter, F. Wolter, and N. Peinecke, "Laplace-Beltrami spectra as 'Shape-DNA' of surfaces and solids," *Computer-Aided Design*, vol. 38, no. 4, pp. 342–366, 2006.

[22] R. Rustamov, "Laplace-beltrami eigenfunctions for deformation invariant shape representation," in *SGP'07 Proceedings of the fifth Eurographics symposium on Geometry processing*, pp. 225–233, 2007.

[23] C. Li and A. Ben Hamza, “A multiresolution descriptor for deformable 3d shape retrieval,” *The Visual Computer*, vol. 29, no. 6, pp. 513–524, 2013.

[24] A. Chaudhari, R. Leahy, B. Wise, N. Lane, R. Badawi, and A. Joshi, “Global point signature for shape analysis of carpal bones,” *Phys. Med. Biol.*, vol. 59, no. 4, pp. 961–973, 2014.

[25] K. Tarmissi and A. Ben Hamza, “Information-theoretic hashing of 3D objects using spectral graph theory,” *Expert Systems with Applications*, vol. 36, no. 5, pp. 9409–9414, 2009.

[26] C. Li and A. Ben Hamza, “Spatially aggregating spectral descriptors for nonrigid 3d shape retrieval: a comparative survey,” *Multimedia Systems*, vol. 20, no. 3, pp. 253–281, 2014.

[27] M. Aubry, U. Schlickewei, and D. Cremers, “The wave kernel signature: A quantum mechanical approach to shape analysis,” in *Proc. Computational methods for the innovative design of electrical devices*, pp. 1626–1633, 2011.

[28] A. Vaxman, M. Ben-Chen, and C. Gotsman, “A multi-resolution approach to heat kernels on discrete surfaces,” *ACM Trans. Graph.*, vol. 29, no. 4, pp. 121:1–121:10, 2010.

[29] J. Sun, M. Ovsjanikov, and L. Guibas, “A concise and provably informative multi-scale signature based on heat diffusion,” *Eurographics Symposium on Geometry Processing 2009*, vol. 28, no. 5, pp. 1383–1392, 2009.

[30] M. M. Bronstein and I. Kokkinos, “Scale-invariant heat kernel signatures for non-rigid shape recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1704–1711, 2010.

[31] Y. Fang, M. Sun, M. Kim, and K. Ramani, “Heat-mapping: A robust approach toward perceptually consistent mesh segmentation,” in *Proc. CVPR*, pp. 2145–2152, 2011.

[32] H. Huang, H. Qin, S. Yoo, and D. Yu, “A new anomaly detection algorithm based on quantum mechanics,” in *IEEE 12th International conference on data mining*, pp. 900–905, 2012.

[33] H. Huang, H. Qin, S. Yoo, and D. Yu, “A robust clustering algorithm based on aggregated heat kernel mapping,” in *11th IEEE International conference on data mining*, pp. 270–279, 2011.

[34] J. Hartigan and M. Wong, “Algorithm as 136: A k-means clustering algorithm,” *Applied Statistics*, vol. 28, no. 1, pp. 100–108, 1978.

[35] Z. Li, J. Liu, S. Chen, and X. Tang, “Noise robust spectral clustering,” in *IEEE 11th International Conference on Computer Vision*, pp. 1–8, 2007.

[36] H. Valizadegan and R. Jin, “Generalized maximum margin clustering and unsupervised kernel learning,” *MIT Press*, pp. 1417–1424, 2007.

[37] Y. Chi, X. Song, D. Zhou, K. Hino, and B. Tseng., “On evolutionary spectral clustering,” *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 4, pp. 1–30, 2009.

[38] A. Jain, M. Murty, and P. Flynn, “Data clustering: a review,” *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[39] A. Jain, “Data clustering: 50 years beyond k-means,” in *Pattern Recognition Letters*, pp. 651–666, 2010.

[40] D. Verma and M. Meila, “Comparison of spectral clustering methods,” Tech. Rep. CSE Technical report, University of Washington Seattle, 2001.

[41] R. Coiefman and S. Lafon, “Diffusion maps,” *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[42] C. Aggarawal, *Data Classification Algorithms and Applications*. Data Mining and Knowledge Discovery Series, 2014.

