Integrative Clustering of Multi-View Data: Subspace Clustering, Graph Approximation to Manifold Learning

A thesis submitted to Indian Statistical Institute
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science

by

Aparajita Khan
Senior Research Fellow

Under the supervision of

Dr. Pradipta Maji, Professor

Machine Intelligence Unit
Indian Statistical Institute, Kolkata

November 2021

To my parents,
who gave me hope. Always.

Acknowledgements

More than a thesis, a PhD is a journey. As I look back to the years that led to the completion of this thesis, I realize how fortunate I am to find such kind people around me whose consistent support, supervision, and confidence in me made this journey possible.

It is a pleasure for me to express my heartfelt gratitude to my thesis supervisor Prof. Pradipta Maji at the very outset for his endless support, guidance, training, and teaching throughout my PhD tenure. I thank him for giving me the opportunity to carry out my thesis work in his Biomedical Imaging and Bioinformatics Laboratory and introducing me to the world of front-line research in pattern recognition and machine learning. The thesis would not have been completed without his vision and technical support. Customary thanks always stand inadequate against the priceless effort he has put in to develop my technical knowledge and inspiring me to become an independent and creative researcher.

My indebtedness to Prof. C. A. Murthy is also beyond words. I consider myself lucky to have him as a teacher for the 'Advanced Pattern Recognition' course. His expositions of several topics in statistics and computer science was immensely helpful to me throughout my research career. I am thankful to Prof. Rajat K. De, Prof. Rana Barua, Prof. Palash Sarkar, Prof. Subhash C. Nandy, and Prof. Bhargab B. Bhattacharya for their teaching and technical insights on some of the core subjects of computer science and mathematics. All the discussions with them and their valuable inputs helped me a lot in several parts of my thesis.

I wish to extend my earnest gratitude to my senior, Dr. Manjari Pradhan, for always being a sweet big sister and her help in several aspects of my hostel life. Without her companionship, this journey would not have been so smooth and cherishable. I express my sincere gratitude to Dr. Sanchayan Santra and Mr. Avisek Gupta for their constant support during different phases of my life. The presence of all three of you has made my life graceful, and I never felt I am away from home. I am fortunate that I met you all. I am thankful to my lab members, especially, Ekta Shah, Ankita Mandal, Sankar Mondal, Suman Mahapatra, Debamita Kumar, Monalisa Pal, Sampa Misra, and Rishika Sen for creating such a pleasant atmosphere for research and collaboration within the lab. I am delighted to acknowledge my friends, Indrani, Arundhati, Sukriti, and Banani with whom I developed strong connections during my undergraduate years and their presence gives me strength and happiness.

I am indebted to the Dean of Studies and the Director of Indian Statistical Institute for providing me the fellowship and grants, and an incomparable infrastructure for research. I would also like to thank all the faculty members of Machine Intelligence Unit, Indian Statistical Institute, for their helpful suggestions and continued support throughout my PhD tenure. I express my sincere thanks to the authorities of the institute for the facilities extended to carry out my research. I also thank office staffs for their efforts to help me out in the official matters and conduct lab work smoothly.

Last but certainly not least, I would like to take this opportunity to express my sincere gratitude to my beloved parents for being the most substantial support not only during my PhD career but also throughout my life. Both of you are my source of inspiration. I am equally thankful for all the supports I got from my other family members. Finally, I owe a debt of gratitude to a very special person, my husband, Abir, for his unfailing love, support and understanding that made the completion of the thesis possible. He was always around at times when I found it hard to continue and helped me to keep things in perspective. I am moved by his simplicity and deeply appreciate his belief in me.

I thank everyone who directly and indirectly supported me throughout my thesis work.

Aparajita Khan

Abstract

Multi-view data clustering explores the consistency and complementary properties of different views to uncover the natural groups present in a data set. While multiple views are expected to provide more information for an improved learning performance, they pose their own set of unique challenges. The most important problems of multi-view clustering are the high-dimensional heterogeneous nature of different views, selection of relevant and complementary views while discarding noisy and redundant ones, preventing the propagation of noise from individual views during data integration, and capturing the lower dimensional non-linear geometry of each view.

In this regard, the thesis addresses the problem of multi-view data clustering, in the presence of high-dimensional, noisy, and redundant views. In order to select the appropriate views for data clustering, some new quantitative measures are introduced to evaluate the quality of each view. While the relevance measures evaluate the compactness and separability of the clusters within each view, the redundancy measures compute the amount of information shared between two views. These measures are used to select a set of relevant and non-redundant views during multi-view data integration.

The "high-dimension low-sample size" nature of different views makes the feature space geometrically sparse and the clustering computationally expensive. The thesis addresses these challenges by performing the clustering in the low-rank joint subspaces, extracted by feature-space, graph, and manifold based approaches. In feature-space based approach, the problem of incremental update of relevant eigenspaces is addressed for multi-view data sets. This formulation makes the extraction of joint subspace computationally less expensive compared to the principal component analysis. The graph based approaches, on the other hand, inherently take care of the data heterogeneity of different views, by modelling each view using a separate similarity graph. In order to filter out the background noise embedded in each view, a novel concept of approximate graph Laplacian is introduced, which captures the de-noised relevant information using the most informative eigenpairs of the graph Laplacian.

In order to utilize the underlying non-linear geometry of different views, the graph-based approach is judiciously integrated with the manifold optimization techniques. The optimization over Stiefel and k-means manifolds is able to capture the non-linearity and orthogonality of the cluster indicator subspaces. Finally, the problem of simultaneous optimization of the graph connectivity and clustering subspaces is addressed by exploiting the geometry and structure preserving properties of Grassmannian and symmetric positive definite manifolds.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Data, the seemingly abundant yet elusive entity, has been the driving force behind the growth of science over the last couple of decades. Nowadays, our daily interactions have majorarily shifted from the physical domain to the digital domain, and as a consequence, every little action generates data. Data pours in from every experiment performed, every file saved, every picture taken, every social media interaction, and every search query submitted to Google. A rough estimate of the amount of data, collected over the last few millennia upto the last decade, is about five exabytes. Nowadays, this same amount of data gets generated and stored every single day. And it is not only the volume of the data that has grown drastically, but also the variety of it. However, just having an abundance of data is not enough, it is also essential to analyze the data and make sense of it.

Data analysis refers to the process of cleaning, organizing, interpreting, and visualizing the data in order to transform it into useful information [88]. Information adds meaning to the data. It is obtained by looking for interesting and non-trivial patterns within the several bytes of numbers, letters, and characters collected as raw data. A pattern refers to a segment of the data that follows an identifiable trend or repeats itself in a discernible way. The huge volume and variety of data available these days necessitate the need for **pattern recognition** which is the automated process of discovering patterns and regularities in data [220]. The massive real-life data sets, along with informative patterns, may also contain measurement errors, imprecision, redundancy, and so on. **Machine learning** [15, 67] plays a significant role in discovering the natural structures within these massive and often noisy data sets. It is the systematic study and design of algorithms for learning useful and non-obvious patterns, and making inference from the data. It addresses the computational aspect of data driven knowledge discovery and decision making.

Depending on the learning strategy, pattern recognition and machine learning algorithms can be broadly classified into following three categories.

- **Supervised learning** algorithms aim to learn a function that maps a set of attributes describing a data instance, also known as object, sample or observation, to a set of labels or target attribute, using a collection of annotated training instances. For example, spam filters used in e-mail servers identify new incoming e-mails as “spam" or “not-spam" based on previously seen annotated instances. In supervised learning, each instance in the set of training examples must be labeled with the corresponding value of the target attribute. This requires a great deal of time and effort to create a data set with labeled instances.

- In **unsupervised learning**, the learners task is to make inference from a set of data instances in absence of labeled training samples. An example of unsupervised learning is to cluster COVID-19 affected patients based on demographics, mortality, and incidence rates, in order to identify vulnerable zones that would benefit from allocating additional resources by the governing authorities. Another example is grouping news articles, hosted in different online news portals, into different categories, such as sports, politics, business, science, etc. Unsupervised learning saves the time and effort invested in labeling the data instances; but lack of supervision makes the problem more challenging to solve.

- **Semi-supervised learning** forms the third category of machine learning algorithms. In several application domains, acquiring data is cheap, but acquiring labeled data turns out to be expensive. For example, in the problem of web page classification, all the web pages hosted in world-wide web are at our disposal, but creating a training set of annotated web pages is a tedious job. In semi-supervised learning, an initial model is developed based on the limited labeled training data and then unlabeled data is used to refine the model.

The learning algorithms traditionally work with two types of data set representations: feature vector based data and relational data [148]. Feature vector based representation consists of numerical, categorical, textual, or binary set of d features for n samples in a d-dimensional measurement space. For example, image data sets can be represented by global or local features (like color histogram) extracted from images or their raw pixel intensities. The relational data, on the other hand, is represented by n^2 pairwise relationships between n samples. For example, in the news article categorization, two articles can be considered to be similar or on related topic if there is a hyperlink connecting the two articles. A set of n samples, represented by d-dimensional feature vectors or by pairwise relationships, is referred to as a "*view*" or "*modality*" of a data set.

In several applications, only a single type of information may not be sufficient to characterize the nature and dynamics of the problem completely. A hyperlink connecting two news articles is not sufficient to claim that both of them belong to the same news category. Similarity between their content profiles also needs to be evaluated for making such a claim. Diverse information can be captured via multiple views for the same set of observations or samples. The thesis addresses the unsupervised learning problem for multi-view data sets.

1.1 Multi-View Data Analysis

Multi-view learning is an emerging machine learning paradigm that focuses on discovering patterns in data represented by multiple distinct views [203]. These data sets are nearly omnipresent in modern real-life applications due to an upsurge in different data collection, measurement, and representation techniques. In image and video processing, color, shape, and texture information generate three different kinds of feature sets, and each of them can be considered as a single view of the given data set. Similarly, in cross-language text classification, the same article can be written in multiple languages. This kind of data

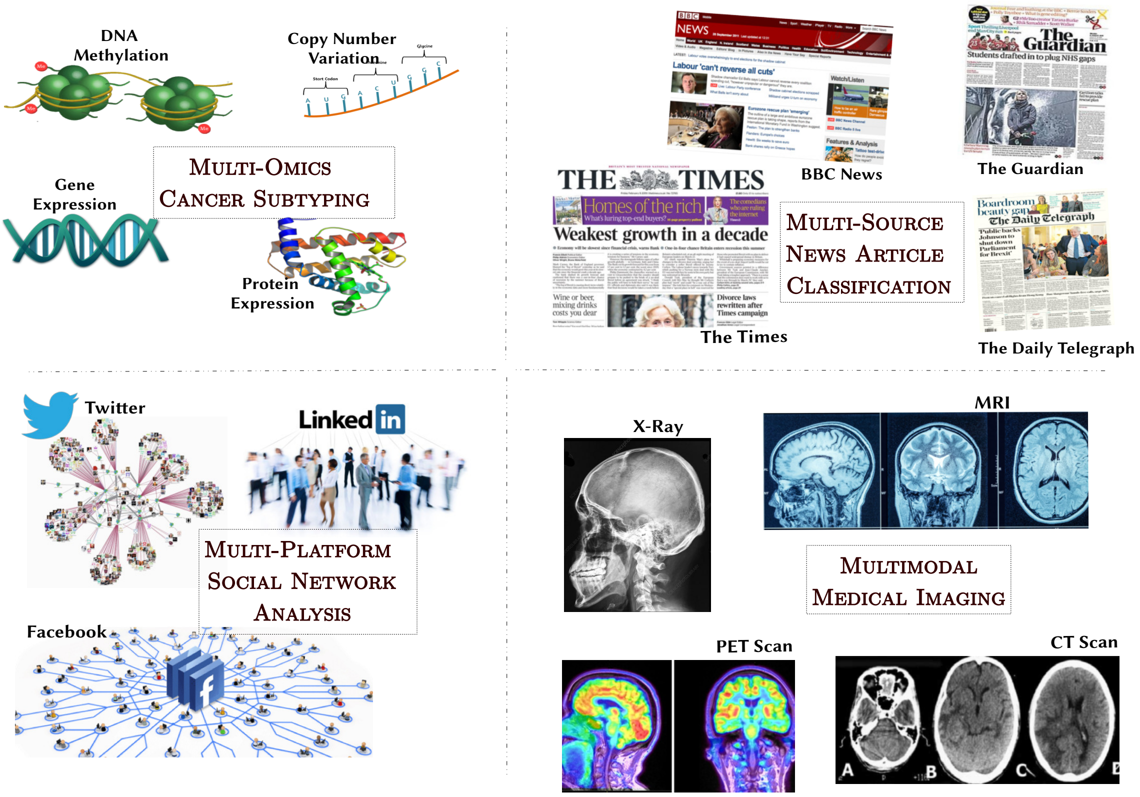


Figure 1.1: Different application areas of multi-view data analysis.

set is known as multi-view data, where each type of feature set or affinity/distance based representation corresponds to a single view.

One of the important unsupervised learning paradigms is clustering. It aims to find coherent groups of samples in the data, such that samples within a group are as similar as possible, while samples in different groups are as dissimilar as possible. Multi-view clustering groups the samples, based on the information of multi-view representation of the data set. Multi-view classification, on the other hand, tries to learn decision boundaries, separating different classes, from the labeled training examples having more than one view. Figure 1.1 shows some of the application areas of multi-view learning. While clustering or classification on each view separately may somewhat reveal the patterns present in the data set, multi-view analysis that utilizes the diverse information of multiple views has the potential to expose more fine-tuned structures that are not revealed by examining only a single view.

The idea of fusing information from multiple sources or views gained importance over traditional single view learning models during the last decade. Although relatively recent, multi-view learning has become an active area of research due to its remarkable success in a wide range of real-world applications, such as multi-camera face recognition, multi-source news article clustering, action recognition, multi-omics cancer stratification, biomedical imaging, and imaging genetics, to name just a few [130, 174, 288].

There are several reasons behind the prominent success of multi-view learning over its single view counterpart. Some of them are listed below.

1. **Comprehensive View of the System**: Different views can reveal different aspects, each giving a glimpse of the underlying dynamics of the whole system. For example, in face recognition, multiple cameras alleviate the limitations of a single camera since there are higher chances of a person being in a favorable frontal pose. Moreover, the facial appearance and features often vary significantly due to variation in lighting conditions, light angles, facial expressions, and head poses. In such a scenario, a multi-camera network obtains multiple images of a face in different poses and lighting conditions, and gives more accurate and robust face recognition results compared to single-camera/single-view analysis.

2. **Complementary Information**: Each view may contain complementary information that is not present in other views, even when all of them capture the same aspect of a system. In multi-omics study, both gene expression and copy number variation data contain genetic information of an individual. The gene expression conveys the overexpression or underexpression of a gene, while copy number variation gives the number of times a gene sequence has been repeated within the DNA of the individual. Utilization of both consistent and complementary information of different views can significantly improve the learning performance.

3. **Resilience to Noise**: Multi-view observations can reduce the effect of experimental or measurement noise in the data. Noisy observations in one view can be compensated by the corresponding observations of other views.

4. **Cross-Platform Analysis**: Due to the availability of multiple views, it is possible to perform a variety of additional analyses, such as drawing associations between variables observed in different views. In imaging genetic studies, given functional magnetic resonance imaging and single nucleotide polymorphism (SNP), it is possible to identify brain region alterations triggered by corresponding SNP changes in genes. Other analysis like estimation of noise content or significance of one view given other views is also possible owing to the availability of multiple views.

Despite these advantages, the abundance of data in multi-view learning comes with several challenges as well [288], which are discussed in the next section.

1.2 Challenges in Multi-View Analysis

The traditional machine learning algorithms, such as artificial neural networks, support vector machines, kernel machines, discriminant analysis, and spectral clustering, are devised to work on single view data. These algorithms do not trivially adapt to the multi-view setting, as the multi-view nature of the data set poses its own set of challenges. Some of them, focused more towards clustering, are listed as follows.

1. **Data Heterogeneity**: The simplest approach to handle the multi-view data sets using the conventional machine learning algorithms is to concatenate the feature sets of all the views to construct a single view. However, this concatenation is not meaningful as each view has its own specific statistical property, and the data in different views is usually measured in different units which are not necessarily compatible.

Different views vary immensely in terms of scale, unit, and variance. For instance, in multi-omics study, RNA sequence based gene expression data is measured in RPM (reads per million) consisting of real values in the order of 10^5, while DNA methylation data consists of β-values which lie in [0, 1]. The concatenation of features from these heterogeneous views is likely to reflect only the properties of views having high variance. Unbiased integration of multiple views requires extracting a transformed feature space or a uniform platform, so that intrinsic properties are equally preserved in all views. Clustering can be performed separately on each heterogeneous view. But, manual integration of clustering solutions from different views can be tedious and may fail to capture cross-platform correlations.

2. **High-Dimension Low-Sample Size Nature**: In real-life data analysis, data sets usually have large number of observed variables, such as several thousands of words in documents, nearly 10^6 pixels in images, 20K genes in DNA microarrays, and so on. The number of samples in these data sets typically ranges within a few thousand. Due to the lack of sufficient training samples, the learning models tend to overfit the data, thus reducing the generalization performance. The multicollinearity issue is also commonly observed in high dimensional settings, in which two or more features are highly correlated. This degrades the consistency properties of the eigenvalues and eigenvectors of the rank deficient sample covariance matrix [109]. In high dimensions, the feature space becomes geometrically sparse; and most of the clustering algorithms become computationally expensive and prone to degraded performance.

3. **Noisy and Redundant Views**: In real-world settings, the observations in different views are often corrupted by noise due to the measurement errors. The noise in different views gets propagated or even exaggerated during the data fusion process, if not explicitly taken care of. Furthermore, most of the multi-view algorithms consider all the available views for learning, under the assumption that each view is informative and provides homogeneous and consistent information about the underlying patterns in the data. However, some views may provide disparate, redundant, or even worse information. Due to the presence of noisy and redundant views, integration of all the available views can degrade the quality of cluster structures and decision boundaries learned from the data.

4. **View Disagreement**: In multi-view learning paradigm, the views are expected to uniformly agree upon an underlying global class/cluster structure. This implies that two samples belonging to a class in one view should belong to the same class in other views as well. However, in realistic settings, data sets are often corrupted by noise and each view is likely to be corrupted by an independent noise process. In such a situation, a set of observations in some views gets corrupted while the corresponding observations in other views may remain unaffected. For example, in multi-sensory data sets, a sensor may temporarily get to an errorneous state before returning back to normal condition. This may lead to disagreement between different views. In case of severe disagreement or corruption, the clusters identified in different views would not conform with each other, and hence arriving at a global consensus becomes hard [43].

5. **Low-Rank Non-Linear Geometry of Views**: In several real-life data sets, most of the views have large number of features. Although the data in these views may appear

to point clouds in a high-dimensional feature space, itâĂŹs meaningful structures often reside on a lower dimensional subspace or manifold embedded in the high-dimensional space. Moreover, in high-dimensions, the ratio between the nearest and farthest points approaches one, that is, the points tend to become uniformly distant from each other [4]. Consequently, the problem of clustering points based on its nearest neighborhood becomes ill-posed, since the contrast between the distances to different data points cease to exist. Hence, clustering in the high-dimensional original feature space usually gives poor performance compared to a transformed space. Even in transformed space learning, extracting a single subspace or manifold for a multi-view data set might not be sufficient. Each view has its own underlying, possibly non-linear, geometry that needs to be captured separately.

6. **Incomplete Views**: Most of the multi-view learning algorithms assume that all samples can be successfully observed on all the views. However, due to measurement and pre-processing errors, the data sets are prone to having incomplete views, where a sample is not observed in one view (missing view), or the sample is partially observed (missing variables). Consideration of only the samples observed in all the views reduces the sample size and makes the model prone to overfitting. The presence of incomplete views necessitates utilization of the connection between the views and restoration of samples in the incomplete views with the help of corresponding samples in the complete views [257].

Some of these challenges like data heterogeneity and incomplete views are inherent to multi-view data, while other challenges like high-dimension low-sample size nature and low-rank geometry exist in single-view data as well. However, presence of multiple heterogeneous views escalates the complexity of these problems. Hence, some new advanced algorithms need to be designed that can efficiently address these challenges and mine meaningful patterns embedded in multi-view data sets.

