Improved *K*-means Clustering Algorithms

A thesis presented in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science

Massey University

Tong Liu

2020

Abstract

K-means clustering algorithm is designed to divide the samples into subsets with the goal that maximizes the intra-subset similarity and inter-subset dissimilarity where the similarity measures the relationship between two samples. As an unsupervised learning technique, *K*-means clustering algorithm is considered one of the most used clustering algorithms and has been applied in a variety of areas such as artificial intelligence, data mining, biology, psychology, marketing, medicine, etc.

K-means clustering algorithm is not robust and its clustering result depends on the initialization, the similarity measure, and the predefined cluster number. Previous research focused on solving a part of these issues but has not focused on solving them in a unified framework. However, fixing one of these issues does not guarantee the best performance. To improve *K*-means clustering algorithm, one of the most famous and widely used clustering algorithms, by solving its issues simultaneously is challenging and significant.

This thesis conducts an extensive research on *K*-means clustering algorithm aiming to improve it.

First, we propose the Initialization-Similarity (IS) clustering algorithm to solve the issues of the initialization and the similarity measure of *K*-means clustering algorithm in a unified way. Specifically, we propose to fix the initialization of the clustering by using sum-of-norms (SON) which outputs the new representation of the original samples and to learn the similarity matrix based on the data distribution. Furthermore, the derived new representation is used to conduct *K*-means clustering.

Second, we propose a Joint Feature Selection with Dynamic Spectral (FSDS) clustering algorithm to solve the issues of the cluster number determination, the similarity measure, and the robustness of the clustering by selecting effective features and reducing the influence of outliers simultaneously. Specifically, we propose to learn the similarity matrix based on the data distribution as well as adding the ranked constraint on the Laplacian matrix of the learned similarity matrix to automatically output the cluster number. Furthermore, the proposed algorithm employs the $L_{2,1}$-norm as the sparse constraints on the regularization term and the loss function to remove the redundant features and reduce the influence of outliers respectively.

Third, we propose a Joint Robust Multi-view (JRM) spectral clustering algorithm that conducts clustering for multi-view data while solving the initialization issue, the cluster number determination, the similarity measure learning, the removal of the redundant features, and the reduction of outlier influence in a unified way.

Finally, the proposed algorithms outperformed the state-of-the-art clustering algorithms on real data sets. Moreover, we theoretically prove the convergences of the proposed optimization methods for the proposed objective functions.

Dedication

In memory of my mother and father

Acknowledgments

I would like to express my sincere gratitude to my supervisors, Associate Professor XiaoFeng Zhu and Distinguished Professor Gaven Martin for all your guidance, encouragements and kindness.

I am deeply thankful for the support and encouragement from Professor Dianne Brunton, Associate Professor Alona Ben-Tal, Associate Professor Evelyn Sattlegger and Ms. Linh Mills.

Publications Arising from this Thesis

Publications included in this thesis:

- Zhou, J., **Liu, T**., & Zhu, J. Weighted adjacent matrix for *K*-means clustering. *Multimedia Tools and Applications*, 2019. **78**: p. 33415–33434, **corresponding author: Liu, T**, incorporated as Chapter 1 and Chapter 2.
- **Liu, T**., Zhu, J., Zhou, J., Zhu, Y., & Zhu, X. Initialization-similarity clustering algorithm. *Multimedia Tools and Applications*, 2019. **78**: p. 33279–33296, incorporated as Chapter 3.
- **Liu, T**., & Martin, G. Joint Feature Selection with Dynamic Spectral Clustering. Revised version submitted to *Neural Processing Letters (Springer)*. DOI: 10.1007/s11063-020-10216-9, incorporated as Chapter 4.
- **Liu, T**., Martin, G., Zhu, Y., Peng, L., & Li, L. Joint Robust Multi-view Spectral Clustering. Submitted to *Neural Processing Letters (Springer)*. DOI: 10.1007/s11063-020-10257-0, incorporated as Chapter 5.

Table of Contents

List of Tables

List of Figures

List of Notations

Symbols	Description
$\mathbf{X}$	Data matrix
$\mathbf{x}$	A vector of $\mathbf{X}$
$\mathbf{x}_i$	The i-th row of $\mathbf{X}$
$x_{i,j}$	The element in the i-th row and j-th column of $\mathbf{X}$
$\|\mathbf{x}\|_2$	L_2-norm of $\mathbf{x}$
$\|\mathbf{X}\|_{2,1}$	$L_{2,1}$-norm of $\mathbf{X}$
$\|\mathbf{X}\|_F$	The Frobenius norm or the Euclidean norm of $\mathbf{X}$
$\mathbf{X}^T$	The transpose of $\mathbf{X}$
K	Cluster number
V	Number of views
v	View index
$\mathbf{X}^v$	Data matrix in the v-th view

Chapter 1

Introduction

Machine learning is a subfield of artificial intelligence that provides machines the ability to learn and improve automatically without being explicitly programmed to do so. If the machine learns to label the data automatically without knowing the pattern of the data beforehand, this type of machine learning is called unsupervised learning. Unsupervised learning is important because it is difficult to know the pattern of the data in advance [1, 2].

As an unsupervised learning technique, clustering divides a given data set into groups with the goal to both maximize the similarity of data points in the same group, and the dissimilarity of data points in different groups [3]. Clustering has been widely applied in scientific data analysis, data mining, biology, psychology, marketing, medicine, and insurance, etc. [4-9]. A search via Google Scholar found over 4.1 million entries with the keyword clustering on Dec 14, 2019. *K*-means clustering algorithm is one of the most popular and widely used clustering algorithms. *K*-means clustering algorithm has been used as part of many other algorithms since it is simple, trustable, promising, and mathematical tractability [10, 11].

1.1 Motivation

K-means clustering algorithm operates in the following steps: First, it initializes cluster centers via randomly selecting *K* data points as the *K* cluster centers. Second, it assigns

each data point to its nearest cluster center according to a similarity measure, e.g., Euclidean distance. Third, it revises the *K* cluster centers as the mean of assigned data points. *K*-means clustering algorithm keeps repeating the last two steps until the algorithm achieves convergence [12]. The flow chart of *K*-means clustering algorithm is shown in Figure 1.1. *K*-means clustering algorithm is considered one of the most used clustering algorithms. It has been successfully applied to broad areas. Previous researches have addressed some of the issues of *K*-means clustering algorithm. But they didn't address the limitations of *K*-means clustering algorithm in a unified manner. Addressing the limitations of *K*-means clustering algorithm in a unified way is challenging and significant.

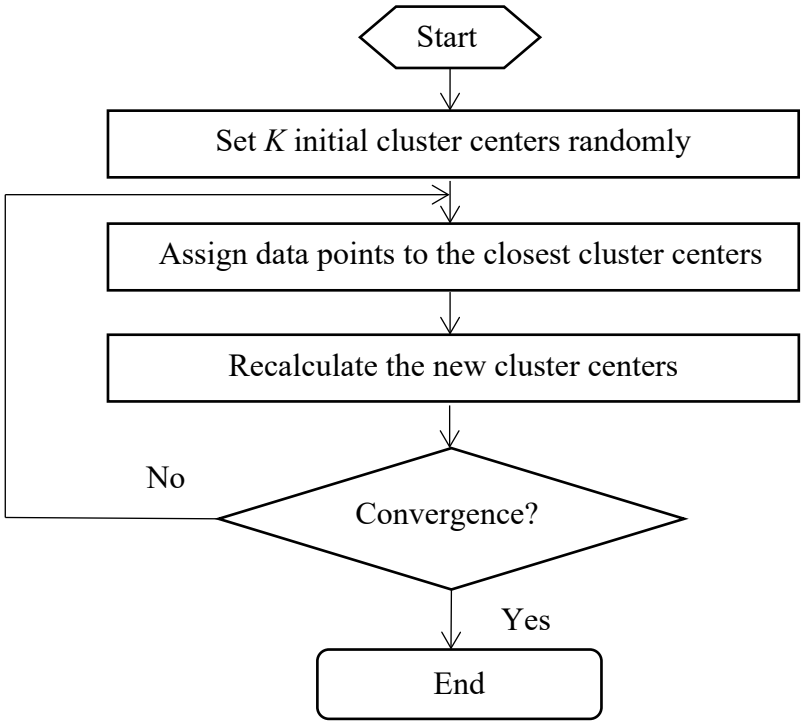


Figure 1.1 *K*-means Flowchart

First, the clustering result of *K*-means clustering algorithm depends on the initialization of cluster centers but random choosing the cluster centers may not lead to an optimal result. It is also difficult to reproduce the clustering results due to the randomness of initialization of *K*-means clustering algorithm. Many of the current clustering algorithms have solved the initialization problem of *K*-means clustering algorithm [4, 13-15]. For example, Duan et al. developed an algorithm to calculate the density to select the initial cluster centers [13]. Lakshmi et al. proposed to use nearest neighbors and feature means to decide the initial cluster centers [14].

Second, the clustering result of *K*-means clustering algorithm depends on the similarity measure. *K*-means clustering algorithm assigns each data point to its closest cluster center based on a similarity measure. Euclidean distance is often used in *K*-means clustering algorithm to determine the similarity by calculating the distance between two data points. However, Euclidean distance measure does not account for the factors such as cluster sizes, dependent features or density [16, 17]. Thus *K*-means clustering algorithm is not good for indistinct or not well-separated data sets [18]. Several works addressed the similarity measure problem of *K*-means clustering algorithm [19-24]. For example, spectral clustering algorithm uses spectral representation to replace original data points, and then conducts *K*-means clustering. To do this, spectral clustering algorithm first generates the similarity matrix and then conducts eigenvalue decomposition on the similarity matrix to obtain the spectral representation. Finally, *K*-means clustering is conducted on the spectral representation.

Third, *K*-means clustering algorithm relies on the given cluster number *K*. As an unsupervised algorithm, *K*-means clustering algorithm is supposed to be used against data which is not labelled. Without knowing the label, the cluster number may not be known beforehand. Robust continuous clustering algorithm is able to automatically calculate the cluster number beforehand [4]. However, this algorithm needs a well calculated similarity matrix beforehand as an input to be able to produce good clustering outcome.

Previous clustering algorithms only fixed part of the issues of the *K*-means clustering algorithm. When a clustering algorithm addresses those problems separately, it is easily to be trapped into the sub-optimal results, which means it is hard to obtain a global optimal solution, for example, even if a best initial value is found to produce optimal results or the best similarity matrix is found to produce optimal results, but the final optimal results may not be obtained. Because the results of the individual steps are not obtained according to the requirements of the next step. It would be challenging and significant if a new clustering algorithm could fix the issues of the initialization, cluster number determination and similarity measure problems of *K*-means clustering algorithm in a unified framework, which means one aspect is reflected in other aspects to achieve global optimal results.

Real-world data sets often contain high-dimensional features, some of which are insignificant for clustering. Data with high-dimensional features, i.e., high-dimensional data, increases the computation cost as well as the “Curse of Dimensionality”. In this circumstance, *K*-means clustering algorithm using Euclidean distance to measure the

similarity is not robust to data with high-dimensional features [25]. Hence, reducing the redundant feature is needed for conduct clustering analysis on high-dimensional data.

Data almost invariably contains noise, outliers and errors due to inadequate data measure, collection, processing or just the inherent variability. Outliers can distort the distribution of the data set. For example, *K*-means clustering algorithm using the mean of all data points in one cluster to decide the new cluster center makes sense when all the data points lie a normal distance from other data points. However, outliers can strongly impact the mean calculation of the whole cluster. As a result, this will push cluster centers closer to the outlier. Outliers could have a strong impact on the final cluster configuration. Hence, to achieve robust clustering performance, it is necessary to reduce the influence of outliers.

Nowadays data could be collected from multiple sources or different aspects. For example, images shared on photo sharing sites such as Instagram or Flickr have complementary information such as description, tags, location, and video, etc. The data collected from multiple views are called multi-view data. Each view of the data set has its own properties to contribute to the understanding of the subject matter. Normally *K*-means clustering algorithm was designed for clustering single-view data, the naive solution for conducting clustering on multi-view data by *K*-means clustering algorithm is to cluster the data with concatenated features across all views of the multi-view data. However, such a simple concatenation approach treats different views equally, even though different views have their own specific properties for their features. Hence, it is

essential to improve *K*-means clustering algorithm on multi-view data clustering as well as solving the aforementioned issues.

1.2 Research Objectives

The aim of this thesis is to design and evaluate new clustering algorithms to overcome the issues of previous *K*-means clustering algorithm. The thesis framework is demonstrated in Figure 1.2.

The specific objectives of this thesis are listed as follows:

- Objective 1: To solve the issues of the initialization and the similarity measure of *K*-means clustering algorithm in a unified way.
- Objective 2: To solve the issues of the cluster number determination, the similarity measure, and to improve the robustness of clustering by selecting effective features and reducing the influence of outliers in a unified way.
- Objective 3: To develop multi-view clustering algorithm while solving the issues of the initialization, the cluster number determination, the similarity measure, feature selection and outlier reduction in a unified way.

1.3 Thesis Structure

This thesis is structured as follows.

- Chapter 2 presents literature review including clustering analysis, feature selection, outlier reduction and evaluation measure.

- Chapter 3 presents Initialization-Similarity (IS) clustering algorithm which solves the issues of initialization and similarity measure of *K*-means clustering algorithm in a unified way. IS clustering algorithm fulfills our objective 1. The proposed IS clustering algorithm outperformed both the classical clustering algorithms *K*-means clustering algorithm and well-known Spectral clustering algorithm.
- Chapter 4 presents Joint Feature Selection with Dynamic Spectral (FSDS) clustering algorithm which solves the issues of cluster number determination, similarity measure, and the robustness of clustering by selecting useful features and reducing the influence of outliers in a unified way. FSDS clustering algorithm fulfills our objective 2. The proposed FSDS clustering algorithm outperformed the classical clustering algorithms *K*-means clustering algorithm, well-known Spectral clustering algorithm, Clustering and projected clustering with adaptive neighbors algorithm (CAN) [24] and Robust continuous clustering algorithm (RCC) [4].
- Chapter 5 presents Joint Robust Multi-view (JRM) Spectral Clustering algorithm solves initialization, cluster number determination, similarity measure, feature selection, and outlier reduction issues for multi-view data in a unified way. JRM clustering algorithm fulfills our objective 3. The proposed JRM clustering algorithm outperformed the classical clustering algorithms *K*-means clustering algorithm, Graph-Based system (GBS) [26], Adaptively weighted Procrustes (AWP) [27], and Multi-view low-rank sparse subspace clustering (MLRSSC) [28].
- Chapter 6 presents the conclusions and future work.

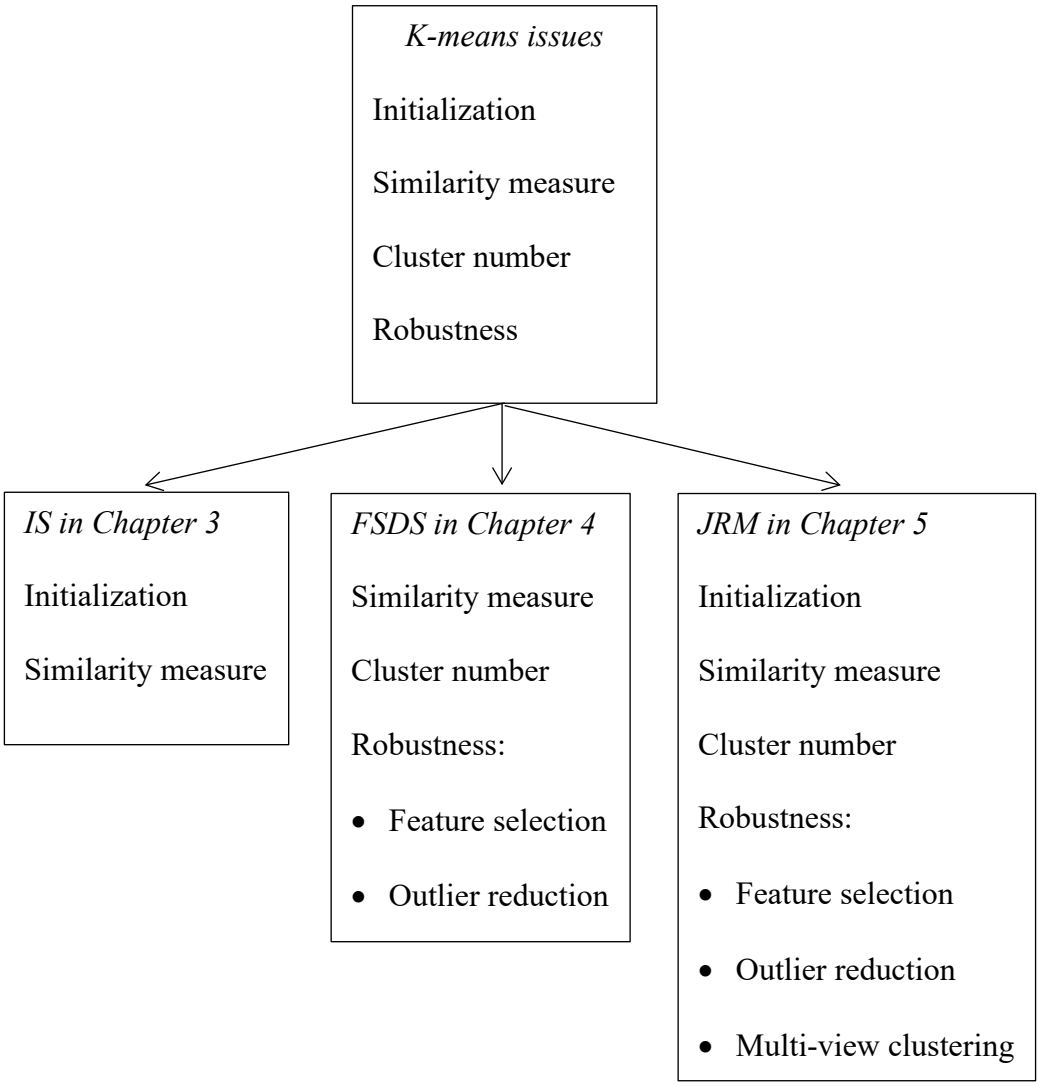


Figure 1.2 Research Framework

Chapter 2

Literature Review

Clustering is an unsupervised learning technique which divides a given data set into groups with the goal to maximize the intra-subset similarity and inter-subset dissimilarity. This chapter reviews the research topics related to this thesis, including the clustering algorithms, feature selection techniques, outlier reduction methods, and evaluation metrics.

2.1 Clustering Algorithms

Clustering algorithms can be classified as single-view clustering algorithms or multi-view clustering algorithms based on if the clustering algorithms aim to cluster single-view data or multi-view data.

2.1.1 Single-view Clustering

Clustering can also be generally categorized into non-graph-based approaches and graph-based approaches, based on whether the clustering algorithm constructs a similarity graph or not.

A. Non-Graph-Based Algorithms

The non-graph-based algorithms conduct clustering directly on the original data without constructing a similarity graph. The non-graph-based clustering algorithms can

be further grouped into different categories such as partitioning-based, hierarchical-based, distribution-based, density-based, nature-based, etc.

Partitioning-based clustering algorithms, also known as centroid-based clustering or distance-based clustering, divide data in one level into a number of partitions, where each partition represents a cluster. The center of the data points in each partition is regarded as the cluster center of the corresponding cluster. *K*-means clustering algorithm is one of the most famous representatives of this kind of clustering algorithms [29]. Specifically, *K*-means clustering algorithm first randomly selects *K* data points as the *K* cluster centers, and then assigns each data point to its nearest cluster center according to Euclidean distance. It keeps recalculating the cluster centers followed by assigning each data point to a cluster until the algorithm achieves convergence [12]. However, *K*-means clustering algorithm needs the cluster number as input, so it is not suitable for a data set with an unknown cluster number. It is also sensitive to the initialization of the cluster centers because the random choice of cluster centers may produce different clustering results on different runs of this algorithm [29]. Furthermore, *K*-means clustering algorithm measures the similarity by using the Euclidean distance which gives the same importance to all the data points without consider other factors such as density, dependent features, shape, patterns or scale of data points [30, 31]. For example, it is difficult for *K*-means clustering algorithm to separate non-convex clusters. There are numbers of other algorithms based on partitioning clustering algorithms, e.g. *K*-medoids, COTCLUS, and Tabu search. *K*-medoids chooses the data points located near their center to represent the clusters. The

rest of remaining data points are clustered with the representative data centers to which they are the most similar based on the minimal sum of the dissimilarities between data points and their corresponding cluster center points [32]. Instead of using only one center for each class, COTCLUS, an improved centroid-based clustering algorithm, uses suitable centroids from another clustering. It finds two centroids from one cluster and replace them by two centroids from the other cluster in such a way that maximum decreases the mean square error of the first clustering. It constructs a clustering from two suboptimal clustering results based on the belief that each suboptimal clustering has benefits regarding to containing some of the correct clusters [33]. After modifying centroids, it applies *K*-means clustering algorithm for final fine-tuning [33]. A Tabu based clustering algorithm employs the center driven approach of the *K*-means clustering algorithm with the guidance of Tabu search, which is a local or neighborhood search algorithm that accepts the worsening searches of no improving search is available and discourages the search from going back to previously visited search [34]. The *K*-medoids, COTCLUS, and Tabu search example like other partitioning-based clustering algorithms need to specify the cluster number *K* before the execution of the algorithms.

In comparison with the partitioning-based clustering, which divides the set of data into un-nested clusters, the hierarchical-based clustering builds a tree of nested clusters. The hierarchical-based algorithms, also known as connectivity-based clustering, build a hierarchical relationship among data points to conduct clustering. Hierarchical clustering is usually represented by a tree structure, where each data point

is identified as a leaf and each node is a cluster. The division and agglomeration are two common approaches of the hierarchical-based clustering. In the division approach, which is also called top-down approach, all the data points are initially in one cluster and then are divided into smaller clusters recursively. Conversely the agglomerative approach also called bottom-up approach which treats each data point as a cluster at the start, and then continuously agglomerate pairs of clusters to build a cluster hierarchy until all clusters have been merged into a single cluster that contains all data points [35, 36]. For example, the hierarchical clustering algorithm for binary data based on cosine similarity (HABOC) uses agglomerative hierarchical clustering procedure [37]. HABOC assesses similarity between data points and computes similarity of data sets containing multiple data points using the cosine similarity, and then exploits hierarchical clustering method to compresses data and merge two clusters based on the cosine feature vector of a set and additivity of the cosine feature vector of a set [37]. HABOC needs the cluster number as an initial parameter. Other hierarchical clustering examples include robust clustering using links (ROCK) and clustering using representatives (CURE) [38-44]. ROCK clustering algorithm draws a number of data points randomly from the original data set as inputs along with the desired cluster number *K*. Instead of using distances to conduct clustering, ROCK uses the number of links which is defined as the number of common neighbors as the similarity measure [42]. The reasoning behind is that the data points belonging to the same cluster most likely have a large number of common neighbours, thus more links. Hence the larger the number of links between data points, the greater likelihood they belong to the same

cluster. But ROCK ignores the possible differences in the similarity measure of different clusters inside the same data set. CURE selects well scattered points from the cluster to represent each cluster, and then shrink them toward the cluster [40]. It chooses more than one representative points from each cluster by using single linkage approaches, the similarity of two clusters is determined by the similarity of their most similar data points. Finally, the clusters with the closest representative points are clustered together. CURE uses random sampling and partitioning to speed up clustering [40]. But it is limited by choosing a fixed amount of scattered data points to represent cluster, and by applying a constant factor to shrink those representatives towards to their cluster centers [45]. CURE also ignores the information about the aggregate interconnectivity of data points in two clusters. Hierarchical clustering algorithms are particularly good when the data has an underlying hierarchical structure [35]. However, the efficiency of hierarchical clustering algorithms is relatively low compared with the linear complexity of partitioning clustering algorithms.

Closely related to statistics, the distribution-based clustering algorithms assume that the data generated from the same distribution belongs to the same cluster. However, not all the data points have several distributions and the parameters have a strong impact on the clustering results [29]. Examples of distribution-based clustering algorithms include incremental local distribution-based clustering algorithm with the Bayesian adaptive resonance theory (ILBART) [46], Gaussian mixture model (GMM) [47] and balanced iterative reducing and clustering using hierarchies (BIRCH) [48, 49]. ILBART first obtains some data patterns with snapshots. Then, the data pattern is

clustered by cluster choosing, matching test, and updating learning three stages. The variation of the covariance determinant, the combining threshold and the choice function are simultaneously considered in determination of the local distribution of the winning cluster [46]. ILBART is sensitive to the data order and its computational stability needs to be improved [46]. GMM uses a probabilistic approach and describes each cluster by its cluster center, covariance, and size. It randomly initializes a fixed number of Gaussian distributions to the data and iteratively optimizes Gaussian distributions parameters such as mean, variance and weight for each cluster. Finally, it calculates the probabilities of data points belonging to each of the clusters [47]. There may be no Gaussian distributions for many real data sets. Besides the issue of Gaussian distributions assumption, choosing the initial number of Gaussian distributions sets and random initialization are also issues of Gaussian mixture model [50]. BIRCH clustering algorithm summarizes the information that retains as much distribution information as possible, and then conducts the clustering on the data summary. Specifically, BIRCH clustering algorithm takes original data set and desired cluster number, and then conducts clustering in the four phases. It first computes the clustering feature tree. Second, it builds a smaller clustering feature tree and regrouping crowded sub-clusters into larger ones. Third, it computes the cluster centers of each cluster and uses an adaptation of the agglomerative clustering to cluster all the leaves of the clustering feature tree. Fourth, it uses the cluster centers to conduct the final clustering. BIRCH is sensitive to the data order and non-spherical clusters [39].

Density-based clustering algorithms partition data points into clusters defined as dense regions of data points separated by low-density regions. Examples of density-based clustering algorithms include density-based spatial clustering of applications with noise (DBSCAN) [51, 52], an attempt at improving density-based clustering algorithms (AIDCA) [53], and two-phase clustering algorithm with a density exploring distance measure (TADEDM)[54]. DBSCAN recognizes each cluster by finding a distinctive density of points by a notably large amount higher than outside of the cluster. Minimum points, core points, border points and neighbourhood are important concepts in DBSCAN. Minimum points define the minimum number of points required to form a cluster. A core point is a point which has at least minimum points within neighbourhood from itself. A border point is a point has at least one core point at a neighbourhood distance. The neighbourhood value defines the cut-off distance of a data point from the core point for it to be clustered as a part of a cluster or not. A point is density-reachability point if it is within neighbourhood distance from the core point. A core point and all the points within a neighbourhood distance form a core set. All the overlapping core sets are grouped together to form a cluster. A point, neither a core nor a border point, and has less than minimum points within neighbourhood distance from itself is a noise point. DBSCAN is not entirely deterministic because some border points could be reachable from more than one cluster. DBSCAN depends on the distance threshold estimation and it cannot handle data sets with large varying densities [51]. AIDCA creates adaptive grids on the data and then merging cells based on local density to form a cluster [53]. It considers each axis of the grid space separately and creates a

number of initial bins for each axis. These uniform bins have size of the data on its axis divided by the number of bins. The density is the sum of the data points in each bin, resulting in a histogram. AIDCA goes through each bin and compares its density with the neighboring one. If the density is less than the set merge-value, the two bins are merged and will be part of the same grid square in the final adaptive grids. The result of this is that neighboring grids are likely to have differing densities. The distribution of grids should result in a small number of denser grids that contain cluster centers surrounded by a number of less dense grid cells that constitute the edges of the cluster that is merged into the core [53]. AIDCA needs to know the number of bins to create and the set merge-value. It is difficult to determine the correct set of parameters. TADEDM is a two-phase clustering method, which applies K-means clustering algorithm in the first phase to obtain the initial clusters which are used as inputs in the second phase [54]. In the second phase, all the data points are clustered using K-means clustering with a density exploring distance measure, which refers to that data points close in distance have high affinity with each other and data points locating in the same cluster have high affinity with each other [54]. Due to using the K-means clustering algorithm in this algorithm, it requires the prior cluster number and it also suffers the initialization problem. The density-based algorithms are based on the assumption that the data points in the high-density region belong to the same cluster. However, the results of density-based algorithms will suffer if the density of data points with large difference. Moreover, most density-based algorithms are also sensitive to the parameters estimation [55].

Imitating the behavior of natural and biological systems, some nature-inspired optimization algorithms have been developed [56]. The nature-inspired optimization algorithms are combined with clustering algorithms to obtain the global optimum solution. The crow search algorithm (CSA) combines the *K*-means clustering algorithm with intelligent behaviour of the crows to obtain the global optimum solution. CSA requires the cluster number to conduct the clustering [57]. The krill herd algorithm (KHA) models the behaviour of individual krill within a larger krill swarm to find the cluster center [58]. It randomly initializes the data structure representing a single krill, then it iteratively generating the fitness function for each krill (data point) of the population, which is similar to calculating the optimized functions for the coordinates of the Krill's position [58]. The flower pollination algorithm (FPA) is another example of nature-inspired optimization procedures. It is inspired by the process of flower pollination. Specifically, to mimic this behavior, FPA employs Levy flight distribution, which is a random walk in which the step lengths have heavier tails than the exponential distribution [58, 59]. CSA, KHA, and FPA are like most of the current nature-inspired algorithms lack of clear mathematical and theoretical proof of convergence [60].

B. Graph-Based Algorithms

Instead of conducting clustering directly on the original data points, most graph-based clustering algorithms will first construct a graph and then apply a clustering algorithm to partition the graph. Graph representation represents the high-order relationship among data points which is easier to interpret the complex relationship inherent in the

data points than to interpret it from the original data points directly. A graph is a set of nodes or vertices with connected edges which have weights associated with them. A node or a vertex of the graph represents a data point and the edge represents the relationship between the data points. The similarity graph represents the similarities between data points. The similarity graph is represented by the similarity matrix, a square symmetric adjacency matrix, where the row and column indices represent the data points, and the entries indicate pairs of data points are connected or not. Two vertices are connected if the similarity between the corresponding data points is larger than a certain threshold. The edges within a cluster should have high weight values because data points within the same cluster are similar to each other. The edges between clusters should have low weight values because data points in different clusters are dissimilar from each other. Then the clustering problem is transformed into the graph cutting problem. The graph is cut into subgraphs, each subgraph being a cluster. The nodes in a cluster are well connected to nodes in the same cluster but not the nodes outside its cluster.

Spectral clustering algorithm is a typical example of graph-based algorithms. It has become increasingly popular. Spectral clustering algorithm first creates a similarity matrix and a diagonal degree matrix, which is the sum of all the weights on each row in a similarity matrix. Then it defines a feature vector by computing the first K eigenvectors of its Laplacian matrix, which is the degree matrix subtracting the similarity matrix. Finally, it runs K-means clustering on these features to separate

objects into *K* clusters [61]. Spectral clustering algorithm is a multi-step algorithm and it requires the cluster number to be predefined.

While some graph-based algorithms construct coefficient vectors of two data points to analyse the similarity between two data points [62], some graph-based algorithms construct hypergraph to represent a set of spatial data [63, 64]. For example, low-rank representation (LRR) identifies the subspace structures from data points and then finds the lowest rank representation among data points to represent the original data points [65]. A low-rank kernel learning graph-based clustering (LKLGC) algorithm is based on a multiple kernel learning with assumption that the consensus kernel matrix is a low-rank matrix and lies in the neighbourhood of the combined kernel matrix [66]. The spectral clustering algorithm is applied to get the final clustering results for LKLGC algorithm, hence the cluster number needs to be predefined [66]. A hybrid clustering algorithm based on minimum spanning tree of natural core points (NCP) first adaptively obtains the number of neighborhood parameter, and finds all the nature core points of datasets, and then it breaks the datasets into subsets and constructs the minimum spanning tree of natural core points. Finally, it cuts the maximum edge of the minimum spanning tree of natural core points iteratively until obtains the desired cluster number [67]. NCP needs the cluster number for its final step of clustering.

The hierarchical clustering using dynamic modeling (CHAMELEON) uses a graph partitioning algorithm to divide the data points into several relatively small sub-clusters initially, and then finds the genuine clusters by repeatedly combining these sub-clusters if they are close together and interconnectivity is high [41]. CHAMELEON is

a graph-based two-phase hierarchical clustering and it requires the predefined cluster number.

Clustering and projected clustering with adaptive neighbors algorithm (CAN) learns the data similarity matrix and then impose the rank constraint to the Laplacian matrix of the data similarity matrix [24]. In the end of the process the connected components in the resulted similarity matrix represent the clusters of the original data points [24]. CAN learns the data similarity matrix and clustering structure simultaneously. But it needs to know the number of the cluster beforehand.

Robust Continuous Clustering (RCC) continuously optimizes a robust objective based on robust estimation [4]. RCC optimizes clustering and its new representation learning jointly [68]. According to RCC algorithm, each data point has a dedicated representative, which locates at the data point initially. Throughout the clustering process, the representatives move and combine into clusters. Despite objective function of RCC is not convex, the optimization is performed by using standard linear least squares solvers [4]. The RCC does not need prior knowledge of the cluster number. However, it needs the similarity matrix beforehand.

Graph-based clustering algorithms improve non-graph-based clustering algorithms by generating the representation of original data points. However, current graph-based clustering algorithms use a multi-stage strategy which learns the similarity matrix, the new representation, or the clustering structure separately. The first stage goal of learning a similarity matrix does not always match the second stage goal of achieving optimal new representation, and thus not guaranteed to always outperform

non-graph-based clustering algorithms. Moreover, most graph-based clustering algorithms still use non-graph-based clustering algorithms in the final stage and thus do not simultaneously solve the initialization, similarity measure or cluster number issues of non-graph-based clustering algorithms.

2.1.2 Multi-view Clustering

The existing multi-view clustering algorithms can be broadly categorized to concatenation-based approach, distribution-based approach, and centralization-based approach.

A concatenation-based multi-view algorithm conducts clustering on the new concatenated feature vectors of each view. Examples of concatenation-based algorithms include concatenation *K*-means clustering and feature concatenation multi-view subspace clustering [69]. *K*-means clustering algorithms was developed for single-view data sets. For multi-view data sets, *K*-means clustering algorithm conducts clustering on the concatenated features across all views. This simple concatenation approach did not consider the unique nature of different views, even though different views have their own specific properties for their features. Furthermore, it may lead to a critical issue of "curse of dimensionality", which refers to a fixed number of data points become increasingly "sparse" as the dimensionality increase. The "curse of dimensionality" affects the clustering results [70].