[43] J. Wang, K. Yu, Y. Gong, and T. Huang, “Linear spatial pyramid matching using sparse coding for image classification,” in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2009.

[44] S. Maji, A. C. Berg, and J. Malik, “Classification using intersection kernel support vector machine is efficient,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[45] A. Z. A. Bosch and X. Munoz, “Scene classification using a hybrid generative/dicriminative approach,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 712 – 727, 2008.

[46] P. Quelhas, F. Monay, J. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Gool, “Modeling scenes with local descriptors and latent aspects,” in *Tenth IEEE International Conference on Computer Vision, 2005*, pp. 883 – 890, 2005.

[47] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2169–2178, 2006.

[48] O. Boiman, E. Shechtman, and M. Irani, “In defense of nearest-neighbor based image classification,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[49] A. Bosch, A. Zisserman, and X. Muoz, “Image classification using random forests and ferns,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[50] L. van der Maaten, E. Postma, and J. van den Herik, “Dimensionality reduction: A comparative review,” Tech. Rep. TiCC TR 2009-005, TiCC, Tilburg University, 2009.

[51] C. Spearman, “”general intelligence,” objectively determined and measured,” *The American Journal of Psychology*, vol. 15, no. 2, pp. 201–292, 1904.

[52] M. Partridge and R. Calvo, “Fast dimensionality reduction and simple pca,” *Intelligent Data Analysis*, vol. 2, no. 3, pp. 292–298, 1997.

[53] S. Agarwal, A. Awan, and D. Roth, “Learning to detect objects in images via a sparse, part based representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 1475–1490, 2004.

[54] E. Pennec and S. Mallat, “Sparse geometric image representations with bandelets,” *IEEE Transactions on Image Processing*, vol. 11, no. 4, pp. 423–438, 2005.

[55] J. Starck, M. Elad, and D. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1570–1582, 2005.

[56] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[57] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *IEEE Computer Soc. Conf. Comput. Vis. Pattern Recognition*, pp. 1794–1801, 2009.

[58] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: a strategy employed by v1?," *Vision Res.*, vol. 37, no. 23, pp. 3311–3325, 1997.

[59] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, 2008.

[60] K. Huang and S. Aviyente, "Sparse representation for signal classification," *Advanced Neural Inf. Process. Syst.*, vol. 19, pp. 609–616, 2007.

[61] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognisition via sparse representation," *IEEE TRans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, 2009.

[62] W. Dong, X. Li, D. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, pp. 457–464, 2011.

[63] J. Tanenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionalty reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[64] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, pp. 3360–3367, 2010.

[65] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, and D. Cai, "Graph regularized sparse coding for image representation," *IEEE Transactions on Image Processing*, vol. 20, no. 5, pp. 1327–1336, 2010.

[66] H. Lee, A. Battle, R. Raina, and A. Ng., “Efficient sparse coding algorithms,” *Adv. Neural Inf. Process. Syst.*, vol. 20, pp. 801–808, 2007.

[67] S. Gao, I. Tsang, , and L. Chia, “Laplacian sparse coding, hypergraph Laplacian sparse coding, and applications,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 92–104, 2013.

[68] H. Lee, A. Battle, R. Raina, and A. Ng., “Efficient sparse coding algorithms,” in *Proc. Neural Information Processing Systems*, 2007.

[69] M. Long, G. Ding, J. Wang, J. Sun, Y. Guo, and P. S. Yu, “Transfer sparse coding for robust image representation,” in *IEEE conference on computer vision and pattern recognition*, pp. 407–414, 2013.

[70] U. Luxburg, “A tutorial on spectral clustering,” Tech. Rep. TR-149, Max Planck Institute for Biological Cybernetics, 2006.

[71] A. Y. Ng., M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in *Advances in Neural Information Processing Systems 14*, pp. 849–856, 2001.

[72] D. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[73] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 2002.

[74] M. Kozak, “A dendrite method for cluster analysis by calinski and harabasz: A classical work that is far too often incorrectly cited,” *Communications in Statistics - Theory and Methods*, vol. 41, no. 12, pp. 2279–2280, 2012.

[75] C. Marzban, “A comment on the roc curve and the area under it as performance measures,” *Weather and Forecasting*, vol. 19, no. 6, pp. 1106–1114, 2004.

[76] D. Powers, “Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation,” *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[77] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, pp. 861–874, 2006.