1.3 Scope and Organization of Thesis

In this regard, the thesis aims at designing a set of algorithms to address some of the problems of multi-view data integration and clustering. One of the major challenges in multi-view clustering is the high-dimension low-sample size nature of each view. For high-dimensional view, the standard approach is to extract a lower dimensional transformed space that captures the cluster structure better than the high-dimensional input space and to perform clustering in that space. The transformed space can be a linear subspace or a general non-linear manifold embedded within the ambient input feature space. Furthermore, depending on the objective function, there can be numerous lower dimensional subspaces and manifolds of the same high dimensional space. The main contribution of this thesis is to design some novel algorithms to extract informative subspaces and manifolds for multi-view data analysis and clustering, and theoretically analyze important properties of these transformed spaces and new algorithms therein.

The outline of the thesis is presented in Figure 1.2. The thesis consists of eight chapters. Chapter 1 provides an introduction to multi-view data analysis and outlines some of it's application areas. It also discusses the major challenges encountered during integrative

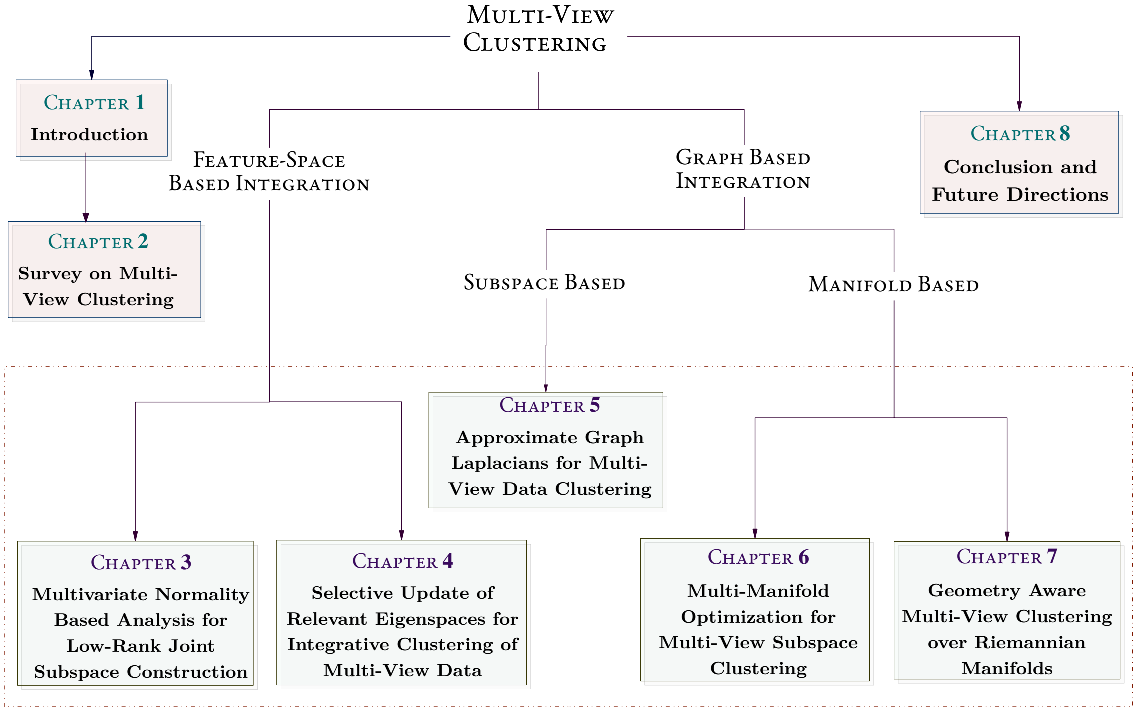

Multi-View Clustering
Chapter 1
Introduction
Chapter 2
Survey on Multi-View Clustering
Feature-Space Based Integration
Graph Based Integration
Chapter 8
Conclusion and Future Directions
Subspace Based
Manifold Based
Chapter 5
Approximate Graph Laplacians for Multi-View Data Clustering
Chapter 3
Multivariate Normality Based Analysis for Low-Rank Joint Subspace Construction
Chapter 4
Selective Update of Relevant Eigenspaces for Integrative Clustering of Multi-View Data
Chapter 6
Multi-Manifold Optimization for Multi-View Subspace Clustering
Chapter 7
Geometry Aware Multi-View Clustering over Riemannian Manifolds

Figure 1.2: Outline of the thesis.

analysis of multi-view data. Chapter 2 describes the problem of multi-view clustering and its basic principles. A brief survey on existing multi-view clustering approaches is also covered in this chapter.

One of the important challenges in multi-view data integration is the appropriate selection of relevant and complementary views over noisy and redundant ones. Another challenge is the high dimension-low sample size nature of each view. Chapter 3 addresses these two challenges by proposing a novel algorithm, which constructs a low-rank joint subspace from the low-rank subspaces of individual high-dimensional views. Statistical hypothesis testing is introduced to effectively estimate the rank of each view by separating the signal component from its noise counterpart. Two quantitative indices are proposed to evaluate the quality of different views. While the first one assesses the degree of relevance of the cluster structure embedded within each view, the second measure evaluates the amount of cluster information shared between two views. To construct the joint subspace, the algorithm selects the most relevant views with maximum shared information. During data integration, the intersection between two subspaces is also considered to select cluster information and filter out the noise from different subspaces. The efficacy of clustering on the joint subspace, extracted by the proposed approach, is compared with that of several existing integrative clustering approaches on real-life multi-omics cancer data sets. Survival analysis is performed to reveal the significant differences between survival profiles of the identified subtypes, while robustness analysis shows that the identified subtypes are not sensitive towards perturbation of the data sets.

Due to the high-dimensional nature of the multi-view data sets, extracting a low-dimensional subspace often becomes computationally very expensive. Extraction of the principal subspace by performing principal component analysis (PCA) on the integrated data set requires eigendecomposition of a considerably higher order covariance matrix. In this regard, Chapter 4 addresses the problem of incrementally updating the singular value decomposition of a higher order data matrix in the context of the multi-view data integration. This analytical formulation enables efficient construction of the joint subspace of integrated data from low-rank subspaces of the individual views. Construction of joint subspace by the proposed method is shown to be computationally more efficient as compared to PCA on the integrated data matrix. New quantitative indices are introduced to theoretically quantify the gap between the joint subspace extracted by the proposed approach and the principal subspace extracted by performing PCA on the integrated data matrix, in terms of the principal angles between these subspaces. Finally, clustering is performed on the extracted joint subspace to identify meaningful clusters. The clustering performance of the proposed approach is studied and compared with that of existing integrative clustering approaches on several real-life multi-view cancer data sets.

Different views of a multi-view data set vary immensely in terms of unit and scale. One of the important approaches of handling data heterogeneity in multi-view data clustering is modeling each modality or view using a separate similarity graph. Information from the multiple graphs is then integrated by combining them into a unified graph. A major challenge here is how to preserve cluster information while removing noise from individual graphs. In this regard, Chapter 5 presents a novel graph-based algorithm that integrates noise-free approximations of multiple similarity graphs. The proposed method first approximates a graph using the most informative eigenpairs of its Laplacian which contain cluster information. The approximate Laplacians are then integrated for the construction of a low-rank subspace that best preserves overall cluster information of multiple graphs. However, this approximate subspace differs from the full-rank subspace which integrates information from all the eigenpairs of each Laplacian. The matrix perturbation theory is used to theoretically evaluate how far approximate subspace deviates from the full-rank one for a given value of approximation rank. Finally, spectral clustering is performed on the approximate subspace to identify the clusters. Extensive experiments are performed on several real-life cancer as well as benchmark multi-view data sets to study and compare the performance of the proposed approach.

The meaningful patterns embedded in high-dimensional multi-view data sets typically tend to have a much more compact representation that often lies close to a low-dimensional manifold. Identification of hidden structures in such data mainly depends on the proper modeling of the geometry of low-dimensional manifolds. In this regard, Chapter 6 presents a manifold optimization based integrative clustering algorithm for multi-view data. To identify consensus clusters, the algorithm uses the approximate joint graph Laplacian, proposed in Chapter 5, to integrate de-noised cluster information from individual views. It then optimizes a joint clustering objective, while reducing the disagreement between the cluster structures conveyed by the joint and individual views. The optimization is performed alternatively over k-means and Stiefel manifolds. The Stiefel manifold helps to model the non-linearities and differential clusters within the individual views, while k-means manifold tries to elucidate the best-fit joint cluster structure of the data. A gradient based movement is performed separately on the manifold of each view, so that individual non-linearity is preserved while looking for shared cluster information. The convergence of the proposed algorithm is established over the manifold, and asymptotic convergence bound is obtained to quantify theoretically how fast the sequence of iterates generated by the algorithm converges to an optimal solution. The performance of the proposed approach, along with a comparison with state-of-the-art multi-view clustering approaches, is demonstrated on synthetic, benchmark and multi-omics cancer data sets.

Simultaneous optimization of the individual graph structures, their weights, and the joint and individual subspaces, is likely to give a more comprehensive idea of the clusters present in the data set. In this regard, Chapter 7 presents another manifold optimization algorithm that harnesses the geometry and structure preserving properties of symmetric positive definite (SPD) manifold and Grassmannian manifold for efficient multi-view clustering. The SPD manifold is used to optimize the graph Laplacians corresponding to the individual views while preserving their symmetricity, positive definiteness, and related properties. The Grassmannian manifold, on the other hand, is used to optimize and reduce the disagreement between the joint and individual clustering subspaces. The geometry preserving property of Grassmannian optimization additionally enforces the clustering solutions to be basis invariant cluster indicator subspaces, such that all cluster indicator matrices whose columns span the same subspace map to the same clustering solution. A gradient based line-search algorithm, that alternates between different manifolds, is proposed to optimize the subspaces and Laplacians. The matrix perturbation theory is used to theoretically bound the disagreement or Grassmannian distance between the joint and individual subpaces at any given iteration of the proposed algorithm. The disagreement is empirically shown to minimize as the algorithm progresses and converges to a local minima. The comparative clustering performance of the proposed and existing approaches is demonstrated on several benchmark and multi-omics cancer data sets.

Finally, Chapter 8 concludes the thesis, and discusses the future scopes and improvements of the proposed research work.

Chapter 2

Survey on Multi-View Clustering

This chapter presents the basics of the multi-view clustering problem. A brief literature survey focused primarily on multi-view clustering, along with it's classification counterpart is also covered in this chapter.

2.1 Multi-View Clustering

A multi-view data set of n samples, $\{x_1, x_2, \ldots, x_n\}$, consists of M views, where $M \geqslant 2$. The term "view" is used interchangeably with the term "modality" throughout the thesis, and accordingly, a multi-view data set is also referred to as a "multimodal data set". The views or modalities can be represented by feature vector based data or by relational data. In case of feature vector based representation, an M-view data set is given by a set of M data matrices $X_1, X_2, \ldots, X_m, \ldots, X_M$, each corresponding to one of the M views. Each X_m is a $(n \times d_m)$ matrix consisting of d_m features for each of the n samples, observed in a d_m-dimensional measurement space. The most commonly encountered space is the Euclidean space, in which case, X_m contains numeric values in $\Re^{n \times d_m}$. The views can contain other types of data as well, like, textual, categorical, binary, and so on. The measurement space, as well as the number of observed variables, d_m, need not be the same across different views. The matrices $X_1, \ldots, X_M$ may vary in terms of their scale, unit, variance, dimension (column-wise), and data distribution. In case of relational data, the M views are typically represented by M similarity (distance) matrices $W_1, W_2, \ldots, W_m, \ldots, W_M$. Each W_m is a $(n \times n)$ matrix given by

$$W_m = [w_m(i,j)]_{n \times n},$$

where $w_m(i,j) \geqslant 0$ is the similarity (distance) between samples x_i and x_j in the m-th view.

Figure 2.1 shows an example of multimodal omics data set with feature vector based representation. The advent of whole genome sequencing technologies have led to the generation of different types of "omics" data from different levels of the genome. As shown in Figure 2.1, the DNA methylation, copy number variation, gene expression, and protein expression data can be observed from the epigenomic, genomic, transcriptomic, and proteomic levels of the genome, respectively. In a multimodal data set, these observations can be made for a common set of n samples or patients whose genome is being sequenced. The resulting data set is a collection of M views, denoted by $X_1, X_2, \ldots, X_m, \ldots, X_M$. Each

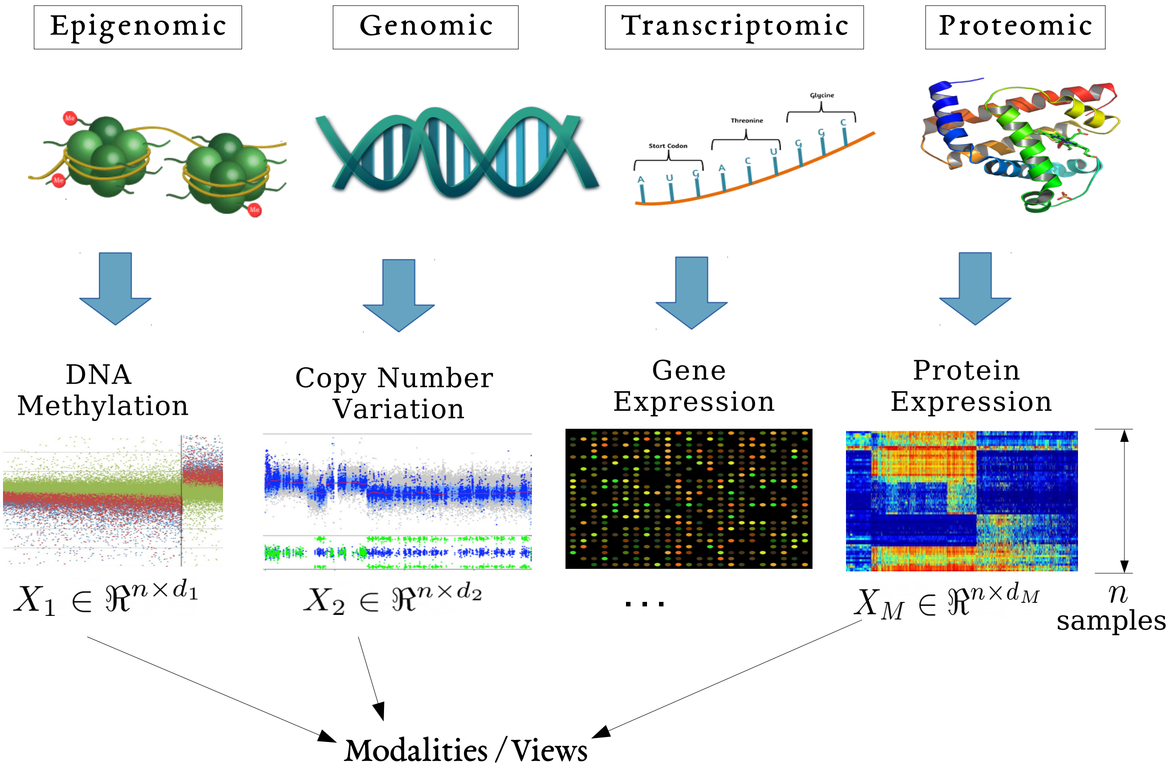


Figure 2.1: Different views of multi-omics data analysis.

X_m, in this case, is a $(n \times d_m)$ data matrix consisting of the expression levels of d_m genes, or micro-RNAs, or proteins for those n samples.

Clustering is an unsupervised learning approach, which discovers the natural groups present in a data set. Multi-view clustering aims at partitioning the n samples, $\{x_i\}_{i=1}^n$, into k subsets $A_1, A_2, \dots, A_k$ based on the feature/ similarity information of multiple views, such that the following three conditions are met:

- $A_j \neq \emptyset$, for $j = 1, 2, \dots, k$.
- $\bigcup_{j=1}^{k} A_j = \{x_1, \dots, x_n\}$.
- $A_j \cap A_l = \emptyset, \forall j \neq l$, and $j, l = 1, 2, \dots, k$.

In addition, the samples contained in a cluster A_j are "more similar" to each other and "less similar" to those in other clusters.

According to the above definition of clustering, each sample can belong to a single cluster. Hence, this type of clustering is termed as "crisp", "hard" or "partitional" clustering. An alternate formulation of clustering, termed as "fuzzy clustering", was introduced by Zadeh [270]. A fuzzy clustering of the samples $\{x_1, \dots, x_n\}$ into k clusters is characterized by k membership functions u_j where

$$u_j : \{x_i\}_{i=1}^n \longrightarrow [0, 1], \text{ for } j = 1, 2, \dots, k,$$

such that

$$\sum_{j=1}^{k} u_j(x_i) = 1, \text{ for } i = 1, 2, \ldots, n, \qquad \text{and} \qquad 0 < \sum_{i=1}^{n} u_j(x_i) < n, \text{ for } j = 1, 2, \ldots, k.$$

Under fuzzy clustering, each sample may belong to more than one cluster “up to some degree". The membership function $u_j(x_i)$ gives the degree of belongingness of sample x_i to the j-th cluster. Fuzzy multi-view clustering is relatively less explored compared to its hard counterpart. This thesis is focused on the design and analysis of hard multi-view clustering algorithms and Section 2.2 primarily covers a brief survey of the same.

The area of multi-view learning is relatively new. However, owing to its state-of-the-art performance in several application areas it quickly came into the limelight of machine learning research and developed a rich literature over the past decade. The literature on multi-view learning can majorarily be divided into multi-view clustering and multi-view classification. Since the thesis focuses on multi-view clustering, the next section describes different multi-view clustering approaches, and then Section 2.3 briefly touches upon it's classification counterpart.

2.2 Multi-View Clustering Approaches

Multi-view clustering algorithms can roughly be classified into seven categories based on their algorithmic approaches, as shown in Figure 2.2. These categories are outlined in the following subsections.

2.2.1 Early Integration Approaches

An early integration approach first concatenates the feature based raw data matrices from all the views and then applies a single-view based clustering algorithm on the concatenated data matrix. This straightforward integration enables the direct application of traditional clustering algorithms to multi-view data. Given the feature based representation of views, $X_1 \ldots X_M$, the concatenated data matrix is formed by

$$\mathbf{X} = \begin{bmatrix} X_1 & X_2 & \ldots & X_m & \ldots & X_M \end{bmatrix},$$
$$\text{where } X_m \in \Re^{n \times d_m} \text{ and } \mathbf{X} \in \Re^{n \times d} \text{ such that } d = \sum_{m=1}^{M} d_m.$$

Then, any single-view clustering algorithm like k-means [139], spectral clustering [230], or Gaussian mixture models [50] can be applied on the raw concatenated matrix $\mathbf{X}$. Figure 2.3 shows a diagrammatic representation of the early integration approach, where the k-means clustering algorithm is applied on $\mathbf{X}$ to obtain the clusters.

The naive concatenation, however, increases the dimension or number of features in the data set, which is a major challenge for some of the single views as well. One baseline solution to the problem of early integration is to perform PCA on concatenated data $\mathbf{X}$ and then perform the single-view clustering, like k-means clustering, on top few principal

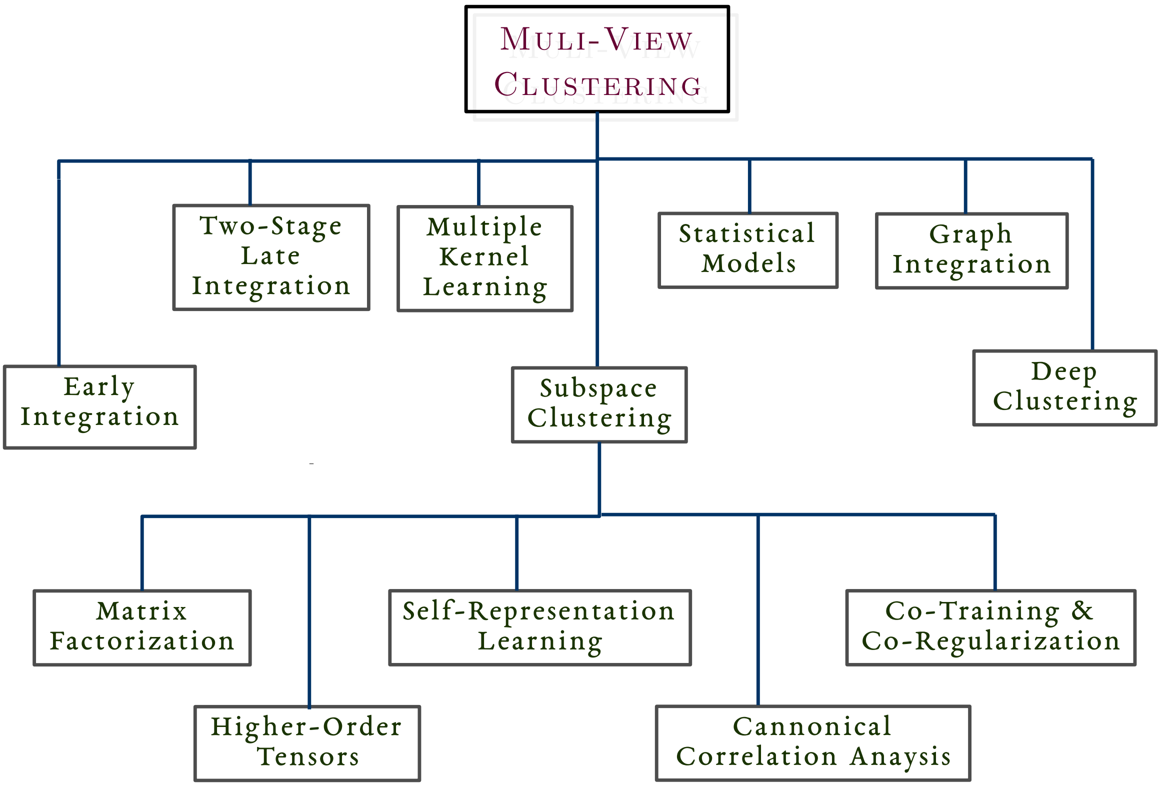


Figure 2.2: Different types of multi-view clustering approaches.

components of **X**. Another approach of handling the high dimension and it's subsequent problem of overfitting is to add regularization to induce data sparsity [221]. In high-dimensional multi-view data integration, even though a majority of features in one view may not be discriminative for a group of samples, a small number of features in the same view can still be highly discriminative. In [235], sparsity inducing $\ell_{2,1}$-norm regularization is imposed on **X** in order to obtain discriminative features from different views. The ℓ_2-norm regularization is imposed within each view to emphasize on view-specific feature weight learning corresponding to each cluster, while ℓ_1-norm is used to enforce sparsity between different views and learn features that are discriminative across multiple clusters.