A distribution-based multi-view algorithm conducts clustering on every view of a multi-view data set individually, and then synthesizes these results from individual views for final clustering. For example, co-regularized spectral clustering algorithm

uses one single objective function for individual view and combines spectral graphs from different views for final *K*-means clustering [71]. A low-rank multi-view matrix completion (lrMMC) algorithm first seeks a low dimensional representation where the common subspace is constrained to be low rank and combination weights which are learned to explore complementarity between different views [72]. Mutual kernel completion algorithm applies different predefined kernels for different views. Then these kernels are combined to an unified kernel [73]. An ensemble approach to multi-view multi-instance learning builds models on multiple heterogeneous data views by combining view learners and pursuing consensus among the weighted class [74]. However distribution-based multi-view algorithms do not fully use the information of multi-view and thus is unavailable to produce reasonable clustering results [75].

Compared with concatenation-based and distribution-based approaches, a centralization-based approach achieves better performance since it takes information from all views of a multi-view data set to conduct clustering [76]. A weighted hybrid fusion method constructs an objective function with rank consistency constraint [77]. Graph-based system (GBS) automatically weights the constructed graph of each view, and then generates a unified graph matrix [26]. Although it dynamically generates the weight of each graph matrix, GBS needs the number of neighbors as a prior. Furthermore the learning of the unified graph and the constructing graphs are in two separate stages. Adaptively weighted Procrustes (AWP) weights each view according to its clustering capacity and forms a weighted Procrustes average problem accordingly [27]. AWP requires spectral embedding matrix calculated beforehand as an input. The

goal of conducting the spectral embedding matrix is different from the second stage of multi-view clustering, and thus not guaranteed to always have optimal performance. Multi-view low-rank sparse subspace clustering (MLRSSC) jointly learns an affinity matrix constrained by sparsity and low-rank, while at the same time balances between the agreements across different views [28]. MLRSSC learns the joint affinity matrix first, and then uses the spectral clustering algorithm to complete the final clustering. The learning of the affinity matrix and final spectral clustering are in two separate stages. Thus, it cannot guarantee to always have optimal clustering results.

2.2 Feature Selection

Real-world data sets are rich in information. They often contain high-dimensional features. However, not all features are effective for clustering algorithms. The high-dimensional features not only increase the computational time for machine learning, but also increasing risk of overfitting. Dimensionality reduction aims to reduce the dimensions of data by obtaining a set of principal data or removing the redundant and dependent features [78]. It transforms the features from a high dimensional space to a low dimensional space. It could be applied to reduce the complexity, avoid overfitting, and reduce the influence of outliers. Feature selection is one of dimensionality reduction approaches. Feature selection is for selecting useful features from the original features or filtering irrelevant or redundant features from the original data set. The feature selection techniques can be broadly categorized into three types: the filter

feature selection methods, the wrapper feature selection methods and the embedded feature selection methods.

The filter feature selection methods filter out unimportant or redundant features from the original data set based on certain criteria [79]. Mutual Information or correlation to select the most relevant features [80]. Feature selection for multi-labeled variables method selects features via maximizing conditional dependency between features [79]. An unsupervised filter feature selection method for mixed data (USFSM) evaluates the relevance of features by their contributions and defines good cluster structures by analysing the changes of spectrum of the normalized Laplacian matrix when a feature is excluded [81]. The filter techniques have advantages of their speed and scalability [82, 83]. Filter methods are useful for selecting a generic set of features for all the machine learning models. The filter techniques have advantages of their speed and scalability. However, in some cases, features selected through filter methods may not be the most optimal set of features for some specific algorithms.

The wrapper feature selection methods are used to select the most optimal features for the specified algorithms [84, 85]. There are different wrapper approaches. A meta-heuristic wrapper method uses random encircling and imitative behavior of the Kestrel bird for optimal selection of features [86]. The sequential approach adds or removes features sequentially; the bio-inspired approach introduces randomness into the process to gain global optima; the iterative approach converts the feature selection problem to an estimation problem [81]. A sequential methods outputs both a ranking of relevant features and an optimal partition by using Mahalanobis metric (multivariate

distance metric which measures the distance between a data point and a distribution) and *K*-means clustering algorithm [84]. Localized feature selection (LFS), an iterative algorithm, uses a randomized rounding approach when weights of regions are fixed [85]. However, traditional wrapper methods usually have poor generalization ability, high complexity, low computational efficiency, and high computational cost [82, 87].

In embedded approaches of feature selection, the feature selection is an integrated part of the learning algorithm. The embedded approaches can be generally divided into two types: decision tree algorithms and regularization techniques. Decision tree algorithms select features recursively during the tree growth process [88, 89]. The tree growth process is also the process of feature selection. Some feature selection methods based on bee colony and gradient boosting decision tree [88]. Some use classification and regression tree-based (CART) decision tree algorithms to select features for 3D depth video [89]. L_1-norm, L_2-norm, or $L_{2,1}$-norm have been used for feature selection in regularization techniques-based algorithms, which objective function is the minimization of the regularized cost. The key difference between the regularization techniques is the regularization term or penalty term. In L_1-norm regularization, the absolute value of the magnitude of the coefficient is the penalty term. In L_2-norm regularization, the squared magnitude of the coefficient is penalty term. In $L_{2,1}$-norm regularization, the penalty term is a non-squared magnitude of the coefficient.

For the matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$, the L_1-norm, L_2-norm, and $L_{2,1}$-norm are defined in Eq. 2.1, Eq. 2.2 and Eq. 2.3 respectively [90]:

$$\|\mathbf{M}\|_1 = \sum_{i=1}^{n} \sum_{j=1}^{m} |m_{i,j}| \tag{2.1}$$

$$\|\mathbf{M}\|_2 = (\sum_{i=1}^{n} \sum_{j=1}^{m} {m_{i,j}}^2) \tag{2.2}$$

$$\|\mathbf{M}\|_{2,1} = \sum_{i=1}^{n} (\sum_{j=1}^{m} {m_{i,j}}^2)^{1/2} \tag{2.3}$$

Specifically, L_1-norm generates element-wise sparsity while $L_{2,1}$-norm generates row-wise sparsity. That is, by using $L_{2,1}$-norm penalty on the regularization term, it makes some rows of the generated projection matrix be 0. The redundant features are filtered out as unrepresentative features corresponding to row-wise sparsity do not participate in the clustering process, $L_{2,1}$-norm-based algorithms have a better interpretability than L_1-norm-based algorithms in feature selection models [91, 92]. L_2 -norm can't generate sparsity, which means it is lack of effectiveness in the feature selection model [93]. The $L_{2,1}$-norm-based approaches are more robust than the L_2-norm-based approaches.

Recently the $L_{2,1}$-norm has been used to improve the robustness of the feature selection algorithms [94, 95]. For instance, the $L_{2,1}$-norm regularization term is imposed to the objective function to achieve feature selection and capture the discriminative structure information [94]. The $L_{2,1}$-norm is used on both reconstruction error and the sparse constraint term to extract representative 2D image features [95]. $L_{2,1}$-norm regularized regression model used for joint feature selection from multiple tasks.

2.3 Outlier Reduction

Real data often contains outliers, which are data points inconsistent with most of the other data points in a given data set [96, 97]. The outliers could be resulted from an

inadequate procedure of data measure, collection, and data handling, or due to inherent variability in the underlying data domain. The outliers could significantly affect the clustering results. Outlier detection and robust clustering algorithms are often used to tackle the outlier problem.

Outlier detection algorithms detect those outliers which are data points deviated from most of the other data points. Most of the existing outlier detection studies focus on unsupervised outlier detection [98]. Examples of outlier detection algorithms include distance-based outlier detection [99], dimension-based outlier detection [100], density-based outlier detection [101], frequent pattern based outlier detection [102, 103], and cluster-based outlier detection [104], etc.

To minimize the impact of outliers, robust clustering has been intended from different areas. Some algorithms learn a robust metric to measure the similarity between points by taking the outliers into account [105, 106]; some algorithms use L_1 or $L_{2,1}$-norm to remove the outliers [107, 108]; some algorithms assign different weights to the data and the outliers during the clustering process [109]; some algorithms decompose outliers into a low-rank part [66, 110]; some algorithms conduct ensemble or fusion-based clustering algorithms combine different partitions results to deliver a more robust result [111, 112]. L_1-norm, L_2-norm, or $L_{2,1}$-norm have been used on the regularization terms of the objective functions of clustering algorithm [113]. A non-convex multi-task generalization of the $L_{2,1}$-norm regularization is used to learn a few features common across multiple tasks [114]. $L_{2,1}$-norm regularized regression model used for joint

feature selection from multiple tasks [115]. $L_{2,1}$-norm regularization encourages multiple predictors to share similar sparsity patterns [115].

Formally, let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \in \mathbb{R}^{p \times n}$, $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n) \in \mathbb{R}^{k \times n}$, and $\mathbf{U} \in \mathbb{R}^{p \times k}$. In L_1-norm-based robust clustering algorithms, the absolute value of the magnitude of the coefficient is used in the loss function.

$$\min_{\mathbf{U},\mathbf{V}} \mathrm{E}_1 (\mathbf{U}, \mathbf{V}) = \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_1 \tag{2.4}$$

Specifically, L_1-norm generates element-wise sparsity. As outliers corresponding to row-wise sparsity instead of element-wise sparsity, L_1-norm based algorithms do not have a good interpretability in the outlier reduction.

In L_2-norm-based robust clustering algorithms, the squared magnitude of the coefficient is penalty term.

$$\min_{\mathbf{U},\mathbf{V}} \mathrm{E}_2 (\mathbf{U}, \mathbf{V}) = \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_F^2 \tag{2.5}$$

The L_2-norm is calculated as the square root of the sum of the squared vector values. For example, an outlier, its residual $\|\mathbf{x}_i - \mathbf{U}\mathbf{v}_i\|$ is larger than residuals of other non-outliers. After squaring, the residual of the outlier could dominate the loss function. The L_2-norm is also used to calculate the Euclidean distance of the vector coordinate from the origin of the vector space. Euclidean distance is often used in clustering algorithm to calculate the similarity. The L_2-norm based is also called Euclidean norm.

The $L_{2,1}$-norm is defined in the following equation:

$$\min_{\mathbf{U},\mathbf{V}} \mathrm{E}_{2,1} (\mathbf{U}, \mathbf{V}) = \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_{2,1} \tag{2.6}$$

While $L_{2,1}$-norm generates row-wise sparsity. As outliers corresponding to row-wise sparsity do not participate in the clustering process, $L_{2,1}$-norm-based algorithms have a better interpretability than L_1-norm-based algorithms in outlier removal [91, 92]. The residual $\|\mathbf{x}_i - \mathbf{U}\mathbf{v}_i\|$ of an outlier is not squared, and thus reduces the influence of the outlier compared to L_2-norm-based loss function. Thus, $L_{2,1}$-norm-based algorithm could achieve more robust clustering results compared to L_2-norm-based algorithm. The $L_{2,1}$ performs more robustly and stable than L_2 when outliers exist [116]. According to the structure of the constraints, the structural sparsity is often obtained by $L_{2,1}$-norm. $L_{2,1}$-norm regularization encourages multiple predictors to share similar sparsity patterns [115]. $L_{2,1}$-norm-based function is robust to outliers [117, 118].

2.4 Evaluation Measure

To assess the performance of the proposed algorithms with related algorithms, we adopted three popular evaluation metrics of clustering algorithms including accuracy (ACC), normalized mutual information (NMI), and Purity [119]. ACC measures the percentage of samples correctly clustered. NMI measures the pairwise similarity between two partitions. Purity measures the percentage of each cluster containing the correctly clustered samples [11, 120]**.** The definitions of these three evaluation metrics are given below**.**

$$\mathrm{ACC} = N_{correct}/N \tag{2.7}$$

where $N_{correct}$ represents the number of correct clustered samples, and N represents total number of samples.

$$\text{NMI}\ (A,B) = \frac{\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} n_{ij}\ \log(n_{ij}n/n_i^A n_j^B)}{\sqrt{\sum_{i=1}^{C_A} n_i^A \log\ (n_i^A/n)\sum_{j=1}^{C_B} n_j^B \log\ (n_j^B/n)}} \tag{2.8}$$

where A and B represents two partitions of n samples into C_A and C_B clusters respectively.

$$\text{Purity} = \sum_{i=1}^{k}(S_i/n)\,P_i \tag{2.9}$$

where k represents number of clusters and n represents total number of samples. S_i represents the number of samples in the *i*-th cluster. P_i represents the distribution of correctly clustered sample.

To rank the performance of different algorithms, we used dense ranking which the highest accuracy rate receives number 1, and the next accuracy rate receives the immediately following ranking number. Same accuracy rates receive the same ranking number. Thus if A ranks ahead of B and C (which compare equal) which are both ranked ahead of D, then A gets ranking number 1 ("first"), B gets ranking number 2 ("joint second"), C also gets ranking number 2 ("joint second") and D gets ranking number 3 ("Third").

2.5 Summary

As one of the most famous and widely used clustering algorithms, *K*-means clustering algorithm still has its limitations. It is difficult to determine the cluster number *K* to

obtain a good clustering result without prior knowledge. Different initializations may obtain completely different clustering results. Using Euclidian distance as similarity measurement is limited for measuring the real-world data. Real-world data contains redundant features and outliers, without considering the reduction of the influence of redundant features and outliers is hard to achieve the optimal results. Existing methods only solved some of these problems. All these issues of *K*-means clustering algorithm are important to be addressed to improve *K*-means clustering algorithm.

Chapter 3

Initialization-Similarity Clustering Algorithm

3.1 Introduction

Due to random initialization and the Euclidian distance as similarity measure, *K*-means clustering algorithm does not guarantee to produce optimal and stable results. Many literatures have solved the part of issues problem of *K*-means clustering algorithm [4, 13-15, 121, 122]. However, previous research focused on solving a part of these issues but has not focused on solving the initialization and the similarity measure in a unified framework. As an innovative clustering method, spectral clustering algorithm has widely applied in the fields such as data mining, computer vision, machine learning, and pattern recognition over recent years [123, 124]. To fix the similarity measure issue of *K*-means clustering algorithm, Spectral clustering algorithm generates the similarity matrix, and then obtain the spectral representation, finally applies *K*-means clustering algorithm to get the final clustering results. Fixing one of the two issues does not guarantee the best performance. Solving similarity and initialization issues of *K*-means clustering algorithm simultaneously can be considered as an improvement over the existing algorithms because it could lead to better outputs.

The proposed Initialization-Similarity (IS) clustering algorithm aims to solving the initialization and the similarity measure issues simultaneously. Specifically, we fix the initialization of the clustering by using sum-of-norms (SON) regularization [125]. Moreover, the SON regularization outputs the new representation of the original

samples. The proposed IS clustering algorithm then learns the similarity matrix based on the data distribution. That is, the similarity is high if the distance of the new representation of the data points is small. Furthermore, the derived new representation is used to conduct *K*-means clustering. Finally, we employ an alternating strategy to solving the proposed objective function. Experimental results on real-world benchmark data sets demonstrate that IS clustering algorithm outperforms the comparison clustering algorithms in terms of three evaluation metrics for clustering algorithm including accuracy (ACC), normalized mutual information (NMI), and Purity.

We briefly summarize the contributions of the proposed IS clustering algorithm as follows:

- IS clustering algorithm fixes the initialization by using the sum-of-norms regularization makes the clustering robust and reproduced. In contrast, the previous clustering algorithm uses randomly selected cluster centers initialization to conduct *K*-means clustering and then outputs unstable or varying clustering results [126].
- Previous spectral clustering algorithm uses spectral representation to replace original representation for conducting *K*-means clustering. To do this, spectral clustering algorithm first generates the similarity matrix and then conducts eigenvalue decomposition on the Laplacian matrix of the similarity matrix to obtain the spectral representation. This is obviously a two-step strategy which the goal of the first step does not guarantee the best clustering result. However, IS clustering algorithm learns the similarity matrix and the new representation

simultaneously. The performance is more promising when the two steps are combined in a unified way.

- The experiment results on ten public data sets show that the proposed IS clustering algorithm outperforms both *K*-means clustering and spectral clustering algorithms. It implies that simultaneously addressing the two issues of *K*-means clustering algorithm is feasible and fitter.

This section has laid the background of the research inquiry. The remainder of the paper is organized as follows: Section 3.2 discusses the motivation behind the development of IS clustering algorithm. Section 3.3 introduces the proposed Initialization-Similarity (IS) algorithm. Section 3.4 provides the optimization process. Section 3.5 provides the convergence analysis. Section 3.6 discusses the experiments we conducted and presents the results of the experiments. The conclusions, limitations and future research direction are presented in Section 3.7.

3.2 Motivation

To discover how other algorithm improves K-means clustering algorithm, we investigated both K-means clustering algorithm and Spectral clustering algorithm, another widely used clustering algorithm, in details.

K-means algorithm aims at minimizing the total intra-cluster variance represented by an objective function known as the squared error function shown in Eq. (3.1).

$$\sum_{j=1}^{K} \sum_{i=1}^{d_j} \left\| x_i - h_j \right\|^2 \tag{3.1}$$

where K is the cluster number, d_j is the number of data points in the j-th cluster, x_i is the i-th data point of cluster j. h_j is the cluster center of cluster j-th cluster. $\left\| x_i - h_j \right\|^2$ is the Euclidean distance between x_i and h_j.

K-means clustering algorithm can be reformulated as the formulation of nonnegative matrix factorization as following Eq. (3.2) [127]:

$$\min_{\mathbf{H},\mathbf{F}} \|\mathbf{X} - \mathbf{F}\mathbf{H}\|_{\mathrm{F}}^2 \tag{3.2}$$

where $\mathbf{F} \in \mathbb{R}^{n \times k}$ is the cluster indicator matrix of $\mathbf{X} \in \mathbb{R}^{n \times k}$ and $\mathbf{H} \in \mathbb{R}^{k \times \mathrm{d}}$ is the cluster center matrix.

K-means clustering algorithm randomly chooses the initial cluster centers. Based on both Eq. (3.1) and Eq. (3.2) , it is obvious that different initialization methods may have different effects on the clustering results [128, 129]. This implies that it is difficult to reproduce the clustering results. Some algorithms were developed to address this issue. For example, the algorithm used for novel centroid selection approaches for K-means-clustering based recommender systems first select one random data point as initial cluster center, then select next cluster center with probability until all K cluster centers are found. The first cluster center is still selected randomly, which will affect the clustering results [128]. Random swap-based algorithms such as an efficiency of random swap clustering algorithm first select the cluster centers randomly, then randomly select one cluster center to be removed and replace it to a randomly selected cluster. This is a trial-and-error approach and it doesn't have clear iteration times [130].

Moreover, Eq. (3.2) also shows that the outcome of the *K*-means clustering objective function only depends on Euclidean distance between the data points and the cluster center, which is how *K*-means clustering algorithm defines the similarity measure between two data points. The smaller the distance between two data points, the more similar the two data points are. The larger the distance between two data points, the more dissimilar the two data points are. Euclidean distance does not reveal other underlying factors such as cluster sizes, shape, dependent features or density [18, 30]. Thus the similarity measure is an issue of *K*-means clustering algorithm. To address the similarity measure issue of *K*-means algorithm, spectral clustering algorithm uses spectral representation to replace original representation. To achieve this, spectral clustering algorithm first builds a similarity matrix and conducts eigenvalue decomposition on its Laplacian matrix to obtain the spectral representation. The pseudo code for *K*-means clustering algorithm is list in Table 3.1.

Table 3.1 The pseudo code for *K*-means clustering algorithm

Input: **X** (data matrix), *K* (the cluster number)

Output: *K* cluster centers and the cluster indicator of each data point

Initialization:

Random selecting *K* cluster centers $h_1, h_2 \ldots h_k$;

Repeat:

1. Assign each data point x_i to nearest cluster *j* using Euclidian distance;
2. Recalculating the new cluster centers $h_1, h_2 \ldots h_k$;

Until convergence (the cluster indicator of each data points unchanged);

A spectral clustering algorithm creates a similarity matrix first and then defines a feature vector. Then it runs the K-means clustering algorithm to conduct clustering

[61]. Thus, a spectral clustering algorithm finds the data similarity matrix and spectral representation in separate stages. Of course, its use of the K-means clustering algorithm requires the cluster number beforehand. Other algorithms e.g. CAN learn the data similarity matrix and clustering structure simultaneously, but again needs to know the cluster number beforehand. In the algorithm RCC, clustering is managed without the prior knowledge of the cluster number by continuously optimizing an objective function based on robust estimation. However, this needs a good similarity matrix calculated beforehand as an input to be able to produce good clustering outcome. The pseudo code for spectral clustering algorithm is shown in Table 3.2.

Table 3.2 The pseudo code for the spectral clustering algorithm

Input: $\mathbf{X} \in \mathbb{R}^{n \times d}$ (data matrix), K (the cluster number)
Output: K cluster center and the cluster indicator of each data point
• Computing $\mathbf{S} \in \mathbb{R}^{n \times n}$ to measure the similarity between any data point pair; • Computing $\mathbf{L} = \mathbf{D} - \mathbf{S}$, where $\mathbf{D} = [d_{ij}]_{n \times n}$ is a diagonal matrix and $d_{ii} = \sum_j (s_{ij} + s_{ji})/2$; • Generating spectral representation using the eigenvectors and eigenvalues of $\mathbf{L}$; • Conducting K-means clustering on the spectral representation;

Obviously, spectral clustering algorithm replacing original representation with spectral representation deals the issue of similarity measure in *K*-means clustering algorithm. However, spectral clustering algorithm separately learns the similarity matrix and the spectral representation, as knowns as a two-stage strategy, where the goal of constructing the similarity matrix in the first stage does not aim at achieving

optimal spectral representation, and thus not guaranteeing to always outperform *K*-means clustering algorithm.

3.3 Proposed Algorithm

This thesis proposes a new clustering algorithm (i.e., Initialization-Similarity (IS)) to simultaneously solve the initialization and similarity measure issues of *K*-means clustering algorithm in a unified framework. Specifically, IS clustering algorithm uses the sum-of-norms regularization to investigate the initialization issue, and jointly learns the similarity matrix and the spectral representation to overcome the issue of the multi-stage strategy of spectral clustering algorithm. To achieve the goal, we form the objective function of the IS clustering algorithm as follows:

$$\min_{\mathbf{S},\mathbf{U}} \frac{1}{2}\|\mathbf{X}-\mathbf{U}\|_F^2 + \frac{\alpha}{2}\sum_{i,j=1}^{n} s_{i,j}\rho(\left\|\mathbf{u}_i-\mathbf{u}_j\right\|_2) + \beta\|\mathbf{S}\|_2^2,\ s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T\mathbf{e} = 1 \tag{3.3}$$

where $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the data matrix, $\mathbf{U} \in \mathbb{R}^{n\times d}$ is the new representation of $\mathbf{X}$, and $\mathbf{S} \in \mathbb{R}^{n\times n}$ is the similarity matrix to measure the similarity among data points. $\rho(\left\|\mathrm{u}_i-\mathrm{u}_j\right\|_2)$ is an implicit function, as known as robust loss function in robust statistics.

Equation. (3.3) aims at learning the new representation **U** and fixes the initialization of clustering. Moreover, Eq. (3.3) learns the new representation **U** as well as considers the similarity among data points, i.e., the higher the similarity $s_{i,j}$ between two data points, the smaller their corresponding new representation ($\mathbf{u}_i$ and $\mathbf{u}_j$) is.

Furthermore, we learn the similarity matrix **S** based on the sample distribution, i.e., iteratively updated by the updated **U**. This makes the new representation reasonable.

Several robust loss functions have been proposed in robust statistics [131, 132]. In this thesis, we employ the Geman-McClure function [133] as follows:

$$\rho\left(\left\|\mathbf{u}_p-\mathbf{u}_q\right\|_2\right)=\frac{\mu\left\|\mathbf{u}_p-\mathbf{u}_q\right\|_2^2}{\mu+\left\|\mathbf{u}_p-\mathbf{u}_q\right\|_2^2} \tag{3.4}$$

Equation. (3.4) is often used to measure how good a prediction model does in terms of being able to predict the expected outcome. The closer the distance is, the smaller value of $\left\|\mathbf{u}_p-\mathbf{u}_q\right\|_2$ is, and the higher the similarity $s_{p,q}$ is. With the update of other parameters in Eq. (3.3), the distance $\left\|\mathbf{u}_p-\mathbf{u}_q\right\|_2$ for some p, q, will be very close, or even $\mathbf{u}_p=\mathbf{u}_q$. In this way, the clusters will be determined.

Algorithm 3.1. The pseudo code for IS clustering algorithm.

Input: $\mathbf{X}\in\mathbb{R}^{n\times d}$

Output: a set of *K* clusters

Initialization: **U** = **X**;

Repeat:

- Update **F** using Eq. (3.13)
- Update **S** using Eq. (3.22)
- Update **U** using Eq. (3.36)

Until U converges

- Apply *K*-means clustering algorithm on **U**

In robust statistics, the optimization of the robust loss function is usually difficult or inefficient. To address this, it is normal for introducing an auxiliary variable $f_{i,j}$ and a penalty item $\varphi(f_{i,j})$ [134-136], and thus Eq. (3.3) is equivalent to:

$$\min_{\mathbf{S},\mathbf{U},\mathbf{F}}\frac{1}{2}\sum_{i=1}^{n}\|\mathbf{x}_i-\mathbf{u}_i\|_2^2+\frac{\alpha}{2}\sum_{i,j=1}^{n}s_{i,j}(f_{i,j}\left\|\mathbf{u}_i-\mathbf{u}_j\right\|_2^2$$

$$+\varphi(f_{i,j})) + \beta \sum_{i=1}^{n} \|\mathbf{s}_i\|_2^2 \;\; s.t., \forall i, s_{i,j} \geq 0, \boldsymbol{s}_i^T \boldsymbol{e} = 1 \tag{3.5}$$

where $\varphi(f_{i,j}) = \mu(\sqrt{f_{i,j}} - 1)^2, i, j = 1 \ldots n$

3.4 Optimization

Equation. (3.5) is not jointly convex on **F**, **U**, and **S**, but is convex on each variable while fixing the rest. To solving the Eq. (3.5), the alternating optimization strategy is applied. We optimize each variable while fixing the rest until the algorithm converges. The pseudo-code of IS clustering algorithm is given in Algorithm 3.1.

1) Update F while fixing S and U

While **S** and **U** are fixed, the objective function can be rewritten in a simplified matrix form to optimize **F**:

$$\min_{\mathbf{F}} \frac{\alpha}{2} \sum_{i,j=1}^{n} s_{i,j} (f_{i,j} \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 + \mu(\sqrt{f_{i,j}} - 1)^2) \tag{3.6}$$

Since the optimization of $f_{i,j}$ is independent of the optimization of other $f_{p,q}, i \neq p, j \neq q$, the $f_{i,j}$ is optimized first as shown in following Eq. (3.7)

$$\min_{f_{i,j}} \frac{\alpha}{2} (s_{i,j} f_{i,j} \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 + s_{i,j} (\mu(f_{i,j} - 2\sqrt{f_{i,j}} + 1)) \tag{3.7}$$

By conducting a derivative on Eq. (3.7) with respect to $f_{i,j}$, we get

$$\frac{\alpha}{2} (s_{i,j} \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 + s_{i,j}\mu - s_{i,j}\mu f_{i,j}^{-\frac{1}{2}}) = 0 \tag{3.8}$$

$$\Rightarrow \frac{\alpha}{2} s_{i,j} \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 + \frac{\alpha}{2} s_{i,j}\mu - \frac{\alpha}{2} s_{i,j}\mu f_{i,j}^{-\frac{1}{2}} = 0 \tag{3.9}$$

$$\Rightarrow \frac{\alpha}{2} s_{i,j}\left\|\mathbf{u}_i - \mathbf{u}_j\right\|_2^2 + \frac{\alpha}{2} s_{i,j}\mu = \frac{\alpha}{2} s_{i,j}\mu {f_{i,j}}^{-\frac{1}{2}} \tag{3.10}$$

$$\Rightarrow {f_{i,j}}^{-\frac{1}{2}} = \frac{\left\|\mathbf{u}_i - \mathbf{u}_j\right\|_2^2 + \mu}{\mu} \tag{3.11}$$

$$\Rightarrow {f_{i,j}}^{\frac{1}{2}} = \frac{\mu}{\mu + \left\|\mathbf{u}_i - \mathbf{u}_j\right\|_2^2} \tag{3.12}$$

$$\Rightarrow f_{i,j} = \left(\frac{\mu}{\mu + \left\|\mathbf{u}_i - \mathbf{u}_j\right\|_2^2}\right)^2 \tag{3.13}$$

2) Update S while fixing U and F

While fixing **U** and **F**, the objective function Eq. (3.5) with respect to **S** is:

$$\min_{\mathbf{S}} \frac{\alpha}{2}\sum_{i,j=1}^{n}(s_{i,j}f_{i,j}\left\|\mathbf{u}_i - \mathbf{u}_j\right\|_2^2 + s_{i,j}(\mu(\sqrt{f_{i,j}} - 1)^2)) + \beta\sum_{i=1}^{n}\|\mathbf{s}_i\|_2^2 \tag{3.14}$$

$$s.t., \forall i, s_{i,j} \geq 0, \mathrm{s}_i^T \mathrm{e} = \mathrm{I}$$

Since the optimization of $\mathbf{s}_\mathrm{i}$ is independent of the optimization of other $\mathbf{s}_j, i \neq \mathrm{j}, i, j = 1, \ldots, \mathrm{n}$, the $\mathbf{s}_i$ is optimized first as shown in following:

$$\min_{\mathbf{s}_i} \frac{\alpha}{2}\sum_{j=1}^{n} s_{i,j}(f_{i,j}\left\|\mathbf{u}_i - \mathbf{u}_j\right\|_2^2 + \mu(\sqrt{f_{i,j}} - 1)^2) + \beta\|\mathbf{s}_i\|_2^2 \tag{3.15}$$

$$s.t., \forall i, s_{i,j} \geq 0, \mathrm{s}_i^T \mathrm{e} = 1$$

Let $b_{i,j} = f_{i,j}\left\|\mathbf{u}_i - \mathbf{u}_j\right\|_2^2$ and $c_{i,j} = \mu(\sqrt{f_{i,j}} - 1)^2$, Eq. (3.15) is equivalent to:

$$\min_{\mathbf{s}_i} \frac{\alpha}{2}\sum_{j=1}^{n} s_{i,j}b_{i,j} + \frac{\alpha}{2}\sum_{j=1}^{n} s_{i,j}c_{i,j} + \beta\|\mathbf{s}_i\|_2^2, s.t., \forall i, s_{i,j} \geq 0, \boldsymbol{s}_i^T e = 1 \tag{3.16}$$

$$\Rightarrow \min_{\mathbf{s}_i} \frac{\alpha}{2}\mathbf{s}_i^T\mathbf{b}_i + \frac{\alpha}{2}\mathbf{s}_i^T\mathbf{c}_i + \beta\|\mathbf{s}_i\|_2^2, \quad s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T e = 1 \tag{3.17}$$

$$\Rightarrow \min_{\mathbf{s}_i} \frac{\alpha}{2}\mathbf{s}_i^T(\mathbf{b}_i + \mathbf{c}_i) + \beta\mathbf{s}_i^T\mathbf{s}_i, \quad s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T e = 1 \tag{3.18}$$

$$\Rightarrow \min_{\mathbf{s}_i} \frac{\alpha}{2\beta}\mathbf{s}_i^T(\mathbf{b}_i + \mathbf{c}_i) + \mathbf{s}_i^T\mathbf{s}_i, \quad s.t., \forall i, s_{i,j} \geq 0, \mathrm{s}_i^T e = 1 \tag{3.19}$$

$$\Rightarrow \min_{\mathbf{s}_i} \mathbf{s}_i^T\mathbf{s}_i + 2\mathbf{s}_i\frac{\alpha}{4\beta}\mathbf{s}_i^T(\mathbf{b}_i + \mathbf{c}_i) + \frac{\alpha}{4\beta}\mathbf{s}_i^T(\mathbf{b}_i + \mathbf{c}_i)^T(\mathbf{b}_i + \mathbf{c}_i)$$

$$- \frac{\alpha}{4\beta}\mathrm{s}_i^T(\mathbf{b}_i + \mathbf{c}_i)^T(\mathbf{b}_i + \mathbf{c}_i), s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T e = 1 \tag{3.20}$$

$$\Rightarrow \min_{\mathbf{s}_i} \left\|\mathbf{s}_i + \frac{\alpha}{4\beta}(\mathbf{b}_i + \mathbf{c}_i)\right\|_2^2, s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T \mathrm{e} = 1 \tag{3.21}$$

According to Karush-Kuhn-Tucker (KKT) [137], the optimal solution $\mathbf{s}_i$ should be

$$\mathrm{S}_{i,j} = \max\{-\frac{\alpha}{4\beta}(\mathrm{b}_{i,j} + \mathrm{c}_{i,j})\} + \theta, 0\}, j = 1, \dots, n \tag{3.22}$$

where $\theta = \frac{1}{\rho}\sum_{j=1}^{\rho}\left(\frac{\alpha}{4\beta}(\mathrm{b}_{i,j} + \mathrm{c}_{i,j}) + 1\right)$, and ω is the descending order of $\frac{\alpha}{4\beta}(\mathrm{b}_{i,j} + \mathrm{c}_{i,j})$. and $\rho = \max_j\{\boldsymbol{\omega}_j - \frac{1}{j}\left(\sum_{r=1}^{j}\boldsymbol{\omega}_r - 1\right), 0\}$.

3) Update U while fixing S and F

While **S** and **F** are fixed, the objective function can be rewritten in a simplified form to optimize **U**:

$$\min_{\mathbf{U}} \frac{1}{2}\sum_{i,j=1}^{n}\|\mathbf{x}_i - \mathbf{u}_i\|_2^2 + \frac{\alpha}{2}\sum_{i,j=1}^{n}s_{i,j}f_{i,j}\left\|\mathbf{u}_i - \mathbf{u}_j\right\|_2^2 \tag{3.23}$$

Let $h_{i,j} = s_{i,j}f_{i,j}$. Eq. (3.23) is equivalent to:

$$\min_{\mathbf{U}} \frac{1}{2}\|\mathbf{X}-\mathbf{U}\|_F^2 \ + \frac{\alpha}{2}\sum_{i,j=1}^{n} h_{i,j}\left\|\mathbf{u}_i-\mathbf{u}_j\right\|_2^2 \tag{3.24}$$

$$\Rightarrow \min_{\mathbf{U}} \frac{1}{2}\|\mathbf{X}-\mathbf{U}\|_F^2 \ + \frac{\alpha}{2} tr(\mathbf{U}^T\mathbf{L}\mathbf{U}) \tag{3.25}$$

$$\Rightarrow \min_{\mathbf{U}} \frac{1}{2} tr((\mathbf{X}-\mathbf{U})^T(\mathbf{X}-\mathbf{U})) + \frac{\alpha}{2} tr(\mathbf{U}^T\mathbf{L}\mathbf{U}) \tag{3.26}$$

$$\Rightarrow \min_{\mathbf{U}} \frac{1}{2} tr((\mathbf{X}^T-\mathbf{U}^T)(\mathbf{X}-\mathbf{U})) + \frac{\alpha}{2} tr(\mathbf{U}^T\mathbf{L}\mathbf{U}) \tag{3.27}$$

$$\Rightarrow \min_{\mathbf{U}} \frac{1}{2} tr(\mathbf{X}^T\mathbf{X}-2\mathbf{U}^T\mathbf{X}+\mathbf{U}^T\mathbf{U}) \ + \frac{\alpha}{2} tr(\mathbf{U}^T\mathbf{L}\mathbf{U}) \tag{3.28}$$

After conducting a derivative on Eq. (3.28) with respect to **U**, we get

$$\Rightarrow \frac{1}{2}(-2\mathbf{X}+2\mathbf{U}) \ + \frac{\alpha}{2}(\mathbf{L}\mathbf{U}+\mathbf{L}^T\mathbf{U}) = 0 \tag{3.29}$$

$$\Rightarrow -\mathbf{X}+\mathbf{U}+\frac{\alpha}{2}\mathbf{L}\mathbf{U} + \frac{\alpha}{2}\mathbf{L}^T\mathbf{U} = 0 \tag{3.30}$$

$$\Rightarrow \mathbf{U}+\frac{\alpha}{2}\mathbf{L}\mathbf{U} + \frac{\alpha}{2}\mathbf{L}^T\mathbf{U} = \mathbf{X} \tag{3.31}$$

$$\Rightarrow (1+\frac{\alpha}{2}\mathbf{L}+\frac{\alpha}{2}\mathbf{L}^T)\mathbf{U} = \mathbf{X} \tag{3.32}$$

$$\Rightarrow (1+\frac{\alpha}{2}(\mathbf{L}+\mathbf{L}^T)\mathbf{U} = \mathbf{X} \tag{3.33}$$

$$\Rightarrow (1+\frac{\alpha}{2}(2\mathbf{L})\mathbf{U} = \mathbf{X} \tag{3.34}$$

$$\Rightarrow (1+\alpha\mathbf{L})\mathbf{U} = \mathbf{X} \tag{3.35}$$

$$\Rightarrow \mathbf{U} = (\mathrm{I}+\alpha\mathbf{L})^{-1}\mathbf{X} \tag{3.36}$$

3.5 Convergence Analysis

In this section, we prove the convergence of the proposed IS clustering algorithm in order to prove the proposed algorithm can reach at least a locally optimal solution, so we apply Theorem 1.

Theorem 1. IS clustering algorithm decreases the objective function value of Eq. (3.5) until it converges.

Proof.

By denoting $\mathbf{F}^{(t)}$, $\mathbf{S}^{(t)}$, and $\mathbf{U}^{(t)}$, the results of the *t*-th iteration of $\mathbf{F}$, $\mathbf{S}$, and $\mathbf{U}$ respectively, we further denote the objective function value of Eq. (3.5) in the *t*-th iteration as $\mathcal{L}\left(\mathbf{F}^{(t)}, \mathbf{S}^{(t)}, \mathbf{U}^{(t)}\right)$.

According to Eq. (3.13) in Section 3.4, $\mathbf{F}$ has a closed-form solution, thus we have the following inequality:

$$\mathcal{L}\left(\mathbf{F}^{(t)}, \mathbf{S}^{(t)}, \mathbf{U}^{(t)}\right) \geq \mathcal{L}\left(\mathbf{F}^{(t+1)}, \mathbf{S}^{(t)}, \mathbf{U}^{(t)}\right) \tag{3.37}$$

According to Eq. (3.22), $\mathbf{S}$ has a closed-form solution, thus we have the following inequality:

$$\mathcal{L}\left(\mathbf{F}^{(t+1)}, \mathbf{S}^{(t)}, \mathbf{U}^{(t)}\right) \geq \mathcal{L}\left(\mathbf{F}^{(t+1)}, \mathbf{S}^{(t+1)}, \mathbf{U}^{(t)}\right) \tag{3.38}$$

According to Eq. (3.36), $\mathbf{U}$ has a closed-form solution, thus we have the following inequality:

$$\mathcal{L}\left(\mathbf{F}^{(t+1)}, \mathbf{S}^{(t+1)}, \mathbf{U}^{(t)}\right) \geq \mathcal{L}\left(\mathbf{F}^{(t+1)}, \mathbf{S}^{(t+1)}, \mathbf{U}^{(t+1)}\right) \tag{3.39}$$

Finally, based on above three inequalities, we get

$$\mathcal{L}\left(\mathbf{F}^{(t)}, \mathbf{S}^{(t)}, \mathbf{U}^{(t)}\right) \geq \mathcal{L}\left(\mathbf{F}^{(t+1)}, \mathbf{S}^{(t+1)}, \mathbf{U}^{(t+1)}\right) \tag{3.40}$$

Equation. (3.40) indicates that the objective function value in Eq. (3.5) decreases after each iteration of Algorithm 3.1. This concludes the proof of Theorem 1.

3.6 Experiments

In this section, we evaluated the performance of the proposed Initialization-Similarity (IS) algorithm, by comparing it with two benchmark algorithms on ten real UCI data sets, in terms of three evaluation metrics [138].

Table 3.3 Description of ten benchmark data sets

Datasets	Samples	Dimensions	Classes
Digital	1797	64	10
MSRA	1799	256	12
Segment	2310	19	7
Solar	323	12	6
USPS	1854	256	10
USPST	2007	256	10
Waveform	5000	21	3
Wine	178	13	3
Wireless	2000	7	4
Yale	165	1024	15

3.6.1 Data Sets

We used ten UCI data sets in the experiments, including the standard data sets for handwritten digit recognition, face data sets, and wine data sets, etc. The details are listed in the following and summarization provide in Table 3.3.

- *Digital* data set is made up of 1797 images (8x8). Each image is a hand-written digit 1-10.
- *MSRA* data set is a face image data set.
- *Segment* contains the instances drawn randomly from a database of 7 outdoor images. It has 2310 instances and 19 continuous attributes describing the images including saturation, Hue, etc.
- *Solar* data set describes the main characteristics of the solar flare.
- *USPS* is one of the standard handwritten digit recognition data sets. It contains the images of number from 0 to 9.
- *USPST* contains 2007 handwritten digit recognition data sets.
- *Waveform* data set has 5000 instances and 3 classes of waves with 21 attributes.
- *Wine* data set is the results of a chemical analysis of wines with three different cultivars. It contains data of 13 constituents found in each of the three types of wines.
- *Wireless* data set collected 2000 instances of the signal strengths of seven WiFi signals visible on a smartphone.
- *Yale* data set contains 165 grayscale images (32x32) of 15 individuals. Each subject has different facial expression or configuration. The decision variable is one of the four rooms.

3.6.2 Comparison Algorithms

Two comparison algorithms are classical clustering algorithms and their details were summarized below.

- *K*-means clustering algorithm (re)assigns data points to their nearest cluster center and recalculates cluster centers iteratively with a goal to minimize the sum of distances between data points and cluster center.
- Spectral clustering algorithm first forms the similarity matrix, and then calculates the first *K* eigenvectors of its Laplacian matrix to define feature vectors. Finally, it runs *K*-means clustering on these features to separate objects into *K* classes. There are different ways to calculate the Laplacian matrix. Instead of using simple Laplacian, we used normalized Laplacian $\mathbf{L} = \mathbf{D} \times \mathbf{L} \times \mathbf{D}$, which have better performance than using simple Laplacian [139].

For the above two algorithms, *K*-means clustering conducts clustering directly on the original data while spectral clustering is a multi-stage based strategy, which constructs a graph first and then applies *K*-means clustering algorithm to partition the graph.

3.6.3 Experiment Setup

In the experiments, firstly, we tested the robustness of the proposed IS clustering algorithm by comparing it with *K*-means clustering and spectral clustering algorithms using real data sets in terms of three evaluation metrics widely used for clustering

research. Due to the sensitivity of *K*-means clustering to its initial cluster centers, we ran *K*-means clustering and spectral clustering algorithms 20 times and chose the average value as the final result. Secondly, we investigated the parameters' sensitivity of the proposed IS clustering algorithm (i.e. α and β in Eq. (3.5)) via varying their values to observe the variations of clustering performance. Thirdly, we demonstrated the convergence of Algorithm 3.1 to solving the proposed objective function Eq. (3.5) via checking the iteration times when Algorithm 3.1 converges.

3.6.4 Experimental Results Analysis

We listed the clustering performance of all algorithms in Table 3.5, which shows that our IS clustering algorithm achieved the best performance on all ten data sets in terms of ACC and NMI, as well as outperformed *K*-means clustering algorithm on all ten data sets in terms of Purity. IS clustering algorithm outperformed spectral clustering algorithm on all eight data sets in terms of Purity but performed slightly worse than spectral clustering algorithm on three data sets USPT, USPST and Yale. The difference in Purity results between IS clustering algorithm and the spectral clustering algorithm was only 1%. More specifically, IS clustering algorithm increased ACC by 6.3% compared to *K*-means clustering algorithm and 3.3% compared to spectral clustering algorithm. IS clustering algorithm increased NMI by 4.6% compared to *K*-means clustering algorithm and 4.5% compared to spectral clustering algorithm. IS clustering algorithm increased Purity by 4.9% compared to *K*-means clustering algorithm and

2.9% compared to spectral clustering algorithm. Other observations were listed in the following sections.

First, both one-step clustering algorithm, e.g. IS clustering algorithm and two-step clustering algorithm, e.g. spectral clustering algorithm outperformed *K*-means clustering algorithm. This implied that constructing the graph or learning a new representation of original data points improved the clustering performance. This means that using new representation can generate better clustering than the methods using original data in clustering tasks. The reason could be that original data generally contains more or less redundant information, which is always true in real data set and the redundancy undoubtedly corrupts the performance of clustering models. In contrast, two similarity matrix-based methods construct the new representation based on original data to conduct clustering, which can relieve the affection of redundancy from original data, so the clustering performance can be improved.

Second, one-step clustering algorithm, e.g. IS clustering algorithm, performed better than two-step clustering algorithms, e.g. spectral clustering algorithm. Compared to the spectral clustering algorithm that first uses the original data to construct the similarity matrix and then uses the orthogonal decomposition onto the similarity matrix to output new representation, our method employed an adaptive learning strategy to dynamically update the similarity matrix and new representation in a unified framework. In this way, both new representation and similarity of our method can capture the intrinsic correlation of data, which means our method can easily output better clustering results than classical spectral clustering methods. This proves that the

goal of the similarity matrix learning and the new representation are the same which leads to optimal clustering results, whereas the two-step clustering algorithm with separate goals achieves sub-optimal results.

Table 3.4 ACC results of IS algorithm on ten benchmark data sets

The highest score of each evaluation metric for each data set is highlighted in bold font.

Datasets	*K*-means	Spectral	IS
Digital	0.73	0.77	**0.80**
MSRA	0.49	0.50	**0.57**
Segment	0.55	0.56	**0.63**
Solar	0.50	0.51	**0.55**
USPS	0.62	0.67	**0.70**
USPST	0.66	0.70	**0.71**
Waveform	0.50	0.51	**0.57**
Wine	0.65	0.69	**0.71**
Wireless	0.94	0.96	**0.97**
Yale	0.39	0.45	**0.46**
Rank	3	2	**1**

Table 3.5 NMI results of IS algorithm on ten benchmark data sets

The highest score of each evaluation metric for each data set is highlighted in bold font.

Datasets	*K*-means	Spectral	IS
Digital	0.73	0.72	**0.78**
MSRA	0.59	0.56	**0.63**
Segment	0.61	0.52	**0.63**
Solar	0.34	0.34	**0.42**
USPS	0.61	0.66	**0.70**
USPST	0.61	0.66	**0.68**
Waveform	0.36	0.37	**0.40**
Wine	**0.43**	0.42	**0.43**
Wireless	0.88	0.89	**0.91**
Yale	0.47	0.51	**0.51**
Rank	2	2	**1**

Table 3.6 Purity results of IS algorithm on ten benchmark data sets

The highest score of each evaluation metric for each data set is highlighted in bold font

Datasets	*K*-means	Spectral	IS
Digital	0.76	0.78	**0.81**
MSRA	0.53	0.53	**0.58**
Segment	0.58	0.58	**0.64**
Solar	0.55	0.55	**0.61**
USPS	0.69	**0.75**	0.74
USPST	0.71	**0.77**	0.76
Waveform	0.53	0.51	**0.59**
Wine	0.69	0.69	**0.71**
Wireless	0.94	0.96	**0.97**
Yale	0.41	**0.47**	0.46
Rank	3	2	**1**

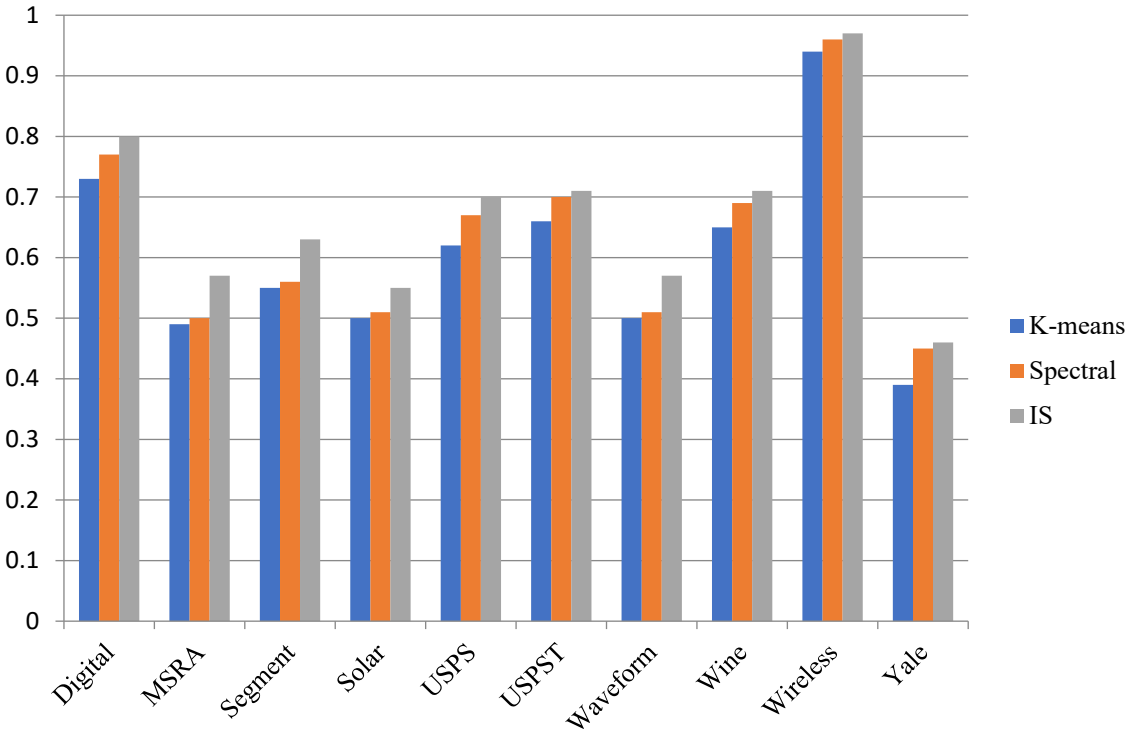


Figure 3.1 ACC results of IS algorithm on ten benchmark data sets

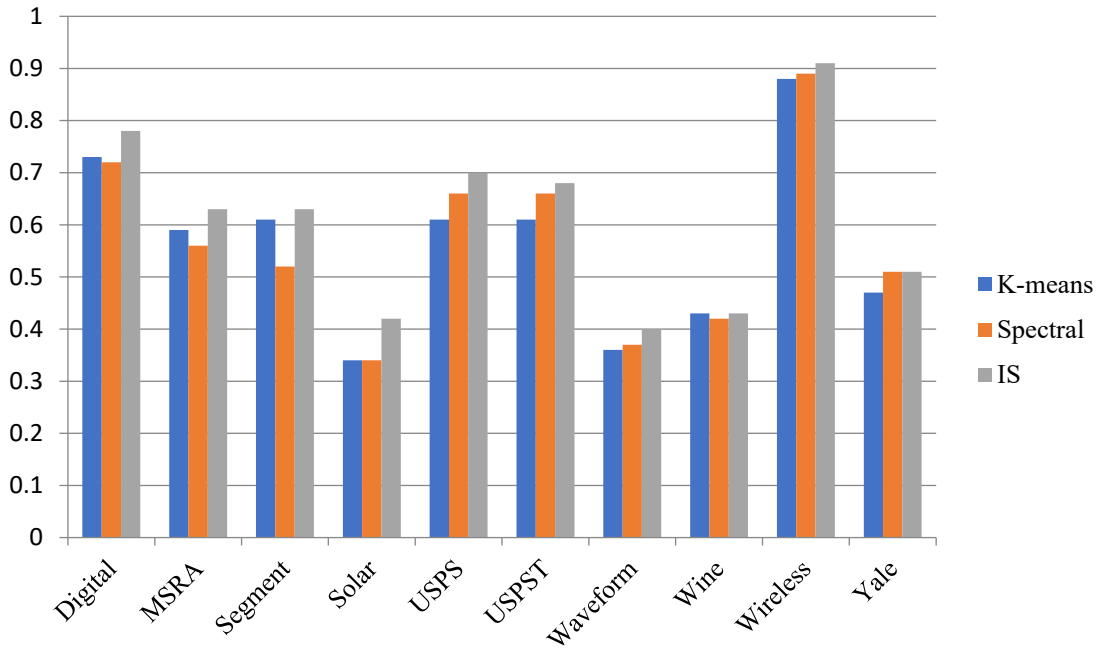


Figure 3.2 NMI results of IS algorithm on ten benchmark data sets

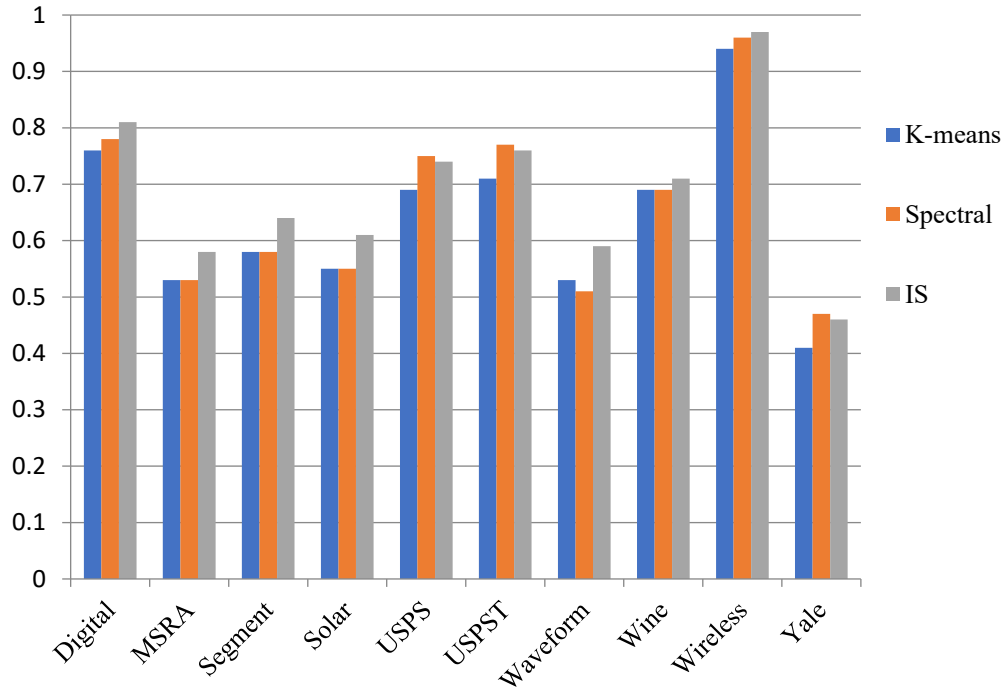


Figure 3.3 Purity results of IS algorithm on ten benchmark data sets

3.6.5 Parameters' Sensitivity

We varied parameters α and β in the range of $[10^{-2},\dots 10^{2}]$, and recorded the values of ACC, NMI and Purity of ten data sets clustering results for IS clustering algorithm in Figures 3.4-3.6.

First, different data sets needed different ranges of parameters to achieve the best performance. For example, IS clustering algorithm achieved the best ACC (97%), NMI (91%) and Purity (97%) on data set Wireless when both parameters α and β were 10. But for the data set Digital, IS clustering algorithm achieved the best ACC (80%), NMI (78%) and Purity (81%) when β = 100 and α =0.1. This indicated that IS clustering algorithm was data-driven.

Second, the clustering ACC results had less than 3% average changes when the parameter α varied in the range of $[10^{-2},\dots 10^{2}]$ in eight out of ten data sets. The lowest average change was 1% (i.e., Wine and Wireless data sets) when the parameter α varied in the range of $[10^{-2},\dots 10^{2}]$. The biggest average change was 5% (e.g., Waveform data set) when the parameter α varied in the range of $[10^{-2},\dots 10^{2}]$. This indicated that IS clustering algorithm was not very sensitive to the parameter α.

Third, the clustering ACC results had less than 3% average changes when the parameter β varied in the range of $[10^{-2},\dots 10^{2}]$ in nine out of ten data sets. The lowest average change was 0 (Wine data set) when the parameter β varied in the range $[10^{-2},\dots 10^{2}]$. The biggest average change was 5% (Waveform data set) when the parameter β varied in the range of $[10^{-2},\dots 10^{2}]$. This indicated that IS clustering algorithm was not very sensitive to the parameter β.

Fourth, even IS clustering algorithm was not very sensitive on parameters α and β, the algorithm was slightly more sensitive on parameter α than it was on the parameter β.

3.6.6 Convergence

Figure. 3.7 showed the trend of objective values generated by the proposed algorithm 3.1 with respect to iterations. The convergence curve indicates the change of the objective function value during the iteration process. From Figure. 3.7, we can see that the algorithm 3.1 monotonically decreased the objective function value until it converged, when applying it to optimize the proposed objective function in Eq. (3.5). That means that the value of the objective function stop changing or only change in a small range e.g. $\left|obj_{(t+1)} - obj_{(t)}\right| / obj_{(t)} \leq 10^{-9}$, At this point, we can obtain the solution. In our proposed optimization algorithm, we have employed an alternating optimization strategy to optimize our objective function, i.e., iteratively updating each parameter until the algorithm converges. Thus, the optimal solution can be worked out by multiple iterations until the demand of minimizing the objective values is satisfied, which means the objective values decline to stable, as shown as the convergence lines. It is worth noting that the convergence rate of the algorithm 3.1 was relatively fast, converging to the optimal value within 20 iterations on all the data sets used. In other words, we can complete the optimization of our model in a fast speed.

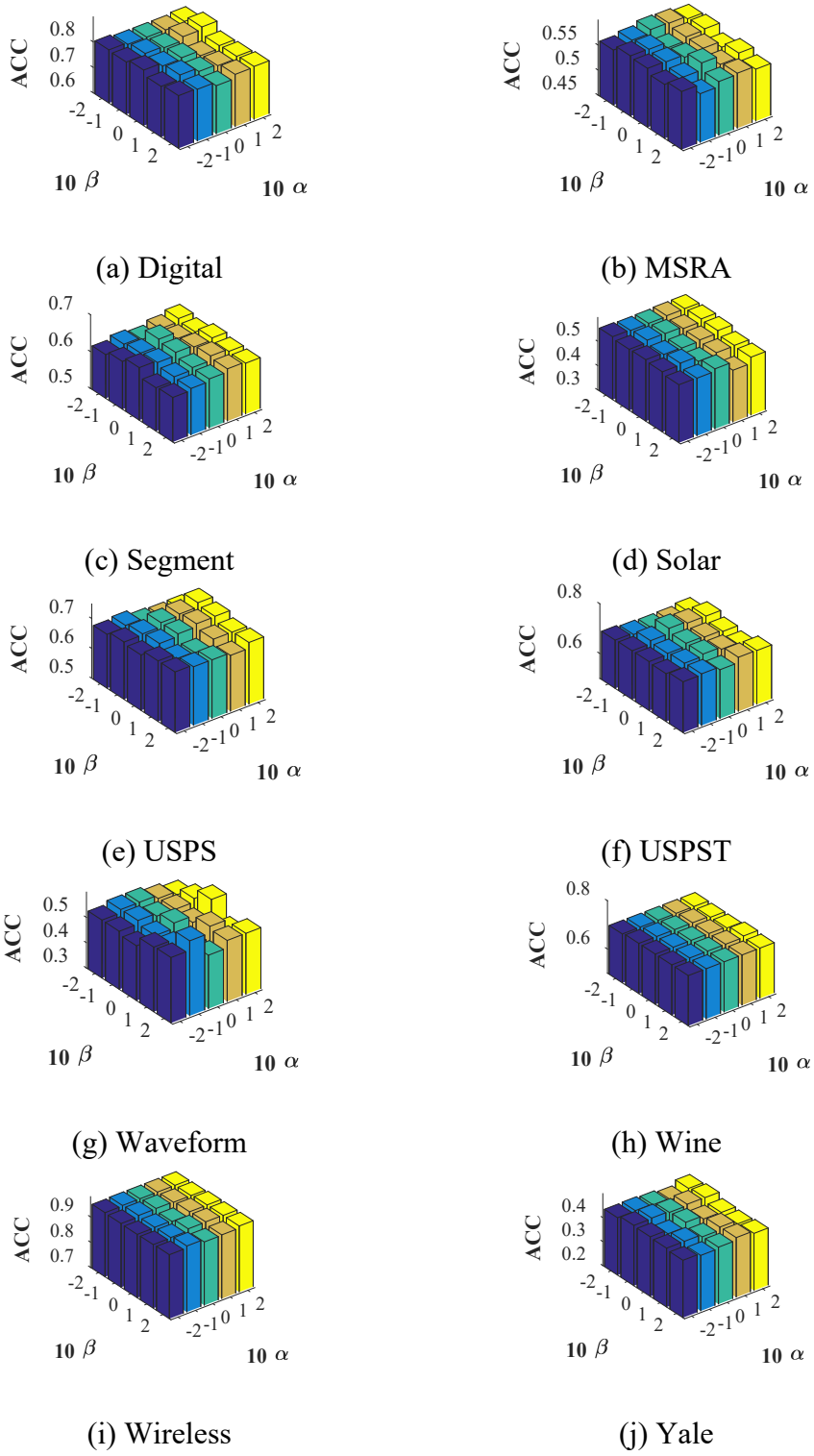


Figure 3.4 ACC results of IS algorithm with respect to different parameter settings

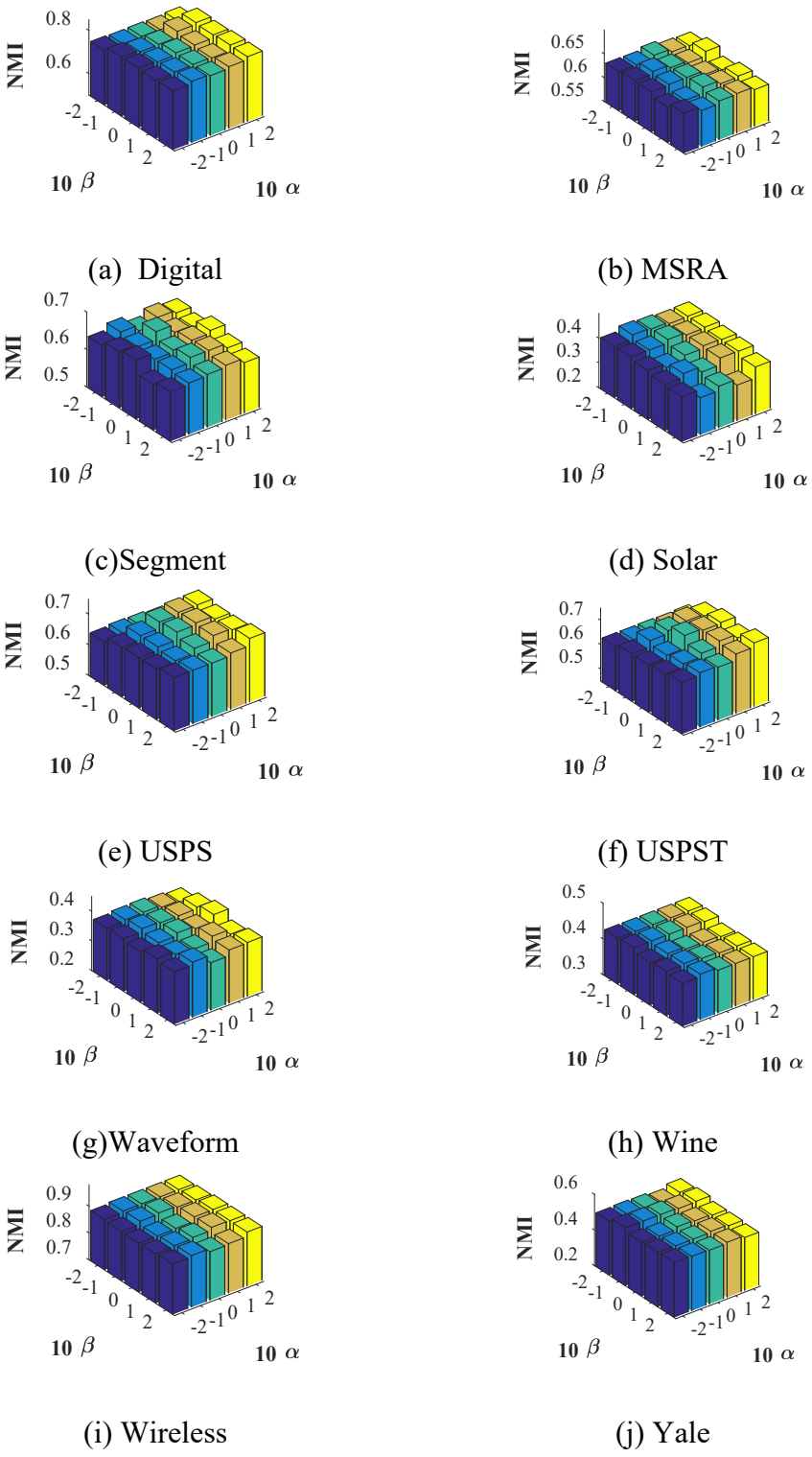
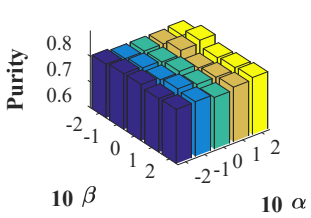
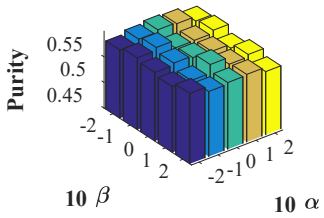


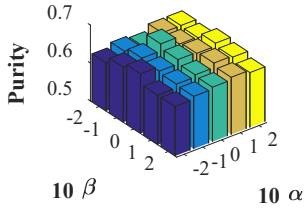
Figure 3.5 NMI results of IS algorithm with respect to different parameter settings



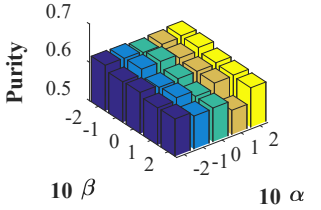
(a) Digital



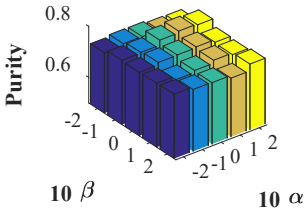
(b) MSRA



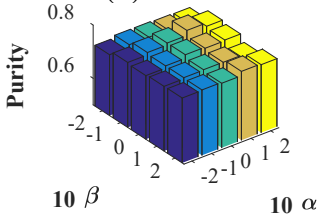
(c)Segment



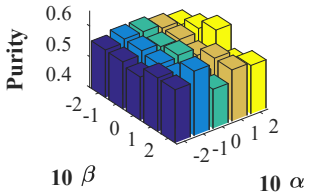
(d) Solar



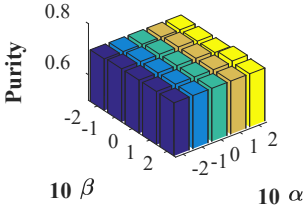
(e) USPS



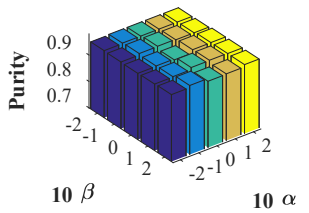
(f) USPST



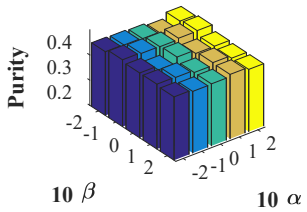
(g)Waveform



(h) Wine



(i) Wireless



(j)Yale

Figure 3.6 Purity results of IS algorithm with respect to different parameter settings

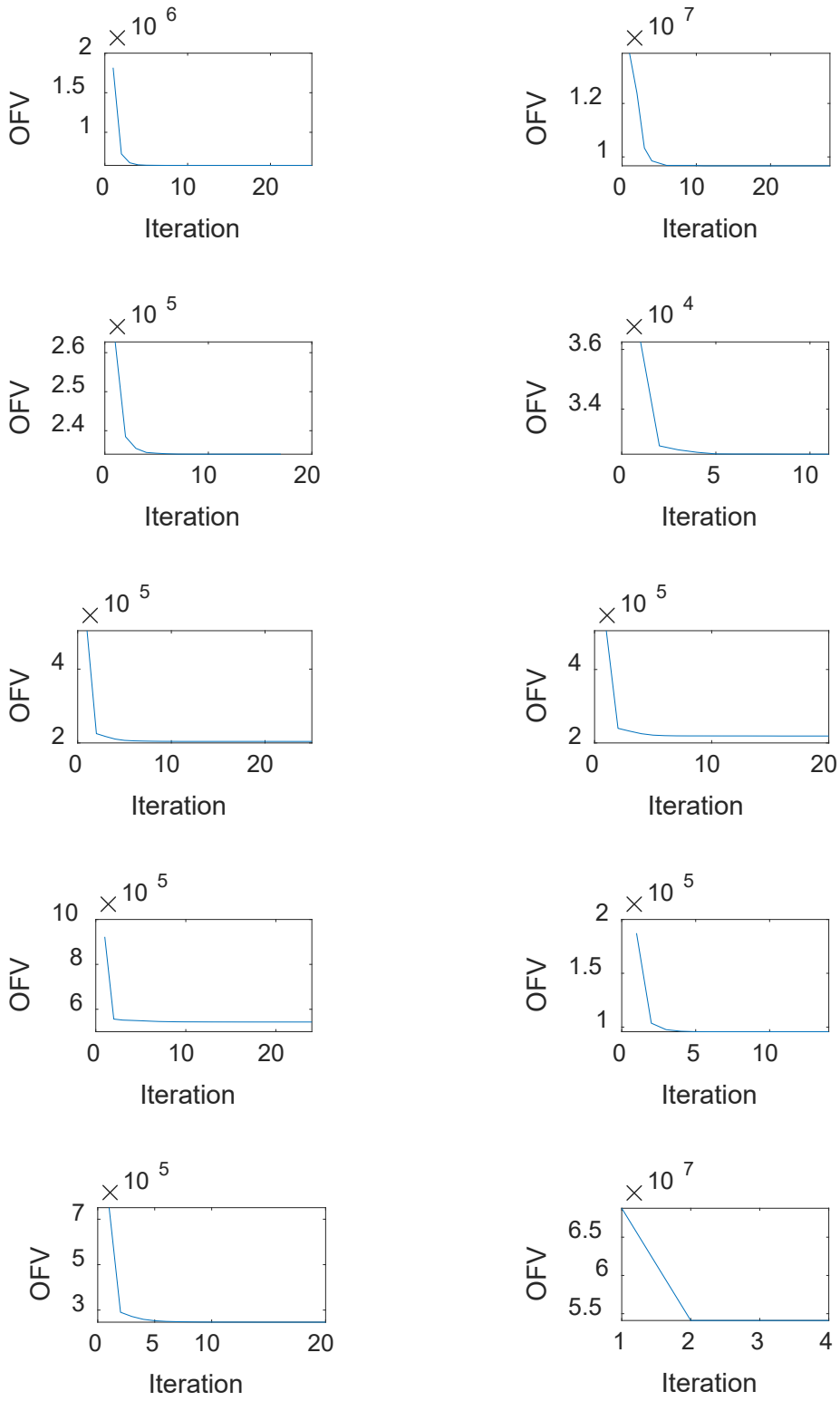


Figure 3.7 Objective function values (OFVs) versus iterations for IS algorithm

3.7 Conclusion

In this chapter we have proposed a new Initialization-Similarity (IS) algorithm to solving the initialization and similarity issues in a unified way. Specifically, we fixed the initialization of the clustering using the sum-of-norms regularization which outputted the new representation of original data points. We then learned the similarity matrix and the new representation simultaneously. Finally, we conducted *K*-means clustering on the derived new representative. Extensive experimental results on real-world benchmark data sets showed that IS clustering algorithm outperformed the related clustering algorithms. Furthermore, IS clustering algorithm is not very parameter sensitive. The fixed initialization of IS clustering algorithm using the sum-of-norms regularization makes the clustering robust.