Although PCA and regularization can somewhat address the curse of dimensionality, there are two more issues with naive concatenation. Firstly, the lack of appropriate normalization is likely to give higher weight to views with larger number of features or higher variance. But, it may not necessarily detect the best possible cluster structure. Secondly, naive feature concatenation does not take into account the difference in the distribution, scale, and unit of measurement of the data in different views. Hence, the concatenation may not be meaningful.

2.2.2 Two-Stage Late Integration Approaches

In the late integration approach, each view is first clustered separately using a single-view clustering algorithm. The per-view clustering solutions are integrated at a second stage

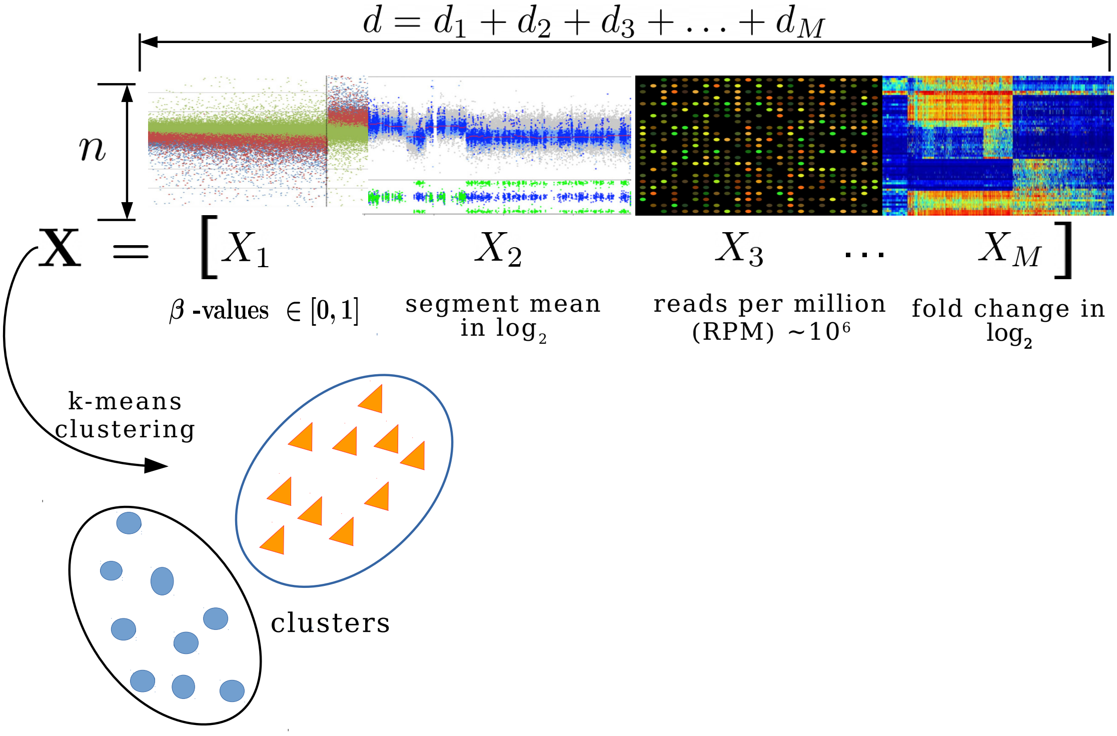

$d = d_1 + d_2 + d_3 + \ldots + d_M$
n
$\mathbf{X} = [X_1 \quad X_2 \quad X_3 \quad \cdots \quad X_M]$
β -values $\in [0,1]$
segment mean in $\log_2$
reads per million (RPM) $\sim 10^6$
fold change in $\log_2$
k-means clustering
clusters

Figure 2.3: Early integration based multi-view clustering.

to identify the integrative clusters [23, 93, 140, 214]. Figure 2.4 shows an example of the two-stage clustering approach, where the final clusters are obtained by taking a global consensus on the individual view-specific clusterings

In the cluster of cluster assignments (COCA) algorithm [93], the clustering solution of a sample x_i, corresponding to view X_m, is encoded by a binary cluster indicator vector that contains a 1 at index j indicating the belongingness of sample x_i to cluster j, and 0 otherwise. The binary vectors for all samples acrosss all views are combined to obtain a multi-view cluster indicator matrix. Consensus clustering [160] on the multi-view indicator matrix reveals the final clustering of the samples. The COCA algorithm has been applied for pan-cancer analysis of multiple genomic modalities. It investigates how tumors in different types of tissues cluster together, and whether the obtained tumor clusters resemble the tissue of the site of cancer [93].

Among the probalilistic approaches, Bruno and Maillet [23] used latent semantic analysis to obtain the final clusters from the multi-view cluster indicator matrix. In Bayesian consensus clustering [140], a Bayesian framework driven by Dirichlet mixture model is developed for simultaneous estimation of the consensus as well as view-specific clusterings. The Dirichlet distribution based modelling has the advantage of incorporating uncertainity within both view-specific and consensus clustering. More flexible methods allow for more general consensus strategies and dependence models. Kirk *et al.* [115] proposed the Bayesian correlated clustering algorithm, which uses a statistical framework to cluster each view while simultaneously modeling the pairwise inter-dependence between two clusterings. In Bergman consensus clustering [126], the disagreement between the consensus clustering result and the input view-specific clusterings is generalized from the traditional Euclidean

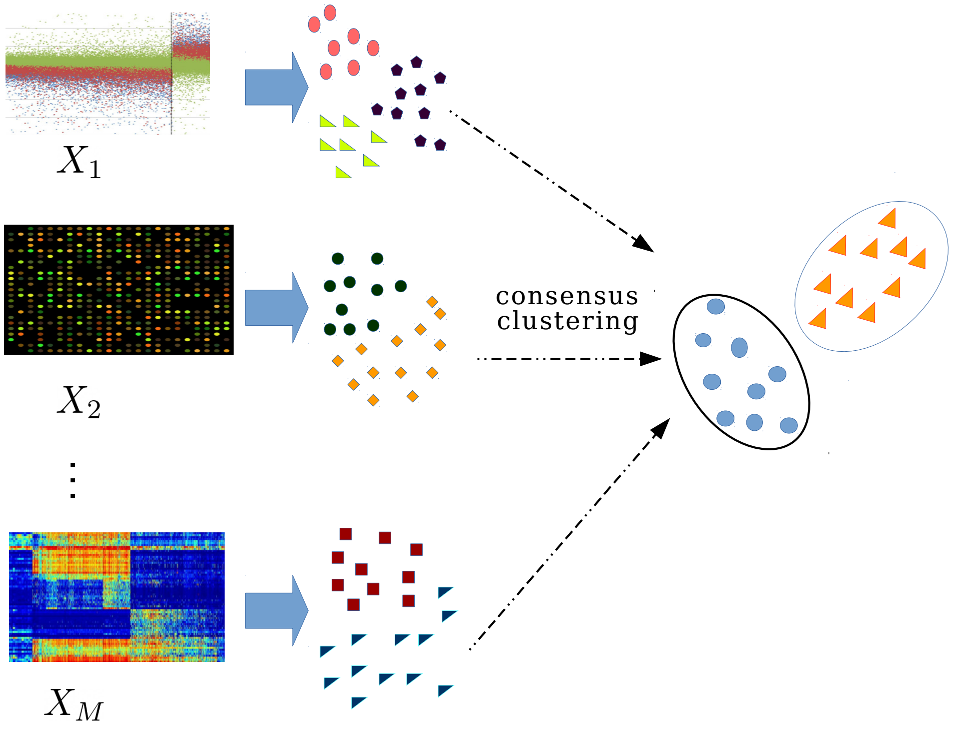


Figure 2.4: Two-stage consensus clustering.

distance to a more general Bergman loss.

The advantage of these late integration approaches is that any clustering algorithm can be used in single view stage. Certain clustering algorithms that are known to work well on certain views can be independently used on those views without having to find a unified model or algorithm that works for all views. However, the major drawback is that the late integration of the view-specific clustering solutions often becomes cumbersome and may fail to capture joint structures shared by different views.

2.2.3 Subspace Clustering Approaches

In several real-world applications, although the observed data is high-dimensional, the essential information of the data can be represented in a much lower dimension. For instance, there can be a large number of pixels in a given image, yet the appearance, geometry, objects, and dynamics of a scene can be described using only a few parameters. Subspace based approaches seek to find a unified latent space from multiple low-dimensional subspaces and afterwards perform clustering in the latent space [137, 141, 289]. Figure 2.5 illustrates the general approach of multi-view subspace clustering. Low-rank subspaces from individual views are merged to obtain a joint subspace, clustering on which gives the final clusters. The mapping of the high-dimensional views to low-dimensional subspaces can be achieved by a variety of methods such as matrix factorization, low-rank approximation, and tensor decomposition. Apart from these methods, subspaces have also been extracted with the idea of preserving different desirable properties in the latent space, like locality and neighborhood, self-representativeness, non-negativity, sparsity, correlation, and so on.

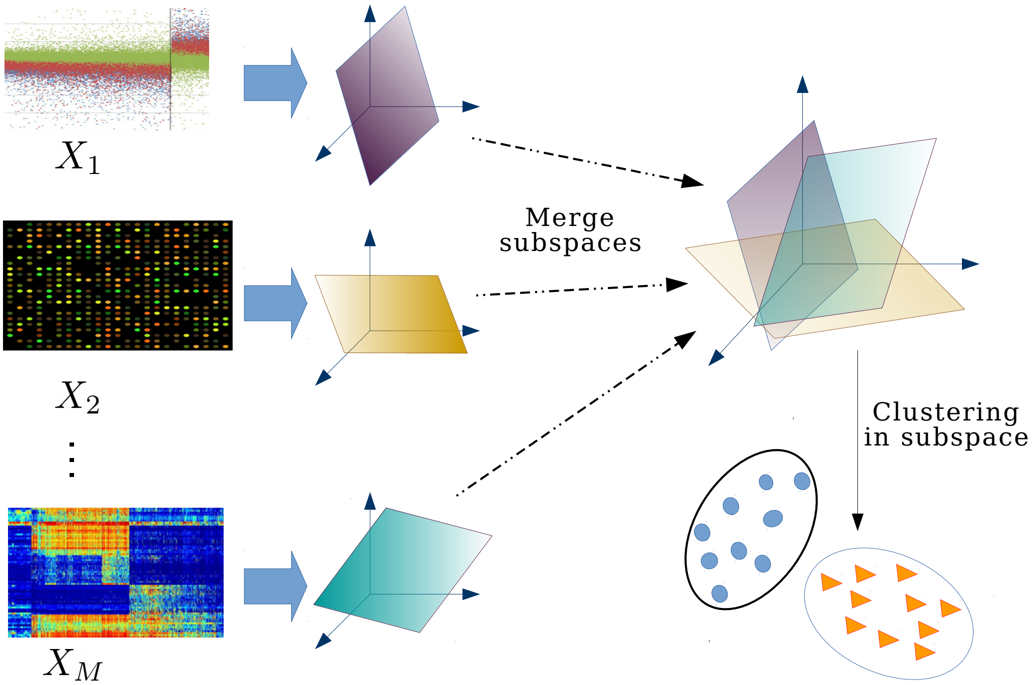


Figure 2.5: Multi-view subspace clustering.

Few mainstream subspace based approaches are described briefly as follows.

2.2.3.1 Matrix Factorization Based Approaches

Matrix factorization approaches use factorization algorithms to obtain low-rank factors that act as representations of the high-dimensional points in a lower dimensional subspace. One of the earliest and widely used factorization algorithm is that of non-negative matrix factorization (NMF) [123]. For a single view X_m, NMF assumes that X_m has an intrinsic lower dimensional non-negative representation, and tries to approximate each X_m as a product of two low-rank non-negative factors $Z_m \in \Re_+^{n\times r}$ and $H_m \in \Re_+^{r\times d_m}$, such that $X_m \approx Z_m H_m$. Among the multi-view extensions of NMF, the MultiNMF [137] algorithm integrates the views by imposing their corresponding low-rank representations Z_ms to be close to a global consensus representation $\bar{Z} \in \Re_+^{n\times r}$. The rows of $\bar{Z}$ are treated as representation of the samples in a r dimensional subspace and the joint clusters are identified using a standard clustering algorithm like k-means on consensus matrix $\bar{Z}$. In another algorithm, termed as JointNMF [137, 281], instead of obtaining a common factor $\bar{Z}$ that is close to all the Z_ms through a two-stage optimization, each view X_m itself is approximated by a common $\bar{Z}$ and a view-specific factor H_m, that is $X_m \approx \bar{Z}H_m$. Locality preserving NMF models have also been proposed, which constrain the pairwise simlarities between the samples in the latent representation $\bar{Z}$ to be proportional to those in the original space X_m [101, 284]. Other variants of NMF based multi-view clustering algorithms have also been proposed, such as, semiNMF [287], graph regularized NMF [146], manifold regularized NMF [283, 299], local patch alignment based NMF [169], and robust neighboring constraint NMF [36].

The reason behind the popularity of NMF is its ability to extract sparse and easily interpretable factors. The NMF latent factors can be viewed as part-based representations because the non-negativity constraint allows only additive, and not subtractive combinations. For instance, in case of an image X_m, the columns of the Z_m factor can be interpreted as basis images, and H_m states how to sum up the basis images in order to reconstruct an approximation of a given image X_m. In the case of facial images, the basis images are features such as eyes, noses, moustaches, and lips, while the columns of H_m indicate which feature is present in which image. The eigenvector based factors in other matrix decompositions like SVD contain both positive and negative entries, which lack the additive factor based meaningful interpretation. However, NMF in it's basic form [123] is applicable only for non-negative data. Also, the factors Z_m and H_m are not unique as they are obtained by alternating optimization, which is sensitive to initialization.

Multi-view clustering has also been addressed using other factorization algorithms, such as tri-matrix factorization [298], anchor graph based non-negative orthogonal factorization [261], bilinear factorization [293], and SVD [91, 141]. Among the SVD based approaches, the joint and individual variance explained (JIVE) [141], and angle based JIVE [63] (A-JIVE) algorithms use SVD to partition each view into a common joint factor and a view-specific individual factor. Clustering on the rows of the joint factor gives the consistent global clustering of the data set, while that on the individual factors give the view-specific clusterings. The different factorization based approaches differ in terms of the interpretation of the low-rank factors, properties preserved in them, and their computational complexity.

2.2.3.2 Tensor Based Approaches

A natural extension of matrix factorization methods for multi-view analysis is the use of tensors, which are higher order matrices. While the conventional matrices can capture pairwise correlations within a view, tensors are capable of capturing high-order correlations among multiple views and extracting information from a multidimensional perspective [39, 245, 253]. Zhang *et al.* [276] proposed the construction of third-order tensor with low-rank constraint to model the cross information among different views and reduce redundancy in the learned subspace. Jia *et al.* [107] imposed structured sparsity and symmetric low-rank constraints on the horizontal and frontal slices of higher-order tensors to model both inter and intra-view relationships. Tensors are also fused with graph and kernel learning to improve multi-view clustering performance. Wu *et al.* [245] proposed to learn a low-rank tensor for spectral clustering [230] directly from multiple similarity graphs. Tensors and affinity graphs are learned simultaneously for multi-view spectral clustering in [41, 42]. Xie *et al.* [252] kernelized the high-dimensional input features using a tensor learning framework to capture non-linear relationships between the samples. Multi-view clustering based on tensor-SVD and its derived tensor nuclear norm are extensively explored in [244, 248, 253, 285].

The low-rank tensor learning approaches generally work by decomposing the input tensor across multiple dimensions into lower order factors. The multi-dimensional decomposition considers excessive combinations of all input features. One major challenge here is to extract useful information from the decomposition and discard useless feature combinations [105]. Moreover, the higher order relations learned from the inherently noisy views can often give misleading information.

2.2.3.3 Self-Representation Based Subspace Learning Approaches

The self-representation based multi-view subspace clustering approaches emerged from two popular baseline approaches, namely, sparse subspace clustering (SSC) [59,60] and low-rank representation (LRR) [135]. Both of these approaches are based on the assumption that high-dimensional data belonging to multiple classes or categories often lies in a union of low-dimensional subspaces. This assumption implies that each sample or data point lying in a union of multiple low-rank subspaces can always be expressed as an affine or linear combination of a few other points belonging to that subspace, referred to as the *'self-representative'* property. These algorithms look for the sparsest combination in order to learn an appropriate basis to fit each group and automatically determine other points lying in the same subspace. The sparse combination coefficients are used to build a similarity matrix from which clusters are identified by spectral clustering.

Multi-view extensions of subspace clustering have been proposed, which directly reconstruct the data points in the original views using the self-representative property and generate view-specific subspace representation [26, 27, 71, 240, 241, 275, 276]. Among them, Zhang *et al.* [276] extended LRR for the multi-view setting using generalized tensor nuclear norm, while Cao *et al.* [27] introduced a diversity term based on Hilbert Schmidt independence criterion, and Wang *et al.* [240] added an exclusivity term to seek complementarity and consistency of the multi-view subspace representations. These approaches reconstruct the data points in each view separately using self-representation. However, a general assumption is that multiple views are generated from a single underlying latent distribution. Based on this assumption, latent multi-view subspace clustering (LMSC) [277] generates a common latent representation for all views rather than that of each individual one. Zhang et al. [275] extended LMSC by introducing neural networks to explore more general relationships between the views. Extensions are also proposed to enable subspace learning in presence of missing samples and features in different views [167, 250, 255].

A majority of this category of subspace clustering approaches learn view-specific self-representations. However, each view is likely to give only partial information about the overall structure of the multi-view data set. Hence, subspaces and clusters reconstructed from view-specific self-representations of samples may not give a complete or even accurate picture of the multi-view data set [212].

2.2.3.4 Cannonical Correlation Analysis Based Approaches

Cannonical correlation analysis (CCA) gained importance in multi-view learning as it naturally extracts two projection vectors from any two views such that the projected data along those vectors is maximally correlated [97]. However, real-world data sets usually have complex structure which is hard to capture via a single pair of covariates. The r-th pair of covariates is obtained by maximizing the correlation between the new pair while constraining it to be orthogonal to the previous ones. Chaudhuri *et al.* [35] theoretically established that when the data is drawn from a mixture of Gaussians or a mixture of log concave distributions, the canonical covariates can be used to cluster the data.

The conventional CCA fails in the high-dimensional low-sample size setting due to the non-invertibility of covariance matrix, multi-collinearity of features, and computational difficulty. To address these issues, sparse CCA [44, 61] is proposed to incorporate feature selection into the CCA model and maximize correlation between only a small subset of features. Other variants of CCA are also proposed for multi-view data integration and clustering, for example, group sparse CCA [133], kernel CCA [17], and cluster CCA [175]. Blaschko and Lampert [17] used kernel CCA to learn non-linear relationships and proposed a generalized spectral clustering algorithm for two-view data.

A major drawback of CCA is that it can extract correlated features from only two feature sets or views. A typical way of generalizing CCA to multiple views is to maximize the sum of pairwise correlations between all pairs of views [229]. However, this approach is unable to capture higher-order correlations obtained by simultaneous examination of all views. In this regard, some generalizations of CCA have been proposed to simultaneously handle an arbitrary number of views. Examples include multi-set CCA [95], generalized regularized CCA [219], graph multi-view CCA [37], and tensor CCA [144], among others.

2.2.4 Co-Training and Co-Regularization Approaches

The co-training and co-regularization approaches are based on the idea that the true underlying clustering of multiple views would assign corresponding points in each view to the same cluster. Based on this assumption, Kumar and Daumé [119] proposed a two-view clustering algorithm that uses clustering result from one view to guide the other view, and vice versa. Specifically, the spectral embedding from graph Laplacian [230] of one view is used to refine the similarity graph used for the other view. By alternately iterating this approach between two views, the clusterings of two views tend to be close to each other. Multi-view extensions of the two-view co-regularization has also been proposed in [120].

Yu *et al.* [268] introduced co-regularization into the self-representation based multi-view subspace clustering framework [60, 71]. Zhao *et al.* [291] combined the simplicity of linear discriminant analysis (LDA) and k-means clustering, along with the co-training approach, to extract discriminative subspaces from one view based on the clustering labels learned in another view. Xu *et al.* [258] imposed tensor nuclear norm constraints on the co-regularization model to capture higher-order relations while looking for consistent clustering across different views. One drawback of the co-training based algorithms is that they co-regularize each view equally which does not make sense when one view is informative, while the other is noisy.

2.2.5 Multiple Kernel Learning Approaches

Kernel functions implicitly map data points into a high (possibly infinite) dimensional space and compute inner-product between images of points without explicitly computing their coordinates in the transformed space [190]. Multiple kernel multi-view clustering approaches pre-define a group of base kernels corresponding to different views and then combine those kernels using a linear or non-linear combination to improve the clustering performance [25, 81, 136, 199, 225].

Equally weighting kernels from different views can degrade the common clustering result due to the presence of low-quality views. Hence, several kernel weighting schemes have been proposed in the multi-view literature. For example, the algorithms proposed in [129, 225, 247] determine the distribution of weights based on some heuristic or model dependent hyper-parameters. Self-weight optimization schemes are proposed in [136, 199, 266] which automatically learn kernel weights without involving extra parameters. Cai and Li [25] used Hilbert Schmidt independence criterion to measure the agreement between a pair of kernels and obtained consensus kernels by agreement maximization. To filter redundant information from views, Yao *et al.* [265] proposed to select a diverse subset of representative kernels from a pre-specified set of kernels corresponding to different views. Trivedi *et al.* [223] used Kernel CCA, while Zhang et al. [279] extended fuzzy c-means algorithm [13] to address the problem of multi-kernel clustering in the presence of incomplete views.

A challenge in multi-kernel learning is obtaining the appropriate choice of kernel function (for example, linear kernel, polynomial kernel, or Gaussian kernel), which maps the input feature space to a high-dimensional Hilbert space.

2.2.6 Statistical Model Based Approaches

Statistical approaches [115, 156, 192, 243] aim to model the probability distribution of the data. These approaches usually assume that the observed data is generated from a mixture of distributions and use expectation maximization [161] to estimate the parameters of the distribution by maximizing the likelihood of the observed data. The statistical approaches have the advantage that it allows incorporating prior knowledge regarding the views while modelling the distribution functions. This can be done using Baysian priors or by specifying the choice of the distribution. Among the statistical approaches, the iCluster algorithm [192] assumes that the views are generated from a misture of Gausssian, MDI [115] assumes a Dirichlet mixture model, while LRAcluster [243] and iCluster+ [156] algorithms allow modeling different views with a different distribution (like, Gaussian distribution for real data, Possion distribution for integer count data).

The other advantage of the statistical approaches is that they can model the uncertainity in the data and make 'soft' probabilistic decisions, like probability of a sample belonging to a cluster. Zhuang *et al.* [297] used probabilistic latent semantic analysis to model the co-occurrence of samples and features in different views and determined cluster assignment based on conditional probability of the samples belonging to different clusters. In spite of the advantages, in most of the statistical formulations, the parameter estimation part turns out to be computationally very expensive on high-dimensional real-life data sets. Moreover, the heuristics and assumptions made regarding the distribution of the data do not always conform with the diverse and noisy real-life data sets, resulting in poor model fitting.

2.2.7 Graph Based Approaches

Graph based models form the most common category of multi-view clustering algorithms [98, 164, 165, 213, 234, 236, 237, 246, 272, 273]. These methods typically take input graphs of all views and find a fused graph or a low-dimensional spectral embedding of the graphs, and then employ an additional clustering algorithm, like k-means or spectral clustering, to produce the final clusters. Figure 2.6 shows an example of clustering based on multi-view graph fusion. Consideration of a separate graph for each view inherently takes care of the heterogeneity within the views in terms unit, variance, and scale. However, the quality of

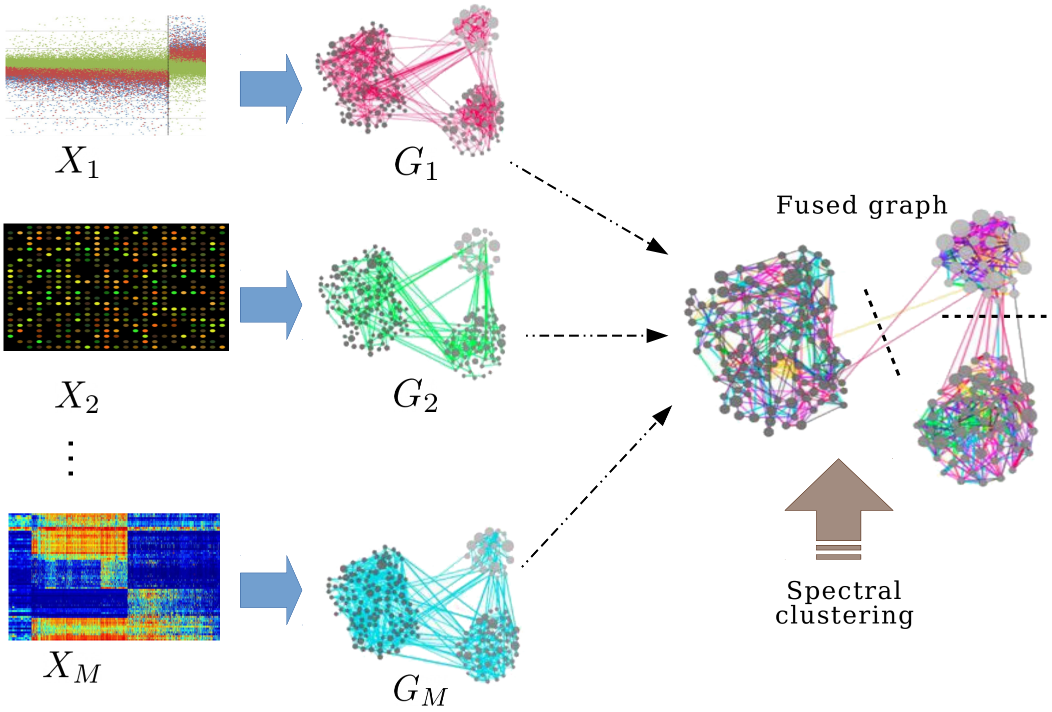


Figure 2.6: Graph based multi-view clustering.

the cluster structure reflected in the graphs vary from one view to another. To incorporate the differences in importance of different views, weighted multi-view graph clustering is proposed in [98, 164, 165]. These approaches first weight each input graph so that different views can have different impact on the unified representation. Several other advanced weight optimization schemes are proposed in [236, 272, 273]. In order to impose consistency between the clusterings reflected in different views, Nie *et al.* [166] proposed construction of a common nearest neighbor graph shared between all the views. The edge weights in the common graph have been assigned based on the similarities between the corresponding samples in all the views.

Most of the graph based approaches perform multi-view clustering on a set of fixed input graphs, and the results are dependent on the quality of input graphs. In this regard, Tao *et al.* [213] proposed an adaptive graph learning strategy where in addition to assigning the importance of the graphs from view level, sample-pair-specific weights are assigned within the views depending on the sample connection across different views. Another set of approaches is based on the hypothesis that each view has a consistent part shared between different views and an inconsistent part that does not appear in other views due to view-specific characteristic traits [19, 94, 132]. These approaches first separate the graph adacency matrices into consistent and inconsistent parts by orthogonality constraints, and then construct a unified matrix for clustering by fusing the consistent parts. Several hybrid approaches have also been proposed that impose graph regularization on NMF [24, 146], CCA [37, 38], tensor [40, 245], and self-representation subspace [267, 292] based multi-view clustering approaches.

One major drawback of the graph based approaches is that the graphs constructed from the inherently noisy real-life views may not be ideal. The noise and misleading edge weights in the individual graphs may propagate during the graph fusion process and distort the cluster structure of the unified graph [113].

2.2.8 Manifold Based Approaches

In several real-world data sets, the features are observed in a high-dimensional Euclidean space but the meaningful structures often lie on a low-dimensional manifold embedded within the input feature space [188]. Intuitively, in case of multi-view data, each view can be regarded as lying on a separate manifold, and the intrinsic structure of the whole data set can be treated as a mixture of manifolds. Based on this hypothesis, several multi-view algorithms have been proposed to identify clusters lying on lower dimensional possibly non-linear manifolds [24, 82, 108, 178, 269, 283, 295]. Cai *et al.* [24] and Zhang *et al.* [283] performed NMF based multi-view clustering with manifold regularization to preserve the local geometrical structures in each view. Zhou *et al.* [295] and Tao *et al.* [108] proposed extensions of this framework to incorporate sparsity and missing data prediction, respectively. Xie *et al.* [249] learns the local manifold structure of each view using Laplacian embedding [73,180] which preserves the neighborhood relationships between the high-dimensional points in the lower dimensional space as well.

In a separate line of approach, a prior assumption is made regarding the form or structure of the manifold and then optimization is performed on that specific manifold to identify the clusters. Yu *et al.* [269] assumed that the lower dimensional representation corresponding to different views belong to the Stiefel manifold [57], while the approaches proposed in [82, 178] assumed that the representations belong to the Grassmannian manifold [3, 57]. The algorithms then optimize and merge the representations on Stiefel or Grassmannian manifolds, using their manifold specific non-Euclidean distance measures. Most of these approaches use manifold induced norm regularizations, embeddings, and distance measures to capture the manifold structure, but they perform optimization over the Euclidean space [24, 178, 249, 283]. The indirect use of manifold may fail to capture the true structure of the underlying manifold. A few approaches [269] which optimize directly on the manifolds can better exploit their inherent structure and properties. However, this optimization is computationally more expensive compared to Euclidean optimization as general non-linear manifolds do not satisfy the vector space assumptions of the Euclidean space. The standard optimization algorithms like gradient descent, Newton's method, conjugate directions, etc., fail to work on non-linear manifolds unless generalized depending on the geometry of the manifold.

2.2.9 Deep Clustering Approaches

The classical machine learning approaches mostly use a shallow and linear (or linear approximation of a non-linear) embedding function to capture the intrinsic structures of multiple views in a lower dimensional subspaces. Inspired by the powerful and non-linear representation ability of deep neural networks [77, 104], recently several deep multi-view clustering approaches have been proposed [7, 33, 131, 211]. The deep clustering approaches are primarily based on CCA [7, 12, 35, 80], matrix factorization [33, 103, 260, 287], self-representation learning [1, 211, 238], and generative adverserial networks (GAN) [131, 239]. Andrew *et al.* [7] proposed DeepCCA, a deep neural network extension of CCA, which extracts non-linear feature embeddings corresponding to each view. The correlation between the feature embeddings are maximized using CCA in the last layer.

In the self-representation category of multi-view clustering, Abavisani and Patel [1] proposed a network consisting of a multi-view encoder, a self-expressive layer, and a multi-view decoder. The encoder constructs a latent representation of the multi-view data, the self-expressive layer enforces the self-representation property to construct a subspace preserving affinity matrix, and the decoder block minimizes the sample reconstruction loss. Spectral clustering on the affinity matrix learned from self-expressive layer generates the clusters. Gao *et al.* [72] proposed a hybrid network combining DeepCCA and the self-representation layer. Among the GAN based approaches, Li *et al.* [131] fused an autoencoder network with an adverserial network consisting of generator and discriminator components for multi-view clustering. The factorization based approaches proposed in [103, 260, 287] learn hierarchical semantics of multi-view data by performing matrix factorization like NMF in a layer-wise fashion. Clustering is performed on the representation learned at the final layer.

The deep multi-view models in general require massive amount of training data to learn the millions of weights and hyper-parameters. Also, in several approaches [1, 131, 238], the network architecture is heavily data dependent, which becomes hard to optimize from a million possible combinations of layers and activation functions.

2.3 Multi-View Classification Approaches

In the problem of multi-view classification, there are n training samples or instances, given by

$$\left\{\left(x_i^{(1)}, \ldots, x_i^{(m)}, \ldots, x_i^{(M)}, y_i\right)\right\}_{i=1}^{n},$$

where $y_i \in \mathcal{Y}$ (the label set) and $x_i^{(m)} \in \mathcal{X}^{(m)}$ (the domain or measurement space corresponding to the m-th view). Each training instance is a $(M+1)$ tuple sampled from an unknown underlying joint distribution over $\mathcal{X}^{(1)} \times \mathcal{X}^{(2)} \times \mathcal{X}^{(M)} \times \mathcal{Y}$. The aim is to find a function

$$f : \mathcal{X}^{(1)} \times \mathcal{X}^{(2)} \times \mathcal{X}^{(M)} \longrightarrow \mathcal{Y}$$

in a hypothesis space $\mathcal{F}$ that can predict the label associated with an unknown instance $x \in \mathcal{X}^{(1)} \times \mathcal{X}^{(2)} \times \mathcal{X}^{(M)}$ by $f(x)$.

The supervised multi-view learning approaches can be divided into three major categories: subspace learning, co-training, and multiple kernel learning. These three approaches are briefly outlined in next three subsections.

2.3.1 Subspace Learning Approaches

Subspace learning approaches aim to extract a common latent space from all the views to perform classification in that subspace. CCA, as discussed in Section 2.2.3.4, is a typical subspace learning approach that maximizes the correlation between pairs of projections from different views. To incorporate the label information in classification problems, Sun *et al.* [208] proposed discriminant CCA that considers inter-class and intra-class similarities of different views during subspace extraction. Elmadany *et al.* [58] optimized discriminant CCA using deep neural networks in order to obtain a nonlinear supervised dimensionality reduction model. Yang and Sun [263] proposed MLDA that combines LDA with CCA to ensure discriminative ability within a single view, while maximizing the correlation between different views. However, high correlation between the canonical vectors computed by MLDA implies that they contain redundant information. To address this, Sun *et al.* [206] further proposed MULDA that combines CCA with uncorrelated or orthogonal LDA. The algorithm proposed in [206] also generalized MULDA to the nonlinear case by replacing CCA with kernel CCA and kernel discriminant CCA. Benton *et al.* [11] incorporated the class information by treating the one-hot encoding matrix of the labels as an additional view. Mandal and Maji [149] integrated supervised information into CCA through the concept of rough hypercuboid. In this work, rough sets [170] are used to handle vagueness within the classes and extract features that maximize the relevance and significance with respect to the given class labels.

2.3.2 Co-Training Approaches

The co-training based approaches typically train separate classifier for each view, and then attempt to minimize the disagreement between the decision boundaries and prediction functions learned in different views by alternating training. The approaches proposed in [121, 196] use co-regularized least squares to perform a joint regularization over the views and minimize the disagreement between the view-specific prediction functions in a least squared sense. Guo and Xiao [83] proposed a subspace based co-regularized training for cross language text classification. Their algorithm jointly minimizes the training error of each classifier in each language while penalizing the distance between the subspace representations of parallel documents. All of these approaches learn a separate prediction function corresponding to each view. Sindhwani and Rosenberg [197] proposed to learn a single prediction function with a data-dependent co-regularization norm that reduces the problem to standard single-view classification problem.

2.3.3 Multi-View Support Vector Machines

Support vector machine is a kernelized classification algorithm that utilizes labeled samples to learn a decision boundary that maximizes the width of the gap between two classes [47]. Due to the popularity and effectiveness of SVMs in classification tasks, several multi-view extensions of SVM have been proposed [62, 79, 100, 124]. Farquhar *et al.* [62] proposed a SVM-2K model that performs two-view classification by combining kernel CCA and SVM into a single optimization model instead of a two-stage model (kernel CCA followed by SVM). Li *et al.* [124] proposed a two-view transductive SVM that utilizes multi-view features to improve performance of classifiers trained on individual views. Incremental multi-view SVM is proposed in [296], which integrates the views one after another in an incremental way instead of processing all views simultaneously. This incremental algorithm is shown to be scalable and is specifically applicable for scenarios with streaming views. Tang *et al.* [210] incorporated the 'learning using privileged information' paradigm [226, 227] into multi-view SVMs to target the complementary information of different views while training. Several co-regularization and co-training style multi-view SVMs have also been proposed. Typical approaches include multiview Laplacian SVMs [254], generalized eigenvalue proximal SVMs [205], sparse multi-view SVMs [204], multi-training SVMs [125], and manifold regularized multi-view vector valued SVMs [145].

Apart from these three major categories, multi-view classification has also been addressed using probabilistic model [207], multiple kernel learning [274], and deep learning [110, 187, 259] based approaches.

2.3.4 Conclusion

The advancement of information acquisition technologies like various sensors, medical and imaging devices, multimedia and networking platforms, and new feature extraction techniques has made multi-view data increasingly common in several real-world applications. The ubiquitous multi-view data has made multi-view learning an active area of research and several algorithms have thus been proposed to understand the natural structure of these data sets. However, these algorithms do not come without limitations. Most algorithms fail to give satisfactory performance in presence of noisy, redundant, and misleading views. However, such views are fairly common in real-life data sets. Furthermore, the algorithms are yet to harness the full non-linear geometry of multiple views and identify the best possible clustering of a data set. This gives a scope for improvement of the multi-view clustering literature by designing algorithms that can truly understand the non-linear dynamics of a multi-view system and be resilient to noisy and redundant views.

As mentioned in Section 1.2 of Chapter 1, one of the important challenges in multi-view data integration is the appropriate selection of relevant and complementary views over noisy and redundant ones. In this regard, the next chapter presents a novel algorithm for constructing a low-rank joint subspace of the multi-view data, taking into consideration the relevance or quality of the cluster structure embedded within each view and the redundancy or amount of cluster information shared between two views.

Chapter 3

Multivariate Normality Based Analysis for Low-Rank Joint Subspace Construction

3.1 Introduction

Advanced high-throughput technologies have expanded the breadth of available omics data from genome sequence data to transcriptomic, methylomic, and proteomic data. Each type of omics data reflects the biological variation at a specific molecular level. However, diseases like cancer involve complex interactions among biological components like genes, microRNAs, and proteins across multiple molecular levels. Therefore, integrative analysis of data from multiple omic modalities like gene expression, DNA methylation, etc, is likely to capture a more accurate picture of dynamic molecular systems. One major objective of integrative analysis is to understand the taxonomy of cancer by identifying the latent disease subtypes. Cancer subtyping provides deeper understanding of disease pathogenesis as well as helps in designing personalized treatments. Data driven subtype discovery is most popularly achieved by clustering data from one or more omic modalities. However, clustering multimodal or multi-view data sets has two major challenges. The main challenge is the selection of appropriate modalities or views, which can provide relevant and shared cluster information over noisy ones. Another challenge is to efficiently handle 'high-dimension low-sample size' nature of the data sets, which reduces the signal-to-noise ratio and makes clustering computationally expensive.

Separate clustering followed by manual integration is a frequently used approach to analyze multiple omics data sets for its simplicity. Cluster-of-cluster assignment (COCA) [93] and Bayesian consensus clustering (BCC) [140] are two such approaches, which first cluster each modality separately and the individual clustering solutions are then combined to get the final cluster assignments. However, the integration of separate clustering solutions fails to capture cross-platform correlations and shared joint structure. On the other hand, some of the direct integrative approaches, like super k-means [282], iCluster [192], iCluster+ [156], LRAcluster [243], joint and individual variance explained (JIVE) [141], and angle-based JIVE (A-JIVE) [63], proceed by concatenating the individual modalities to get the integrated data which is then used for clustering. As the naive concatenation of different modalities may degrade the signal-to-noise ratio of the data, most of the direct integrative approaches first extract a low-rank subspace representation of the high dimensional integrated data and then clustering is performed in the reduced subspace [141,156,192,243]. A brief survey of two-stage consensus and subspace based multimodal clustering approaches is provided in Sections 2.2.2 and 2.2.3, respectively, of Chapter 2.