Although the proposed IS clustering algorithm achieved significant clustering results, but we used *K*-means clustering in the final stage clustering. Similar to all *K*-means based clustering algorithms, this is the main limitation of IS clustering algorithm. Hence, future research needs to develop new clustering algorithms to learn the clustering number *K,* initialization and similarity automatically in a unified way.

Chapter 4

Joint Feature Selection with Dynamic Spectral Clustering

4.1 Introduction

Chapter 3 mainly solve the problems of initialization and similarity measurement issues of *K*-means clustering algorithm, which however can not specify the cluster number and is not robust to outliers and redundant features. Many of the current clustering algorithms need priori knowledge of the cluster number beforehand to conduct clustering. Some clustering algorithms learn this cluster number by continuously optimizing an objective function based on robust estimation [4]. Also, many clustering algorithms use Euclidean distance (in one form or another) to calculate similarity without considering factors such as the cluster number, sizes, dependent features or density. Some clustering algorithms are able to learn the similarity matrix [24, 66]. Current clustering algorithms either learn the similarity matrix only or learn the cluster number only. As an unsupervised learning approach, a clustering algorithm would be more useful if it could learn the cluster number and similarity measure simultaneously, and was less dependent on the Euclidean norm, which is prone to outlier issues. In this chapter, we propose a new improved algorithm called joint feature selection with dynamic spectral (FSDS) clustering algorithm, which considers the predefined cluster

number *K* and similarity measurement, feature selection and outlier reduction to further improve K-means clustering algorithm.

Real-world data sets often contain high-dimensional features, some of which are meaningless or irrelevant for clustering. Data with high-dimensional features could increase computational time and risk overfitting. Feature selection is a way to reduce the dimension of a data set. It is achieved either by selecting more useful features from an original feature list or by filtering irrelevant or superfluous features from the original data set. Feature selection techniques can be broadly classified into three groups: filter methods, wrapper methods, and embedded methods. Filter methods are usually too general and wrapper methods usually have a high computational cost. Embedded methods are more effective. In an embedded approach, a feature selection algorithm is an integral part of the learning algorithm. Recently the $L_{2,1}$-norm has been used in embedded approaches to improve the robustness and effectiveness of feature selection algorithms. The proposed embedded robust clustering algorithm adopts an $L_{2,1}$-norm minimization with sparse constraints on the regularization term to conduct feature selection.

Data almost invariably contains noise, outliers and errors due to inadequate data measure, collection, handling or just the inherent variability in the underlying data domain. Skewed data points which lie an abnormal distance from other data points are called outliers and these can distort the representativeness of the data set. To alleviate the significant influence of outliers, outlier detection and robust clustering algorithms are often used and a $L_{2,1}$-norm-based function has been shown to be robust with respect

to outliers [117, 118]. Thus the proposed robust joint feature selection with dynamic spectral clustering algorithm applies $L_{2,1}$-norm minimization with sparse constraints to the objective function to reduce the influence of outliers.

Previous research only focused on solving a few of the many clustering issues. These include the cluster number determination, the similarity measure, feature selection, and outlier reduction, but typically have not focused on solving all these issues in a unified framework. Clearly fixing only one or two of these issues does not guarantee the optimal results. Solving cluster number determination, similarity measure, feature selection, and outlier reduction issues of clustering algorithms simultaneously represents a big improvement over the existing algorithms because it could lead to better outputs.

The proposed FSDS clustering algorithm aims to solving cluster number determination, similarity measure, feature selection, and outlier reduction issues of *K*-means clustering algorithm in a unified way. Specifically, the proposed FSDS clustering algorithm learns the similarity matrix based on the data distribution, and then adds the ranked constraint on the Laplacian matrix of the learned similarity matrix to solving the cluster number issue. Furthermore, we employ the $L_{2,1}$-norm as the sparse constraints on both loss function and regularization term to reduce the influence of outliers and remove the redundant features. Constraining the normalized solution with the $L_{2,1}$-norm leads to clear cluster structures. Finally, we utilize an alternating strategy to solving the proposed objective function. We briefly summarize the contributions of the proposed FSDS clustering algorithm as follows:

- The proposed clustering algorithm learns the cluster number automatically.
- The proposed clustering algorithm learns the data similarity matrix, clustering structure and the cluster number simultaneously. The optimal performance could be reached when the separated stages are combined in a unified way.
- The proposed clustering algorithm employs $L_{2,1}$-norm minimization sparse constrains on the objective function and regularization term to reduce the influence of outliers and to select useful features.
- The experiment results on eight public data sets show that the proposed clustering algorithm outperforms four clustering algorithms [4, 24, 140] in terms of two evaluation metrics for clustering algorithms including accuracy (ACC) and Purity. It proves that simultaneously addressing the four primary issues (cluster number determination, similarity measure, feature selection and outlier reduction) for clustering algorithms is feasible and robust.

The remainder of the paper is organized as follows: Section 4.2 discusses the motivation behind the development of IS clustering algorithm. Section 4.3 provides the optimization process. Section 4.4 provides the convergence analysis. Section 4.5 discusses the experiments we conducted and presents the results of the experiments. The conclusions, limitations and future research direction are presented in Section 4.6.

4.2 Motivation

Chapter 3 mainly solve the problems of initialization and similarity measurement, but needs to specify the number of the clusters, which is unpractical in real applications.

Besides, most methods ignore the importance of reducing the influence of redundant features and outliers when conducting clustering task, so that the clustering performance easily get corrupted. To find out how other algorithms improve K-means clustering algorithm by automatically generating cluster number and improving robustness, we investigated K-means clustering algorithm, Clustering and projected clustering with adaptive neighbors algorithm (CAN) and Robust continuous clustering algorithm (RCC) in details.

As one of the most famous examples of partitioning clustering algorithms, the *K*-means clustering algorithm aims at minimizing the total intra-cluster variance represented by an objective function shown in Eq. (4.1).

$$\sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \tag{4.1}$$

where *K* is the cluster number, *n* is the number of data points, $x_i^{(j)}$ is the *i*-th data point of cluster *j*. c_j is the cluster center for cluster *j*, $\left\| x_i^{(j)} - c_j \right\|^2$ is the Euclidean distance between $x_i^{(j)}$ and c_j.

K-means clustering randomly selects *K* cluster centers first, and then iteratively recalculates the mean, reassigns and relocates data points to the clusters until convergence. The outcome of the *K*-means clustering objective function only depends on Euclidean distance between the data points and the cluster center, but the Euclidean distance does not reveal other underlying structures of the data such as cluster sizes, shape, dependent features or density, etc. [18, 30]. Thus the similarity

measure is an issue of *K*-means clustering. *K*-means clustering algorithm requires the cluster number *K* as an input. For some simple low dimensional data sets, the cluster number *K* could be abstained manually. In real applications, the cluster number *K* is not always known. There are a number of literatures have focused on solving this issue [141, 142]. For example, Elbow method determines the value of cluster number *K* based on the vision of a generated graph. But not all the data generated graph show any elbows. The rule of thumb method uses square root of the number of data divided by 2 to estimate the cluster number. For real clustering, the value gets from rule of thumb usually is unreasonably large. As an unsupervised machine learning technology, *K*-means clustering algorithm would be more powerful if it could calculate the cluster number *K* automatically. *K*-means clustering algorithm treats all data points equally without considering the characteristics of each data point, thus it is susceptible to the redundant features and the outliers.

Many algorithms have been constructed to try solving the issues of *K*-means clustering algorithm. The spectral clustering algorithm resolves the similarity issue by creating a similarity matrix first and computing the first *K* eigenvectors of its Laplacian matrix to define a feature vector. Then it runs *K*-means clustering on these features to separate data points into *K* clusters [61]. The spectral clustering algorithm conducts the data similarity matrix and spectral representation in two separate stages, where the goal of the first stage of constructing the similarity matrix disconnects from the goal of the second stage of achieving optimal spectral representation, and thus not guaranteed to always perform better than *K*-means clustering algorithm.

Clustering and projected clustering with adaptive neighbors algorithm (CAN) learns the data similarity matrix and clustering structure simultaneously [24]. The objective function shown in Eq. (4.2) is used to achieve the assignment of neighbors with the clustering structure.

$$\min_{\mathbf{S}} \sum_{i,j=1}^{n} \left(\left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|_2^2 s_{i,j} + r s_{ij}^2 \right) \tag{4.2}$$

$$s.t., \forall i, s_i^T \mathbf{1} = 1, 1 \geq s_i \geq 0, rank(L_s) = n - c$$

where $\mathbf{X} \in \mathbb{R}^{\mathrm{n} \times \mathrm{d}}$ is the data matrix of a dataset $\{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n\}$. $\mathrm{x}_i \in \mathbb{R}^{\mathrm{d} \times 1}$ is the *K*-nearest data points in the dataset to x_i while $\mathrm{x}_j \in \mathbb{R}^{\mathrm{d} \times 1}$ is the *K*-nearest data points in the dataset to x_j. $\mathrm{s}_{i,j}$ is the probability of the data point $j = 1, ..., n$ connected *i*-th data point x_i. r is the regularization parameter.

But again it needs to know *K*, the cluster number, beforehand. It also uses L_2-norm in its objective function. The $L_{2,1}$-norm performs more robustly and stable than L_2-norm when outliers exist [116].

Robust continuous clustering algorithm (RCC) optimizes an objective based on the following form [4]:

$$\mathbf{C}(\mathbf{U}) = \frac{1}{2} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{u}_i\|_2^2 + \frac{\lambda}{2} \sum_{(p,q) \in \varepsilon} w_{p,q} \rho(\left\| \boldsymbol{u}_p - \boldsymbol{u}_q \right\|_2) \tag{4.3}$$

where $\mathbf{X} = [x_1, x_2, \ldots, x_n]$ $\mathrm{x}_i \in \mathbb{R}^D$ is the input, $\mathbf{U} = \left[u_{1,} u_{2,} \ldots, u_{n,}\right], u_i \in \mathbb{R}^{\boldsymbol{D}}$ is the respresentatives. ε is set of edges of connected data points in a graph. $w_{p,q}$ balances the contribution of each data point to the pairwise terms.

RCC does not need prior knowledge of the cluster number. However, it needs the similarity matrix calculated beforehand as an input. The goal of constructing the similarity matrix and the goal of learning the cluster number K are different, and thus RCC does not guarantee an optimal solution, nor indeed outperform other algorithms such as K-means clustering, spectral clustering and CAN. It uses L_2-norm which is susceptive to high-dimensional features, noise, and outliers.

4.3 Proposed Algorithm

We propose a new clustering algorithm (i.e., Joint Feature Selection with Dynamic Spectral (FSDS) clustering algorithm) to concurrently address the challenges of clustering algorithms i.e., determination of the cluster number K, the similarity measure, the feature selection and outlier reduction of clustering algorithms in a unified framework. Specifically, the proposed clustering algorithm jointly learns the cluster number K, similarity matrix and the data representation to overcome the issue of current clustering algorithms, and applies $L_{2,1}$-norm to both the loss function and the regularization term. Minimizing the $L_{2,1}$-norm usually generates sparse solutions. With sparse constraints the $L_{2,1}$-norm forces many rows of the projection matrix to be zero, which leads the solution to take on discrete values and have more zero elements. Thus the most relevant data points are selected more efficiently. Hence, to reduce the influence of high-dimensional data, outliers, and noise, the loss function and the regularization term of the proposed FSDS clustering algorithm are all $L_{2,1}$-norm-based.

To achieve our goals, we form the objective function of the proposed clustering algorithm as follows:

$$\min_{\mathbf{W},\mathbf{U},\mathbf{S}} \frac{1}{2}\|\mathbf{XW}-\mathbf{U}\|_{2,1} + \frac{\alpha}{2}\sum_{i,j=1}^{n} s_{i,j}\rho(\left\|\mathbf{u}_i-\mathbf{u}_j\right\|_2) + \beta\|\mathbf{S}\|_F^2 + r\|\mathbf{W}\|_{2,1} \qquad (4.4)$$

$$s.t., \forall i, s_{i,j} \geq 0, s_i^T e = 1$$

where $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the data matrix, $\mathbf{W} \in \mathbb{R}^{d\times d}$ is the weight matrix to balance the contribution of each data point, $\mathbf{U} \in \mathbb{R}^{n\times d}$ is the new representation of $\mathbf{X}$, and $\mathbf{S} \in \mathbb{R}^{n\times n}$ is the similarity matrix to measure the similarity among data points, and $\rho(\left\|\mathbf{u}_i-\mathbf{u}_j\right\|_2)$ is a robust loss function, used for automatically generating clusters. The smaller the value of $\left\|\mathbf{u}_i-\mathbf{u}_j\right\|_2$ is, the closer the distance is, and the higher the similarity $\mathbf{s}_i$ and $\mathbf{s}_j$is. With the update of other parameters in Eq. (4.4), the distance $\left\|\mathbf{u}_i-\mathbf{u}_j\right\|_2$ for some i and j, will be very close, or even $\mathbf{u}_i = \mathbf{u}_j$. The clusters will be determined. $\boldsymbol{e} = [\mathbf{1}, \ldots, \mathbf{1}]^T$ **.** Both the capacity of the loss function and the regularization term are controlled by the $L_{2,1}$ norm, which is especially suitable for noise reduction, outliers removal and feature selection.

Equation. (4.4) automatically learns the new representation $\mathbf{U}$, the weight matrix $\mathbf{W}$, and the similarity matrix $\mathbf{S}$. The similarity matrix $\mathbf{S}$ learning is based on the data distribution, i.e., iteratively updated by the updated $\mathbf{U}$. This produces an intelligent new representation of the original data matrix.

Minimizing the $L_{2,1}$-norm usually generates sparse solutions [117, 118] so the residue $\|\mathbf{XW}-\mathbf{U}\|_{2,1}$ and regularization $\|\mathbf{W}\|_{2,1}$ take on discrete values with more zero elements. Moreover, Eq. (4.4) will keep the distance of indicator vectors similar if

the data belongs to the same cluster, possibly making them equal. The distance of indicator vectors is as separated as possible if data belongs to the different clusters.

A number of robust loss functions have been proposed to avoid the influence of noise and outliers in robust statistics [131, 132]. Here we employ the Geman-McClure function [133]:

$$\rho\left(\left\|\mathbf{u}_p-\mathbf{u}_q\right\|_2\right)=\frac{\mu\left\|\mathbf{u}_p-\mathbf{u}_q\right\|_2^2}{\mu+\left\|\mathbf{u}_p-\mathbf{u}_q\right\|_2^2} \tag{4.5}$$

The literature of half-quadratic minimization and robust statistics explains the reason for selecting Geman–McClure loss function instead of other loss functions [143]. Eq. (4.5) measures how well a model predicts the expected outcome. The smaller the value of $\left\|\mathbf{u}_p-\mathbf{u}_q\right\|_2^2$ is, the closer the distance is, and the higher the similarity s_p and $s_q$is. With the update of other parameters in Eq. (4.4), the distance $\left\|\mathbf{u}_p-\mathbf{u}_q\right\|_2^2$ for some p and q, will be very close, or even $\mathbf{u}_p=\mathbf{u}_q$. The clusters will be determined.

The optimization of the robust loss function is challenging. To address this, it is normal practice to introduce an auxiliary variable $f_{i,j}$ and a penalty item $\varphi(f_{i,j})$ [134-136], and thus Eq. (4.4) is rewritten to:

$$\min_{\mathbf{W},\mathbf{U},\mathbf{F},\mathbf{S}} \frac{1}{2}\|\mathbf{XW}-\mathbf{U}\|_{2,1}+\frac{\alpha}{2}\sum_{i,j=1}^{n} s_{i,j}\left(f_{i,j}\left\|\mathbf{u}_i-\mathbf{u}_j\right\|_2+\varphi(f_{i,j})\right)$$

$$+\beta\|\mathbf{S}\|_F^2+r\|\mathbf{W}\|_{2,1}, s.t., \forall i, s_{i,j}\geq 0, s_i^T e=1 \tag{4.6}$$

where $\varphi\left(f_{i,j}\right)=\mu(\sqrt{f_{i,j}}-1)^2, i,j=1\ldots n.$

This objective function is still challenging to solve. An iterative optimization process is adopted to tackle this challenge. In the next section, we will show how iterative optimization is utilized to solving the problem.

Algorithm 4.1. The pseudo code for proposed FSDS clustering algorithm

Input: $\mathbf{X} \in \mathbb{R}^{n \times d}$ (data set **X** with n instances and d features)

Output: a set of *K* clusters

Initialization:

U = **X**;

Repeat:

- Update **W** using Eq. (4.20);
- Update **F** using Eq. (4.23);
- Update **S** using Eq. (4.27);
- Update **U** using Eq. (4.38);

Until U converges

4.4 Optimization

Equation. (4.6) is convex on each variable of **W**, **F**, **S**, and **U** while fixing the rest. The alternating optimization strategy is applied to solving the Eq. (4.6). Specifically, we optimize each variable while fixing the rest until the objective function converges. The pseudo-code of the proposed clustering algorithm is given in Algorithm 4.1.

1) Update W while fixing F, S and U

While **F**, **S** and **U** are fixed, the objective function is transformed to a simplified matrix form to optimize **W**:

$$\min_{\mathbf{W}} \frac{1}{2}\|\mathbf{XW}-\mathbf{U}\|_{2,1} + r\|\mathbf{W}\|_{2,1} \tag{4.7}$$

$$\Rightarrow d_{i,i} = \frac{1}{2\|\mathbf{XW}-\mathbf{U}\|_2}, i = 1,\dots,n \tag{4.8}$$

$$\Rightarrow m_{i,i} = \frac{1}{2\|\mathbf{W}\|_2}, i = 1,\dots,n \tag{4.9}$$

$$\Rightarrow \min_{\mathbf{W}} \frac{1}{2}\mathrm{tr}((\mathbf{XW}-\mathbf{U})^T\mathbf{D}(\mathbf{XW}-\mathbf{U})) + r\mathrm{tr}(\mathbf{W}^T\mathbf{MW}) \tag{4.10}$$

$$\min_{\mathbf{W}} \frac{1}{2}\mathrm{tr}(\mathbf{W}^T\mathbf{X}^T-\mathbf{U}^T)\mathbf{D}(\mathbf{XW}-\mathbf{U}) + r\mathrm{tr}(\mathbf{W}^T\mathbf{MW}) \tag{4.11}$$

$$\Rightarrow \min_{\mathbf{W}} \frac{1}{2}\mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{DXW}-\mathbf{W}^T\mathbf{X}^T\mathbf{DU}-\mathbf{U}^T\mathbf{DXW}+\mathbf{U}^T\mathbf{DU}) + r\mathrm{tr}(\mathbf{W}^T\mathbf{MW}) + r\mathrm{tr}(\mathbf{W}^T\mathbf{MW}) \tag{4.12}$$

$$\Rightarrow \min_{\mathbf{W}} \frac{1}{2}(\mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{DXW}) - \mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{DU}) - \mathrm{tr}(\mathbf{U}^T\mathbf{DXW})^T + \mathrm{tr}(\mathbf{U}^T\mathbf{DU}) + r\mathrm{tr}(\mathbf{W}^T\mathbf{MW})) \tag{4.13}$$

$$\Rightarrow \min_{\mathbf{W}} \frac{1}{2}\mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{DXW}) - \mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{DU}) - \mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{D}^T\mathbf{U}) + \mathrm{tr}(\mathbf{U}^T\mathbf{DU}) + r\mathrm{tr}(\mathbf{W}^T\mathbf{MW}) \tag{4.14}$$

Due the $\mathbf{D}$ is diagonal matrix. $\mathbf{D}^T = \mathbf{D}$

$$\Rightarrow \min_{\mathbf{W}} \frac{1}{2}(\mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{DXW}) - 2\mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{DU}) + \mathrm{tr}(\mathbf{U}^T\mathbf{DU})) + r\mathrm{tr}(\mathbf{W}^T\mathbf{MW})) \tag{4.15}$$

$$\Rightarrow \mathcal{L}(\mathbf{W}) = \frac{1}{2}(\mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{DXW}) - 2\mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{DU}) + \mathrm{tr}(\mathbf{U}^T\mathbf{DU})) + r\mathrm{tr}(\mathbf{W}^T\mathbf{MW})) \tag{4.16}$$

By taking a derivative of $\mathcal{L}(\mathbf{W})$ on Eq. (4.16) with respect to $\mathbf{W}$ and setting the derivative to zero we see:

$$\frac{1}{2}(2\mathbf{X}^T\mathbf{D}\mathbf{X}\mathbf{W} - 2\mathbf{X}^T\mathbf{D}\mathbf{U}) + r2\mathbf{M}\mathbf{W} = 0 \tag{4.17}$$

$$\Rightarrow \mathbf{X}^T\mathbf{D}\mathbf{X}\mathbf{W} - \mathbf{X}^T\mathbf{D}\mathbf{U} + 2r\mathbf{M}\mathbf{W} = 0 \tag{4.18}$$

$$\Rightarrow (\mathbf{X}^T\mathbf{D}\mathbf{X}\mathbf{W} + 2r\mathbf{M}\mathbf{W}) = \mathbf{X}^T\mathbf{D}\mathbf{U} \tag{4.19}$$

$$\Rightarrow \mathbf{W} = (\mathbf{X}^T\mathbf{D}\mathbf{X} + 2r\mathbf{M})^{-1}\mathbf{X}^T\mathbf{D}\mathbf{U} \tag{4.20}$$

Algorithm 4.2. Algorithm to solving the problem described in Eq. (4.7)

Input: $\mathbf{X} \in \mathbb{R}^{n\times d}$, $\mathbf{U} \in \mathbb{R}^{n\times d}$

Output: Projection matrix **W**

Repeat:

- With current $\mathbf{U}, \mathbf{M}, \mathbf{D}$, **W** is obtained by solving problem (4.20)
- With current **W**, **U** is obtained by Eq. (4.38)
- With current **W** and **U**, **D** is obtained by Eq. (4.8)
- With current **W**, **M** is obtained by Eq. (4.9)

Until W converges

2) Update F while fixing W, S and U

While $\mathbf{W}, \mathbf{S}$ and $\mathbf{U}$ are fixed, the objective function of Eq. (4.6) can be rewritten in a simplified matrix form to optimize **F**:

$$\min_{\mathbf{F}} \frac{\alpha}{2}\sum_{i,j=1}^{n} s_{i,j}(f_{i,j}\left\|\mathbf{u}_i - \mathbf{u}_j\right\|_2^2 + \mu(\sqrt{f_{i,j}} - 1)^2), \quad s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T\mathbf{e} = 1 \tag{4.21}$$

Since the optimization of $f_{i,j}$ is independent of the optimization of other $f_{p,q}, i \neq p, j \neq q$, the $f_{i,j}$ is optimized first as shown in following

$$\frac{\alpha}{2}(s_{i,j}f_{i,j}\left\|\mathbf{u}_i - \mathbf{u}_j\right\|_2^2 + s_{i,j}\mu(f_{i,j} - 2\sqrt{f_{i,j}} + 1)) \tag{4.22}$$

By conducting a derivative on Eq. (4.23) with respect to $f_{i,j}$, we get

$$f_{i,j} = \left(\frac{\mu}{\mu+\|\mathbf{u}_i-\mathbf{u}_j\|_2^2}\right)^2 \tag{4.23}$$

3) Update S while fixing W, U and F

While fixing **W**, **U** and **F**, the objective function Eq. (4.6) with respect to **S** is:

$$\min_{\mathbf{S}} \frac{\alpha}{2}\sum_{i,j=1}^{n} s_{i,j}(f_{i,j}\|\mathbf{u}_i-\mathbf{u}_j\|_2^2+\mu(\sqrt{f_{i,j}}-1)^2)+\beta\|\mathbf{S}\|_F^2 \tag{4.24}$$

$$s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T\mathbf{e}=1$$

Since the optimization of $\mathbf{s}_i$ is independent of the optimization of other $\mathbf{s}_j, i \neq j, i, j = 1, \ldots, n$, the $\mathbf{s}_i$ is optimized first as shown in following:

$$\min_{\mathbf{S}} \frac{\alpha}{2}\sum_{i,j=1}^{n} s_{i,j}(f_{i,j}\|\mathbf{u}_i-\mathbf{u}_j\|_2^2+\mu(\sqrt{f_{i,j}}-1)^2)+\beta\sum_{i=1}^{n}\|\mathbf{s}_i\|_2^2 \tag{4.25}$$

$$s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T\mathbf{e}=1$$

Let $b_{i,j} = f_{i,j}\|\mathrm{u}_i-\mathrm{u}_j\|_2^2$ and $c_{i,j} = \mu(\sqrt{f_{i,j}}-1)^2$, Eq. (4.25) is equivalent to:

$$\min_{\mathbf{s}_i} \left\|\mathbf{s}_i+\frac{\alpha}{4\beta}(\mathbf{b}_i+\mathbf{c}_i)\right\|_2^2, s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T\mathbf{e}=1 \tag{4.26}$$

According to Karush-Kuhn-Tucker (KKT) [137], the optimal solution $\mathbf{s}_i$ should be

$$\mathrm{s}_{i,j} = \max\{-\frac{\alpha}{4\beta}(\mathrm{b}_{i,j}+\mathrm{c}_{i,j})+\theta, 0\}, j = 1, \ldots, n \tag{4.27}$$

where $\theta = \frac{1}{\rho}\sum_{j=1}^{\rho}\left(\frac{\alpha}{4\beta}(\mathrm{b}_{i,j} + \mathrm{c}_{i,j}) + 1\right)$, and $\rho = max_j\{\omega_j - \frac{1}{j}\left(\sum_{r=1}^{j}\omega_r - 1\right), 0\}$ and ω is the descending order of $\frac{\alpha}{4\beta}(\mathrm{b}_{i,j} + \mathrm{c}_{i,j})$.

4) Update U while fixing W, S and F

While $\mathbf{W}$, $\mathbf{S}$ and $\mathbf{F}$ are fixed, the objective function can be rewritten in a simplified form to optimize $\mathbf{U}$:

$$\min_{\mathbf{U}} \frac{1}{2}\|\mathbf{XW} - \mathbf{U}\|_{2,1} + \frac{\alpha}{2}\sum_{i,j=1}^{n} s_{i,j}(f_{i,j}\left\|\mathbf{u}_i - \mathbf{u}_j\right\|_2) \tag{4.28}$$

$$s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T\mathbf{e} = 1$$

where $\mathbf{F} \in \mathbb{R}^{n\times c}$ and Let $\mathbf{S}_{SF} = \frac{(\mathbf{S}\odot\mathbf{F})^T + (\mathbf{S}\odot\mathbf{F})}{2}$. The degree matrix $\mathbf{D}_s = diag(\mathbf{S}_{SF}\mathbf{1})$ is a diagonal matrix. The Laplacian Matrix $\mathbf{L}$ is defined below:

$$\mathbf{L} = \mathbf{D}_s - \mathbf{S}_{SF} \tag{4.29}$$

Eq. (4.29) is equivalent to:

$$\min_{\mathbf{U}} \frac{1}{2}\|\mathbf{XW} - \mathbf{U}\|_{2,1} + \frac{\alpha}{2}\mathrm{tr}(\mathbf{U}^T\mathbf{LU}) \tag{4.30}$$

After applying Eq. (4.8), Eq. (4.30) is equivalent to:

$$\min_{\mathbf{U}} \frac{1}{2}\mathrm{tr}(\mathbf{W}^\mathrm{T}\mathbf{X}^\mathrm{T}\mathbf{DXW} - \mathbf{W}^\mathrm{T}\mathbf{X}^\mathrm{T}\mathbf{DU} - \mathbf{U}^\mathrm{T}\mathbf{DXW} + \mathbf{U}^\mathrm{T}\mathbf{DU}) + \frac{\alpha}{2}\mathrm{tr}(\mathbf{U}^\mathrm{T}\mathbf{LU}) \tag{4.31}$$

$$\Rightarrow \min_{\mathbf{U}} \frac{1}{2}\mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{DXW}) - \mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{DU}) - \mathrm{tr}(\mathbf{U}^T\mathbf{DXW})^T + \mathrm{tr}(\mathbf{U}^T\mathbf{DU}) + \frac{\alpha}{2}\mathrm{tr}(\mathbf{U}^T\mathbf{LU}) \tag{4.32}$$

$$\Rightarrow \min_{\mathbf{U}} \frac{1}{2}\mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{DXW}) - \mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{DU}) - \mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{D}^T\mathbf{U}) + \mathrm{tr}(\mathbf{U}^T\mathbf{DU})$$

$$+\frac{\alpha}{2}\mathrm{tr}(\mathbf{U}^T\mathbf{L}\mathbf{U}) \tag{4.33}$$

$$\Rightarrow \min_{\mathbf{U}} \ \frac{1}{2}\mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{D}\mathbf{X}\mathbf{W}) - 2\mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{D}\mathbf{U}) \ + \mathrm{tr}(\mathbf{U}^T\mathbf{D}\mathbf{U}) + \frac{\alpha}{2}\mathrm{tr}(\mathbf{U}^T\mathbf{L}\mathbf{U}) \tag{4.34}$$

After taking a derivative of $\mathcal{L}(\mathbf{U})$ on Eq. (4.34) with respect to **U** and setting the derivative to zero, we get

$$\frac{1}{2}(-2\mathbf{D}\mathbf{X}\mathbf{W} + 2\mathbf{D}\mathbf{U}) \ + \frac{\alpha}{2}2\mathbf{L}\mathbf{U} = 0 \tag{4.35}$$

$$\Rightarrow -\mathbf{D}\mathbf{X}\mathbf{W} + \mathbf{D}\mathbf{U} + \ \alpha\mathbf{L}\mathbf{U} = 0 \tag{4.36}$$

$$\Rightarrow \ \mathbf{D}\mathbf{U} + \ \alpha\mathbf{L}\mathbf{U} = \mathbf{D}\mathbf{X}\mathbf{W} \tag{4.37}$$

The term **U** can be efficiently obtained by solving the Eq. (4.37):

$$\mathbf{U} = (\mathbf{D} + \alpha\mathbf{L})^{-1}\mathbf{D}\mathbf{X}\mathbf{W} \tag{4.38}$$

We adopt an iterative algorithm to obtain the solution of **U** such that Eq. (4.38) is satisfied. We will prove that the proposed algorithm converges in the following subsection.

Algorithm 4.3. Algorithm to solving the problem described in Eq. (4.30)

Input: $\mathbf{X} \in \mathbb{R}^{n\times d}$, Data matrix $\mathbf{W} \in \mathbb{R}^{d\times d}$ $\mathbf{D} \in \mathbb{R}^{d\times d}$, $\mathbf{S} \in \mathbb{R}^{n\times n}$, $\mathbf{L} \in \mathbb{R}^{n\times n}$

Output: Projection matrix $\mathbf{U} \in \mathbb{R}^{n\times d}$

Repeat:

- With current **S**, **D**, **L** is obtained by Eq. (4.29)
- With current **W**, **D**, **L, U** is obtained by Eq. (4.38)

Until U converges

4.5 Convergence Analysis

In this section, we will provide the convergence analysis for the proposed FSDS clustering algorithm reaching an optimal solution. The convergence of the proposed clustering algorithm is summarized in the following theorems. To prove the convergence, we need the lemma proposed by Nie et al. [144].