An important parameter of the low-rank based approaches is the rank or dimension of the low-rank subspace to be extracted. The relation between the number of clusters k in a data set and rank r of the low-rank subspace has already been established in literature [52, 271]. The k centroids of k clusters in a d-dimensional input space lie in an affine subspace of dimension at most $(k-1)$ [69]; and when the data is well clustered, the affine subspace determined by the k centroids is parallel to the k principal components of the data [151]. Zha *et al.* [271] showed that the top k principal components are the continuous solutions to the discrete cluster membership indicators in the k-means clustering problem. However, given the indicators for the $(k-1)$ clusters, the cluster membership indicators for the k-th cluster can be retrieved. This indicates the presence of redundancy in the top k principal components. Consequently, Ding and He [52] showed that the continuous solution to the discrete k-means cluster indicators is given by the $(k-1)$ principal components. When the number of clusters in a data set is known, low-rank approaches like iCluster [192], sparse iCluster [193], iCluster2 (iCluster with variance-weighted shrinkage) [191], and iCluster+ [156] use the relation between the cluster subspace spanned by the k centroids and the $(k-1)$ principal components to estimate the required rank parameter. Other low-rank based approaches like LRAcluster [243] and JIVE [141] do not use the relation between the number of clusters and rank parameter. While LRAcluster uses a likelihood based index, JIVE uses permutation tests to estimate the rank of the reduced subspace. In general, the existing integrative clustering algorithms after estimating the rank, use all the available modalities to construct the final joint subspace. The relevance of the individual modalities as well as the amount of shared information contained within them are not considered explicitly for the selection of modalities. However, some of the omic modalities may provide only noisy information [278], which can degrade the underlying cluster structure.

In this regard, this chapter presents a novel algorithm, termed as NormS (Normality based Subspace), to extract a low-rank joint subspace of the integrated data from the low-rank subspaces of the individual modalities. The proposed algorithm uses Roystons's H-statistic [182] for multivariate normality to estimate the ranks of the individual subspaces. A normality based measure of relevance of an individual modality and a orthogonality based measure of shared information or dependency between two modalities are introduced in this work. The relevance measure gives a linear ordering of the modalities, indicating the quality of cluster information embedded within them, while the dependency measure is used to asses the overlap between the information provided by two modalities. The modalities with maximum relevance and shared cluster structure are used to construct the joint subspace. Furthermore, during integration of low-rank individual subspaces, intersection between the subspaces is considered to select the cluster information only and filter out the noise from each subspace. The performance of the clustering on the joint subspace extracted by the proposed method is studied and compared with the existing low-rank and consensus based approaches on several real-life multimodal omics data sets. The efficiency of rank estimation and appropriate modality selection, based on the proposed relevance and dependency measures, is established over existing approaches of naive integration of all the modalities. Finally, the identified clusters are shown to be robust and stable against perturbation of the data set. Some of the results of this chapter are reported in [111].

The rest of the chapter is organized as follows: Section 3.2 describes the proposed multivariate normality based approach for multi-view data clustering. It also introduces two quantitative measures proposed to evaluate the quality of different modalities. Experimental results on different multimodal cancer data sets and comparative performance analysis with existing approaches are presented in Section 3.3. Finally, Section 3.4 concludes the chapter.

3.2 NormS: Proposed Method

This section presents a new algorithm, based on multivariate normality, to construct the joint subspace of the integrated data from the low-rank subspaces of individual modalities. A multimodal or multi-view data set consists of $M \geqslant 2$ different sets of observations corresponding to the same set of n samples. Let M different modalities or views be given by $X_1, \ldots, X_m, \ldots, X_M$, where each $X_m \in \Re^{n \times d_m}$ and d_m is the number of features in X_m. The proposed algorithm assumes a signal-plus-noise model of the data [262], where each X_m can be decomposed as

$$X_m = \Xi_m + Z_m, \tag{3.1}$$

where Ξ_m is the signal component and Z_m is the noise component consisting of independent error terms. Z_m is assumed to follow $\mathcal{N}(0, \Lambda)$ distribution, where $\Lambda = diag(\sigma_1^2, \ldots, \sigma_{d_m}^2)$ and $\sigma_{d_i}^2$ is the variance of noise along the i-th feature. The signal component Ξ_m represents the inherent structure of the data. For a data set having embedded cluster structure, the signal component Ξ_m is considered to be a mixture of Gaussian. If the rank of the latent Ξ_m matrix is r_m, then the r_m-dimensional principal subspace of X_m represents the structural information embedded in Ξ_m. The r_m-dimensional principal subspace of a modality X_m is a linear subspace of $\Re^n$ spanned by the first r_m left singular vectors of X_m or the first r_m eigenvectors of $X_m X_m^T$ as basis. The principal subspace has the advantage of explaining the maximal possible variance of the data using r_m components.

3.2.1 Principal Subspace Model

The principal subspace of modality X_m is generally extracted using SVD of the mean centered X_m since $n << d_m$. Let $\mu(X_m) \in \Re^{d_m}$ be mean of X_m and $\mathbf{1}$ be a column vector of length n of all ones. Let the SVD of X_m be given by

$$X_m - \mathbf{1}\mu(X_m)^T = U(X_m)\Sigma(X_m)V(X_m)^T, \tag{3.2}$$

where $U(X_m)$ and $V(X_m)$, in their columns, contain the left and right singular vectors, respectively, and $\Sigma(X_m)$ is a diagonal matrix of corresponding singular values arranged in decreasing order. The principal components of X_m are obtained by scaling the projections in the columns of $U(X_m)$ by the corresponding spread values in $\Sigma(X_m)$, given by

$$Y(X_m) = U(X_m)\Sigma(X_m). \tag{3.3}$$

Therefore, the r_m-dimensional principal subspace representation of X_m is given by the two-tuple:

$$\Psi(X_m) = \langle U(X_m), \Sigma(X_m)\rangle, \tag{3.4}$$

where $U(X_m)$ is truncated to store the top r_m left singular vectors and $\Sigma(X_m)$ contains the corresponding r_m largest singular values.

3.2.2 Rank Estimation of Individual Modality

The proposed algorithm assumes that the data in a modality X_m is generated from a mixture of Gaussian. It uses a statistical hypothesis test to estimate the rank r_m of its principal subspace. The estimation of r_m proceeds as follows: for each possible value of $r = 1, 2, 3, ...$, it is tested whether the r-dimensional principal subspace encodes better cluster structure compared to the $(r - 1)$-dimensional subspace. Under the assumption of normally distributed noise, that is $Z_m \sim \mathcal{N}(0, \Lambda)$, a subspace of dimension r would be normally distributed only if it does not reflect cluster structure. On the other hand, if the r-dimensional subspace encodes better cluster structure compared to the $(r-1)$-dimensional subspace, then the r-dimensional subspace would deviate more from normality as compared to the $(r-1)$-dimensional subspace. However, once all the meaningful variations due to the clusters are summarized in the principal subspace of dimension r, the remaining variation can be attributed to the normally distributed Gaussian noise of the Z_m component, which gets reflected in the subspace of dimension $(r + 1)$. In this case, the $(r + 1)$-dimensional subspace has higher normality compared to the r-dimensional subspace, and the rank r_m of modality X_m is considered to be r. In this regard, it should be noted that Hamerly and Elkan [87] proposed to use the normality test for estimating the number of clusters in a data set.

In the proposed algorithm, normality of a subspace is tested using Royston's multivariate normality test [182]. It is an extension of the Shapiro-Wilk's test [189] of univariate normality, which has been shown to be the most powerful normality test for all types of distributions and sample sizes [153, 176]. Moreover, the H-statistic of Royston's normality test is found to have good power properties [150] against many alternative distributions. For a certain value of rank r, the first r left singular vectors in $U(X_m)$ span the r-dimensional principal subspace of X_m. Let the r left singular vectors be given by $U(X_m) = [U^1, \ldots, U^i, \ldots, U^r]$, where $U(X_m) \in \Re^{n\times r}$. Let the univariate data corresponding to the i-th left singular vector be $U^i = (u^i_1, \ldots, u^i_j, \ldots, u^i_n)^T$. The Shapiro-Wilk W-statistic for univariate normality computes the correlation between the order statistics of the given data and the expected standard normal order statistics. It has the following form:

$$W^i = \left(\sum_{j=1}^{n} a_j u^i_{(j)}\right)^2 \Bigg/ \left(\sum_{j=1}^{n} \left(u^i_j - \bar{u}\right)^2\right); \tag{3.5}$$

where $u^i_{(1)}, \ldots, u^i_{(n)}$ are the order statistic of $u^i_1, \ldots, u^i_n$ and $\bar{u}$ is the mean of U^i. The a_j's are given by [189]:

$$(a_1, \ldots, a_n) = \left(f^T V^{-1}\right)\left(f^T V^{-1} V^{-1} f\right)^{-1/2}, \tag{3.6}$$

where $f = (f_1, \ldots, f_n)^T$. The f_j's are the expected values of the order statistic of independent and identically distributed random variables sampled from the standard normal distribution and V is the covariance matrix of those order statistics. The value of W-statistic lies in $(0, 1]$ and a value close to 1 suggests a good fit to normality.

Royston [181] showed that W^i could be transformed to an approximately standard normal variate T^i, using the following transformation:

$$T^i = \sigma^{-1}\left[\left(1 - W^i\right)^{\lambda} - \mu\right], \tag{3.7}$$

where λ, μ, and σ are the functions of n, calculated based on polynomial approximations given by [183]. Let $T^1, \ldots, T^i, \ldots, T^r$ be obtained using (3.7), where T^i is the transformation of the i-th component U^i of the principal subspace of modality X_m. Let R^i be defined by

$$R^i = \left\{\Phi^{-1}\left[\frac{1}{2}\Phi\left(-T^i\right)\right]\right\}^2, \; i = 1, \ldots, r, \tag{3.8}$$

where $\Phi(.)$ denotes the cumulative distribution function of the standard normal distribution. Since T^i is approximately standard normal, therefore, $R^i \sim \chi_1^2$ individually. Here χ_d^2 denotes the χ^2-distribution with d degrees of freedom. For any r-dimensional subspace, where the variables R^i's are not necessarily uncorrelated, Royston's H-statistic is given by

$$H_r = \frac{e}{r}\sum_{i=1}^{r} R^i. \tag{3.9}$$

The H_r-statistic follows approximately χ_e^2 distribution, where $e \leqslant r$ is called the *effective degrees of freedom* of the χ^2-distribution. The parameter e is estimated, using the method of moments as described by [182], as follows:

$$e = \frac{r}{1 + (r-1)\bar{c}}, \quad \text{where} \quad \bar{c} = \frac{1}{r^2 - r}\sum_{i \neq j}\sum c_{ij} \tag{3.10}$$

and c_{ij} is the correlation between variables R^i and R^j.

In order to estimate the rank of modality X_m, for each possible value of rank r, the difference between the normality of r- and $(r-1)$-dimensional subspaces is evaluated. Two alternative hypotheses are as follows :

$\mathcal{H}_0$: r-dimensional subspace does not deviate more from normality compared to the $(r-1)$-dimensional one.

$\mathcal{H}_1$: r-dimensional subspace deviates more from normality compared to the $(r-1)$-dimensional one.

The H_r-statistic in (3.9) measures the normality of the r-dimensional principal subspace of a modality X_m. The above hypothesis is tested using the following statistic:

$$\gamma_r = H_r - H_{r-1}, \tag{3.11}$$

where γ_r measures the difference between the normalities of the r- and $(r-1)$-dimensional principal subspaces. However, in a principal subspace, the left singular vectors are orthogonal to each other. So, the correlation $c_{ij} = 0$ in (3.10) for $i \neq j$. Therefore, for a principal subspace, the effective degrees of freedom e of the H_r-statistic is equal to the dimension r of the subspace, and $H_r \sim \chi^2_r$. Hence, the γ_r-statistic reduces to

$$\gamma_r = \sum_{i=1}^{r} R^i - \sum_{j=1}^{r-1} R^j = R^r \text{ and } \gamma_r \sim \chi^2_1. \tag{3.12}$$

The relation in (3.12) signifies that if the value of R^r corresponding to the r-th left singular vector itself shows deviation from normality, the r-dimensional subspace deviates more from normality compared to the $(r-1)$-dimensional subspace. This is similar to the use of univariate normality for identification of significant principal components as in [91]. Acceptance of null hypothesis $\mathcal{H}_0$ at rank r implies that the r-th left singular vector reflects the noise from the normally distributed noise component Z_m of modality X_m. On the other hand, failure to accept $\mathcal{H}_0$ implies that the r-th singular vector reflects structural variation from the mixture of Gaussian component Ξ_m representing the clusters. To estimate the rank r_m of the signal component Ξ_m, the hypothesis $\mathcal{H}_0$ is tested sequentially for each value of r starting from one. The minimum value of r for which $\mathcal{H}_0$ is accepted implies that the $(r-1)$-dimensional principal subspace has summarized all the meaningful variations due to the structural component Ξ_m, while the r-dimensional subspace additionally includes noisy variation from the Z_m component. Following this argument, for a given significance level α, the rank r_m is determined by the relation:

$$r_m = \min\{r : p_r \geqslant \alpha\} - 1, \tag{3.13}$$

where p_r is the p-value of hypothesis test $\mathcal{H}_0$ at rank r. This implies that the rank of a modality X_m is estimated to be the smallest integer r_m such that the (r_m+1)-th principal component follows a normal distribution. However, real-life omics data sets are often corrupted with high proportions of noise. Consequently, a principal component may abruptly follow a normal distribution depicting the high noise content, while being preceded and succeeded by components that deviate from normality. To make the rank estimation robust to such noisy artifacts, the rank is estimated to be the smallest integer r_m such that its two consecutive components $(r_m + 1)$ and $(r_m + 2)$ are normally distributed, indicating that r_m-th component is the last meaningful one depicting the clusters.

For a certain modality, if the first two components U^1 and U^2 are normally distributed, then the null hypothesis $\mathcal{H}_0$ gets accepted at rank 1. This implies that the rank of the signal component Ξ_m is 0 and the modality does not encode any relevant structural information other than the random Gaussian noise. This gives the advantage of automatically filtering out noisy modalities, where the underlying subtype structure is not reflected at all. For the remaining modalities with rank $r_m > 0$, the cluster structure, encoded by the modalities, varies from one modality to another. Some modalities may reflect compact and well separated clusters, while others may reflect poor cluster structure. Moreover, two modalities may either provide shared cluster information or completely disjoint noisy information. Thus, appropriate choice of modalities, providing relevant and shared cluster information, is expected to provide better cluster structure in the final low-rank subspace. The relevance and dependency measures, proposed in this work, to evaluate the quality of each modality are described next.

3.2.3 Relevance and Dependency Measures

Let r_m be the rank of a modality X_m estimated using the hypothesis testing as described in Section 3.2.2. The r_m-dimensional principal subspace of X_m consists of the top r_m left singular vectors of X_m in $U(X_m)$ and their corresponding singular values in $\Sigma(X_m)$. The left subspaces $U(X_m)$'s from different modalities have varying ranks and are not directly comparable. So, a uniform measure of relevance is introduced next to assess the quality of cluster information provided by different modalities with varying ranks. This measure is based on the distribution of the principal components and the spread of the data along them.

3.2.3.1 Relevance

Let H_{r_m} be the value of Royston's H-statistic for the r_m-dimensional principal subspace of X_m. It approximately follows χ^2-distribution with r_m degrees of freedom. However, the H-statistic values across different modalities are not comparable as the degrees of freedom of the χ^2-distribution vary for different modalities. Let $F^{-1}_{\chi^2}(p, df)$ be the inverse of the χ^2 cumulative distribution function with *df* degrees of freedom for the corresponding probability p. At the significance level of α, $F^{-1}_{\chi^2}((1-\alpha), r_m)$ gives the minimum value of H-statistic for which the multivariate normality assumption of the r_m-dimensional principal subspace of X_m can be rejected. Therefore, the difference between $H_{r_m}$and $F^{-1}_{\chi^2}((1-\alpha), r_m)$ evaluates how far the normality of the principal subspace of X_m is with respect to the minimum threshold for rejection of normality. Moreover, the singular values in $\Sigma(X_m)$ give the spread of the data in the principal subspace of X_m. Amongst different modalities, higher the value of spread, better is the separability of clusters reflected in its corresponding principal subspace. The relevance of a modality X_m is defined as the product of two factors:

$$\mathcal{R}_l(X_m) = \Phi(H_{r_m}, r_m, \alpha) \times \Theta_m, \tag{3.14}$$

$$\text{where } \Theta_m = tr(\Sigma(X_m)) \Big/ \sum_{j=1}^{M} tr(\Sigma(X_j)),$$

$$\Phi(H_{r_m}, r_m, \alpha) = \frac{1}{2}\left[1 + \frac{H_{r_m} - F^{-1}_{\chi^2}((1-\alpha), r_m)}{\max\{H_{r_m}, F^{-1}_{\chi^2}((1-\alpha), r_m)\}}\right],$$

and $tr(A)$ denote the trace of a matrix A. The first factor $\Phi(H_{r_m}, r_m, \alpha)$ evaluates the distribution of the data along the principal subspace of X_m, while the second factor Θ_m measures the fraction of variance/spread explained by the modality X_m out of the total variance explained by the principal subspaces of all modalities.

In $\Phi(H_{r_m}, r_m, \alpha)$, the H_{r_m} values are not comparable for different modalities X_m's. So, the difference between H_{r_m} and its own minimum threshold $F^{-1}_{\chi^2}((1-\alpha), r_m)$ is considered for significance analysis. This makes the H-statistic values H_{r_m}'s comparable across different modalities. This is illustrated in Figure 3.1. Let there be three modalities X_1, X_2, and X_3 with ranks r_1, r_2, and r_3, respectively. Without loss of generality, let $r_1 \neq r_2 \neq r_3$. For modality X_m, $m = 1, 2$, and 3, Royston's H-statistic H_{r_m} for the r_m-dimensional principal subspace of X_m follows χ^2 distribution with r_m degrees of freedom. Figure 3.1 shows the χ^2 distributions for three modalities. The shaded areas in three χ^2 curves show the regions where the H-statistic shows statistically significant deviation from multivariate normality. At a significance level of α, τ_m gives the minimum value for which the H-statistic of the respective $\chi^2_{r_m}$-distribution can be called statistically significant. Let us assume that δ_m be the difference between H_{r_m} and τ_m. A value of $\delta_m \geqslant 0$ implies that the corresponding H_{r_m} is statistically significant. Figure 3.1 shows that H_{r_1} and H_{r_2} are statistically significant, whereas H_{r_3} is not. Moreover, as $\delta_2 > \delta_1 > \delta_3$, $\Phi(H_{r_2}, r_2, \alpha) > \Phi(H_{r_1}, r_1, \alpha) > \Phi(H_{r_3}, r_3, \alpha)$. This implies that principal components of modality X_2 deviate further away from normality as compared to those of X_1, so X_2 has better cluster structure compared to X_1.

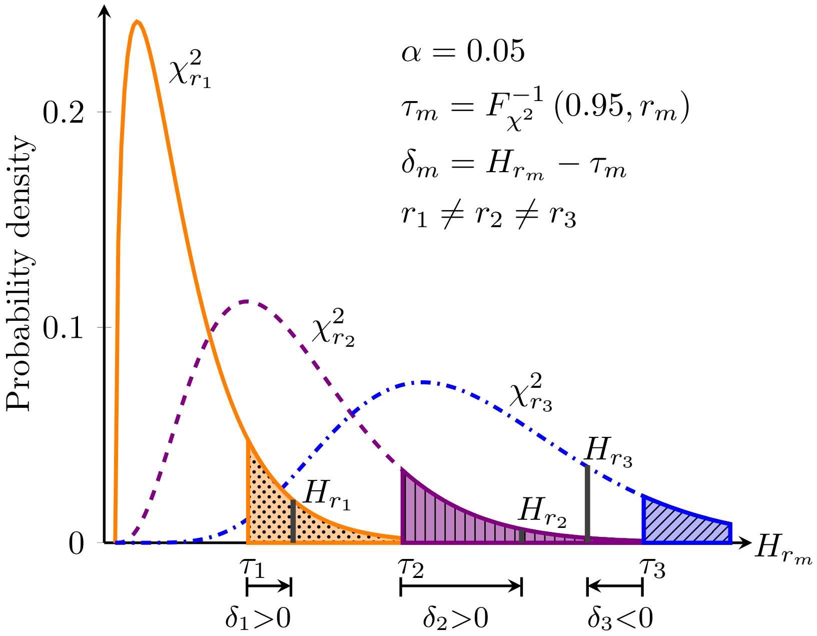


Figure 3.1: χ^2 distributions for H-statistic of three modalities.

The following properties can be stated about the relevance measure $\mathcal{R}_l(X_m)$:

1. $0 \leqslant \mathcal{R}_l(X_m) \leqslant 1$.
2. $\mathcal{R}_l(X_m) = 0$ if $H_{r_m} = 0$ or $tr(\Sigma(X_m)) = 0$.

For $H_{r_m} = 0$,

$$\Phi(0, r_m, \alpha) = \frac{1}{2}\left[1 + \frac{0 - F^{-1}_{\chi^2}\left((1-\alpha), r_m\right)}{\max\{0, F^{-1}_{\chi^2}\left((1-\alpha), r_m\right)\}}\right] = 0.$$

3. $\mathcal{R}_l(X_m) \rightarrow 1$ when $H_{r_m} \rightarrow \infty$ and $tr(\Sigma(X_m)) \rightarrow \sum\limits_{j=1}^{M} tr(\Sigma(X_j))$.

   This is because

$$\begin{aligned}\lim_{H_{r_m}\rightarrow\infty} \Phi(H_{r_m}, r_m, \alpha) &= \lim_{H_{r_m}\rightarrow\infty} \frac{1}{2}\left[1 + \frac{H_{r_m} - F^{-1}_{\chi^2}\left((1-\alpha), r_m\right)}{\max\{H_{r_m}, F^{-1}_{\chi^2}\left((1-\alpha), r_m\right)\}}\right] \\ &= \frac{1}{2}\left[1 + \lim_{H_{r_m}\rightarrow\infty} \frac{H_{r_m}\left(1 - F^{-1}_{\chi^2}\left((1-\alpha), r_m\right)/H_{r_m}\right)}{H_{r_m}}\right] \\ &= \frac{1}{2}\left[2 - \lim_{H_{r_m}\rightarrow\infty} \frac{F^{-1}_{\chi^2}\left((1-\alpha), r_m\right)}{H_{r_m}}\right] = 1,\end{aligned}$$

$$\text{and} \qquad \lim_{tr(\Sigma(X_m))\rightarrow \sum\limits_{j=1}^{M} tr(\Sigma(X_j))} \Theta_m = 1.$$

When H_{r_m} equals to $F^{-1}_{\chi^2}\left((1-\alpha), r_m\right)$, the modality X_m is at the minimum threshold for rejecting the null hypothesis of normality and the value of $\Phi(H_{r_m}, r_m, \alpha)$ is 0.5. Therefore, the value of $\Phi(H_{r_m}, r_m, \alpha) < 0.5$ implies that the principal subspace of X_m has a multivariate normal distribution, which reflects the presence of only random Gaussian noise from the Z_m component. On the other hand, the value of $\Phi(H_{r_m}, r_m, \alpha) \geqslant 0.5$ indicates statistically significant deviation from multivariate normality and the presence of signal component Ξ_m. Hence, a modality X_m is considered irrelevant if $\Phi(H_{r_m}, r_m, \alpha) < 0.5$, and is not considered for joint subspace construction. On the other hand, $\Phi(H_{r_m}, r_m, \alpha) \geqslant 0.5$ implies that X_m has relevant cluster information. For a modality X_m, the first factor $\Phi(H_{r_m}, r_m, \alpha)$ tends to 1 when its H-statistic H_{r_m} tends to ∞, while the second factor Θ_m tends to 1 when only X_m has non-zero variance and all the other modalities have variance close to 0. Higher value of $\Phi(H_{r_m}, r_m, \alpha)$ implies further deviation from the normally distributed noise component, while higher value of Θ_m implies a larger fraction of explained variance. Taking both the factor together, a higher value of $\mathcal{R}_l$ implies better cluster information.

3.2.3.2 Dependency

Let X_i and X_j be two modalities with left subspaces $U(X_i)$ and $U(X_j)$, and ranks r_i and r_j, respectively. The dependency of X_j on X_i is measured by the proportion of the subspace $U(X_j)$ that can be spanned by the subspace $U(X_i)$. This is obtained by projecting the singular vectors in $U(X_j)$ onto the column space of $U(X_i)$. As $U(X_i)$ is orthogonal, the projection matrix onto the column space of $U(X_i)$ is given by $U(X_i)U(X_i)^T$. Using this projection matrix, the projection of $U(X_j)$ onto the subspace $U(X_i)$ is given by

$$\mathcal{P} = U(X_i)U(X_i)^T U(X_j) \tag{3.15}$$

The proportion of $U(X_j)$ that can be spanned by $U(X_i)$ is given by the ratio of norm of projection $\mathcal{P}$ to the norm of $U(X_j)$ itself. So, dependency of modality X_j on X_i is given by

$$\mathcal{D}(X_j|X_i) = \frac{||\mathcal{P}||_F^2}{||U(X_j)||_F^2} \tag{3.16}$$

where $||A||_F^2$ is the squared Frobenius norm of the matrix A. Some properties of the dependency measure can be stated as follows:

1. $0 \leqslant \mathcal{D}(X_j|X_i) \leqslant 1$.
2. If $U(X_i)$ and $U(X_j)$ are orthogonal to each other, then the projection $\mathcal{P} = 0$ and dependency $\mathcal{D}(X_j|X_i) = 0$ (Figure 3.2(a)).
3. If all the left singular vectors in $U(X_j)$ are linear combinations of those in $U(X_i)$, then $\mathcal{P} = U(X_j)$ and dependency $\mathcal{D}(X_j|X_i) = 1$ (Figure 3.2(b)).
4. $\mathcal{D}(X_j|X_i) \neq \mathcal{D}(X_i|X_j)$ (asymmetric).

Dependency $\mathcal{D}$, thus, measures the amount of shared information present in modality X_j, given the information in modality X_i. The possible cases of dependency of a modality X_j on a modality X_i are depicted in Figure 3.2. In Figure 3.2(a), the subspace $U(X_j)$ is orthogonal to $U(X_i)$, so its dependency on $U(X_i)$ is 0. On the other hand, if $U(X_j)$ is linearly dependent on $U(X_i)$ as in Figure 3.2(b), its dependency on $U(X_i)$ is 1. For any other arbitrary orientation of two subspaces (Figure 3.2(c)), dependency $\mathcal{D}(X_j|X_i)$ lies in between 0 and 1.

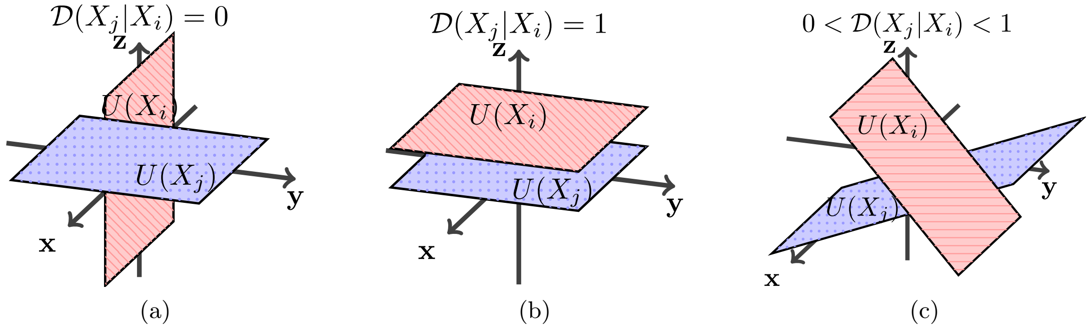

$\mathcal{D}(X_j|X_i) = 0$
z
$U(X_i)$
$U(X_j)$
y
x
(a)
$\mathcal{D}(X_j|X_i) = 1$
z
$U(X_i)$
$U(X_j)$
y
x
(b)
$0 < \mathcal{D}(X_j|X_i) < 1$
z
$U(X_i)$
$U(X_j)$
y
x
(c)

Figure 3.2: Dependency of modality X_j on X_i: (a) Orthogonal subspaces (b) Linearly dependent subspaces (c) Arbitrary subspaces.

3.2.4 Proposed Algorithm

The proposed algorithm is described next for the construction of joint subspace from the principal subspaces of individual modalities. For each X_m, its rank r_m is estimated. A modality X_m with rank $r_m = 0$ consists of only the noisy component Z_m and gets automatically filtered out at the first stage. The relevance $\mathcal{R}_l(X_m)$ is computed according to (3.14) for each modality X_m having rank $r_m > 0$. Let

$$\Psi(\mathbf{X}_m) = \langle U(\mathbf{X}_m), \Sigma(\mathbf{X}_m) \rangle \tag{3.17}$$

denote the joint subspace obtained at step m of the proposed algorithm. The process of joint subspace construction is initiated from the modality X_π having maximum relevance value. Thus, at step 1, the initial joint subspace is given by

$$\Psi(\mathbf{X}_1) = \Psi(X_\pi) = \langle U(X_\pi), \Sigma(X_\pi) \rangle. \tag{3.18}$$

At step $(m+1)$, each remaining modality X_j may have some shared cluster information with respect to the current joint subspace $\Psi(\mathbf{X}_m)$. To assess the amount of shared information, the dependency $\mathcal{D}(\mathbf{X}_m|X_j)$ of X_j on the current joint subspace $U(\mathbf{X}_m)$ is computed for each of the remaining modalities X_j's. Higher the value of dependency, stronger is the presence of shared structure in that modality. So, the modality X_ω having maximum dependency on the current subspace $U(\mathbf{X}_m)$ is chosen for integration. At step $(m + 1)$, the principal subspace $\Psi(X_\omega)$ is integrated with the joint subspace $\Psi(\mathbf{X}_m)$ obtained at step m.

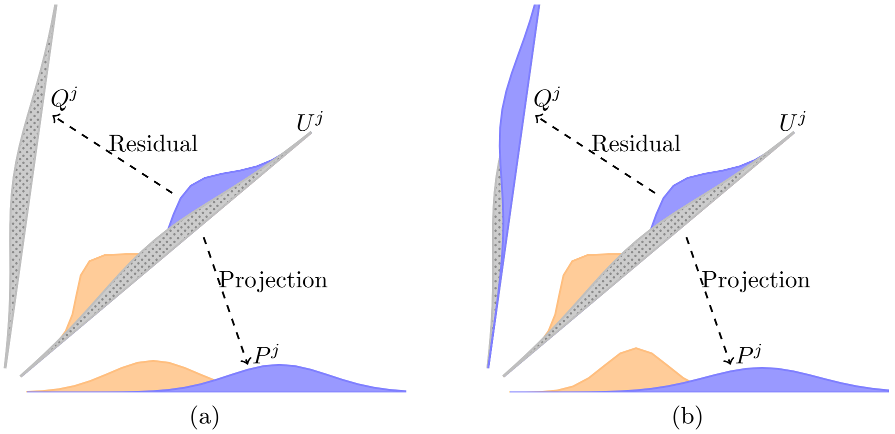


Figure 3.3: Two different cases of residual component Q^j after the projection of U^j on the current joint subspace: (a) Residual follows normal distribution (b) Residual shows divergence from normal distribution.

Both $U(\mathbf{X}_m)$ and $U(X_\omega)$ are subspaces of $\Re^n$ with their column vectors as their basis. The intersection between these two subspaces reflects common structures encoded by both the subspaces. Intersection is computed by projecting the columns of $U(X_\omega)$ onto the basis spanned by the columns of $U(\mathbf{X}_m)$, and is given by

$$\mathcal{I} = U(\mathbf{X}_m)^T U(X_\omega). \tag{3.19}$$

The projection P of $U(X_\omega)$, lying in the subspace $U(\mathbf{X}_m)$, is obtained by the product of the basis $U(\mathbf{X}_m)$ and the projection magnitudes in $\mathcal{I}$, and is given by

$$P = U(\mathbf{X}_m)\mathcal{I}. \tag{3.20}$$

The residual Q of $U(X_\omega)$ is obtained by subtracting the projection P from $U(X_\omega)$ itself, which is given by

$$Q = U(X_\omega) - P. \tag{3.21}$$

The projection P reflects the shared structure, while residual Q contains the extra information of X_ω. Let the columns of Q be given by $Q = [Q^1, \ldots, Q^j, \ldots, Q^{r_\omega}]$. If a residual vector Q^j contains cluster information, then the data in Q^j shows divergence from the normality. But, if Q^j contains only noisy information, then it should be normally distributed. The main idea is to incorporate only remaining cluster information of modality X_ω. So, the singular vectors of $U(X_\omega)$, whose residuals show significant divergence from normality, are considered for the construction of joint subspace $U(\mathbf{X}_{m+1})$. This is given by the set

$$S = \{U^j : p_{Q^j} < \alpha\}, \tag{3.22}$$

where U^j denotes the j-th column of $U(X_\omega)$ and p_{Q^j} denotes the p-value corresponding to the Shapiro-Wilk normality test on the residual column Q^j of Q. Therefore, for each component $U^j \in U(X_\omega)$, there are two possible cases: either its residual component Q^j is normally distributed (Figure 3.3(a)), or the residual shows significant divergence from the normality (Figure 3.3(b)). These two cases are illustrated in Figure 3.3. In the figure, let component U_j consists of two clusters and the noise. In Figure 3.3(a), the projected component P^j of U^j reflects both the clusters, and the residual component Q^j depicts only the noise. Thus, Q^j is normally distributed. On the other hand, in Figure 3.3(b), the residual Q^j depicts some cluster information along with noise. So, Q^j shows divergence from the normality. The set S is formed using only those U^j's having cluster information in their residuals. Finally, the components of the joint subspace $\Psi(\mathbf{X}_{m+1})$ at step $(m+1)$ are obtained as follows:

$$U(\mathbf{X}_{m+1}) = \begin{bmatrix} U(\mathbf{X}_m) & S \end{bmatrix} \quad \text{and} \tag{3.23}$$

$$\Sigma(\mathbf{X}_{m+1}) = diag\left(\Sigma(\mathbf{X}_m), \Sigma(S)\right), \tag{3.24}$$

where $U(\mathbf{X}_{m+1})$ is formed by column-wise concatenation of $U(\mathbf{X}_m)$ and vectors of S, and $\Sigma(S)$ is the diagonal matrix of singular values corresponding to vectors in S.

For $\mathcal{M}$ modalities having rank greater than 0, the final joint subspace $\Psi(\mathbf{X}_\mathcal{M})$ is obtained in $\mathcal{M}$ steps using the above procedure. The principal components $Y(\mathbf{X}_\mathcal{M})$ are then obtained from $\Psi(\mathbf{X}_\mathcal{M})$ using (3.3). Finally, k-means clustering is performed on the rows of $Y(\mathbf{X}_\mathcal{M})$ to get the sample clusters or the cancer subtypes. The proposed algorithm to extract a joint subspace of a multimodal data set is given in Algorithm 3.1.

3.2.4.1 Computational Complexity of Proposed Algorithm

The proposed algorithm begins by performing SVD on each of the modalities to extract its left subspace and singular values. For a single modality, this complexity is bounded by

Algorithm 3.1 Proposed Algorithm: **NormS**

```
Input: $X_1, \dots, X_m, \dots, X_M$, $X_m \in \Re^{n \times d_m}$
Output: Joint subspace $\Psi(\mathbf{X})$
 1: for $m \leftarrow 1$ to $M$ do
 2:     Estimate rank $r_m$ of each modality $X_m$ using hypothesis test.
 3:     Compute the principal subspace $\Psi(X_m)$ of rank $r_m$ using (3.4).
 4:     Compute relevance $\mathcal{R}_l(X_m)$ for each modality $X_m$ using (3.14).
 5: end for
 6: Let $\Gamma$ be the set of $\mathcal{M} \leqslant M$ modalities having rank greater than 0.
 7: Find the modality $X_\pi$ having maximum relevance.
 8: Set $\Psi(\mathbf{X}_1) = \Psi(X_\pi)$
 9: Remove modality $X_\pi$ from $\Gamma$, that is, $\Gamma = \Gamma \backslash \{X_\pi\}$.
10: for $m \leftarrow 1$ to $(\mathcal{M} - 1)$ do
11:     Compute the dependency $\mathcal{D}(\mathbf{X}_m | X_j)$ of each of the remaining modalities
         $X_j \in \Gamma$ on the current joint subspace $\mathbf{X}_m$.
12:     Select $X_\omega$ having the maximum dependency or shared structure.
13:     Compute intersection $\mathcal{I}$, projection $P$, and residual $Q$ using (3.19), (3.20),
         and (3.21), respectively.
14:     Test the normality of each residual vector $Q^j \in Q$ using Shapiro-Wilk test.
15:     Compute the set $S$ of residuals using (3.22) which show significant divergence
         from normality.
16:     Update the current joint subspace as follows:
        $U(\mathbf{X}_{m+1}) = \begin{bmatrix} U(\mathbf{X}_m) & S \end{bmatrix}$
        $\Sigma(\mathbf{X}_{m+1}) = diag(\Sigma(\mathbf{X}_m), \Sigma(S))$
        $\Psi(\mathbf{X}_{m+1}) = \langle U(\mathbf{X}_{m+1}), \Sigma(\mathbf{X}_{m+1}) \rangle$
17:     Remove modality $X_\omega$ from $\Gamma$, that is, $\Gamma = \Gamma \backslash \{X_\omega\}$.
18: end for
19: Set $\Psi(\mathbf{X}) = \Psi(\mathbf{X}_{\mathcal{M}})$.
20: Return $\Psi(\mathbf{X})$.
```

$\mathcal{O}(n^2 d_{max})$, where d_{max} is the maximum number of features among the modalities. This is followed by the computation of rank. The rank is computed by consecutively performing normality test on the left singular vectors. Each normality test has a complexity of $\mathcal{O}(n \log n)$ attributed to the computation of order statistics in (3.5). For a modality, the complexity of rank estimation is bounded by $\mathcal{O}(r_{max} n \log n)$, where r_{max} is the maximum rank among the modalities. The computation of relevance $\mathcal{R}_l$ takes $\mathcal{O}(M)$ time. Therefore, for M modalities, the time complexity of individual subspace construction followed by rank and relevance estimation is bounded by $\mathcal{O}\left(M\left(n^2 d_{max} + r_{max} n \log n + M\right)\right) = \mathcal{O}(M n^2 d_{max})$, considering $M, r_{max} << n$. The subspaces can also be constructed parallelly for different modalities as the problems are independent of each other. Then, the most relevant modality is selected in $\mathcal{O}(M)$ time and the initial joint subspace is constructed in $\mathcal{O}(1)$ time.

At each step of the joint subspace construction (steps 10-18), the dependency of the remaining left subspaces on the current joint subspace is computed. This computation is upper bounded by $\mathcal{O}\left(M n^2 r_{max}\right)$. For the modality with maximum shared structure, the projection P and the residual Q are computed in $\mathcal{O}(n^2 r_{max})$ time. Evaluation of normality of the residuals is bounded by $\mathcal{O}\left(r_{max} n \log n\right)$. Depending on the residuals, the next selected modality is updated into joint subspace in $\mathcal{O}(1)$ time by concatenation. Hence, the time complexity of a single updation step of the algorithm is $\mathcal{O}\big(Mn^2 r_{max} + n^2 r_{max} + r_{max} n \log n + 1\big) = \mathcal{O}(Mn^2 r_{max})$. Finally, the overall complexity of the proposed algorithm is bounded by $\mathcal{O}\big(Mn^2 d_{max} + Mn^2 r_{max}\big) = \mathcal{O}(Mn^2 d_{max})$. This shows that the computational complexity of the proposed algorithm is dominated by the initial steps of individual subspace construction and rank estimation.

3.3 Experimental Results and Discussion

This section presents the clustering performance of the joint subspace extracted by the proposed algorithm and its comparison with the existing integrative clustering algorithms.

3.3.1 Data Sets and Experimental Setup

The multimodal omics data for four types of cancer, namely, cervical carcinoma (CESC), lower grade glioma (LGG), ovarian carcinoma (OV), and breast invasive carcinoma (BRCA), are obtained from The Cancer Genome Atlas (TCGA) (`https://cancergenome.nih.gov/`). By comprehensive integrated analysis, TCGA research network has identified three subtypes of CESC [218] and LGG [217], and four subtypes of OV [215] and BRCA [214]. These tumor subtypes have been shown to be clinically relevant and reveal new potential therapeutic targets for the cancer. The four data sets CESC, LGG, OV, and BRCA consist of 124, 267, 334, and 398 samples, respectively. All the data sets have four different omic modalities, namely, gene expression (RNA), DNA methylation (mDNA), microRNA expression (miRNA), and reverse phase protein array expression (RPPA). These four modalities are measured on different platforms and represent different biological information. The details of the data sets, their subtypes, and the pre-processing steps are described in the Appendix A.