Lemma 1. *The following inequality holds for any positive real number a and b* [144].

$$\sqrt{a} - \frac{a}{2\sqrt{b}} \leq \sqrt{b} - \frac{b}{2\sqrt{b}} \tag{4.39}$$

The convergence of Algorithm 4.2 can be proven by the following theorem.

Theorem 1. *In Algorithm 4.2, updated* **W** *will decrease the objective value of problem described in (4.7) until converge*.

Proof. Eq. (4.20) is the solution to the following problem:

$$\min_{\mathbf{W}} \ \frac{1}{2}\mathrm{tr}(\mathbf{XW}-\mathbf{U})^T\mathbf{D}(\mathbf{XW}-\mathbf{U}) + r\mathrm{tr}(\mathbf{W}^T\mathbf{MW}) \tag{4.40}$$

In the *t*-th iteration:

$$\mathbf{W}_{t+1} = \mathrm{argmin}_{\mathbf{W}} \frac{1}{2}\mathrm{tr}\Big((\mathbf{XW}_{t+1}-\mathbf{U}_t)^{\mathrm{T}}\mathbf{D}_t(\mathbf{XW}_{t+1}-\mathbf{U}_t)\Big)$$

$$+ r\mathrm{tr}({\mathbf{W}_{t+1}}^T\mathbf{M}_{t+1}\mathbf{W}_{t+1}) \tag{4.41}$$

The following equation can be already established

$$\frac{1}{2}\mathrm{tr}\big((\mathbf{XW}_{t+1}-\mathbf{U}_t)^T\mathbf{D}_t(\mathbf{XW}_{t+1}-\mathbf{U}_t)\big) + r\mathrm{tr}({\mathbf{W}_{t+1}}^T\mathbf{M}_{t+1}\mathbf{W}_{t+1})$$

$$\leq \frac{1}{2}\mathrm{tr}\big((\mathbf{XW}_t-\mathbf{U}_t)^T\mathbf{D}_t(\mathbf{XW}_t-\mathbf{U}_t)\big) + r\mathrm{tr}({\mathbf{W}_t}^T\mathbf{M}_t\mathbf{W}_t) \tag{4.42}$$

We substitute the definition of **D** and **M** in Eq. (4.8) and (4.9), then inequality (4.42) can be rewritten as:

$$\sum_{i=1}^{n} \frac{\|(\mathbf{X}\mathbf{W}_{t+1}-\mathbf{U}_t)^i\|_2^2}{2\|(\mathbf{X}\mathbf{W}_t-\mathbf{U}_t)^i\|_2} + r\sum_{i=1}^{n} \frac{\|\mathbf{W}_{t+1}{}^i\|_2^2}{2\|\mathbf{W}_t{}^i\|_2}$$

$$\leq \sum_{i=1}^{n} \frac{\|(\mathbf{X}\mathbf{W}_t-\mathbf{U}_t)^i\|_2^2}{2\|(\mathbf{X}\mathbf{W}_t-\mathbf{U}_t)^i\|_2} + r\sum_{i=1}^{n} \frac{\|\mathbf{W}_t{}^i\|_2^2}{2\|\mathbf{W}_t{}^i\|_2} \quad (4.43)$$

Based on Lemma 1, we get Eq. (4.44) and Eq. (4.45).

$$\sum_{i=1}^{n}\|(\mathbf{X}\mathbf{W}_{t+1}-\mathbf{U}_t)^i\|_2 - \sum_{i=1}^{n} \frac{\|(\mathbf{X}\mathbf{W}_{t+1}-\mathbf{U}_t)^i\|_2^2}{2\|(\mathbf{X}\mathbf{W}_t-\mathbf{U}_t)^i\|_2}$$

$$\leq \sum_{i=1}^{n}\|(\mathbf{X}\mathbf{W}_t-\mathbf{U}_t)^i\|_2 - \sum_{i=1}^{n} \frac{\|(\mathbf{X}\mathbf{W}_t-\mathbf{U}_t)^i\|_2^2}{2\|(\mathbf{X}\mathbf{W}_t-\mathbf{U}_t)^i\|_2} \quad (4.44)$$

$$\sum_{i=1}^{n}\|\mathbf{W}_{t+1}{}^i\|_2 - \sum_{i=1}^{n} \frac{\|\mathbf{W}_{t+1}{}^i\|_2^2}{2\|\mathbf{W}_t{}^i\|_2} \leq \sum_{i=1}^{n}\|\mathbf{W}_t{}^i\|_2 - \sum_{i=1}^{n} \frac{\|\mathbf{W}_t{}^i\|_2^2}{2\|\mathbf{W}_t{}^i\|_2} \quad (4.45)$$

Sum over the inequality Eq. (4.43) , inequality Eq. (4.44) and inequality Eq. (4.45), we arrive at

$$\sum_{i=1}^{n} \tfrac{1}{2}\|(\mathbf{X}\mathbf{W}_{t+1}-\mathbf{U}_t)^i\|_2 + r\sum_{i=1}^{n} \tfrac{1}{2}\|\boldsymbol{W}_{t+1}{}^i\|_2$$

$$\leq \sum_{i=1}^{n} \tfrac{1}{2}\|(\mathbf{X}\mathbf{W}_t-\mathbf{U}_t)^i\|_2 + r\sum_{i=1}^{n} \tfrac{1}{2}\|\boldsymbol{W}_t{}^i\|_2 \quad (4.46)$$

This is to say,

$$\tfrac{1}{2}\|\mathbf{X}\mathbf{W}_{t+1}-\mathbf{U}_t\|_{2,1} + r\|\mathbf{W}_{t+1}\|_{2,1} \leq \tfrac{1}{2}\|\mathbf{X}\mathbf{W}_t-\mathbf{U}_t\|_{2,1} + r\|\mathbf{W}_t\|_{2,1} \quad (4.47)$$

This completes the proof Algorithm 4.2. The convergence of Algorithm 4.3 can be proven by the following theorem.

Theorem 2. *In Algorithm 4.3, updated* **U** *will decrease the objective value of problem (4.30) until converge.*

Proof. Eq. (4.38) is the solution to the following problem:

$$\min_{\mathbf{U}} \frac{1}{2}\mathrm{tr}(\mathbf{W}^T\mathbf{X}^T - \mathbf{U}^T)\mathbf{D}(\mathbf{XW} - \mathbf{U})) + \frac{\alpha}{2}\mathrm{tr}(\mathbf{U}^T\mathbf{LU})) \tag{4.48}$$

In the *t*-th iteration,

$$\mathbf{U}_{t+1} = \underset{\mathbf{U}}{\mathrm{argmin}} \ \frac{1}{2}\mathrm{tr}((\mathbf{XW}_t - \mathbf{U}_{t+1})^T\mathbf{D}_{t+1}(\mathbf{XW}_t - \mathbf{U}_{t+1}) + \frac{\alpha}{2}\ \mathrm{tr}(\mathbf{U}_{t+1}^T\mathbf{LU}_{t+1})) \tag{4.49}$$

We substitute the definition of **D** in Eq. (4.8), and then inequality Eq. (4.49) can be rewritten as:

$$\begin{aligned} &\frac{1}{2}\mathrm{tr}((\mathbf{XW}_t - \mathbf{U}_{t+1})^T\mathbf{D}_{t+1}(\mathbf{XW}_t - \mathbf{U}_{t+1}) + \frac{\alpha}{2}\ \mathrm{tr}(\mathbf{U}_{t+1}^T\mathbf{LU}_{t+1})) \\ &\leq \frac{1}{2}\mathrm{tr}((\mathbf{XW}_t - \mathbf{U}_t)^{\mathrm{T}}\mathbf{D}_t(\mathbf{XW}_t - \mathbf{U}_t) + \frac{\alpha}{2}\ \mathrm{tr}(\mathbf{U}_t^T\mathbf{LU}_t)) \end{aligned} \tag{4.50}$$

We substitute the definition of **D** in Eq. (4.8) and **L** in Eq. (4.29), and then inequality (4.50) can be rewritten as:

$$\begin{aligned} &\frac{1}{2}\sum_{i=1}^{n}\frac{\left\|(\mathbf{XW}_t-\mathbf{U}_{t+1})^i\right\|_2^2}{2\left\|(\mathbf{XW}_t-\mathbf{U}_t)^i\right\|_2}+\frac{\alpha}{2}\mathrm{tr}(\mathbf{U}_t^T\mathbf{LU}_t) \\ \leq &\frac{1}{2}\sum_{i=1}^{n}\frac{\left\|(\mathbf{XW}_t-\mathbf{U}_t)^i\right\|_2^2}{2\left\|(\mathbf{XW}_t-\mathbf{U}_t)^i\right\|_2}+\frac{\alpha}{2}\mathrm{tr}(\mathbf{U}_t^T\mathbf{LU}_t) \end{aligned} \tag{4.51}$$

Based on Lemma 1, we know

$$\begin{aligned}&\frac{1}{2}\sum_{i=1}^{n}\left\|((\mathbf{X}\mathbf{W}_t-\mathbf{U}_{t+1})^i\right\|_2-\frac{\left\|(\mathbf{X}\mathbf{W}_t-\mathbf{U}_{t+1})^i\right\|_2^2}{2\left\|(\mathbf{X}\mathbf{W}_t-\mathbf{U}_t)^i\right\|_2}\\&\leq\frac{1}{2}\sum_{i=1}^{n}\left\|((\mathbf{X}\mathbf{W}_t-\mathbf{U}_t)^i\right\|_2-\frac{\left\|(\mathbf{X}\mathbf{W}_t-\mathbf{U}_t)^i\right\|_2^2}{2\left\|(\mathbf{X}\mathbf{W}_t-\mathbf{U}_t)^i\right\|_2}\end{aligned} \tag{4.52}$$

Sum over the inequality Eq. (4.51) and inequality Eq. (4.52), we could arrive at inequality Eq. (4.53).

$$\begin{aligned}&\frac{1}{2}\sum_{i=1}^{n}\left\|((\mathbf{X}\mathbf{W}_t-\mathbf{U}_{t+1})^i\right\|_2+\frac{\alpha}{2}\mathrm{tr}\left(\mathbf{U}_{t+1}^T\mathbf{L}\mathbf{U}_{t+1}\right)\\&\leq\frac{1}{2}\sum_{i=1}^{n}\left\|((\mathbf{X}\mathbf{W}_t-\mathbf{U}_t)^i\right\|_2+\frac{\alpha}{2}\mathrm{tr}(\mathbf{U}_t^T\mathbf{L}\mathbf{U}_t)\end{aligned} \tag{4.53}$$

This is to say,

$$\frac{1}{2}\|(\mathbf{X}\mathbf{W}_t-\mathbf{U}_{t+1})\|_{2,1}+\frac{\alpha}{2}\mathrm{tr}(\mathbf{U}_{t+1}^T\mathbf{L}\mathbf{U}_{t+1})\leq\|(\mathbf{X}\mathbf{W}_t-\mathbf{U}_t)\|_{2,1}+\frac{\alpha}{2}\mathrm{tr}(\mathbf{U}_t^T\mathbf{L}\mathbf{U}_t) \tag{4.54}$$

This completes the proof of Algorithm 4.3.

Theorem 3. *FSDS clustering algorithm decreases the objective function value of Eq. (4.6) until it converges.*

According to Theorem 1,

$$\mathcal{L}(\mathbf{W}_{t+1},\mathbf{U}_t,\mathbf{F}_t,\mathbf{S}_t)\leq\mathcal{L}(\mathbf{W}_t,\mathbf{U}_t,\mathbf{F}_t,\mathbf{S}_t) \tag{4.55}$$

According to Theorem 2,

$$\mathcal{L}(\mathbf{W}_{t+1},\mathbf{U}_{t+1},\mathbf{F}_t,\mathbf{S}_t)\leq\mathcal{L}(\mathbf{W}_{t+1},\mathbf{U}_t,\mathbf{F}_t,\mathbf{S}_t) \tag{4.56}$$

According to Eq. (4.23) in Section 4.4, $\mathbf{F}$ has a closed-form solution, thus we have the following inequality:

$$\mathcal{L}(\mathbf{W}_{t+1},\mathbf{U}_{t+1},\mathbf{F}_{t+1},\mathbf{S}_t)\leq\mathcal{L}(\mathbf{W}_{t+1},\mathbf{U}_{t+1},\mathbf{F}_t,\mathbf{S}_t) \tag{4.57}$$

According to Eq. (4.27) in Section 4.4, $\mathbf{S}$ has a closed-form solution, thus we have the following inequality:

$$\mathcal{L}(\mathbf{W}_{t+1}, \mathbf{U}_{t+1}, \mathbf{F}_{t+1}, \mathbf{S}_{t+1}) \leq \mathcal{L}(\mathbf{W}_{t+1}, \mathbf{U}_{t+1}, \mathbf{F}_{t+1}, \mathbf{S}_t) \tag{4.58}$$

Sum up Eq.(4.55), Eq.(4.56), Eq.(4.57), and Eq.(4.58), we get:

$$\mathcal{L}(\mathbf{W}_{t+1}, \mathbf{U}_{t+1}, \mathbf{F}_{t+1}, \mathbf{S}_{t+1}) \leq \mathcal{L}(\mathbf{W}_t, \mathbf{U}_t, \mathbf{F}_t, \mathbf{S}_t) \tag{4.59}$$

Hence Algorithm 4.1 will converge to the global optimum for the problem (4.6). Empirical results also show that the objective function convergences.

4.6 Experiments

In this section, we first evaluate the performance of the proposed FSDS algorithm by comparing it with four benchmark algorithms on eight real UCI datasets in terms of two evaluation metrics for clustering research, accuracy (ACC) and Purity. Then we investigated parameter sensitivity of the proposed algorithm (i.e. α, r and β in Eq. (4.6)) via varying their values to observe the variations of clustering algorithm's performance. Finally we demonstrated the convergence of Algorithm 1 to solve the proposed objective function Eq. (4.6) via checking the iteration times when Algorithm 4.1 converges.

4.6.1 Data Sets

We ran the proposed algorithm and four comparison algorithms on eight data sets including Cardiotocography, Diabetic Retinopathy Debrecen, Parkinson Speech,

German Credit, Australian Credit Approval, Balance Scale, Credit Approval, and Musk. The eight UCI data sets in the experiments are summarized in the following and are shown in Table 4.1.

- *Cardiotocography.* Data set measures the respective diagnostic features of the fetal cardiotocograms. It has 2126 instances and 41 features.
- *Diabetic Retinopathy Debrecen.* Data set contains features to predict whether a Messidor image has signs of diabetic retinopathy. It has 1151 instances and 19 attributes.
- *Parkinson Speech.* Data set has multiple types of sound recordings. It has 1040 in-stances and 28 features.
- *German Credit.* Data set contains a set of attributes to classify people as good or bad credit risks. It has 1000 instances and 20 attributes.
- *Australian Credit Approval.* Data set contains data about credit card applications. There are 690 instances and 14 attributes including continuous, nominal with small numbers of values, and nominal with larger numbers of values. There are also a few missing values.
- *Balance Scale.* Data set was generated to model psychological experimental results. It has 625 instances and 4 attributes.
- *Cedit Approval.* Data set concerns credit card applications. It has 690 instances and 15 mixed attributes and missing values.
- *Musk (Version 2).* Data set contains features of molecules. It has 6598 instances and 166 features.

Table 4.1 Description of benchmark datasets

Datasets	Instances	Features	Classes
Cardiotocography	2126	41	3
Diabetic Retinopathy Debrecen	1151	19	2
Parkinson Speech	1040	28	2
German Credit	1000	20	2
Australian Credit Approval	690	14	2
Balance Scale	625	4	3
Credit Approval	690	15	2
Musk (Version 2)	6598	166	2

4.6.2 Comparison Algorithms

We tested the robustness of the proposed Joint Feature Selection with Dynamic Spectral (FSDS) clustering algorithm by comparing it with *K*-means clustering algorithm, spectral clustering algorithm, clustering with adaptive neighbors (CAN) [24] , and robust continuous clustering (RCC) [4].

4.6.3 Experiment Setup

In the experiments, firstly, we evaluate the performance of the proposed FSDS algorithm by comparing it with four benchmark algorithms on eight real UCI data sets in terms of two evaluation metrics for clustering research, accuracy (ACC) and Purity. Then we investigated parameter sensitivity of the proposed algorithm (i.e. α, r and β in Eq. (4.6)) via varying their values to observe the variations of clustering algorithm's performance. Finally we demonstrated the convergence of Algorithm 4.1 to solving the

proposed objective function Eq. (4.6) via checking the iteration times when Algorithm 4.1 converges.

4.6.4 Experimental Results Analysis

The performance of all the algorithms are listed in Table 4.2 and Table 4.3, which show that the proposed clustering algorithm achieved the best overall performance on the eight data sets in terms of ACC and Purity. More specifically, in terms of average ACC results of all eight data sets, the proposed FSDS clustering algorithm increased ACC by 12.56%, 4.43%, 5.79%, and 11.68% respectively compared to *K*-means clustering algorithm, spectral clustering, CAN, and RCC. In terms of average Purity results on all eight data sets, the FSDS algorithm increased the average Purity by 7.13%, 8.26%, 7.85%, and 6.90% respectively compared to *K*-means clustering, spectral clustering, CAN, and RCC. The FSDS algorithm performed best for data sets that have high dimensions such as the Musk data set with 166 features. For this data set, the algorithm increased ACC by 30.09%, 4.20%, 16.59%, and 42.90% respectively compared to *K*-means clustering, spectral clustering, CAN [24], and RCC [4]. The FSDS algorithm performed better than IS algorithm in terms of ACC and purity. IS algorithm performed better than other comparison algorithms including *K*-mean clustering, spectral clustering, CAN, and RCC. Other observations are listed below.

First, being similar to our first proposed method IS, the proposed FSDS clustering algorithm use the unified framework to adaptively update the new

presentation and similarity matrix, which can reduce the influence of redundancy of original data and can more accurately capture the intrinsic correlation of original data. So, our method easily gets better clustering performance than other clustering algorithms. By contrast, other methods separately address each issues step by step, easily trapping into the sub-optimal results, which means it is hard to output the optimal clustering results.

Second, compared to IS that use a unified framework and other methods that do not use a unified framework, FSDS has further employed the L21-norm minimization for the loss function. As shown as former content, the L21-norm can conduct feature selection in the process of clustering tasks, which means that we can more easily find the intrinsic correlation of data by removing the redundant features from original data. So, our method achieved the best clustering results. Besides, our method outperformed both L2-norm-based clustering algorithms [4, 24], which indicate that our method is robust in handling the influence of outlies.

Furthermore, we can observe that our algorithm achieved excellent improvement compared to algorithm [4] on a high-dimensional data set. This supports the idea that $L_{2,1}$-norm-based clustering algorithms reduce dimension and remove irrelevant features to improve performance.

4.6.5 Parameters' Sensitivity

To consider the "parameter sensitivity" of FSDS algorithm, we varied the parameters α, γ and β of the objective function and recorded the clustering results in terms of ACC and Purity for the eight data sets in Figure 4.3 and Figure 4.4.

First, different data sets needed different ranges of parameters to achieve the best performance. For example, the algorithm achieved the best ACC (85%) and Purity (85%) on data set Musk when parameters α =1, γ = 0.001 and β = 0.001. But for the data set Cardiotocography, the proposed clustering algorithm achieved the best ACC (77.89%) and Purity (77.89%) when α = 1, γ =7 and β = 1. Thus the proposed clustering algorithm is data-driven. Since the algorithm is sensitive to the parameters, the performance depends on parameter combinations. The parameter α is used to tune the auxiliary variable **F**. The parameter γ tunes the sparsity of the transfer matrix **W**, so different γ produce different levels of sparsity of **W**, and so in turn different percentages of redundant features are removed from the original data set. The parameter β is used to tradeoff the importance of similarity matrix **S**. Finally from Figure 4.3-4.4 we can perceive that parameter α and γ are more sensitive than β on the eight benchmark data sets.

4.6.6 Convergence

Figure 4.5 shows the trend of objective values generated with respect to iterations. We set the stopping criteria of the proposed clustering algorithm to $\left|obj_{(t+1)} - obj_{(t)}\right| / obj_{(t)} \leq 10^{-9}$, where $obj_{(t)}$ represents the objection function

value of Eq. (4.6) after the *t*-th iteration. From Figure. 4.5, we see that the value of the objective function monotonically decreases until it converges, when we optimize the proposed objective function in Eq. (4.6). The convergence rate of Algorithm 4.1 is relatively fast. It converges to the optimal value within 40 iterations on all the eight data sets. It actually converged to the optimal value even faster for some data sets such as Diabetic Retinopathy Debrecen or Balance.

4.7 Conclusion

In this chapter we have proposed a new Joint Feature Selection with Dynamic Spectral (FSDS) clustering algorithm to solve the cluster number *K* estimation, similarity matrix learning, feature selection, and outlier reduction issues of clustering algorithms in a unified way. Specifically, the proposed clustering algorithm learns the similarity matrix based on the data distribution. Then it adds the rank constraint on the Laplacian matrix of the learned similarity matrix to solving the cluster number *K* determination issue. At the same time, the proposed clustering algorithm applies the $L_{2,1}$-norm as the sparse constraints to minimize both loss function and regularization term of the objective function to reduce the influence of outliers and to remove the redundant features. Experimental results on eight real-world benchmark data sets showed that the proposed clustering algorithm performed better than the related clustering algorithms.

Although the proposed FSDS clustering algorithm achieved good clustering results overall, we haven't tested multi-view data sets. Hence, future research needs to find a new clustering algorithm to learn the clustering number *K*, and similarity

automatically in a unified way and have capability of feature selection and outlier reduction for multi-view data sets.

Table 4.2 ACC results of FSDS algorithm on eight benchmark data sets

Datasets	*K*-means	Spectral	CAN	RCC	IS	FSDS
Cardiotocography	0.5176	0.7785	0.7775	0.7785	0.7785	**0.7789**
Diabetic Retinopathy	0.5439	0.5421	0.5361	0.5308	0.5730	**0.5752**
Parkinson Speech	0.5094	**0.6083**	0.5010	0.5000	0.5219	0.5385
German Credit	0.6256	0.6990	0.6800	**0.7000**	**0.7000**	**0.7000**
Australian Credit	0.6258	0.5864	0.6899	0.6246	0.6800	**0.6900**
Balance Scale	0.5593	0.5449	0.6432	0.4608	0.5632	**0.6848**
Credit Approval	0.5732	0.5870	0.5333	0.5580	0.5841	**0.6913**
Musk (Version 2)	0.5450	0.8039	0.6800	0.4169	**0.8459**	**0.8459**
Rank	4	3	3	5	2	1

Table 4.3 Purity results of FSDS algorithm on eight benchmark data sets

Datasets	*K*-means	Spectral	CAN	RCC	IS	FSDS
Cardiotocography	0.7785	0.7790	**0.7850**	0.7785	0.7788	0.7789
Diabetic Retinopathy	0.5439	0.5421	0.5361	0.5308	0.6455	**0.6785**
Parkinson Speech	0.5094	0.6083	0.501	0.6911	0.7423	**0.7654**
German Credit	**0.7000**	**0.7000**	**0.7000**	**0.7000**	**0.7000**	**0.7000**
Australian Credit	0.6258	0.5864	0.6899	**0.9151**	0.8478	0.6900
Balance Scale	0.6724	0.5247	**0.7392**	0.4608	0.6137	0.6848
Credit Approval	0.5884	0.5870	0.5551	**0.7910**	0.6464	0.6913
Musk (Version 2)	**0.8459**	**0.8459**	0.7000	0.4150	**0.8459**	**0.8459**
Rank	4	5	3	3	2	1

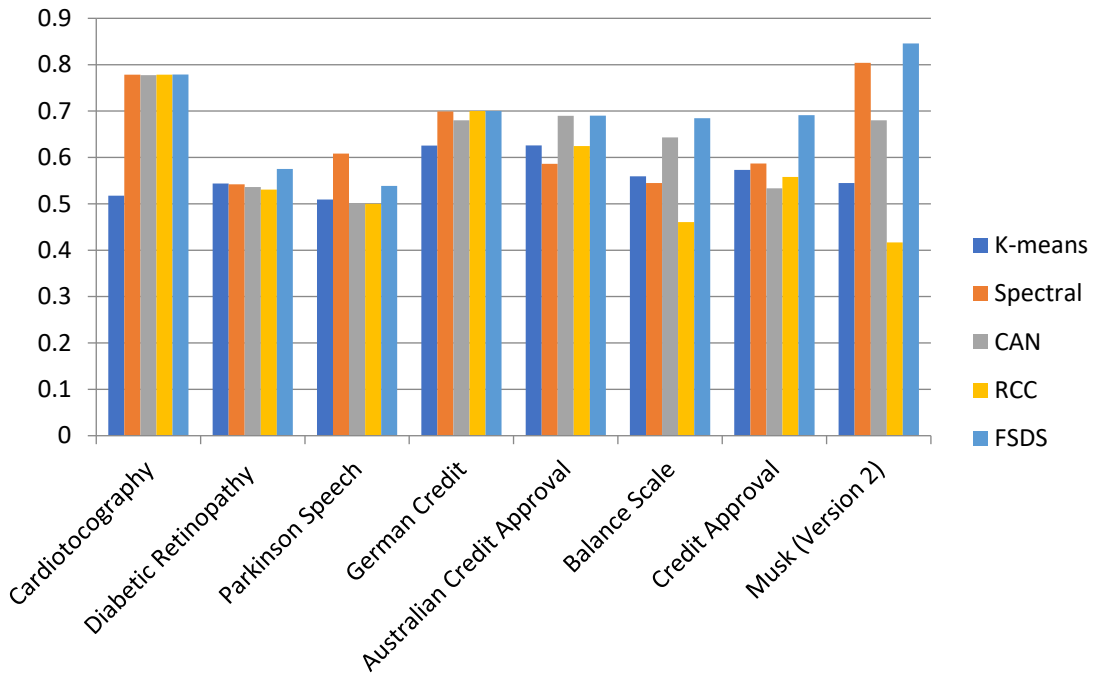


Figure 4.1 ACC results of FSDS algorithm on eight benchmark data sets

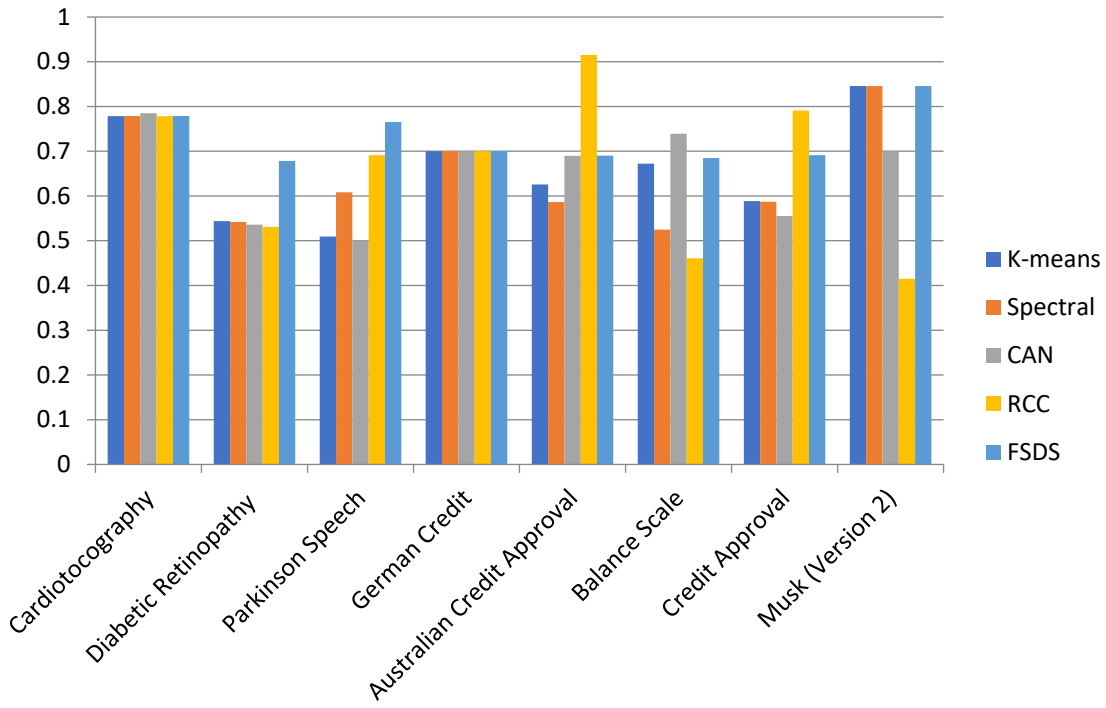
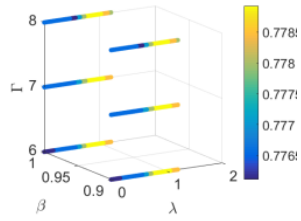
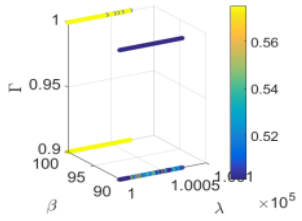


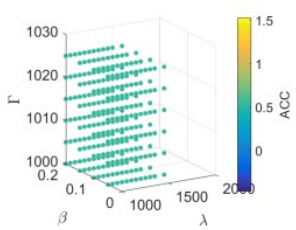
Figure 4.2 Purity results of FSDS algorithm on eight benchmark data sets



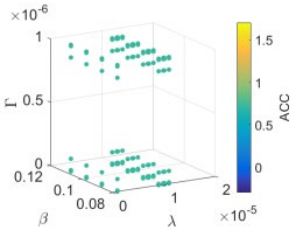
(a) Cardiotocography



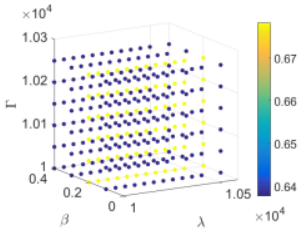
(b) Diabetic Retinopathy Debrecen



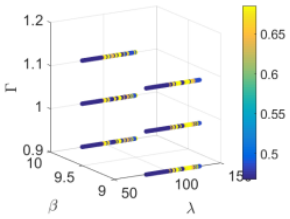
(c) Parkinson Speech



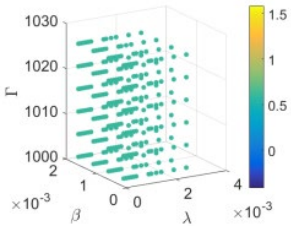
(d) German Credit



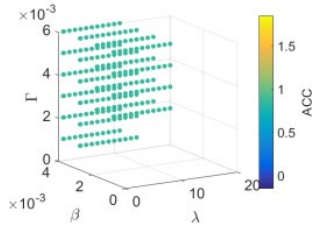
(e) Australian Credit Approval



(f) Balance Scale

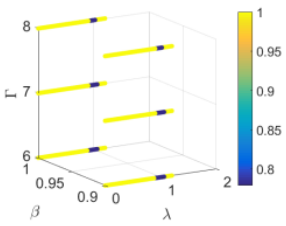


(g) Credit Approval

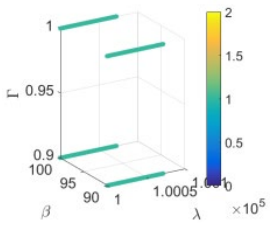


(h) Musk (Version 2)

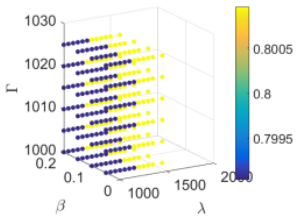
Figure 4.3 ACC results of FSDS algorithm with respect to different parameter settings



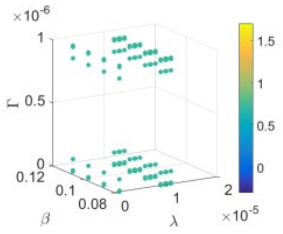
(a) Cardiotocography



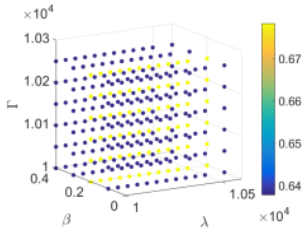
(b) Diabetic Retinopathy Debrecen



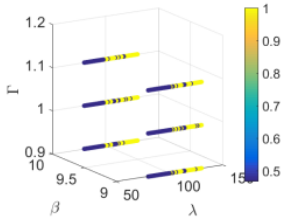
(c) Parkinson Speech



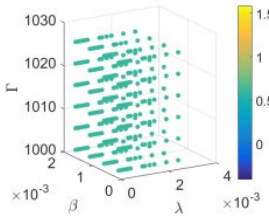
(d) German Credit



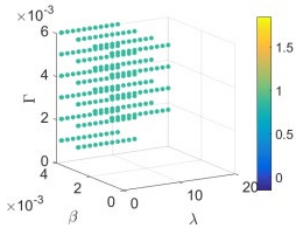
(e) Australian Credit Approval



(f) Balance Scale

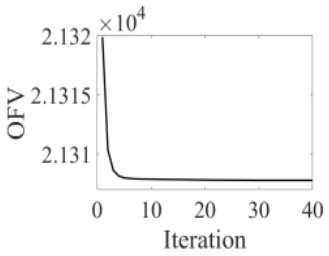


(g) Credit Approval

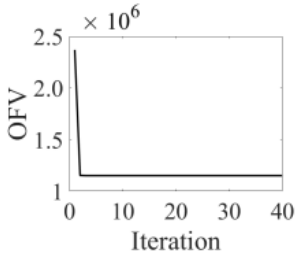


(h) Musk (Version 2)

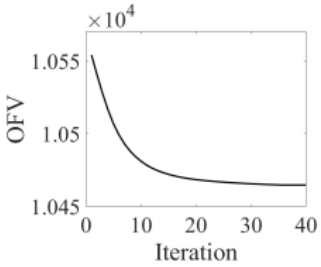
Figure 4.4 Purity results of FSDS algorithm with respect to different parameter settings



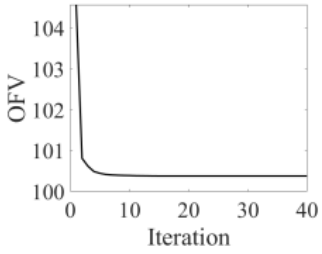
(a) Cardiotocography



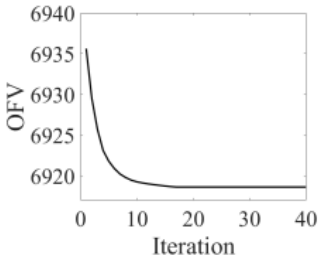
(b) Diabetic Retinopathy Debrecen



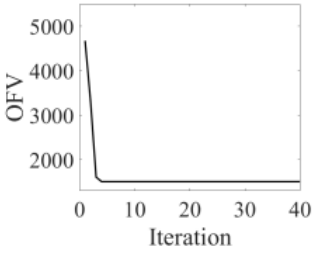
(c) Parkinson Speech



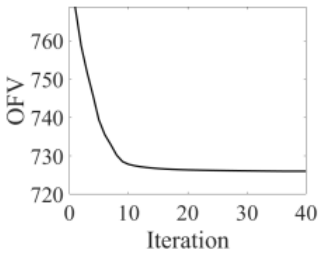
(d) German Credit



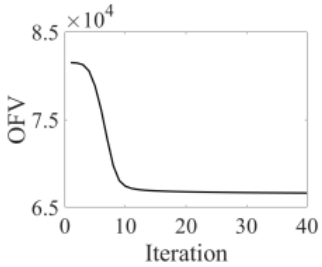
(e) Australian Credit Approval



(f) Balance Scale



(g) Credit Approval



(h) Musk (Version 2)

Figure 4.5 Objective function values (OFVs) versus iterations for FSDS algorithm

Chapter 5

Joint Robust Multi-view Spectral Clustering

5.1 Introduction

Chapter 4 improved the predefined cluster number *K* and similarity measurement, feature selection and outlier reduction of *K*-means clustering algorithm. However, it is designed for single view dataset. In real world, data is often collected from multiple sources or from different aspects of the data. A data set containing information from multiple views is called a multi-view data set. Each view of the data has its own properties to contribute to the understanding of the subject matter. Different views provide complementary information, which helps our information discovery purpose such as clustering. Many clustering algorithms were designed for single-view data set, which was the most available data set in the past [145]. A concatenation-based algorithm uses single-view clustering algorithm on the concatenated features from each view of the multi-view data set [146]. It may not lead to an optimal result because it treats different views equally even though they have their own special characteristics. Furthermore, it suffers the "curse of dimensionality" [70]. A distribution-based multi-view algorithm synthesizes the clustering results from individual view to get final clustering result. Similar to concatenation-based approach, distribution-based approach is unavailable to yield optimal results as it still not fully use the information from multi-view data set [75]. Compare to both concatenation-based and distribution-based approaches, a centralization-based approach achieves better performance because it

considers information from all views to conduct clustering [76]. For instance, Graph-based system (GBS) [26], Adaptively weighted Procrustes (AWP) [27], and Multi-view low-rank sparse subspace clustering (MLRSSC) [28] are centralization-based multi-view clustering algorithms. But these three algorithms all use a multi-stage clustering strategy. GBS extracts data feature matrix of each view in the first stage, and then constructs graph matrices of all view in the second stage, finally conducts clustering on the unified graph matrix generated in the last stage [26]. AWP constructs embedding matrix in the first stage and conducts the clustering in the final stage [27]. MLRSSC learns the joint affinity matrix in the first stage, and then uses the spectral clustering algorithm to complete clustering in the final stage [28]. But the goal of the first stage may not guarantee the optimal clustering result for the second stage. Thus, algorithms using multi-stage approaches may not guarantee an optimal clustering result. So, in this chapter, we further improved *K*-means clustering algorithm by developing a new centralization-based multi-view clustering algorithm addressing initialization, similarity measurement, cluster number determination, outliers reduction and feature selection issues in a unified way.