The performance of the proposed method is compared with that of two existing consensus based approaches, namely, cluster of cluster analysis (COCA) [93], and Bayesian consensus clustering (BCC) [140], two statistical model based low-rank approaches, namely, LRAcluster [243], and iCluster [192], and three SVD based low-rank approaches, namely, PCA on concatenated data (PCA-con) [6], joint and individual variance explained (JIVE) [141] with rank estimation based on permutation tests (JIVE-Perm) and Bayesian information criteria (JIVE-BIC), and angle based JIVE, termed as A-JIVE [63]. The details of the experimental setup and parameter tuning used for the existing algorithms are specified in the supplementary material of [111]. The performance of different algorithms is evaluated by comparing their identified subtypes with the clinically established TCGA subtypes using six external cluster validity indices, namely, clustering accuracy, normalized mutual information (NMI), adjusted Rand index (ARI), F-measure, Rand index, and purity. The definitions of these external indices are provided in Appendix B. Experimental results corresponding to Jaccard and Dice coefficients are provided in [111]. To study the clinical implications of the identified subtypes, survival analysis is performed using Cox log-rank test [96] and Peto and Peto's modification of Gehan-Wilcoxon test [172]. These tests determine the statistical significance of differences in survival profiles of the identified subtypes. The source code of the proposed NormS algorithm, written in R language, is available at `https://github.com/Aparajita-K/NormS`.

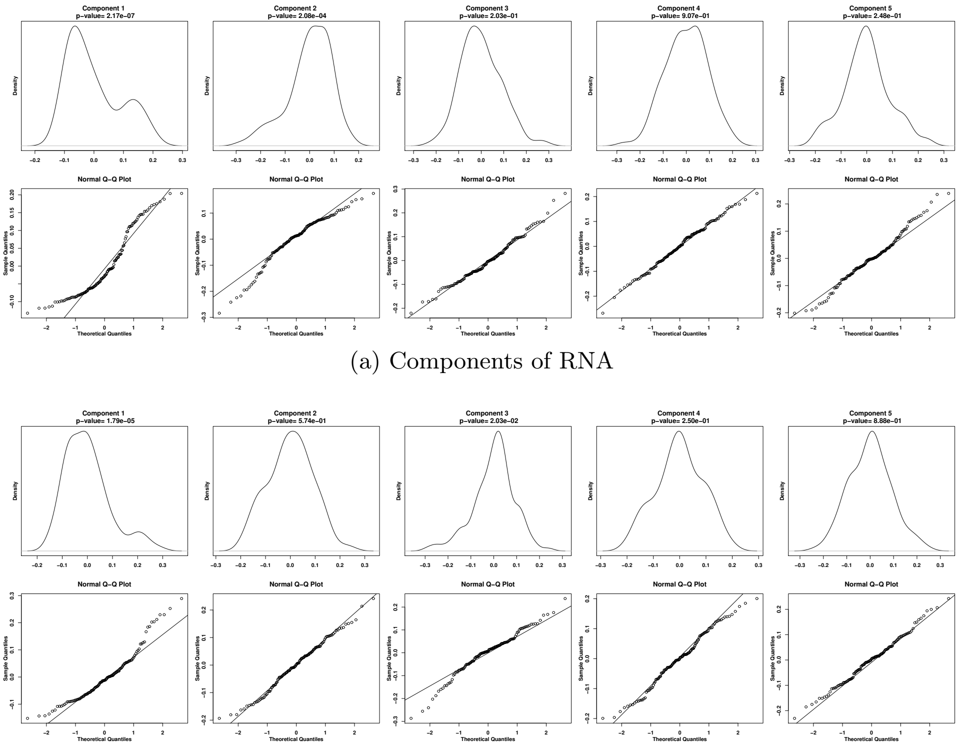
Table 3.1: Relevance and Rank of Each Modality and Modalities Selected by the Proposed Algorithm

Modality		Relevance	Rank	Selected		Relevance	Rank	Selected
mDNA	**CESC**	0.1884817	3	RNA,	**LGG**	0.4320317	10	
RNA		0.2921399	2	mDNA,		0.0289518	0	mDNA,
miRNA		0.1990886	5	miRNA,		0.0056958	0	RPPA
RPPA		0.2006048	4	RPPA		0.2428867	6	
mDNA	**OV**	0.0230986	0	RNA, miRNA, RPPA	**BRCA**	0.2373227	5	RNA,
RNA		0.4936741	3			0.2947759	3	mDNA,
miRNA		0.2474369	5			0.1602746	4	miRNA,
RPPA		0.0579902	2			0.2464338	6	RPPA

3.3.2 Illustration of Proposed Algorithm

The proposed algorithm uses multivariate normality to estimate the rank and relevance of the individual modalities. The rank and relevance of different modalities, as well as the modalities selected by the proposed algorithm on different data sets are reported in Table 3.1. Table 3.1 shows that the relevance and rank of the modalities vary among the data sets, and hence different subsets of modalities are selected for different data sets. A modality having zero rank indicates that its first two principal components are normally distributed, and the modality contains only the noise component. This automatically eliminates noisy modalities having zero rank and low relevance values (like RNA and miRNA modalities of LGG, and mDNA modality of OV) from integrating into the joint subspace. For CESC data set, initially all the modalities have non-zero ranks and all are considered for joint subspace construction. However, during integration, a majority of the residual components from different modalities turn out to be normal with respect to the existing joint subspace. Hence, they are not integrated into the final subspace, thus performing a second level of noise removal.

The working principle of the proposed algorithm is illustrated using the CESC data set as an example. Table 3.1 shows that for the CESC data set, the rank r of mDNA, RNA, miRNA, and RPPA are 3, 2, 5, and 4, respectively. Figures 3.4(a) and 3.4(b) show density plots, quantile-quantile (Q-Q) plots, and p-values for the first 5 principal components of RNA and mDNA modalities, respectively of CESC data set. These figures show that third, fourth, and fifth components of the RNA, and fourth and fifth components of mDNA are normally distributed, depicting the random Gaussian noise component of these modalities. On the other hand, the first two components of RNA in Figure 3.4(a) show deviation from normality, indicating the presence of clusters. For mDNA, Figure 3.4(b) shows that the second principal component abruptly follows a normal distribution, while both first and third components show deviation from normality. Additionally, the remaining components from 4 onwards are normally distributed. So, the rank of mDNA is estimated to be 3. The density plots in Figures 3.4(a) and 3.4(b) also show that the first component of both RNA



(a) Components of RNA

(b) Components of mDNA

Figure 3.4: Density and Q-Q plots for first five principal components of RNA and mDNA modalities of CESC data set.

and mDNA have a bimodal distribution, indicating multiple clusters. According to the relevance values in Table 3.1, four modalities of the CESC data set can be ordered as RNA followed by RPPA, miRNA, and mDNA. Therefore, the joint subspace construction begins with RNA. Although mDNA is the modality with lowest relevance, it has the maximum shared information with RNA, according to the dependency measure. So, mDNA is selected next for integration. Figure 3.5 shows the density and Q-Q plots of the residuals of mDNA with respect to the current joint subspace of RNA. The figure shows that the residuals of the first and second component of mDNA are normally distributed with p-values 0.284 and 0.246, respectively, while the third component deviates from normality (p-value is 0.0348). Therefore, only the third principal component of mDNA is integrated into the joint subspace.

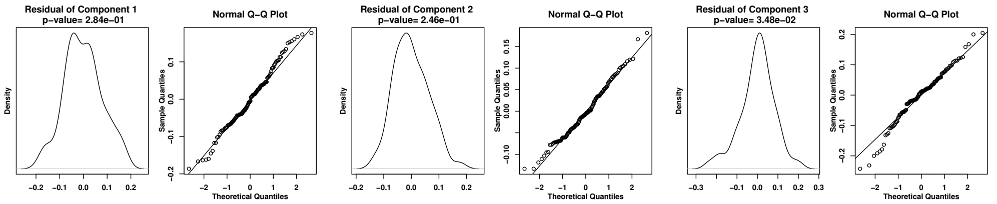
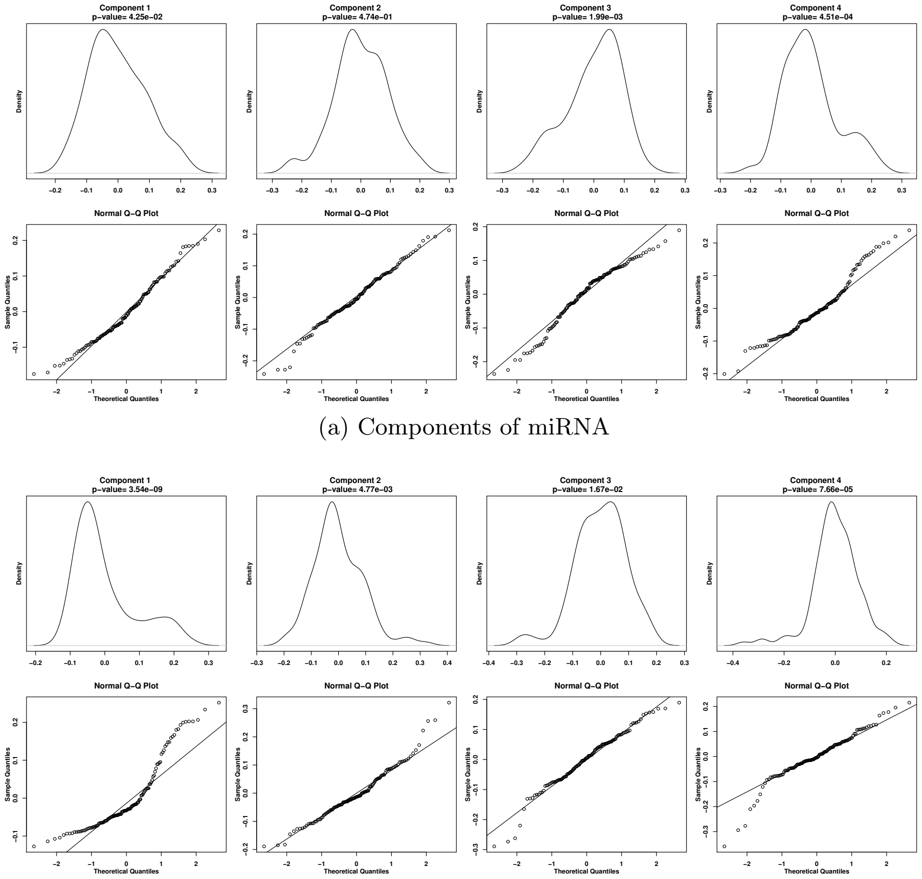


Figure 3.5: Density and Q-Q plots for the residual components of mDNA for CESC data.

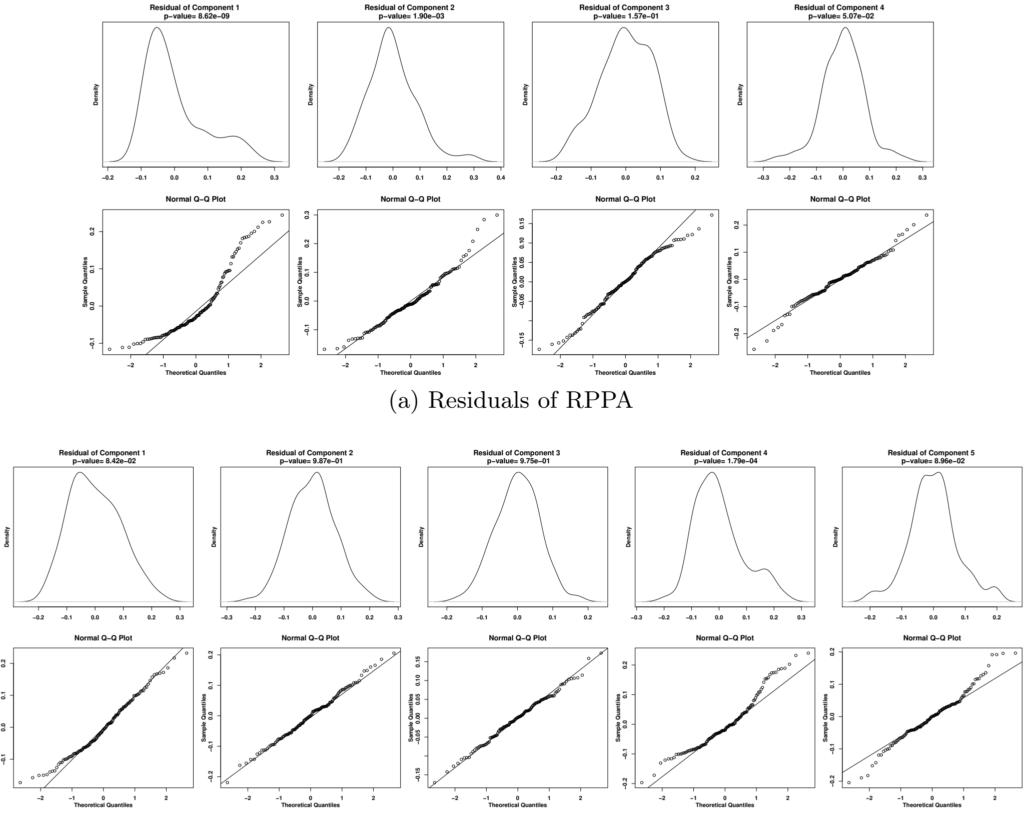


(a) Components of miRNA

(b) Components of RPPA

Figure 3.6: Density and Q-Q plots for first four principal components of miRNA and RPPA for CESC data set.

The modality selected next for integration is miRNA whose estimated rank is 5. The density and Q-Q plots for principal components of miRNA and their residuals with respect to the current joint subspace are given in Figures 3.6(a) and 3.7(b), respectively. The plots of the residuals in Figure 3.7(b) show that the residual of only the fourth principal component of miRNA shows significant divergence from normality and is selected for integration into the joint subspace. Finally, RPPA is selected for integration whose estimated rank is 4. The density and Q-Q plots for principal components of RPPA and their residuals are given in Figures 3.6(b) and 3.7(a), respectively. Figure 3.7(a) shows that out of the top four principal components of RPPA, the residuals of only the first and second components show deviation from normality. Thus only these two components of the RPPA modality are integrated into the joint subspace and the rest are eliminated as noisy ones, thus forming a six dimensional joint subspace for the CESC data set.



(a) Residuals of RPPA

(b) Residuals of miRNA

Figure 3.7: Density and Q-Q plots for the residual components the miRNA and RPPA for CESC data set.

3.3.3 Effectiveness of Proposed Algorithm

This subsection illustrates the importance of rank estimation, relevance and dependency measures introduced in this work. It also highlights the significance of selecting non-normal residuals only during data integration.

3.3.3.1 Importance of Relevance

The proposed relevance measure $\mathcal{R}_l(X_m)$ estimates the relevance of a modality X_m based on the distribution of its r_m principal components and the spread of the data along those components. The relevance measure provides an ordering of the modalities, and the process of integration starts with the most relevant one. To establish the importance of the proposed relevance measure and the ordering, the performance of clustering is studied for three different cases where the process of integration is initiated with the second, third, and fourth most relevant modalities, keeping all other components of the algorithm fixed. The starting modality for the three different cases and its comparative performance with the proposed algorithm are reported in Table 3.2 for different data sets. The rank is estimated to be 0 for both RNA and miRNA modalities of LGG and DNA methylation modality of OV. Hence, the comparative performance of starting with these three modalities is not

Table 3.2: Importance of Relevance

Data Set	Different Measures	2nd Most Relevant	3rd Most Relevant	4th Most Relevant	Proposed Algorithm
CESC	Starting Modality	RPPA	miRNA	mDNA	RNA
	Rank of Modality	4	5	3	2
	Relevance	0.2006048	0.1990886	0.1884817	**0.2921399**
	Accuracy	**0.8870968**	0.5403226	0.7016129	**0.8870968**
	NMI	**0.7085741**	0.1488846	0.2518516	0.6854921
	ARI	**0.7118294**	0.1282493	0.2987082	0.7004411
	F-measure	0.8781377	0.5677140	0.6852036	**0.8801172**
	Rand	**0.8644112**	0.5774980	0.6556517	0.8587726
	Purity	**0.8870968**	0.6129032	0.7016129	**0.8870968**
LGG	Starting Modality	RPPA	RNA	miRNA	mDNA
	Rank of Modality	6	0	0	10
	Relevance	0.2428867	0.0289518	0.0056958	**0.4320317**
	Accuracy	0.7228464	-	-	**0.7940075**
	NMI	0.4739360	-	-	**0.5325030**
	ARI	0.3557794	-	-	**0.4649223**
	F-measure	0.7236789	-	-	**0.7916535**
	Rand	0.6978401	-	-	**0.7465292**
	Purity	0.7228464	-	-	**0.7940075**
OV	Starting Modality	miRNA	RPPA	mDNA	RNA
	Rank of Modality	5	2	0	3
	Relevance	0.2474369	0.0579902	0.0230986	**0.4936741**
	Accuracy	0.5568862	0.5568862	-	**0.6976048**
	NMI	0.2717504	0.2717504	-	**0.4504552**
	ARI	0.2015805	0.2015805	-	**0.4142200**
	F-measure	0.5552212	0.5552212	-	**0.6910392**
	Rand	0.6949524	0.6949524	-	**0.7766269**
	Purity	0.5568862	0.5568862	-	**0.6976048**
BRCA	Starting Modality	RPPA	mDNA	miRNA	RNA
	Rank of Modality	6	5	4	3
	Relevance	0.2464338	0.2373227	0.1602746	**0.2947759**
	Accuracy	**0.7688442**	0.7160804	**0.7688442**	**0.7688442**
	NMI	**0.5540359**	0.4142157	0.5437267	0.5437267
	ARI	**0.5179618**	0.4007992	0.5090183	0.5090183
	F-measure	0.7692316	0.7228568	**0.7699789**	**0.7699789**
	Rand	**0.8033746**	0.7593636	0.7999063	0.7999063
	Purity	**0.7688442**	0.7185930	**0.7688442**	**0.7688442**

reported in Table 3.2. The results in Table 3.2 show that the proposed algorithm gives the better performance compared to the cases where integration begins with any modality other than the most relevant one, for both LGG and OV data sets. For BRCA data set, when integration is initiated with the most relevant modality, that is RNA, then miRNA is the least redundant one that is chosen at next step of integration. Vice-versa, when integration is initiated with miRNA, then RNA is the least redundant one that is selected next. So, for BRCA data set, the performance of the proposed algorithm is same as the case where integration begins with miRNA. For the CESC and BRCA data sets, the proposed algorithm gives best performance for F-measure compared to the other three cases. For Rand index, ARI, and NMI, the proposed algorithm gives the second best performance. However, the best performance for majority of the indices is obtained in the case where integration begins with the second most relevant modality, that is RPPA. In brief, the proposed method of integrating modalities based on their relevance ordering gives better performance compared to that of some arbitrary ordering in majority of the cases.

3.3.3.2 Importance of Rank Estimation

The proposed method uses normality tests to separately estimate the rank of each individual modality, independent of the number of clusters in the data set. The estimated ranks of the individual modalities are given in Table 3.2 for different data sets. Existing low-rank based approaches like iCluster [192], iCluster2 [191], iCluster+ [156], use the fixed relation between the number of clusters k, and the $(k-1)$ principal components [52], to determine the rank of their respective subspaces. To establish the importance of the proposed method of rank estimation, the performance of clustering in the subspace extracted by the proposed algorithm is compared with that of the subspace formed by concatenating $(k-1)$ principal components of each modality. The comparative results reported in Table 3.3 for all the data sets show that the proposed algorithm, with variable number of selected components from the individual modalities, gives better performance compared to the case where fixed $(k-1)$ components are selected from each modality, except for NMI and Rand index in BRCA data set. This shows that for real life data sets where the clusters are not well-separated, the strict relation between the number of clusters k and rank $(k-1)$ does not necessarily hold.