To alleviate the significant influence of outliers, the L_1-norm, L_2-norm, or $L_{2,1}$-norm are often used in objective functions [117, 147]. The L_1-norm-based algorithms tend to give unstable or multiple solutions. Many current clustering algorithms used the L_2-norm, but the L_2-norm-based algorithms tend to give not very robust solution. The proposed algorithm adopts the $L_{2,1}$-norm minimization with sparse constraints on the

objective function to reduce the influence of outliers, at the same time adopts the $L_{2,1}$-norm on the regularization term to conduct feature selection.

Compared to previous algorithms using multi-stage strategies to conduct clustering, the proposed algorithm aims to solving initialization, cluster number determination, similarity measure, feature selection, and outlier reduction issues around clustering for multi-view data set in a unified way. The optimal performance is reached when the separated stages are combined in a unified way. We utilize an alternating strategy to solving the proposed objective function. Experiments performed on six real-world benchmark data sets show that the proposed algorithm outperforms the comparison clustering algorithms in terms of two evaluation metrics for clustering algorithms including accuracy (ACC) and Purity.

We briefly summarize the contributions of the proposed clustering algorithm as follows:

- It is a new centralization-based multi-view clustering algorithm. It achieves better performance compared to both concatenation-based and distribution-based approaches because it considers information from all views to conduct clustering.
- A unified way addresses initialization, similarity matrix learning, and cluster number determination issues around clustering. The performance is more promising comparing to clustering algorithm GBS [26], AWP [27], and MLRSSC [28] when the multiple stages are combined in a unified way.

- The cluster number is automatically generated. Many of the current clustering algorithms need a priori knowledge of the cluster number beforehand to conduct clustering.
- The similarity measure is automatically generated based on the data distribution instead of using Euclidean distance like *K*-means clustering algorithm does.
- $L_{2,1}$-norm minimization with sparse constrains employed on the objective function and regularization term to reduce the influence of outliers and select useful features. Compare to algorithms based on L_1-norm and L_2-norm, the proposed clustering is more effective for outlier reduction and feature selection.
- The proposed clustering algorithm outperforms four clustering algorithms. It implies that simultaneously addressing the five issues (initialization, cluster number determination, similarity measure, feature selection and outlier reduction) of multi-view clustering algorithm is feasible and robust.

This section has laid the background of this paper. The remainder of the paper is organized as follows: Section 5.2 discusses the motivation behind the development of the Joint Robust Multi-view (JRM). Section 5.3 presents the proposed JRM spectral clustering algorithm. Section 5.4 provides the optimization process. Section 5.5 provides the convergence analysis. Section 5.6 presents the experiments we conducted and discusses the results of the experiments. The conclusions, limitations and future research direction are presented in Section 5.7.

5.2 Motivation

In the former two chapters, we respectively proposed two clustering methods to consider the problems of initialization, similarity measurement, cluster number determination, and feature selection and outlier deduction. However, both two methods are designed to conduct clustering on single-view data. To find how other algorithms improves K-means clustering algorithm using for the multi-view dataset, we investigated K-means clustering algorithm, Graph-based system (GBS), Adaptively weighted Procrustes (AWP), and Multi-view low-rank sparse subspace clustering (MLRSSC) in details.

K-means clustering algorithm is one of the most famous classic clustering algorithms. The *K*-means clustering algorithm aims at minimizing a sum of squared loss function shown in Eq. (5.1).

$$\sum_{i=1}^{N} \sum_{k=1}^{K} \delta_{ik} \|\mathrm{x}_i - \mathrm{v}_k\|_2^2 \tag{5.1}$$

Where N is the total number of data points, K is number of clusters, x_i is *i-th* data point, δ_{ik} is an indicator variable, C_k is data points in the *K-th* cluster, $\delta_{ik} = 1$ if $\mathrm{x}_i \in \mathrm{C}_k$; $\delta_{ik} = 0$ if $\mathrm{x}_i \notin \mathrm{C}_k$, v_k is the *K-th* cluster center. $\|\mathrm{x}_i - \mathrm{v}_k\|$ is the Euclidean distance between x_i and v_k. For multi-view clustering, the features are concatenated across all views into a long vector before the *K*-means clustering is applied. The *K*-means clustering relies on the given cluster number *K*. As an unsupervised machine learning algorithm, the *K*-means clustering is used against data which is not labelled. Without known label or pattern, the cluster number may not be known prior. The

similarity measure of the *K*-means clustering algorithm only depends on the Euclidean distance. Euclidean distance measure does not account for factors such as cluster sizes, dependent features or density [18, 30].

Graph-Based system (GBS) automatically assigns weights to the constructed graph of each view, and then generates a unified graph matrix [26]. The objective function is shown in Eq. (5.2).

$$\min_{\mathbf{U}} \sum_{v=1}^{V} \boldsymbol{w}_v \|\mathbf{U} - \mathbf{S}^v\|_{\mathrm{F}}^2 + 2\lambda Tr(\mathbf{F}^T \mathbf{L}_u \mathbf{F}) \tag{5.2}$$

$$s.t. s_{ii}^v = 0, s_{ij}^v \geq 0, \mathbf{1}^T \boldsymbol{s}_i^v = \mathbf{1}, u_{ij} \geq 0, \mathbf{1}^T \boldsymbol{u}_i = \mathbf{1}, \mathbf{F}^T \mathbf{F} = \mathbf{I}$$

where $\boldsymbol{w}_v$ is weight of the *v-th* view. $\mathbf{U} \in \mathbb{R}^{n \times n}$ is the unified matrix, $\mathbf{S}$ is the similarity-induced graph matrices $\{\mathbf{S}^1 \dots \mathbf{S}^v\}$. $\mathbf{F} = \{\boldsymbol{f}_1, \dots \boldsymbol{f}_c\}$ is the embedding matrix. $\mathbf{L}_u$ is graph Laplacian matrix of $\mathbf{U}$ and it dynamically generates the weight of each graph matrix. But it needs the number of neighbors prior as well as constructing the graph of each view separately and the constructed graphs are unable to update. The learning of the unified graph and the constructing graphs are in two separate stages.

Adaptively weighted Procrustes (AWP) assigns weights to each view with its clustering capacity and forms a weighted Procrustes average problem accordingly [27]. The objective function of AWP is presented in Eq. (5.3).

$$\min_{\mathbf{Y}, \{\mathbf{R}^{(v)}\}_V} \sum_{v=1}^{V} \left\|\mathbf{Y} - \mathbf{F}^{(i)} \mathbf{R}^{(i)}\right\|_F \tag{5.3}$$

$$s.t. \mathbf{Y} \in \mathbf{Ind}, \left(\mathbf{R}^{(i)}\right)^T \mathbf{R}^{(i)} = \mathbf{I}, \ \forall i = 1 \dots V$$

where $\mathbf{Y} \in \mathbf{Ind}$ is an indicator matrix, $\mathbf{F}^{(v)} \in \mathbb{R}^{n\times k}$ is the spectral embedding, $\mathbf{R}^{(v)} \in \mathbb{R}^{k\times k}$ is a rotation matrix.

AWP requires spectral embedding matrix calculated prior as an input. The goal of conducting the spectral embedding matrix is different from the second stage goal of multi-view clustering, and thus not guaranteed to always perform well.

Multi-view low-rank sparse subspace clustering (MLRSSC) jointly learns an affinity matrix constrained by sparsity and low-rank, while at the same time balances between the agreements across different views [28]. The objective function of MLRSSC is shown in Eq. (5.4).

$$\min_{\mathbf{C}^{(1)},\mathbf{C}^{(2)},\ldots,\mathbf{C}^{(V)}} \sum_{v=1}^{V} \left(\frac{1}{2}\left\|\Phi\left(\mathbf{X}^{(v)}\right) - \Phi\left(\mathbf{X}^{(v)}\right)\mathbf{C}^{(v)}\right\|_{\mathrm{F}}^{2}\right) + \beta_1\left\|\mathbf{C}^{(v)}\right\|_{*} + \beta_2\left\|\mathbf{C}^{(v)}\right\|_{1}$$
$$+ \lambda^{(v)}\left\|\mathbf{C}^{(v)} - \mathbf{C}^{*}\right\|_{F}^{2}),\ s.t., \mathrm{diag}(\mathrm{C}^{(v)}) = 0, v = 1, \ldots V. \quad (5.4)$$

Where $\Phi(\mathbf{X}^{(v)})$ is a function that maps the original input space $\mathbf{X}^{(v)} = \{\mathbf{X}_i^{(v)} \in \mathbb{R}^{D}\}_{i=1}^{N}$ in *v*-th view into a high-dimensional feature space. $\mathbf{C}^{(v)} \in \mathbb{R}^{N\mathrm{x}N}$ is the representation matrix for *v-th* view. $\mathbf{C}^{*} \in \mathbb{R}^{N\mathrm{x}N}$ denotes cluster center matrix.

MLRSSC learns the joint affinity matrix first, and then uses the spectral clustering algorithm to complete the final clustering. The learning of the affinity matrix and final spectral clustering are in two separate stages.

5.3 Proposed Algorithm

This paper proposes a new centralization-based multi-view clustering algorithm (i.e., Joint Robust Multi-view (JRM) spectral clustering) to concurrently address the

challenges of clustering algorithms i.e., initialization, automatic cluster numbers determination, similarity matrix learning, the feature selection and the outliers reduction for multi-view clustering algorithms in a unified framework. To achieve our goal, we initialize the new representative as the original multi-view data, applies sum-of-square error estimation to minimize the difference between the original data and its new representative, applies sum-of-norm regularization to control model fit and automatically generate the cluster number, learns the similarity matrix based on the data distribution, and at the same time uses $L_{2,1}$-norm to select the important features and reduce the outliers. We form the objective function of the proposed clustering algorithm in Eq. (5.5).

$$\min_{\mathbf{S},\mathbf{U},\mathbf{W}^v} \ \frac{1}{2}\sum_{v=1}^{V}\|\mathbf{X}^v\mathbf{W}^v-\mathbf{U}\|_{2,1}+\frac{\alpha}{2}\sum_{i,j=1}^{n}s_{i,j}\rho(\left\|\mathbf{u}_i-\mathbf{u}_j\right\|_2)$$

$$+r\sum_{v=1}^{V}\|\mathbf{W}^v\|_{2,1}+\beta\|\mathbf{S}^v\|_F^2, s.t., \forall i, s_{i,j}\geq 0, \boldsymbol{s}_i^T\boldsymbol{e}=1 \tag{5.5}$$

where $\{v=1,\dots,V\},\{i=1,\dots,n\},\{j=1,\dots,n\}$, V is the total number of views, n is the number of data points, $\mathbf{X}^v\in\mathbb{R}^{n\times d^v}$ is data matrix in the *v-th* view. $d^v$is the features of data in the *v-th* view. $\mathbf{W}^v\in\mathbb{R}^{d^v\times d^v}$is the weight matrix of *v-th* view to balance the contribution of *v-th* data view, $\mathbf{U}\in\mathbb{R}^{n\times c}$ is the common representation of $\mathbf{X}^v$, and $\mathbf{S}^v\in\mathbb{R}^{n\times n}$ is the similarity matrix to measure the similarity among data points, $\rho(\left\|\mathbf{u}_i-\mathbf{u}_j\right\|_2)$ is a robust loss function, which is used for generating cluster number automatically. $L_{2,1}$-norm enforces sparse in rows, making it especially suitable for the outliers reduction and feature selection.

Eq. (5.5) learns the new common representation **U** and learns the similarity matrix **S** based on the data distribution, i.e., iteratively updated by the updated **U**. Furthermore, Eq. (5.5) learns weight matrix $\mathbf{W}^v$ for each view. This produces an intelligent new common representation of the original multi-view data matrix. The $L_{2,1}$-norm usually generates sparse solutions [117, 118]. That is to say, the residue $\sum_{v=1}^{V}\|\mathbf{X}^v\mathbf{W}^v - \mathbf{U}\|_{2,1}$ and regularization $\sum_{v=1}^{V}\|\mathbf{W}^v\|_{2,1}$ will take on discrete values and have more zero elements. Moreover, Eq. (5.5) will keep the distance of indicator vectors similar if data belongs to the same cluster, possibly making them equal. The distance of indicator vectors is separated if data belongs to the different clusters.

Several robust loss functions have been proposed to automatically generate cluster numbers [131, 132]. Here we employ the Geman-McClure function [133]:

$$\rho\left(\left\|\mathbf{u}_p - \mathbf{u}_q\right\|_2\right) = \frac{\mu\left\|\mathbf{u}_p - \mathbf{u}_q\right\|_2^2}{\mu + \left\|\mathbf{u}_p - \mathbf{u}_q\right\|_2^2} \tag{5.6}$$

where ρ (.) is robust estimator constructed by the half-quadratic theory [143, 148]. Eq. (5.6) measures how well our model predicts the expected outcome. The smaller the value of $\left\|\mathbf{u}_p - \mathbf{u}_q\right\|_2^2$ is, the closer the distance between two data points is, and the higher the similarity between two data points is. With the update of other variables in Eq. (5.5), the distance $\left\|\mathbf{u}_p - \mathbf{u}_q\right\|_2^2$ for data points p and q, will be very close, or even $\mathbf{u}_p = \mathbf{u}_q$, and the clusters will be formed.

It is a normal practice to introduce an auxiliary variable $f_{i,j}$ and a penalty item $\varphi(f_{i,j})$ to a robust loss function, due to the difficult of the optimization [134-136]. Thus Eq. (5.5) is rewritten as is equivalent to:

$$\min_{\mathbf{S},\mathbf{U},\mathbf{F},\mathbf{W}^v} \frac{1}{2}\sum_{v=1}^{V}\|\mathbf{X}^v\mathbf{W}^v - \mathbf{U}\|_{2,1} + \frac{\alpha}{2}\sum_{i,j=1}^{n} s_{i,j}\left(f_{i,j}\|\mathbf{u}_i - \mathbf{u}_j\|_2^2 + \varphi(f_{i,j})\right) + r\sum_{v=1}^{V}\|\mathbf{W}^v\|_{2,1} + \beta\|\mathbf{S}^v\|_F^2, \quad s.t., \forall i, s_{i,j} \geq 0, \boldsymbol{s}_i^T\boldsymbol{e} = 1 \tag{5.7}$$

Where $\varphi(f_{i,j}) = \mu(\sqrt{f_{i,j}} - 1)^2, i, j = 1 \ldots n$

Algorithm 5.1. The pseudo code for our proposed JRM clustering algorithm

Input: $\mathbf{X}^v \in \mathbb{R}^{n \times d^v}$

Output: a set of *K* clusters

- Update $\mathbf{W}^v$ using Eq. (5.17)
- Update **F** using Eq. (5.20)
- Update **S** using Eq. (5.24)
- Update U using Eq. (5.36)

Until U converges

This objective function is still difficult to solve. An iterative optimization algorithm is adopted to address the difficulties of the proposed method. Thus, in the next section, we will introduce how we solve the problem using iterative optimization algorithm.

5.4 Optimization

Equation. (5.7) is convex on each variable of $\mathbf{W}^v$, U, **F**, and **S** while fixing the rest. The alternating optimization strategy is applied to solving the Eq. (5.7). Specifically,

we optimize each variable while fixing the rest until the objective function converges. The pseudo-code of the proposed clustering algorithm is given in Algorithm 5.1.

1) Update W^v **while fixing F, S and** U

While **F, S and** U are fixed, the objective function is transformed to a simplified matrix form to optimize $\mathbf{W}^v$:

$$\min_{\mathbf{W}^v} \frac{1}{2}\sum_{v=1}^{V}\|\mathbf{X}^v\mathbf{W}^v - \mathbf{U}\|_{2,1} + \ r\sum_{v=1}^{V}\|\mathbf{W}^v\|_{2,1} \tag{5.8}$$

Let $\mathbf{D}^v$, $\mathbf{M}^v$ be the diagonal matrix, and they are defined in Eq. (5.9) and Eq. (5.10), respectively.

$$d_{ii}^v = \frac{1}{2\|(\mathbf{X}^v\mathbf{W}^v - \mathbf{U})^i\|_2},\ i = 1,\ldots,n \tag{5.9}$$

$$m_{ii}^v = \frac{1}{2\|(\mathbf{W}^v)^i\|_2}, i = 1,\ldots,n \tag{5.10}$$

After applied Eq. (5.9) and Eq. (5.10), Eq. (5.8) is rewritten in the following forms:

$$\min_{\mathbf{W}^v} \frac{1}{2}tr\left(\mathbf{W}^{vT}\mathbf{X}^{v\mathrm{T}}\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v - \mathbf{W}^{vT}\mathbf{X}^{vT}\mathbf{D}^v\mathbf{U} - \mathbf{U}^T\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v + \mathbf{U}^T\mathbf{D}^v\mathbf{U}\right) + rtr(\mathbf{W}^{vT}\mathbf{M}^v\mathbf{W}^v) \tag{5.11}$$

$$\Rightarrow \min_{\mathbf{W}^v}\frac{1}{2}(tr(\mathbf{W}^{vT}\mathbf{X}^{v\mathrm{T}}\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v) - tr(\mathbf{W}^{vT}\mathbf{X}^{vT}\mathbf{D}^v\mathbf{U}) - tr(\mathbf{U}^T\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v) + tr(\mathbf{U}^T\mathbf{D}^v\mathbf{U}) + rtr(\mathbf{W}^{vT}\mathbf{M}^v\mathbf{W}^v)) \tag{5.12}$$

$$\Rightarrow \min_{\mathbf{W}^v} \frac{1}{2}(tr(\mathbf{W}^{vT}\mathbf{X}^{v\mathrm{T}}\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v) - 2tr(\mathbf{U}^T\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v) + tr(\mathbf{U}^T\mathbf{D}^v\mathbf{U}) + rtr(\mathbf{W}^{vT}\mathbf{M}^v\mathbf{W}^v)) \tag{5.13}$$

By taking a derivative of $\mathcal{L}(\mathbf{W}^v)$ on Eq. (5.13) with respect to $\mathbf{W}^v$ and setting the derivative to be zero, we see:

$$\frac{1}{2}\left(2\mathbf{X}^{v^T}\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v - 2\mathbf{X}^{v^T}\mathbf{D}^v\mathbf{U}\right) + r2\mathbf{M}^v\mathbf{W}^v = 0 \tag{5.14}$$

$$\Rightarrow (\mathbf{X}^{v^T}\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v - \mathbf{X}^{v^T}\mathbf{D}^v\mathbf{U} + 2r\mathbf{M}^v\mathbf{W}^v) = 0 \tag{5.15}$$

$$\Rightarrow \left(\mathbf{X}^{v^T}\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v + 2r\mathbf{M}^v\mathbf{W}^v\right) = \mathbf{X}^{v^T}\mathbf{D}^v\mathbf{U} \tag{5.16}$$

The solution is shown as the following:

$$\Rightarrow \mathbf{W}^v = (\mathbf{X}^{v^T}\mathbf{D}^v\mathbf{X}^v + 2r\mathbf{M}^v)^{-1}\mathbf{X}^{v^T}\mathbf{D}^v\mathbf{U} \tag{5.17}$$

The problem (5.8) has been solved to get $\mathbf{W}^v$. The detail of the algorithm is described in Algorithm 5.2. Later, we will prove that Algorithm 5.2 can make problem (5.8) converge.

Algorithm 5.2. Algorithm to solve the problem described in Eq. (5.8)

Input: $\mathbf{X}^v \in \mathbb{R}^{n\times d^v}$, $\mathbf{U} \in \mathbb{R}^{n\times C}$

Output: Projection matrix $\mathbf{W}^v$

Repeat:

- With current $\mathbf{U}, \mathbf{M}^v$, $\mathbf{D}^v$ the optimal solution $\mathbf{W}^v$ is obtained by Eq. (5.17)
- With current $\mathbf{W}^v$and $\mathbf{D}^v$, $\mathbf{U}$ is obtained by Eq. (5.36)
- With current $\mathbf{W}^v$ and $\mathbf{U}$, $\mathbf{D}^v$ is obtained by Eq. (5.9)
- With current $\mathbf{W}^v$, $\mathbf{M}$ is obtained by Eq. (5.10)

Until $\mathbf{W}^v$ converges

2) Update F while fixing $\mathbf{W}^v$, S and U

While $\mathbf{W}^v$, $\mathbf{S}$, and $\mathbf{U}$ are fixed, the objective function on Eq. (5.7) can be rewritten in a simplified matrix form to optimize $\mathbf{F}$:

$$\min_{\mathbf{F}} \frac{\alpha}{2} \sum_{i,j=1}^{n} s_{i,j} (f_{i,j} \left\| \mathbf{u}_i - \mathbf{u}_j \right\|_2^2 + \mu(\sqrt{f_{i,j}} - 1)^2) \tag{5.18}$$

Since the optimization of $f_{i,j}$ is independent of the optimization of other $f_{p,q}, i \neq p, j \neq q$, the $f_{i,j}$ is optimized first as shown in following

$$\frac{\alpha}{2} (s_{i,j} f_{i,j} \left\| \mathbf{u}_i - \mathbf{u}_j \right\|_2^2 + s_{i,j} (\mu(f_{i,j} - 2\sqrt{f_{i,j}} + 1)) \tag{5.19}$$

By conducting a derivative on Eq. (5.19) with respect to $f_{i,j}$, we get

$$f_{i,j} = \left(\frac{\mu}{\mu + \left\| \mathbf{u}_i - \mathbf{u}_j \right\|_2^2} \right)^2 \tag{5.20}$$

3) Update S while fixing $\mathbf{W}^v$, U and F

While fixing $\mathbf{W}^{\boldsymbol{v}}$, $\mathbf{U}$, and $\mathbf{F}$, the objective function Eq. (5.7) with respect to $\mathbf{S}$ is:

$$\min_{\mathbf{S}} \frac{\alpha}{2} \sum_{i,j=1}^{n} s_{i,j} (f_{i,j} \left\| \mathbf{u}_i - \mathbf{u}_j \right\|_2^2 + \mu(\sqrt{f_{i,j}} - 1)^2) + \beta \|\mathbf{S}\|_F^2 \tag{5.21}$$

$$s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T \mathbf{e} = 1$$

Since the optimization of $\mathbf{S}_i$is independent of the optimization of other $\mathbf{S}_j, i \neq j, i, j = 1, \ldots, n$, the $\mathbf{s}_i$ is optimized as shown in following:

$$\min_{\mathbf{s}_i} \frac{\alpha}{2} \sum_{j=1}^{n} s_{i,j} (f_{i,j} \left\| \mathbf{u}_i - \mathbf{u}_j \right\|_2^2 + \mu(\sqrt{f_{i,j}} - 1)^2) + \beta \sum_{i=1}^{n} \|\mathbf{s}_i\|_2^2 \tag{5.22}$$

$$s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T \mathbf{e} = 1$$

Let $b_{i,j} = f_{i,j} \left\| \mathbf{u}_i - \mathbf{u}_j \right\|_2^2$ and $c_{i,j} = \mu(\sqrt{f_{i,j}} - 1)^2$, Eq. (5.22) is equivalent to:

$$\min_{\mathbf{s}_i} \left\| \mathbf{s}_i + \frac{\alpha}{4\beta}(\mathbf{b}_i + \mathbf{c}_i) \right\|_2^2, s.t., \forall i, s_{i,j} \geq 0, \mathbf{s}_i^T \mathbf{e} = 1 \quad (5.23)$$

According to Karush-Kuhn-Tucker (KKT) [137], the optimal solution s_i should be

$$\mathrm{s}_{i,j} = \max\{-\frac{\alpha}{4\beta}(\mathrm{b}_{i,j} + \mathrm{c}_{i,j}) + \theta, 0\}, \ j = 1, \dots, n \quad (5.24)$$

where $\theta = \frac{1}{\rho}\sum_{j=1}^{\rho}\left(\frac{\alpha}{4\beta}(\mathrm{b}_{i,j} + \mathrm{c}_{i,j}) + 1\right)$, and $\rho = max_j\{\omega_j - \frac{1}{j}\left(\sum_{r=1}^{j}\omega_r - 1\right), 0\}$ and ω is the descending order of $\frac{\alpha}{4\beta}(\mathrm{b}_{i,j} + \mathrm{c}_{i,j})$.

4) Update U while fixing $\mathbf{W}^v$, S and F

While $\mathbf{W}^v$, $\mathbf{S}$,and $\mathbf{F}$ are fixed, the objective function can be rewritten in a simplified form to optimize $\mathbf{U}$:

$$\min_{\mathbf{U}} \frac{1}{2}\sum_{v=1}^{V}\|\mathbf{X}^v\mathbf{W}^v - \mathbf{U}\|_{2,1} + \frac{\alpha}{2}\sum_{i,j=1}^{n} s_{i,j}(f_{i,j}\left\|\mathbf{u}_i - \mathbf{u}_j\right\|_2) \quad (5.25)$$

Let $\mathbf{S}_{SF} = \frac{(\mathbf{S}\odot\mathbf{F})^T + (\mathbf{S}\odot\mathbf{F})}{2}$. The degree matrix $\mathbf{D}_s = diag(\mathbf{S}_{SF}\mathbf{1})$.The Laplacian Matrix $\mathbf{L}$ is defined below

$$\mathbf{L} = \mathbf{D}_s - \mathbf{S}_{SF} \quad (5.26)$$

After applied Eq.(5.26), Eq. (5.25) is equivalent to:

$$\min_{\mathbf{U}} \frac{1}{2}\sum_{v=1}^{V}\|\mathbf{X}^v\mathbf{W}^v - \mathbf{U}\|_{2,1} + \frac{\alpha}{2} tr(\mathbf{U}_t^T \mathbf{L}\, \mathbf{U}_t) \quad (5.27)$$

Let $d_{ii}^v = \frac{1}{2\|(\mathbf{X}^v\mathbf{W}^v - \mathbf{U})^i\|_2}$, and Eq. (5.27) is equivalent to:

$$\min_{\mathbf{U}} \frac{1}{2}\sum_{v=1}^{V}\mathrm{tr}((\mathbf{W}^{vT}\mathbf{X}^{vT} - \mathbf{U}^T)\mathbf{D}^v(\mathbf{X}^v\mathbf{W}^v - \mathbf{U})) + \frac{\alpha}{2}\mathrm{tr}(\mathbf{U}^T\mathbf{L}\mathbf{U}) \quad (5.28)$$

$$\Rightarrow \min_{\mathbf{U}} \ \frac{1}{2} \sum_{v=1}^{V} \mathrm{tr}(-\mathbf{W}^{v^T}\mathbf{X}^{v^T}\mathbf{D}^v\mathbf{U} - \mathbf{U}^T\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v + \mathbf{U}^T\mathbf{D}^v\mathbf{U}) + \frac{\alpha}{2} \mathrm{tr}(\mathbf{U}^T\mathbf{L}\mathbf{U}) \tag{5.29}$$

$$\Rightarrow \min_{\mathbf{U}} \ \frac{1}{2} \sum_{v=1}^{V} \mathrm{tr}(-2\mathbf{U}^T\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v + \mathbf{U}^T\mathbf{D}^v\mathbf{U}) + \frac{\alpha}{2} \mathrm{tr}(\mathbf{U}^T\mathbf{L}\mathbf{U}) \tag{5.30}$$

$$\Rightarrow \min_{\mathbf{U}} \ \frac{1}{2} \sum_{v=1}^{V} (\mathrm{tr}(-2\mathbf{U}^T\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v) + \mathrm{tr}(\mathbf{U}^T\mathbf{D}^v\mathbf{U})) + \frac{\alpha}{2} \mathrm{tr}(\mathbf{U}^T\mathbf{L}\mathbf{U}) \tag{5.31}$$

After taking a derivative of $\mathcal{L}(\mathbf{U})$ on Eq. (5.31) with respect to **U** and setting the derivative to be zero, we get

$$\frac{1}{2} \sum_{v=1}^{V} (-2\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v + 2\mathbf{D}^v\mathbf{U}) + \alpha\mathbf{L}\mathbf{U} = 0 \tag{5.32}$$

$$\Rightarrow \sum_{v=1}^{V} (-\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v + \mathbf{D}^v\mathbf{U}) + \alpha\mathbf{L}\mathbf{U} = 0 \tag{5.33}$$

$$\Rightarrow \sum_{v=1}^{V} (-\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v) + \sum_{v=1}^{V} \mathbf{D}^v\mathbf{U} + \alpha\mathbf{L}\mathbf{U} = 0 \tag{5.34}$$

$$\Rightarrow \sum_{v=1}^{V} (-\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v) + (\sum_{v=1}^{V} \mathbf{D}^v + \alpha\mathbf{L})\mathbf{U} = 0 \tag{5.35}$$

The term **U** can be efficiently obtained by solving the Eq. (5.35):

$$\Rightarrow \mathbf{U} = (\sum_{v=1}^{V} \mathbf{D}^v + \alpha\mathbf{L})^{-1} \sum_{v=1}^{V} (\mathbf{D}^v\mathbf{X}^v\mathbf{W}^v) \tag{5.36}$$

Algorithm 5.3. Algorithm to solve the problem described in Eq. (5.27)

Input:

$\mathbf{X}^v \in \mathbb{R}^{n \times d^v}$, Data matrix $\mathbf{W}^v \in \mathbb{R}^{n \times d^v}$, $\mathbf{S} \in \mathbb{R}^{n \times n}$

Output: Projection matrix $\mathbf{U} \in \mathbb{R}^{n \times C}$

Repeat:

- With current **S**, the Laplacian Matrix **L** is obtained by Eq. (5.26)
- With current $\mathbf{W}^v$ and **U,** $\mathbf{D}^v$ is obtained by Eq. (5.9)
- With current $\mathbf{W}^v$, $\mathbf{D}^v$, **L, U** is obtained by Eq. (5.36)

Until U converges

We adopted an iterative optimization algorithm to obtain the solution **U** such that Eq. (5.36) is satisfied, and prove that the proposed iterative algorithm 5.3 will converge in the following subsection.

5.5 Convergence Analysis

In this section, we will prove the convergence analysis of Algorithm 5.2 and Algorithm 5.3. To prove the convergence, we need the lemma proposed by Nie et al. [144].

Lemma 1. *The following inequality holds for any positive real number a and b [144].*

$$\sqrt{a} - \frac{a}{2\sqrt{b}} \leq \sqrt{b} - \frac{b}{2\sqrt{b}} \tag{5.37}$$

The convergence of Algorithm 5.2 can be proven by the following theorem.

Theorem 1. *In Algorithm 5.2, updated* $\mathbf{w}^v$ *will decrease the objective value of problem described in (5.8) until converge.*

Proof. Eq. (5.17) is the solution to the following problem:

$$min_{\mathbf{W}^v} \ \frac{1}{2} tr\left(\left(\mathbf{W}^{v^T}\mathbf{X}^{v\mathrm{T}} - \mathbf{U}\right)^T \mathbf{D}^v(\mathbf{X}^v\mathbf{W}^v - \mathbf{U}) \ + rtr(\mathbf{W}^{v^T}\mathbf{M}^v\mathbf{W}^v)\right. \tag{5.38}$$

Thus after the *t-th* iteration,

$$\mathbf{W}_{t+1}^v = \underset{\mathbf{W}_{t+1}^v}{\operatorname{argmin}} \ \frac{1}{2}\mathbf{tr}((\mathbf{X}^v\,\mathbf{W}_{t+1}^v \ - \mathbf{U}_t)^{\mathbf{T}}\mathbf{D}_t^v(\mathbf{X}^v\,\mathbf{W}_{t+1}^v - \mathbf{U}_t)\ )$$

$$+r\mathbf{tr}(\mathbf{W}_{t+1}^{(v)}\mathbf{M}_{t+1}^v\mathbf{W}_{t+1}^{(v)}) \tag{5.39}$$

The following equation can be established

$$\frac{1}{2}tr((\mathbf{X}^v\,\mathbf{W}_{t+1}^v-\mathbf{U}_t)^T\mathbf{D}_t^v\,(\mathbf{X}^v\,\mathbf{W}_{t+1}^v-\mathbf{U}_t)\;)+rtr(\mathbf{W}_{t+1}^v\mathbf{M}_{t+1}^v\mathbf{W}_{t+1}^v)$$

$$\leq\frac{1}{2}tr((\mathbf{X}^v\mathbf{W}_t^v-\mathbf{U}_t)^T\mathbf{D}_t^v\,(\mathbf{X}^v\,\mathbf{W}_t^v-\mathbf{U}_t)\;)+rtr(\mathbf{W}_t^v\mathbf{M}_t^v\mathbf{W}_t^v)\qquad(5.40)$$

We substitute the definition of $\mathbf{D}^v$ in Eq. (5.9) and $\mathbf{M}^v$ in Eq. (5.10), and then inequality Eq. (5.40) can be rewritten as:

$$\frac{1}{2}\sum_{i=1}^{n}\frac{\|(\mathbf{X}^v\,\mathbf{W}_{t+1}^v-\mathbf{U}_t)^i\;\|_2^2}{2\|(\mathbf{X}^v\,\mathbf{W}_t^v-\mathbf{U}_t)^i\;\|_2}+r\sum_{i=1}^{n}\frac{\left\|\mathbf{W}_{t+1}^{v\,i}\right\|_2^2}{2\left\|\mathbf{W}_t^{v\,i}\right\|_2}$$

$$\leq\frac{1}{2}\sum_{i=1}^{n}\frac{\|(\mathbf{X}^v\,\mathbf{W}_t^v-\mathbf{U}_t)^i\;\|_2^2}{2\|(\mathbf{X}^v\,\mathbf{W}_t^v-\mathbf{U}_t)^i\;\|_2}+\mathrm{r}\sum_{i=1}^{n}\frac{\left\|\mathbf{W}_t^{v\,i}\right\|_2^2}{2\left\|\mathbf{W}_t^{v\,i}\right\|_2}\qquad(5.41)$$

Based on Lemma 1, we know

$$\sum_{i=1}^{n}\left\|(\mathbf{X}^v\,\mathbf{W}_{t+1}^v-\mathbf{U}_t)^i\;\right\|_2-\sum_{i=1}^{n}\frac{\|(\mathbf{X}^v\,\mathbf{W}_{t+1}^v-\mathbf{U}_t)^i\;\|_2^2}{2\|(\mathbf{X}^v\,\mathbf{W}_t^v-\mathbf{U}_t)^i\;\|_2}$$

$$\leq\;\sum_{i=1}^{n}\left\|(\mathbf{X}^v\,\mathbf{W}_t^v-\mathbf{U}_t)^i\;\right\|_2-\sum_{i=1}^{n}\frac{\|(\mathbf{X}^v\,\mathbf{W}_t^v-\mathbf{U}_t)^i\;\|_2^2}{2\|(\mathbf{X}^v\,\mathbf{W}_t^v-\mathbf{U}_t)^i\;\|_2}\qquad(5.42)$$

$$\sum_{i=1}^{n}\left\|(\mathbf{W}_{t+1}^v)^i\;\right\|_2-\sum_{i=1}^{n}\frac{\|\mathbf{W}_{t+1}^v)^i\,\|_2^2}{2\|\mathbf{W}_t^v)^i\,\|_2}\;\leq\;\sum_{i=1}^{n}\left\|(\mathbf{W}_t^v)^i\;\right\|_2-\sum_{i=1}^{n}\frac{\|\mathbf{W}_t^v)^i\,\|_2^2}{2\|\mathbf{W}_t^v)^i\,\|_2}\qquad(5.43)$$

Divide inequality Eq. (5.42) by 2, and sum over with the inequality Eq. (5.41), and then sum over with inequality Eq. (5.43) multiplied by r, we obtain the following inequality

$$\sum_{i=1}^{n}\frac{1}{2}\left\|(\mathbf{X}^v\,\mathbf{W}_{t+1}^v-\mathbf{U}_t)^i\;\right\|_2+r\sum_{i=1}^{n}\left\|(\mathbf{W}_{t+1}^v)^i\;\right\|_2$$

$$\leq\sum_{i=1}^{n}\frac{1}{2}\left\|(\mathbf{X}^v\,\mathbf{W}_t^v-\mathbf{U}_t)^i\;\right\|_2+r\sum_{i=1}^{n}\left\|(\mathbf{W}_t^v)^i\;\right\|_2\qquad(5.44)$$

Hence the theorem 1 is proven,

$$\frac{1}{2}\|\mathbf{X}^v\mathbf{W}_{t+1}^v - \mathbf{U}\|_{2,1} + \ r\|\mathbf{W}_{t+1}^v\|_{2,1} \leq \frac{1}{2}\|\mathbf{X}^v\mathbf{W}_t^v - \mathbf{U}\|_{2,1} + \ r\|\mathbf{W}_t^v\|_{2,1} \quad (5.45)$$

The convergence of Algorithm 5.3 can be proven by the following theorem.

Theorem 2. *In Algorithm 5.3, updated* $\mathbf{U}$ *will decrease the objective value of problem (5.27) until converge.*

Proof. Eq. (5.36) is the solution to the problem Eq. (5.28). The *t-th* iteration of Eq. (5.28), is shown as following:

$$\mathbf{U}_{t+1} = \underset{\mathbf{U}}{\operatorname{argmin}} \ \frac{1}{2}\operatorname{tr}((\mathbf{X}^v\,\mathbf{W}^v - \mathbf{U}_t)^T\mathbf{D}_t^v(\mathbf{X}^v\,\mathbf{W}^v - \mathbf{U}_t) + \frac{\alpha}{2}\ \operatorname{tr}(\mathbf{U}_t^T\mathbf{L}\,\mathbf{U}_t)) \quad (5.46)$$

Suppose $\mathbf{D}_{t+1}^v$ is the updated $\mathbf{D}_t^v$, Eq. (5.46) indicates that

$$\frac{1}{2}\operatorname{tr}\big((\mathbf{X}^v\,\mathbf{W}^v\ - \mathbf{U}_{t+1})^T\mathbf{D}_{t+1}^v(\mathbf{X}^v\,\mathbf{W}^v - \mathbf{U}_{t+1})\big) + \frac{\alpha}{2}\ \operatorname{tr}(\mathbf{U}_{t+1}^T\mathbf{L}\,\mathbf{U}_{t+1})$$

$$\leq \frac{1}{2}\operatorname{tr}\big((\mathbf{X}^v\,\mathbf{W}^v\ - \mathbf{U}_t)^T\mathbf{D}_t^v(\mathbf{X}^v\,\mathbf{W}^v - \mathbf{U}_t)\big) + \frac{\alpha}{2}\ \operatorname{tr}(\mathbf{U}_t^T\mathbf{L}\,\mathbf{U}_t) \quad (5.47)$$

We substitute the definition of $\mathbf{D}^v$ and $\mathbf{L}$, then inequality Eq. (5.47) can be rewritten as:

$$\frac{1}{2}\sum_{i=1}^{n}\frac{\|(\mathbf{X}^v\mathbf{W}^v - \mathbf{U}_{t+1})^i\|_2^2}{2\|(\mathbf{X}^v\mathbf{W}^v - \mathbf{U}_t)^i\|_2} + \frac{\alpha}{2}\operatorname{tr}(\mathbf{U}_{t+1}^T\mathbf{L}\,\mathbf{U}_{t+1})$$

$$\leq \frac{1}{2}\sum_{i=1}^{n}\frac{\|(\mathbf{X}^v\mathbf{W}^v - \mathbf{U}_t)^i\|_2^2}{2\|(\mathbf{X}^v\mathbf{W}^v - \mathbf{U}_t)^i\|_2} + \frac{\alpha}{2}\operatorname{tr}(\mathbf{U}_t^T\mathbf{L}\,\mathbf{U}_t) \quad (5.48)$$

Based on Lemma 1, we know

$$\sum_{i=1}^{n}\big\|(\mathbf{X}^v\mathbf{W}^v - \mathbf{U}_{t+1})^i\big\|_2 - \frac{\|(\mathbf{X}^v\mathbf{W}^v - \mathbf{U}_{t+1})^i\|_2^2}{2\|(\mathbf{X}^v\mathbf{W}^v - \mathbf{U}_t)^i\|_2}$$

$$\leq \sum_{i=1}^{n} \left\| (\mathbf{X}^v \mathbf{W}^v - \mathbf{U}_t)^i \right\|_2 - \frac{\left\| (\mathbf{X}^v \mathbf{W}^v - \mathbf{U}_t)^i \right\|_2^2}{2\left\| (\mathbf{X}^v \mathbf{W}^v - \mathbf{U}_t)^i \right\|_2} \tag{5.49}$$

Divide inequality (5.49) by 2, then sum over with the inequality Eq. (5.48), we arrive at

$$\frac{1}{2}\sum_{i=1}^{n} \left\| (\mathbf{X}^v \mathbf{W}^v - \mathbf{U}_{t+1})^i \right\|_2 + \frac{\alpha}{2} tr(\mathbf{U}_{t+1}^T \mathbf{L}\, \mathbf{U}_{t+1})$$

$$\leq \frac{1}{2}\sum_{i=1}^{n} \left\| (\mathbf{X}^v \mathbf{W}^v - \mathbf{U}_t)^i \right\|_2 + \frac{\alpha}{2}\, tr(\mathbf{U}_t^T \mathbf{L}\, \mathbf{U}_t) \tag{5.50}$$

Hence theorem 2.is proven,

$$\frac{1}{2}\sum_{i=1}^{n} \left\| (\mathbf{X}^v \mathbf{W}^v - \mathbf{U}_{t+1})^i \right\|_2 + \frac{\alpha}{2} tr(\mathbf{U}_{t+1}^T \mathbf{L}\, \mathbf{U}_{t+1})$$

$$\leq \frac{1}{2}\sum_{i=1}^{n} \left\| (\mathbf{X}^v \mathbf{W}^v - \mathbf{U}_t)^i \right\|_2 + \frac{\alpha}{2}\, tr(\mathbf{U}_t^T \mathbf{L}\, \mathbf{U}_t)) \tag{5.51}$$

Theorem 3. *JRM clustering algorithm decreases the objective function value of Eq. (5.7) until it converges.*

According to Theorem 1,

$$\mathcal{L}(\mathbf{W}_{t+1}^v, \mathbf{U}_t, \mathbf{F}_t, \mathbf{S}_t) \leq \mathcal{L}(\mathbf{W}_t^v, \mathbf{U}_t, \mathbf{F}_t, \mathbf{S}_t) \tag{5.52}$$

According to Theorem 2,

$$\mathcal{L}(\mathbf{W}_{t+1}^v, \mathbf{U}_{t+1}, \mathbf{F}_t, \mathbf{S}_t) \leq \mathcal{L}(\mathbf{W}_{t+1}^v, \mathbf{U}_t, \mathbf{F}_t, \mathbf{S}_t) \tag{5.53}$$

According to Eq. (5.20) in Section 5.4, $\mathbf{F}$ has a closed-form solution, thus we have the following inequality:

$$\mathcal{L}(\mathbf{W}_{t+1}^v, \mathbf{U}_{t+1}, \mathbf{F}_{t+1}, \mathbf{S}_t) \leq \mathcal{L}(\mathbf{W}_{t+1}^v, \mathbf{U}_{t+1}, \mathbf{F}_t, \mathbf{S}_t) \tag{5.54}$$

According to Eq. (5.24) in Section 5.4, $\mathbf{S}$ has a closed-form solution, thus we have the following inequality:

$$\mathcal{L}(\mathbf{W}_{t+1}^{v}, \mathbf{U}_{t+1}, \mathbf{F}_{t+1}, \mathbf{S}_{t+1}) \leq \mathcal{L}(\mathbf{W}_{t+1}^{v}, \mathbf{U}_{t+1}, \mathbf{F}_{t+1}, \mathbf{S}_{t}) \tag{5.55}$$

Sum up inequality Eqs.(5.52-5.55), we get:

$$\mathcal{L}(\mathbf{W}_{t+1}^{v}, \mathbf{U}_{t+1}, \mathbf{F}_{t+1}, \mathbf{S}_{t+1}) \leq \mathcal{L}(\mathbf{W}_{t}^{v}, \mathbf{U}_{t}, \mathbf{F}_{t}, \mathbf{S}_{t}) \tag{5.56}$$

This completes the proof for theorem 3. Empirical results also show that the objective function convergences.

5.6 Experiments

In this section, we evaluate the performance of the proposed JRM algorithm, by comparing it with the state-of-the-art multi-view algorithms and one single-view benchmark clustering algorithm on six real data sets, in terms of two evaluation metrics for clustering algorithm accuracy (ACC) and Purity.

5.6.1 Data Sets

The six data sets used in the experiments are Flowers, Texas, Wisconsin, Cornell, 3Sources, and Washington [149, 150]. The summary of the data sets is provided in Table 5.1.

5.6.2 Comparison Algorithms

We tested the robustness of the proposed multi-view clustering algorithm by comparing it with the *K*-means clustering algorithm, Graph-based system (GBS) [26], Adaptively

weighted Procrustes (AWP) [27], Multi-view low-rank sparse subspace clustering (MLRSSC) [28], and Joint Feature Selection with Dynamic Spectral (FSDS) algorithm.

For the above five algorithms, *K*-means clustering and FSDS algorithm conduct clustering directly on each view of the original data and the concatenated features across all views while the rest clustering algorithms conduct clustering directly on the multi-view data.

Table 5.1 The six multi-view benchmark data sets

Datasets	Samples	Views	Classes	Descriptions
Flowers	1360	4	17	80 Images Views: large scale, pose and light variations
Texas	187	4	5	1703 Words 578 Links Views: content, inbound, outbound, cites
Wisconsin	265	4	5	1703 Words 938 Links Views: content, inbound, outbound, cites
Cornell	195	4	5	1703 Words 569 Links Views: content, inbound, outbound, cites
3Source	294	3	6	948 News Articles Views: BBC, Reuters, and Guardian
Washington	230	4	5	1703 Words 783 Links Views: content, inbound, outbound, cites

5.6.3 Experiment Setup

In the experiments, firstly, we tested the robustness of the proposed multi-view clustering algorithm by comparing it with the four clustering algorithms on real data sets in terms of two widely used evaluation metrics for clustering research. Secondly, we investigated the parameters' sensitivity of the proposed clustering algorithm (i.e. α, r and β in Eq. (5.7)) via varying their values to observe the variations of clustering performance. Thirdly, we demonstrated the convergence of Algorithm 5.1 to solving the proposed objective function Eq. (5.7) via checking the iteration times when Algorithm 5.1 converges.

5.6.4 Experimental Results Analysis

The performances of all algorithms are listed in Tables 5.2-5.3 and Figures 51-5.2, which showed that the proposed clustering algorithm achieved the best overall performance on each of the six data sets in terms of ACC and Purity. More specifically, on the average ACC results of all six data sets, the proposed algorithm increased it by 45.05%, 41.95% , 33.49%, 40.01%, 34.38%, and 39.32% respectively, compared to worst *K*-means clustering result, best *K*-means clustering result, concatenation-based *K*-means clustering result, GBS, AWP, and MLRSSC. Besides, on the average Purity results on all six data sets, the proposed algorithm increased it by 37.55%, 37.24%, 33.58%, 33.36%, 34.40%, and 31.73% compared to worst *K*-means clustering result, best *K*-means clustering result, concatenation-based *K*-means clustering result, GBS, AWP, and MLRSSC. JRM algorithm performed better than FSDS algorithm on multi-

view data sets. FSDS algorithm performed better than *K*-mean clustering algorithm in terms of ACC and purity. The worst FSDS algorithm result, best FSDS algorithm result, and concatenation-based FSDS algorithm result increased the average ACC by 2.23%, 10.73%, and 8.54% respectively, compared to the worst *K*-means clustering, the best *K*-means clustering, and the concatenation-based *K*-means clustering. The worst FSDS, best FSDS, and concatenation-based FSDS increased the average Purity by 10.69%, 31.42%, and 19.83% respectively, compared to the worst *K*-means clustering, the best *K*-means clustering, and the concatenation-based *K*-means clustering. Other observations are listed below.

First, as a centralization-based multi-view approach, the proposed clustering algorithm outperformed both the distribution-based and the concatenation-based *K*-means clustering approach. Especially it increased ACC by 48.34% compared to the best result of *K*-means cluster algorithm on different view of data set Cornell. The proposed clustering algorithm increased ACC by 44.95% compared to the clustering result of *K*-means cluster algorithm on concatenated features from all the views of the data set Texas. The reason is that concatenation approach not only disregards the unique nature of different views, but also easily cause the problem of curse of dimensionality by concatenating features of different view to form an extremely high-dimensional data, so it is hard to achieve good clustering results. Differently, the distribution-based approach takes the partial information across multi-view data into account, however, it cannot outperform our method, because centralization-based multi-view approaches have considered both common information and distinguish information cross views of

multi-view data. This observation supports the idea that it is unable to produce reasonable clustering performance without fully using the information of multi-view data sets.

Second, by simultaneously addressing the major issues of clustering algorithms, our algorithm performed better than multi-stage clustering algorithms. Especially the proposed clustering algorithm increased ACC by 39.00%, 35.22%, and 68.05% compared to GBS, AWP, and MLRSSC algorithms which are multi-stage clustering algorithms on data set Wisconsin. The reason being that addressing these issues in a unified way seeks one global goal leading to optimal clustering results, whereas the multi-stage clustering algorithms with separate goals in each stage achieve sub-optimal results.

Third, our algorithm employs $L_{2,1}$-norm minimization for the loss function achieved better results compared to GBS, AWP, and MLRSSC algorithms which use L_2-norm minimization for their loss functions. E.g., the proposed clustering algorithm increased ACC by 22.96%, 36.52%, and 33.25% compared to L_2-norm-based clustering algorithms GBS, AWP, and MLRSSC on data set Washington. This supports the idea that $L_{2,1}$-norm could reduce the influence of outliers and improved the performance of the clustering. In this way, the clustering results won't be corrupted by the redundant features of the original data, so the clustering accuracy of our proposed method can be improved.

Finally, our algorithm employs $L_{2,1}$-norm minimization for regularization term achieved better results compared to MLRSSC clustering algorithm which use L_2-norm

on its regularization term, e.g., the proposed clustering algorithm increased ACC by 63.63% compared to clustering algorithm MLRSSC whose regularization term is L_2-norm-based on data set Flowers. This supports the idea that $L_{2,1}$-norm could reduce the dimension and select relevant features to improve performance of the clustering.

Table 5.2 ACC results of JRM algorithm on six multi-view data sets

	Flowers	*Texas*	*Wisconsin*	*Cornell*	3Source	*Washington*
Worst *K*-means	0.3301	0.5532	0.4677	0.4141	0.3034	0.1343
Best *K*-means	0.3361	0.5572	0.4874	0.4192	0.3757	0.2133
Con *K*-means	0.4417	0.5452	0.5357	0.4597	0.3420	0.5724
GBS	0.4308	0.4759	0.4226	0.3231	0.4304	0.4226
AWP	0.7995	0.5508	0.4604	0.4256	0.3197	0.2870
MLRSSC	0.1765	0.7380	0.1321	0.7385	0.4422	0.3197
Worst FSDS	0.4876	0.3830	0.4038	0.4256	0.2585	0.3783
Best FSDS	0.5897	0.4920	0.5434	0.4256	0.4388	0.5435
Con FSDS	0.6853	0.5492	0.5774	0.6410	0.4388	0.5174
JRM	**0.8128**	**0.9947**	**0.8126**	**0.9026**	**0.7313**	**0.6522**

Table 5.3 Purity results of JRM algorithm on six multi-view data sets

	Flowers	*Texas*	*Wisconsin*	*Cornell*	3Source	*Washington*
Worst *K*-means	0.3551	0.5751	0.4926	0.4433	0.3298	0.6361
Best *K*-means	0.3584	0.5807	0.5074	0.4477	0.4027	0.5539
Con *K*-means	0.4653	0.6086	0.5598	0.4844	0.3517	0.6007
GBS	0.4906	0.5775	0.4906	0.5641	0.4739	0.4868
AWP	0.7995	0.5508	0.4604	0.4256	0.3197	0.4652
MLRSSC	0.3846	0.5936	0.5283	0.5231	0.4558	0.6957
Worst FSDS	0.4743	0.6684	0.7434	0.6462	0.3197	0.6217
Best FSDS	0.5912	0.6684	0.8067	0.9649	0.7789	0.9261
Con FSDS	0.6949	0.6738	0.8049	0.6564	0.7262	0.7043
JRM	**0.8082**	**0.7099**	**0.8075**	**0.9695**	**0.83418**	**0.95615**

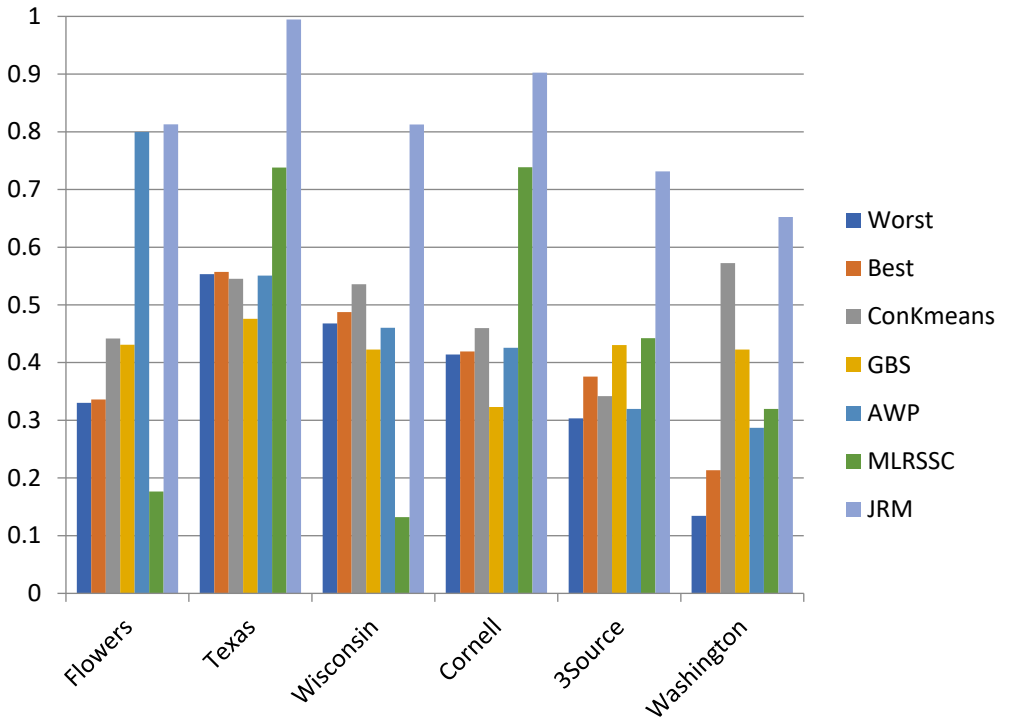


Figure 5.1 ACC results of JRM algorithm on six real data sets

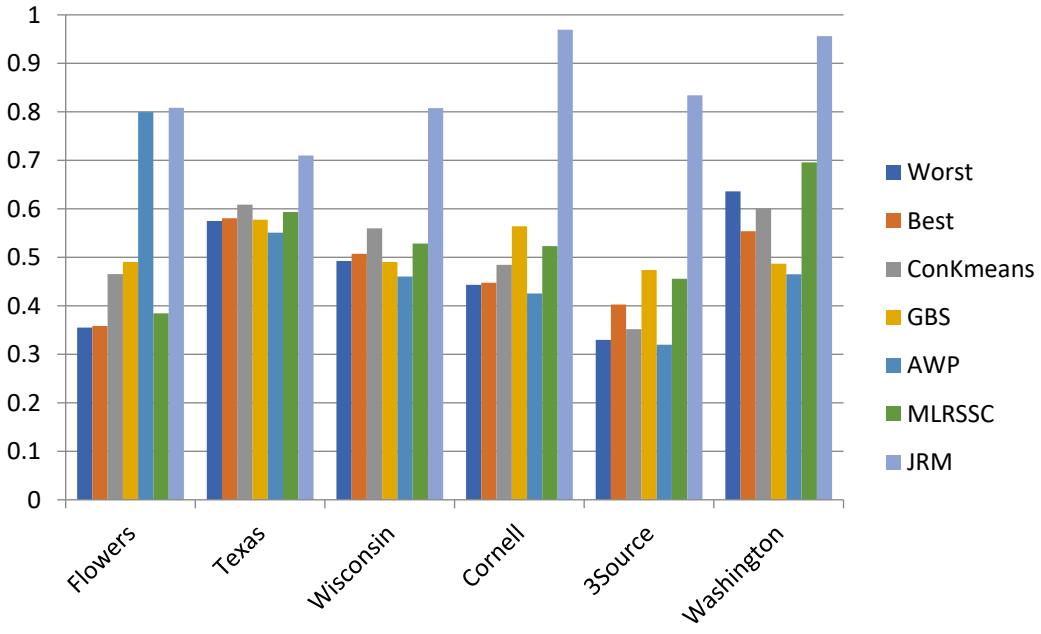


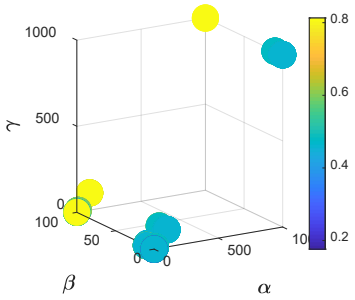
Figure 5.2 Purity results of JRM algorithm on four real data sets

5.6.5 Parameters' Sensitivity

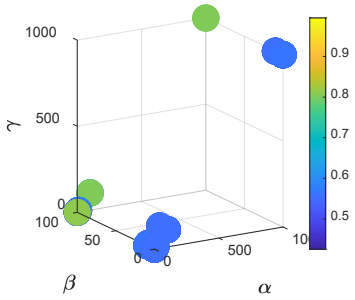
To investigate the parameters' sensitivity of our algorithm, we varied the parameters α, γ and β of our objective function from 0 to 1000 and recorded the clustering results in terms of ACC and Purity for the six data sets in Figures 5.3-5.4.

First, different data sets needed different ranges of parameters to achieve the best performance. For example, our algorithm achieved the best ACC (99.47%) on data set Texas when parameters $\alpha = 0.1$, $\gamma = 0.1$ and $\beta = 10$. For the data set Flowers, our algorithm achieved the best ACC (81.28%) when $\alpha = 0.001$, $\gamma = 0.001$ and $\beta = 100$. For the data set Cornell, our algorithm achieved the best ACC (90.26%) when $\alpha = 1000$, $\gamma = 1000$ and $\beta = 0.01$. For the data set Wisconsin, our algorithm achieved the best ACC (81.28%) when $\alpha = 0.001$, $\gamma = 0.001$ and $\beta = 100$. For the data set Washington, our algorithm achieved the best ACC (65.22%) when $\alpha = 10$, $\gamma = 2$ and $\beta = 100$. Thus the proposed clustering algorithm is data-driven.

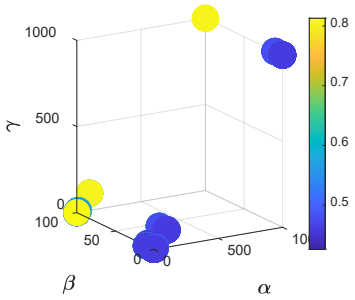
Since the algorithm is sensitive to the parameters, the performance depends on parameter combinations. The parameter γ tunes the sparsity of the transfer matrix $\mathbf{W}^v$**.** Different γ produces different level of sparsity of $\mathbf{W}^v$, i.e., different percentage of redundant features are removed from the original data set. The parameter α and β are used to tradeoff the importance of **F** and **S**. Finally, from Figures. 5.3-5.4 we can perceive that parameter α and γ are more sensitive than β on the six benchmark multi-view data sets.



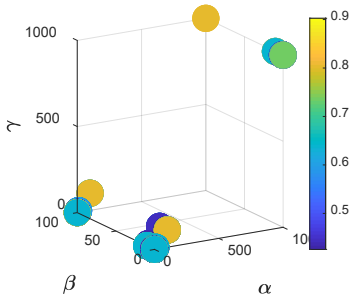
(a) Flowers



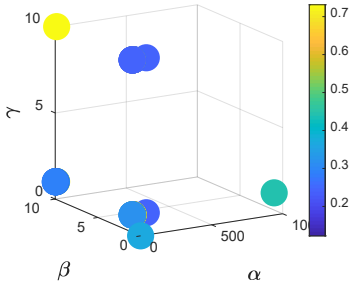
(b) Texas



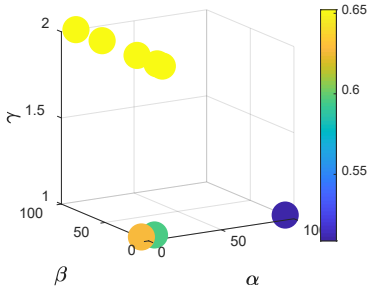
(c) Wisconsin



(d) Cornell



(e) 3Source



(f) Washington

Figure 5.3 ACC results of JRM algorithm with respect to different parameter settings

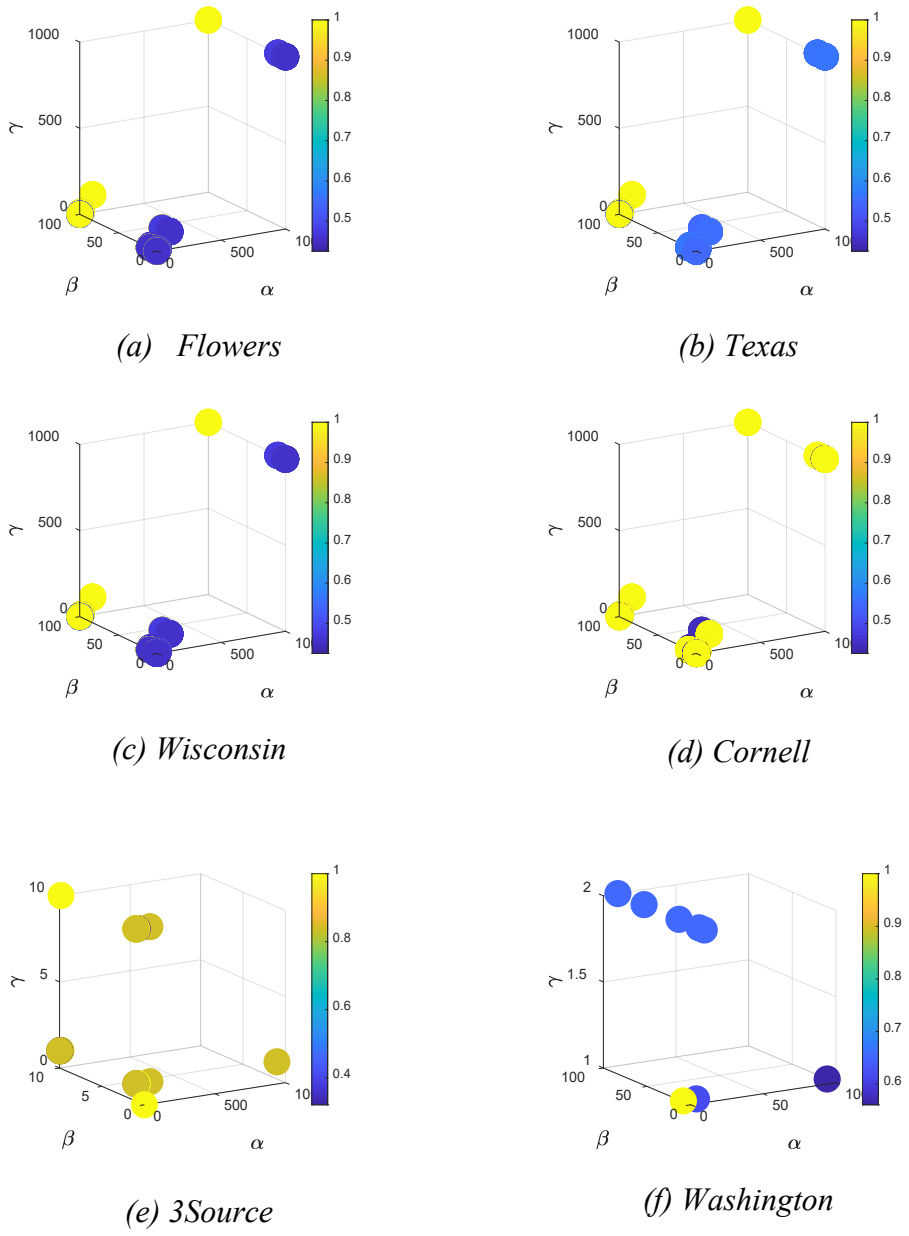


Figure 5.4 Purity results of JRM algorithm with respect to different parameter settings

5.6.6 Convergence

Figure. 5.5 shows the trend of objective values generated with respect to iterations. We set the stopping criteria of the proposed clustering algorithm to

$\left|obj_{(t+1)} - obj_{(t)}\right| / obj_{(t)} \leq 10^{-9}$, where $obj_{(t)}$ represents the objection function value of Eq. (5.7) in the *t-th* iteration.

From Figure 5.5, we see that our algorithm monotonically decreased the value of objective function until it converged when we optimized the proposed objective function in Eq. (5.7). Our algorithm converged to the optimal value within 100 iterations on all the data sets used. This shows that Algorithm 5.1 can make problem Eq. (5.7) converge.

5.7 Conclusion

In this chapter we have proposed a new Joint Robust Multi-view (JRM) spectral clustering algorithm which aims to solving initialization, cluster number determination, similarity measure, feature selection, and outlier reduction issues for multi-view data in a unified way.

As a centralization-based multi-view algorithm, JRM considers information from all views of the multi-view data set to conduct clustering. The optimal performance could be reached when the separated stages are combined in a unified way. The $L_{2,1}$-norm is applied to both loss function and regularization term to reduce the influence of outliers and select relevant features. Experiments have been performed on six real-world benchmark data sets and JRM outperforms the comparison clustering algorithms in terms of two evaluation metrics for clustering algorithm including accuracy (ACC) and Purity.

In the future, we plan to extend our JRM algorithm to handle incomplete data.

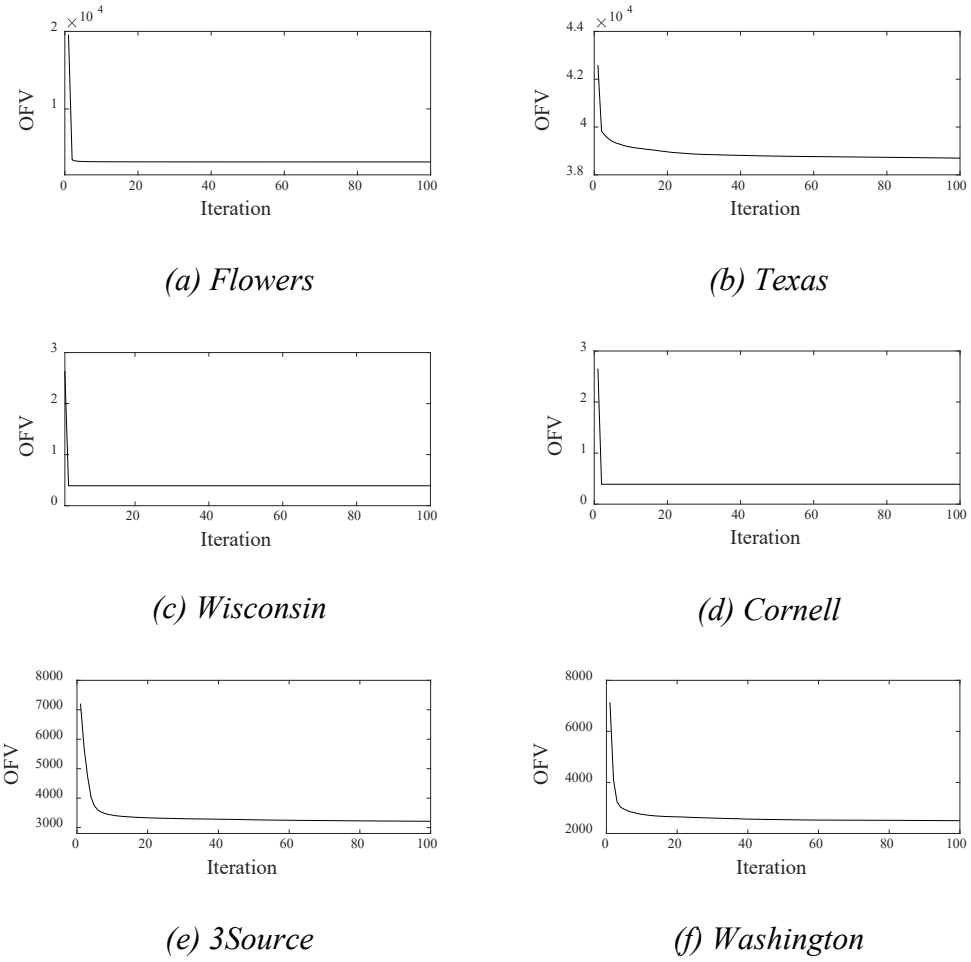


Figure 5.5 Objective function values (OFVs) versus iterations for JRM algorithm

Chapter 6

Conclusion and Future Work

6.1 Conclusion

K-means clustering algorithm is one of the most widely used unsupervised machine learning techniques. This thesis focused on the problems related to *K*-means clustering: initialization, the cluster number determination, the similarity measure, feature selection, outlier reduction, and multi-view clustering.

First, Chapter 3 solved the issues of initialization and similarity measure of *K*-means clustering algorithm in a unified way. We fixed the initialization of the *K*-means clustering algorithm using sum-of-norms, which also outputs the new representation of the original samples. Concurrently, we fixed the similarity measure of *K*-means clustering algorithm by learning the similarity matrix based on the data distribution. Furthermore, the derived new representation is used to conduct *K*-means clustering. The proposed IS clustering algorithm outperformed both the classical clustering algorithms K-means clustering algorithm and well-known Spectral clustering algorithm.

Second, Chapter 4 solved the issues of cluster number determination, similarity measure, and the robustness of clustering by selecting useful features and reducing the influence of outliers in a unified way. Specifically, the similarity matrix was learnt based on the data distribution while the cluster number was automatically generated by the ranked constraint on the Laplacian matrix of the learned similarity matrix.

Furthermore, to select the useful features and reduce the influence of outliers, we employed the $L_{2,1}$-norm as the sparse constraints on the regularization term and the loss function. The proposed FSDS clustering algorithm outperformed the classical clustering algorithms *K*-means clustering algorithm, well-known Spectral clustering algorithm, Clustering and projected clustering with adaptive neighbors algorithm (CAN) [24] and Robust continuous clustering algorithm (RCC) [4].

Third, Chapter 5 considered information from all views of the multi-view data set to conduct clustering while solving the issues of the initialization, the cluster number determination, the similarity measure, feature selection, and outlier reduction in a unified way. Instead of concatenating the features across all views of the multi-view data set or treating each view independently, we considered information from all views of the multi-view data set to conduct clustering. The proposed JRM clustering algorithm outperformed the classical clustering algorithms K-means clustering algorithm, Graph-Based system (GBS) [26], Adaptively weighted Procrustes (AWP) [27], and Multi-view low-rank sparse subspace clustering (MLRSSC) [28] using real datasets in terms of two widely used evaluation metrics for clustering research.

Finally, we evaluated the proposed algorithms by comparing them with the state-of-the-art clustering algorithms on real data sets. The proposed clustering algorithm outperformed the comparison clustering algorithms in terms of evaluation metrics for clustering algorithms including ACC and Purity. Moreover, we theoretically proved the convergences of the proposed optimization methods for the objective functions of the proposed algorithms.

6.2 Future Directions

This research conducted an extensive study on *K*-means clustering literature to find the limitations of the current *K*-means clustering researches. We solved the key limitations of the K-means clustering algorithm. However, there are still spaces to improve the proposed algorithms in this thesis.

- It is not uncommon that real data contains missing values for some features. A data set with some missing feature values is referred to an incomplete data set. Many current clustering algorithms including *K*-means clustering algorithm cannot efficiently perform with incomplete data. The imputation approach replaces the missing values with the estimations of these values. The missing values could be imputed as the degree of difference [151] and the degree of belongingness [152]. The imputation approach could be applied to develop a probabilistic fuzzy clustering algorithm for incomplete data for future research. Hence, conducting clustering analysis on the incomplete data sets is also one of our future works.
- Imbalanced data exists in many real-world applications. When the data is imbalanced, the number of data points in minority class is much smaller than the number of data points in majority class. Due to the strong influence of the majority classes, traditional clustering algorithms including *K*-means clustering algorithm may not achieve good results especially for minority classes [153]. Attempting to imitate the human neural networks in the brain, deep learning uses multiple layers of machine learning algorithms to process data [154]. In the

future, we would like to focus on conducting clustering analysis on imbalanced data sets.

References

1. Ford, M., *Architects of Intelligence: The truth about AI from the people building it*. 2018: Packt Publishing Ltd.
2. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning.* nature, 2015. **521**(7553): p. 436.
3. Zhou, X., et al., *Graph Convolutional Network Hashing.* IEEE transactions on cybernetics, 2018: p. 1-13.
4. Shah, S.A. and V. Koltun, *Robust continuous clustering.* Proceedings of the National Academy of Sciences, 2017. **114**(37): p. 9814-9819.
5. Song, J., et al., *From deterministic to generative: Multimodal stochastic RNNs for video captioning.* IEEE transactions on neural networks and learning systems, 2018(99): p. 1-12.
6. Bin, Y., et al., *Describing video with attention-based bidirectional LSTM.* IEEE transactions on cybernetics, 2018(99): p. 1-11.
7. Parlett, C., *Exploring Age-Related Metamemory Differences Using Modified Brier Scores and Hierarchical Clustering.* Open Psychology, 2019. **1**(1): p. 215-238.
8. Khalili-Damghani, K., F. Abdi, and S. Abolmakarem, *Solving customer insurance coverage recommendation problem using a two-stage clustering-classification model.* International Journal of Management Science and Engineering Management, 2019. **14**(1): p. 9-19.
9. Sato, Y., et al., *Data mining based on clustering and association rule analysis for knowledge discovery in multiobjective topology optimization.* Expert Systems with Applications, 2019. **119**: p. 247-261.
10. Saxena, A., et al., *A review of clustering techniques and developments.* Neurocomputing, 2017. **267**: p. 664-681.
11. Zhu, X., et al., *One-step multi-view spectral clustering.* IEEE Transactions on Knowledge and Data Engineering, 2018. **31**(10): p.2022-2034.
12. Hartigan, J.A. and M.A. Wong, *Algorithm AS 136: A k-means clustering algorithm.* Journal of the Royal Statistical Society. Series C (Applied Statistics), 1979. **28**(1): p. 100-108.
13. Duan, Y., Q. Liu, and S. Xia. *An improved initialization center k-means clustering algorithm based on distance and density*. in *AIP,* https://doi.org/10.1063/1.5033710
14. Lakshmi, M.A., G.V. Daniel, and D.S. Rao, *Initial Centroids for K-Means Using Nearest Neighbors and Feature Means*, in *Soft Computing and Signal Processing*. 2019, Springer. p. 27-34.
15. Motwani, M., N. Arora, and A. Gupta, *A Study on Initial Centroids Selection for Partitional Clustering Algorithms*, in *Software Engineering*. 2019, Springer. p. 211-220.
16. Femi, P.S. and S.G. Vaidyanathan. *Comparative Study of Outlier Detection Approaches*. in *ICIRCA*. 2018. IEEE. p. 366-371.
17. Buczkowska, S., N. Coulombel, and M. de Lapparent, *A comparison of euclidean distance, travel times, and network distances in location choice mixture models.* Networks and spatial economics, 2019: p. 1-34.
18. Doad, P.K. and M.B. Mahip, *Survey on Clustering Algorithm & Diagnosing Unsupervised Anomalies for Network Security.* International Journal of Current Engineering and Technology ISSN, 2013: p. 2277-410.
19. Yan, Q., et al., *A discriminated similarity matrix construction based on sparse subspace clustering algorithm for hyperspectral imagery.* Cognitive Systems Research, 2019. **53**: p. 98-110.
20. Bian, Z., H. Ishibuchi, and S. Wang, *Joint Learning of Spectral Clustering Structure and Fuzzy Similarity Matrix of Data.* IEEE Transactions on Fuzzy Systems, 2019. **27**(1): p. 31-44.

21. Satsiou, A., S. Vrochidis, and I. Kompatsiaris. *A Hybrid Recommendation System Based on Density-Based Clustering*. in *INSCI*, 2017. p. 49-57
22. Radhakrishna, V., et al., *A novel fuzzy similarity measure and prevalence estimation approach for similarity profiled temporal association pattern mining.* Future generation computer systems, 2018. **83**: p. 582-595.
23. Rong, H., et al., *A novel subgraph K^+-isomorphism method in social network based on graph similarity detection.* Soft Computing, 2018. **22**(8): p. 2583-2601.
24. Nie, F., X. Wang, and H. Huang. *Clustering and projected clustering with adaptive neighbors*. in *SIGKDD*, 2014. p. 977-986.
25. Wang, X.-D., et al., *Fast adaptive K-means subspace clustering for high-dimensional data.* IEEE Access, 2019. **7**: p. 42639-42651.
26. Wang, H., et al., *A study of graph-based system for multi-view clustering.* Knowledge-Based Systems, 2019. **163**: p. 1009-1019.
27. Nie, F., L. Tian, and X. Li. *Multiview clustering via adaptively weighted procrustes*. in *SIGKDD*, 2018, ACM. p. 2022-2030.
28. Brbić, M. and I. Kopriva, *Multi-view low-rank sparse subspace clustering.* Pattern Recognition, 2018. **73**: p. 247-258.
29. Xu, D. and Y. Tian, *A comprehensive survey of clustering algorithms.* Annals of Data Science, 2015. **2**(2): p. 165-193.
30. Singh, A., A. Yadav, and A. Rana, *K-means with Three different Distance Metrics.* International Journal of Computer Applications, 2013. **67**(10): p. 13-17.
31. Zhu, X., et al., *Graph PCA hashing for similarity search.* IEEE Transactions on Multimedia, 2017. **19**(9): p. 2033-2044.
32. Saraswathi, S. and M.I. Sheela, *A comparative study of various clustering algorithms in data mining.* International Journal of Computer Science and Mobile Computing, 2014. **11**(11): p. 422-428.
33. Rezaei, M., *Improving a Centroid-Based Clustering by Using Suitable Centroids from Another Clustering.* Journal of Classification, 2019: p. 1-14.
34. Lu, Y., et al., *A Tabu Search based clustering algorithm and its parallel implementation on Spark.* Applied Soft Computing, 2018. **63**: p. 97-109.
35. Ieva, C., et al., *Discovering Program Topoi via Hierarchical Agglomerative Clustering.* IEEE Transactions on Reliability, 2018. **67**(3): p. 758-770.
36. Tie, J., et al., *The application of agglomerative hierarchical spatial clustering algorithm in tea blending.* Cluster Computing, 2018: p. 1-10.
37. Gao, X. and S. Wu. *Hierarchical Clustering Algorithm for Binary Data Based on Cosine Similarity*. in *LISS*, 2018. p. 1-6.
38. Cheng, D., et al., *A hierarchical clustering algorithm based on noise removal.* International Journal of Machine Learning and Cybernetics, 2019. **10**(7): p. 1591-1602.
39. Sisodia, D., et al., *Clustering techniques: a brief survey of different clustering algorithms.* International Journal of Latest Trends in Engineering and Technology (IJLTET), 2012. **1**(3): p. 82-87.
40. Franco, M. and J.-M. Vivo, *Cluster Analysis of Microarray Data*, in *Microarray Bioinformatics*. 2019, Springer. p. 153-183.
41. Karypis, G., E.-H.S. Han, and V. Kumar, *Chameleon: Hierarchical clustering using dynamic modeling.* Computer, 1999(8): p. 68-75.
42. Guha, S., R. Rastogi, and K. Shim, *ROCK: A robust clustering algorithm for categorical attributes.* Information systems, 2000. **25**(5): p. 345-366.
43. Guha, S., R. Rastogi, and K. Shim, *Cure: an efficient clustering algorithm for large databases.* Information systems, 2001. **26**(1): p. 35-58.
44. Majumdar, J., S. Udandakar, and B.M. Bai, *Implementation of Cure Clustering Algorithm for Video Summarization and Healthcare Applications in Big Data*, in *Emerging Research in Computing, Information, Communication and Applications*. 2019, Springer. p. 553-564.
45. Guha, S., R. Rastogi, and K. Shim. *CURE: an efficient clustering algorithm for large databases*. in *ACM Sigmod Record*, 1998, 27(2): p.73-84.

46. Wang, L., et al., *Incremental Local Distribution-Based Clustering Using Bayesian Adaptive Resonance Theory.* IEEE transactions on neural networks and learning systems, 2019. **30**(11): p. 3496-3504.
47. Rasmussen, C.E. *The infinite Gaussian mixture model*. in *Advances in neural information processing systems*. 2000.
48. Zhang, T., R. Ramakrishnan, and M. Livny. *BIRCH: an efficient data clustering method for very large databases*. in *ACM Sigmod Record*, 1996, 25(2): p.103-114.
49. Ciccolella, S., et al., *Effective clustering for single cell sequencing cancer data. in ACM-BCB,* 2019. p. 437-446.
50. Viroli, C. and G.J. McLachlan, *Deep gaussian mixture models.* Statistics and Computing, 2019. **29**(1): p. 43-51.
51. Ghosh, P. and K. Mali, *Image Segmentation by Grouping Pixels in Color and Image Space Simultaneously using DBSCAN Clustering Algorithm.* Journal of Remote Sensing & GIS, 2019. **4**(3): p. 52-60.
52. Xu, S., et al., *DBSCAN Clustering Algorithm for Detection of Nearby Open Clusters Based on Gaia-DR2.* Acta Astronomica Sinica, 2018. **59**.
53. Brown, D., A. Japa, and Y. Shi. *An Attempt at Improving Density-based Clustering Algorithms*. in *ACM SE*, 2019, p. 172-175.
54. Ma, J., X. Jiang, and M. Gong, *Two-phase clustering algorithm with density exploring distance measure.* CAAI Transactions on Intelligence Technology, 2018. **3**(1): p. 59-64.
55. Yan, J., et al., *Applying Machine Learning Algorithms to Segment High-Cost Patient Populations.* Journal of general internal medicine, 2019. **34**(2): p. 211-217.
56. Dhal, K.G., et al., *A survey on nature-inspired optimization algorithms and their application in image enhancement domain.* Archives of Computational Methods in Engineering, 2019. **26**(5): p. 1607-1638.
57. Lakshmi, K., N.K. Visalakshi, and S. Shanthi, *Data clustering using K-Means based on Crow Search Algorithm.* Sādhanā, 2018. **43**(11): p. 190-202.
58. Kowalski, P.A., et al., *Nature Inspired Clustering–Use Cases of Krill Herd Algorithm and Flower Pollination Algorithm*, in *Interactions Between Computational Intelligence and Mathematics Part 2*. 2019, Springer. p. 83-98.
59. Wang, R., et al., *Optimising discrete dynamic berth allocations in seaports using a Levy Flight based meta-heuristic.* Swarm and evolutionary computation, 2019. **44**: p. 1003-1017.
60. ODILI, J.B., et al., *A Critical Review of Major Nature-Inspired Optimization Algorithms.* The Eurasia Proceedings of Science, Technology, Engineering & Mathematics, 2018. **2**: p. 376-394.
61. Zhu, X., et al., *Low-rank sparse subspace for spectral clustering.* IEEE Transactions on Knowledge and Data Engineering, 2019. **31**(8): p. 1532-1543.
62. Wu, S., X. Feng, and W. Zhou, *Spectral clustering of high-dimensional data exploiting sparse representation vectors.* Neurocomputing, 2014. **135**: p. 229-239.
63. Cherng, J.-S. and M.-J. Lo. *A hypergraph based clustering algorithm for spatial data sets*. in *ICDM*, 2001, p. 83-90.
64. Estivill-Castro, V. and I. Lee. *Amoeba: Hierarchical clustering based on spatial proximity using delaunay diagram*. in *ISSDH,* 2000, p. 1-16.
65. Liu, G., et al., *Robust recovery of subspace structures by low-rank representation.* IEEE transactions on pattern analysis and machine intelligence, 2013. **35**(1): p. 171-184.
66. Kang, Z., et al., *Low-rank kernel learning for graph-based clustering.* Knowledge-Based Systems, 2019. **163**: p. 510-517.
67. Huang, J., et al., *A Novel Hybrid Clustering Algorithm Based on Minimum Spanning Tree of Natural Core Points.* IEEE Access, 2019. **7**: p. 43707-43720.
68. Min, E., et al., *A survey of clustering with deep learning: From the perspective of network architecture.* IEEE Access, 2018. **6**: p. 39501-39514.
69. Zheng, Q., et al., *Feature Concatenation Multi-view Subspace Clustering.* 2019, arXiv preprint arXiv:1901.10657.

70. Li, J., et al., *Feature selection: A data perspective.* ACM Computing Surveys (CSUR), 2018. **50**(6): p. 94.
71. Kumar, A., P. Rai, and H. Daume. *Co-regularized multi-view spectral clustering*. in *NIPS,*2011. p. 1413-1421.
72. Liu, M., et al. *Low-rank multi-view learning in matrix completion for multi-label image classification*. in *AAAI,* 2015, p. 2778-2784.
73. Liu, X., et al., *Multiple kernel k-means with incomplete kernels.* IEEE transactions on pattern analysis and machine intelligence, 2019. DOI: 10.1109/TPAMI.2019.2892416
74. Cano, A., *An ensemble approach to multi-view multi-instance learning.* Knowledge-Based Systems, 2017. **136**: p. 46-57.
75. Sun, S., *A survey of multi-view machine learning.* Neural computing and applications, 2013. **23**(7-8): p. 2031-2038.
76. Yin, Q., et al., *Multi-view clustering via pairwise sparse subspace representation.* Neurocomputing, 2015. **156**: p. 12-21.
77. Wang, S., et al. *A Novel Weighted Hybrid Multi-View Fusion Algorithm for Semi-Supervised Classification*. in *ISCAS,* 2019, p. 1-5.
78. Roweis, S.T. and L.K. Saul, *Nonlinear dimensionality reduction by locally linear embedding.* science, 2000. **290**(5500): p. 2323-2326.
79. Sekeh, S.Y. and A.O. Hero. *Feature Selection for Mutlti-labeled Variables via Dependency Maximization*. in *ICASSP,* 2019, pp. 3127-3131.
80. Ryu, U., et al., *Construction of traffic state vector using mutual information for short-term traffic flow prediction.* Transportation Research Part C: Emerging Technologies, 2018. **96**: p. 55-71.
81. Solorio-Fernández, S., J.F. Martínez-Trinidad, and J.A. Carrasco-Ochoa, *A new unsupervised spectral feature selection method for mixed data: a filter approach.* Pattern Recognition, 2017. **72**: p. 314-326.
82. Solorio-Fernández, S., J.A. Carrasco-Ochoa, and J.F. Martínez-Trinidad, *A review of unsupervised feature selection methods.* Artificial Intelligence Review, 2019: p. 1-42.
83. Das, S. *Filters, wrappers and a boosting-based hybrid for feature selection*. in *ICML*, 2001. p. 74-81.
84. Breaban, M. and H. Luchian, *A unifying criterion for unsupervised clustering and feature selection.* Pattern Recognition, 2011. **44**(4): p. 854-865.
85. Hoseininejad, F.S., Y. Forghani, and O. Ehsani, *A fast algorithm for local feature selection in data classification.* Expert Systems, 2019. **36**(3): p. e12391.
86. Agbehadji, I.E., et al., *Integration of Kestrel-based search algorithm with artificial neural network for feature subset selection.* International Journal of Bio-Inspired Computation, 2019. **13**(4): p. 222-233.
87. Lee, S.-J., et al., *A novel bagging C4. 5 algorithm based on wrapper feature selection for supporting wise clinical decision making.* Journal of biomedical informatics, 2018. **78**: p. 144-155.
88. Rao, H., et al., *Feature selection based on artificial bee colony and gradient boosting decision tree.* Applied Soft Computing, 2019. **74**: p. 634-642.
89. Jing, R., et al., *CART-based fast CU size decision and mode decision algorithm for 3D-HEVC.* Signal, Image and Video Processing, 2019. **13**(2): p. 209-216.
90. Li, R., et al., *L2,1-Norm Based Loss Function and Regularization Extreme Learning Machine.* IEEE Access, 2018. **7**: p. 6575-6586.
91. Zhu, X., et al., *A novel relational regularization feature selection method for joint regression and classification in AD diagnosis.* Medical image analysis, 2017. **38**: p. 205-214.
92. Mo, D. and Z. Lai, *Robust Jointly Sparse Regression with Generalized Orthogonal Learning for Image Feature Selection.* Pattern Recognition, 2019. **93**: p. 164-178.
93. Lopez-Martinez, D., *Regularization approaches for support vector machines with applications to biomedical data.* 2017, arXiv preprint arXiv:1710.10600.
94. Zhao, M., et al., *Trace Ratio Criterion based Discriminative Feature Selection via l2, p-norm regularization for supervised learning.* Neurocomputing, 2018. **321**: p. 1-16.

95. Zhang, Z., et al., *Robust neighborhood preserving projection by nuclear/L2, 1-norm regularization for image feature extraction.* IEEE Transactions on Image Processing, 2017. **26**(4): p. 1607-1622.
96. Knox, E.M. and R.T. Ng. *Algorithms for mining distancebased outliers in large datasets*. in *VLDB*, 1998, pp. 392-403.
97. Suri, N.N.R.R., M.N. Murty, and G. Athithan, *Research Issues in Outlier Detection*, in *Outlier Detection: Techniques and Applications*, J. Kacprzyk and L.C. Jain, Editors. 2019, Springer. p. 29-51.
98. Liu, H., et al., *Clustering with Outlier Removal.* 2018, arXiv preprint arXiv:1801.01899.
99. Wahid, A. and A.C.S. Rao, *A distance-based outlier detection using particle swarm optimization technique*, in *Information and Communication Technology for Competitive Strategies*. 2019, Springer. p. 633-643.
100. Trittenbach, H. and K. Böhm, *Dimension-based subspace search for outlier detection.* International Journal of Data Science and Analytics, 2019. **7**(2): p. 87-101.
101. Liu, C., D. Zhang, and J. Qi. *Outlier Detection Based on Cluster Outlier Factor and Mutual Density*. in *ISICA,* 2018, *Vol 986,* p. 319-329.
102. Kawanobe, S. and T. Ozaki. *Experimental study of characterizing frequent itemsets using representation learning*. in *WAINA,*2018.p. 170-174.
103. Yuan, G., S. Cai, and S. Hao. *A Novel Weighted Frequent Pattern-Based Outlier Detection Method Applied to Data Stream*. in *ICCCBDA*, 2019, p. 503-510.
104. Mahajan, M., S. Kumar, and B. Pant, *A Novel Cluster Based Algorithm for Outlier Detection*, in *Computing, Communication and Signal Processing*. 2019, Springer. p. 449-456.
105. Al-Obaidi, S.A.R., et al., *Robust Metric Learning based on the Rescaled Hinge Loss.* 2019, arXiv preprint arXiv:1904.11711.
106. Ren, Z., et al., *Simultaneous learning of reduced prototypes and local metric for image set classification.* Expert Systems with Applications, 2019. **134**: p. 102-111.
107. CHEN, M., et al., *Capped l1-Norm Sparse Representation Method for Graph Clustering.* IEEE Access.2019, **7**: p. 54464-54471.
108. Du, L., et al. *Robust multiple kernel k-means using l21-norm*. in *IJCAI*, 2015, p. 3476-3482.
109. Yang, C., et al., *Joint correntropy metric weighting and block diagonal regularizer for robust multiple kernel subspace clustering.* Information Sciences, 2019. **500**: p. 48-66.
110. You, C.-Z., V. Palade, and X.-J. Wu, *Robust structure low-rank representation in latent space.* Engineering Applications of Artificial Intelligence, 2019. **77**: p. 117-124.
111. Mojarad, M., et al., *A fuzzy clustering ensemble based on cluster clustering and iterative Fusion of base clusters.* Applied Intelligence, 2019. **49**(7): p. 2567-2581.
112. Sohn, S.Y. and S.H. Lee, *Data fusion, ensemble and clustering to improve the classification accuracy for the severity of road traffic accidents in Korea.* Safety Science, 2003. **41**(1): p. 1-14.
113. Yu, Y.-F., et al., *Joint Transformation Learning via the L2, 1-Norm Metric for Robust Graph Matching.* IEEE transactions on cybernetics, 2019: p. 1-13.
114. Argyriou, A., T. Evgeniou, and M. Pontil. *Multi-task feature learning*. in *Advances in neural information processing systems*. in *NIPS*, 2007, p. 41-48.
115. Liu, J., S. Ji, and J. Ye. *Multi-task feature learning via efficient l 2, 1-norm minimization*. 2012, *arXiv preprint arXiv:1205.2631*.
116. Jiang, B. and C. Ding, *Outlier Regularization for Vector Data and L21 Norm Robustness.* 2017,arXiv preprint arXiv:1706.06409.
117. Nie, F., et al. *Efficient and robust feature selection via joint ℓ2, 1-norms minimization*. in *NIPS*, 2010, p. 1813-1821.
118. Nie, F., et al. *Unsupervised and semi-supervised learning via ℓ 1-norm graph*. in in *ICCV,* 2011, p. 2268-2273.
119. Wang, C., et al., *Multiple Kernel Clustering With Global and Local Structure Alignment.* IEEE Access, 2018. **6**: p. 77911-77920.
120. Domeniconi, C. and M. Al-Razgan, *Weighted cluster ensembles: Methods and analysis.* ACM Transactions on Knowledge Discovery from Data (TKDD), 2009. **2**(4): p. 17.

121. Deelers, S. and S. Auwatanamongkol, *Enhancing K-means algorithm with initial cluster centers derived from data partitioning along the data axis with the highest variance.* International Journal of Computer Science, 2007. **2**(4): p. 247-252.
122. Likas, A., N. Vlassis, and J.J. Verbeek, *The global k-means clustering algorithm.* Pattern recognition, 2003. **36**(2): p. 451-461.
123. Janani, R. and S. Vijayarani, *Text document clustering using spectral clustering algorithm with particle swarm optimization.* Expert Systems with Applications, 2019. **134**: p. 192-200.
124. Suryanarayana, S., G.V. Rao, and G.V. Swamy, *A Survey: Spectral Clustering Applications and its Enhancements.* International Journal of Computer Science and Information Technologies, 2015. **6**(1), p. 185-189.
125. Lindsten, F., H. Ohlsson, and L. Ljung. *Clustering using sum-of-norms regularization: With application to particle filter output computation*. in *SSP,* 2011, p. 201-204.
126. Kuncheva, L.I. and D.P. Vetrov, *Evaluation of stability of k-means cluster ensembles with respect to random initialization.* IEEE transactions on pattern analysis and machine intelligence, 2006. **28**(11): p. 1798-1808.
127. Wang, J., et al., *Fast Approximate K-Means via Cluster Closures*, in *Multimedia data mining and analytics*. 2015, Springer. p. 373-395.
128. Zahra, S., et al., *Novel centroid selection approaches for KMeans-clustering based recommender systems.* Information sciences, 2015. **320**: p. 156-189.
129. Pavan, K.K., A.D. Rao, and G. Sridhar, *Single pass seed selection algorithm for k-means.* Journal of Computer Science, 2010. **6**(1): p. 60-66.
130. Fränti, P., *Efficiency of random swap clustering.* Journal of Big Data, 2018. **5**(1): p. 13.
131. Barron, J.T., *A more general robust loss function.* arXiv preprint arXiv:1701.03077, 2017.
132. Zheng, W., et al., *Unsupervised feature selection by self-paced learning regularization.* Pattern Recognition Letters, 2018: p. 438-446
133. Geman, S. and D.E. McClure, *Statistical Methods for Tomographic Image Reconstruction.* Bulletin of the International statistical Institute, 1987. **52**(4): p. 5-21.
134. Black, M.J. and A. Rangarajan, *On the unification of line processes, outlier rejection, and robust statistics with applications in early vision.* International Journal of Computer Vision, 1996. **19**(1): p. 57-91.
135. Zheng, W., et al., *Dynamic graph learning for spectral feature selection.* Multimedia Tools and Applications, 2018. **77**(22): p. 29739-29755.
136. Lei, C. and X. Zhu, *Unsupervised feature selection via local structure learning and sparse learning.* Multimedia Tools and Applications, 2018. **77**(22): p. 29605-29622.
137. Voloshinov, V.V., *A generalization of the Karush–Kuhn–Tucker theorem for approximate solutions of mathematical programming problems based on quadratic approximation.* Computational Mathematics and Mathematical Physics, 2018. **58**(3): p. 364-377.
138. Dua, D. and C. Graff, *UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences,* 2019.
139. Das, A. and P. Panigrahi, *Normalized Laplacian spectrum of some subdivision-joins and R-joins of two regular graphs.* AKCE International Journal of Graphs and Combinatorics, 2018. **15**(3): p. 261-270.
140. Park, S. and H. Zhao, *Spectral clustering based on learning similarity matrix.* Bioinformatics, 2018. **34**(12): p. 2069-2076.
141. Cheung, Y.-M., *k∗-Means: A new generalized k-means clustering algorithm.* Pattern Recognition Letters, 2003. **24**(15): p. 2883-2893.
142. Bholowalia, P. and A. Kumar, *EBK-means: A clustering technique based on elbow method and k-means in WSN.* International Journal of Computer Applications, 2014. **105**(9).
143. Nikolova, M. and R.H. Chan, *The equivalence of half-quadratic minimization and the gradient linearization iteration.* IEEE Transactions on Image Processing, 2007. **16**(6): p. 1623-1627.
144. Nie, F., W. Zhu, and X. Li. *Unsupervised feature selection with structured graph optimization*. in *AAAI,*2016. p.1302-1308.

145. Yu, H., et al., *An active three-way clustering method via low-rank matrices for multi-view data.* Information Sciences, 2019. **507**: p. 823-839.
146. Wang, N., et al., *Structured sparse multi-view feature selection based on weighted hinge loss.* Multimedia Tools and Applications, 2019. **78**(11): p. 15455-15481.
147. Nie, F., et al., *Auto-weighted multi-view learning for image clustering and semi-supervised classification.* IEEE Transactions on Image Processing, 2017. **27**(3): p. 1501-1511.
148. Huber, P.J., *Robust statistics*. International Encyclopedia of Statistical Science. 2011: Springer.
149. Greene, D., *3-sources*, U.C. Dublin, Editor.
150. Grimal, C., *WebKB*, U.o.C. LINQS, Santa Cruz, Editor.
151. Zhang, C., et al., *Three-way clustering method for incomplete information system based on set-pair analysis.* Granular Computing, 2019: p. 1-10.
152. Bodyanskiy, Y., A. Shafronenko, and D. Rudenko. *Online Neuro Fuzzy Clustering of Data with Omissions and Outliers based on Completion Strategy*. in *CMIS*. 2019. P.18-27.
153. Tao, X., et al., *Real-value negative selection over-sampling for imbalanced data set learning.* Expert Systems with Applications, 2019. **129**: p. 118-134.
154. Pouyanfar, S., et al., *A survey on deep learning: Algorithms, techniques, and applications.* ACM Computing Surveys (CSUR), 2019. **51**(5): p. 92.