3.3.3.3 Significance of Dependency

At each iteration of the data integration, the proposed algorithm selects a modality that has maximum dependency or shared information with respect to the current joint subspace. To assess the significance of prioritizing modalities having maximum shared structure, all the modalities are naively integrated based on their relevance ordering, and the clustering performance of the resulting subspace is studied. The comparative performance of this relevance-based subspace (without dependency) and the proposed one is reported in Table 3.3. The results imply that for CESC and BRCA data sets, the proposed approach of considering dependency or shared structure along with relevance extracts better overall cluster information compared to relevance alone. For LGG and OV data sets however, both approaches have the same performance. This because, for LGG data set, apart from the most relevant modality, mDNA, RPPA is the only other modality with non-zero rank and is also the one with maximum shared structure. Similarly, for OV data set, mDNA is automatically eliminated due to its zero-rank. Amongst the remaining modalities, the ordering obtained considering relevance and dependency together is identical to the one obtained based on relevance alone. Hence, both approaches have the same performance for LGG and OV data sets. Thus, when a large number of modalities having non-zero ranks are available, considering relevance and dependency together during integration gives better performance compared to only relevance based integration.

Table 3.3: Importance of Rank Estimation, Dependency Measure, and Selection of Non-normal Residuals

Data Set	Different Measures	Fixed Rank ($k-1$)	Without Dependency	Taking All Residuals	Proposed Algorithm
CESC	Accuracy	0.8387097	0.8790323	0.8387097	**0.8870968**
	NMI	0.6579328	0.6707970	0.6579328	**0.6854921**
	ARI	0.6040195	0.6875915	0.6040195	**0.7004411**
	F-measure	0.8181978	0.8706213	0.8181978	**0.8801172**
	Rand	0.8080252	0.8526095	0.8080252	**0.8587726**
	Purity	0.8387097	0.8790323	0.8387097	**0.8870968**
LGG	Accuracy	0.6479401	**0.7940075**	0.7228464	**0.7940075**
	NMI	0.3181501	**0.5325030**	0.4739360	**0.5325030**
	ARI	0.2801154	**0.4649223**	0.3557794	**0.4649223**
	F-measure	0.6471171	**0.7916535**	0.7236789	**0.7916535**
	Rand	0.6512630	**0.7465292**	0.6978401	**0.7465292**
	Purity	0.6479401	**0.7940075**	0.7228464	**0.7940075**
OV	Accuracy	0.6916168	**0.6976048**	**0.6976048**	**0.6976048**
	NMI	0.4431124	**0.4504552**	**0.4504552**	**0.4504552**
	ARI	0.4089271	**0.4142200**	**0.4142200**	**0.4142200**
	F-measure	0.6834049	**0.6910392**	**0.6910392**	**0.6910392**
	Rand	0.7742893	**0.7766269**	**0.7766269**	**0.7766269**
	Purity	0.5568862	**0.6976048**	**0.6976048**	**0.6976048**
BRCA	Accuracy	0.7638191	0.7638191	0.7613065	**0.7688442**
	NMI	**0.5556963**	0.5231492	0.5517217	0.5437267
	ARI	0.5082782	0.4958426	0.5052503	**0.5090183**
	F-measure	0.7651760	0.7642758	0.7626970	**0.7699789**
	Rand	**0.8002101**	0.7928053	0.7989950	0.7999063
	Purity	0.7638191	0.7638191	0.7613065	**0.7688442**

3.3.3.4 Importance of Selecting Non-normal Residuals

Once the modality with maximum shared information gets selected, the proposed method examines the distribution of the residuals of the selected modality with respect to the current subspace. Out of all the components, only the components whose residuals depict the presence of cluster structure are integrated with the current subspace. The residuals following normal distribution depict noise and are eliminated from the integration. To establish the importance of integrating only non-normal residual components, a joint subspace is constructed where all the components of the selected modality are integrated irrespective of the distribution of their residuals. The comparative performance of this subspace and the proposed one, is studied and reported in Table 3.3. Comparative analysis in Table 3.3 show that for CESC, LGG, and BRCA data sets, selection of only non-normal residual components yields a lower-dimensional subspace with better cluster structure compared to the one which integrates all the residuals. For the OV data set, however, all the residuals at each of the proposed algorithm show deviance from normality, and hence all the residuals are selected for integration. Hence, for OV data set, the two subspaces are identical, giving the same performance. Thus, elimination of components having noisy residuals, used in the proposed method preserves better cluster structure in all the data sets.

3.3.4 Comparative Performance Analysis

This section compares the performance of the proposed algorithm with that of eight existing integrative clustering approaches, namely, COCA [93], BCC [140], LRAcluster [243], iCluster [192], JIVE-Perm and JIVE-BIC [141], A-JIVE [63], and PCA-con [6].

Table 3.4: Comparative Performance Analysis of Proposed and Existing Approaches

Data Set	Different Algorithms	Rank of Subspace	External Evaluation Index					
			Accuracy	NMI	ARI	F-measure	Rand	Purity
CESC	COCA	-	0.6693548	0.4172592	0.3677157	0.6870510	0.6971282	0.6774194
	BCC	-	0.6895161	0.2854917	0.3144526	0.6795619	0.6687779	0.6935484
	JIVE-Perm	24	0.7177419	0.4425848	0.3860367	0.7097880	0.7164962	0.7177419
	JIVE-BIC	4	0.8064516	0.5296325	0.5229385	0.8011385	0.7791765	0.8064516
	A-JIVE	48	0.6500000	0.3700238	0.3355826	0.6511586	0.6857724	0.6814516
	iCluster	2	0.5483871	0.1737526	0.1017765	0.5568753	0.5731707	0.5645161
	LRAcluster	1	0.8145161	0.5176602	0.5384740	0.8123256	0.7867821	0.8145161
	PCA-con	3	0.8548387	0.6750978	0.6333073	0.8390298	0.8237608	0.8548387
	NormS	6	**0.8870968**	**0.6854921**	**0.7004411**	**0.8801172**	**0.8587726**	**0.8870968**
LGG	COCA	-	0.6591760	0.2772248	0.2533847	0.6608123	0.6454901	0.6591760
	BCC	-	0.6340824	0.2737596	0.248606	0.63111660	0.6382755	0.6355805
	JIVE-Perm	8	0.5617978	0.2299551	0.1606599	0.5757978	0.6056715	0.5730337
	JIVE-BIC	8	0.6741573	0.3441747	0.3050874	0.6679019	0.6642730	0.6741573
	A-JIVE	48	0.7168539	0.4267241	0.3376560	0.7172792	0.6869055	0.7168539
	iCluster	2	0.4382022	0.1379678	0.0996867	0.5187438	0.5821858	0.5355805
	LRAcluster	2	0.4719101	0.1240057	0.1030798	0.5137382	0.5831714	0.5280899
	PCA-con	3	0.6666667	0.3438738	0.3031312	0.6574834	0.6616823	0.6666667
	NormS	14	**0.7940075**	**0.5325030**	**0.4649223**	**0.7916535**	**0.7465292**	**0.7940075**
OV	COCA	-	0.5943114	0.3131466	0.2810761	0.6068513	0.7039183	0.5943114
	BCC	-	0.4610778	0.1567582	0.1254690	0.4755846	0.6268706	0.4622754
	JIVE-Perm	32	0.5718563	0.2629523	0.2027605	0.5653910	0.6885005	0.5718563
	A-JIVE	64	0.5191617	0.2124862	0.1981556	0.5111353	0.6942997	0.5221557
	iCluster	3	0.5089820	0.2249889	0.2005886	0.4808256	0.6916078	0.5119760
	LRAcluster	2	0.6287425	0.3745173	0.2999204	0.6384046	0.7322472	0.6287425
	PCA-con	4	0.6946108	0.4424701	0.4068449	0.6868295	0.7734621	0.6946108
	Proposed	10	**0.6976048**	**0.4504552**	**0.4142200**	**0.6910392**	**0.7766269**	**0.6976048**
BRCA	COCA	-	0.7434673	0.5002408	0.4864778	0.7457304	0.7905295	0.7434673
	BCC	-	0.6251256	0.3169187	0.3049874	0.6242493	0.7055783	0.6334171
	JIVE-Perm	12	0.6859296	0.4287142	0.3772649	0.6889363	0.7464906	0.6859296
	JIVE-BIC	4	0.6608040	0.4372675	0.3603942	0.6678438	0.7286432	0.6608040
	A-JIVE	64	0.6140704	0.4482479	0.3710317	0.6707575	0.7363682	0.6841709
	iCluster	3	0.7638191	0.5176193	0.4745746	0.7658865	0.7842867	0.7638191
	LRAcluster	2	0.7110553	0.4368520	0.4035040	0.7101385	0.7521740	0.7110553
	PCA-con	4	0.7587940	**0.5506612**	0.5038795	0.7601317	0.7984380	0.7587940
	Proposed	11	**0.7688442**	0.5437267	**0.5090183**	**0.7699789**	**0.7999063**	**0.7688442**

3.3.4.1 Cluster Analysis

Table 3.4 compares the performance of clustering on the joint subspace extracted by the proposed algorithm with that of existing integrative clustering approaches, in terms of the external cluster evaluation indices. The results in Table 3.4 show that the proposed method outperforms all the existing algorithms for CESC, LGG, and OV data sets, in terms of all six external evaluation indices. For BRCA data set, the proposed method gives the best results for accuracy, ARI, Rand, F-measure, and purity and second-best performance for NMI. PCA-con gives the best performance for NMI. PCA-con also has the second best performance in CESC and OV data sets for all the external indices. For LGG, A-JIVE has the second best performance for across all the indices. The JIVE algorithm gives better performance with BIC based rank estimation compared to permutation test based approach for both CESC and LGG data sets. For LGG, the joint rank estimated by JIVE is the same using both BIC and permutation tests. However, the overall performance differs due to difference in rank of the individual modalities estimated by the two criteria. For the OV data set, permutation test based JIVE algorithm extracts a 8-dimensional joint structure for each modality, which are concatenated to form the 32-dimensional final joint structure. However, BIC based JIVE algorithm estimates the rank of joint structure to be 0, which implies that the four different modalities do not share any correlated information among them. The iCluster algorithm uses regularized joint Gaussian latent variable model with standard lasso penalty to estimate the low-rank subspace. The performance of iCluster is heavily dependent on the choice of the penalty parameter. The lower performance of iCluster for all data sets, except BRCA, is attributed to poor model fitting and penalty parameter tuning. LRAcluster has relatively good performance for CESC, OV, and BRCA data sets, but for LGG its performance is very poor. This is primarily due to error in estimation of optimal rank, as better performance of LRAcluster is observed for ranks higher than the optimal one selected by the algorithm. The performance of Bayesian consensus based approach, BCC, is relative poor compared to the low-rank based approaches for CESC, OV, and BRCA data sets. This is mainly due to the poor estimation of distribution parameters based on Gibbs sampling approach, used by the algorithm. The results also show that the proposed approach gives better performance compared to all the existing low-rank approaches like JIVE, A-JIVE, iCluster, LRAcluster, and PCA-con for all data sets. This implies that the proposed method of rank estimation, and selection of modalities with high relevance and shared information preserve better cluster structure in the joint subspace compared to the low-rank subspaces extracted by the existing algorithms, which consider all the available modalities irrespective of their information content.

3.3.4.2 Survival Analysis

The log-rank and Wilcoxon test *p*-values from survival analysis of proposed and existing approaches are reported in Table 3.5. The survival difference of the proposed subtypes is compared with that of the previously identified TCGA subtypes. The Kaplan-Meier survival plots for the subtypes identified by the proposed approach are given in Figure 3.8 for different data sets. The median survival time and change in survival rate of the subtypes are observed over 2, 5, and 7 years of diagnosis of cancer. Median survival time is a statistic that refers to how long patients are expected to survive with a disease. The median survival time for a disease subtype is given by the time period where the Kaplan-

Table 3.5: Survival p-values and Execution Times of Proposed and Existing Approaches

Different Algorithms		Survival Analysis (p-value)		Time (in sec)		Survival Analysis (p-value)		Time (in sec)
		Log-Rank	Wilcoxon			Log-Rank	Wilcoxon	
COCA	**CESC**	5.563e-02	3.126e-02	6.01	**LGG**	1.166e-04	2.805e-05	18.61
BCC		5.318e-01	4.572e-01	10.33		3.721e-06	3.434e-07	12.78
JIVE-Perm		**4.074e-02**	**2.479e-02**	575.95		3.736e-04	1.310e-04	622.28
JIVE-BIC		8.295e-02	8.341e-02	69.08		3.156e-08	**5.134e-10**	1636.94
A-JIVE		3.463e-01	2.469e-01	251.77		3.784e-07	1.922e-08	462.65
iCluster		1.448e-01	1.212e-01	1054.89		4.201e-03	7.864e-03	1241.97
LRAcluster		2.404e-01	2.418e-01	9.29		9.278e-02	1.682e-01	25.09
PCA-con		1.243e-01	9.175e-02	**0.23**		**3.144e-08**	9.196e-10	1.61
Proposed		1.352e-01	1.064e-01	1.09		2.473e-07	6.000e-09	**1.05**
COCA	**BRCA**	1.159e-02	7.210e-03	26.90	**OV**	6.042e-02	2.699e-01	25.40
BCC		5.174e-01	5.433e-01	17.94		2.333e-01	3.957e-01	40.38
JIVE-Perm		**1.137e-02**	1.435e-02	934.13		**7.982e-03**	**8.471e-03**	1491.21
JIVE-BIC		5.314e-01	4.693e-01	734.10		-	-	-
A-JIVE		2.358e-01	2.206e-01	761.76		1.825e-01	2.489e-01	557.67
iCluster		1.409e-02	**4.282e-03**	511.87		5.831e-01	6.338e-01	2076.36
LRAcluster		1.513e-01	2.320e-01	23.53		1.583e-01	2.305e-01	15.35
PCA-con		2.765e-02	2.047e-02	**1.06**		7.744e-02	2.583e-01	**1.07**
Proposed		6.887e-02	5.397e-02	1.47		4.296e-02	1.516e-01	1.72

Meier curve for the subtype crosses the survival probability of 0.5, and it is not available for subtypes whose survival curves end before the survival probability of 0.5 due to low sample count or presence of censored samples. The observations are reported in Table 3.6. For OV and BRCA data sets, the p-values from both log-rank and Wilcoxon tests on proposed subtypes are lower than that of the previously identified TCGA subtypes. This implies that subtypes identified by the proposed method have larger difference in survival profiles compared to the already identified subtypes of these data sets. For the LGG data set, the already identified subtypes have lower p-values as compared to that of the proposed subtypes. However, the survival difference is statistically significant for both proposed and TCGA subtypes.

For the CESC data set, the p-value of pairwise log-rank test, comparing subtypes 1 and 2, is 0.04706108, which implies a significant difference between their survival profiles. This is also visible from their distinctly separate survival curves reported in Figure 3.8(a). However, the p-values from log-rank test are not significant for the other two pairs. Table 3.6 shows that proposed subtype 1 for CESC data set has a median survival time of 5.57 years, and its survival probability drops to 0.547 after 5 years of diagnosis of the cancer. However, for subtypes 2 and 3, the survival probability is as high as 0.804 and 0.771, respectively, after 5 years of diagnosis. This implies that subtype 1 shows poor prognosis compared to subtypes 2 and 3, and its survival probability is also less than 0.5 after 7 years of diagnosis. For the LGG data set, Figure 3.8(b) shows that subtypes 1 and 2 have close and intersecting survival curves, which implies lower difference between their survival profiles. However, subtype 3 is significantly different from subtypes 1 and 2, as the p-values from pairwise log-rank test on subtypes 1 and 3, and subtypes 2 and 3 are 4.162-06 and 4.359e-05, respectively. Moreover, subtype 3 has a very low median survival

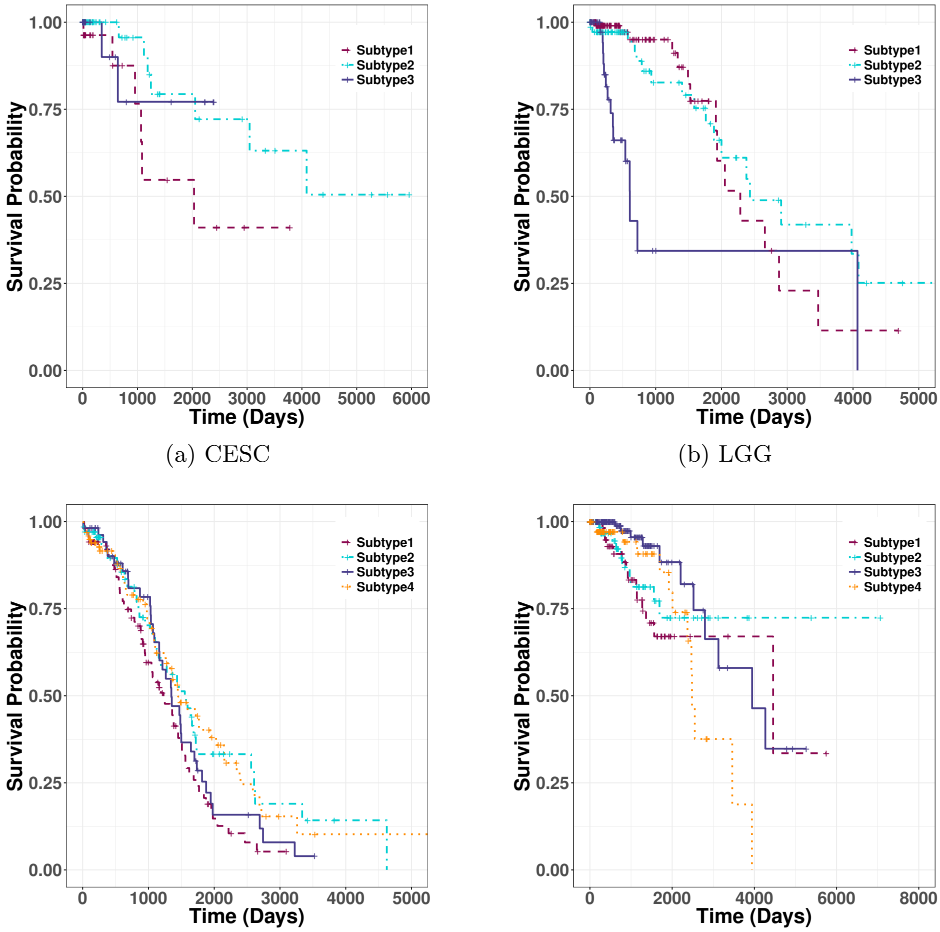
Table 3.6: Survival Analysis of Cancer Subtypes Identified by Proposed Algorithm

Different Data Sets	Different Subtypes	No. of Samples	Survival Probability After			Median Survival Time (Years)
			2 Years	5 Years	7 Years	
CESC	Subtype1	32	0.875	0.547	0.410	5.57
	Subtype2	68	0.957	0.804	0.721	-
	Subtype3	24	0.771	0.771	-	-
		p-values: log-rank= 1.352176e-01 Wilcoxon= 1.064875e-01				
LGG	Subtype1	139	0.950	0.774	0.430	6.26
	Subtype2	77	0.889	0.709	0.489	6.67
	Subtype3	51	0.343	0.343	0.343	1.66
		p-values: log-rank= 2.473056e-07 Wilcoxon= 6.06793e-09				
OV	Subtype1	106	0.748	0.224	0.079	3.37
	Subtype2	70	0.814	0.333	0.333	4.25
	Subtype3	56	0.809	0.254	0.158	3.69
	Subtype4	100	0.790	0.402	0.246	3.98
		p-values: log-rank= 4.296905e-02 Wilcoxon= 1.516501e-01				
BRCA	Subtype1	84	0.908	0.670	0.670	12.2
	Subtype2	77	0.922	0.725	0.725	-
	Subtype3	150	0.988	0.884	0.746	10.8
	Subtype4	87	0.971	0.855	0.376	6.8
		p-values: log-rank= 6.887602e-02 Wilcoxon= 5.397618e-02				

time of only 1.66 years, as compared to 6.26 and 6.67 years, respectively, for subtypes 1 and 2. This implies higher survival risk for patients belonging to subtype 3. For the OV data set, Figure 3.8(c) shows that the survival curves of the subtypes are close to each other, and the survival durations are not significantly different, which is also the case for the established TCGA subtypes [215]. However, the change in survival rate of the subtypes over the years shows that subtype 1 of the OV data set has the highest risk of survival after 7 years of diagnosis compared to the other three subtypes. For BRCA, Figure 3.8(d) shows that proposed subtypes 1 and 3 have fairly high median survival time. However, for subtype 4 the survival probability drops sharply from 0.855 at 5 years of diagnosis to 0.376 after 7 years of diagnosis, implying its poor prognosis.

3.3.4.3 Execution Efficiency

The execution times reported in Table 3.5 show that the proposed approach is computationally much faster than other statistical and SVD based low-rank approaches like iCluster, JIVE, A-JIVE, and LRAcluster. The reduced computation time of the proposed approach is due to the iterative updation of the joint subspace from the individual subspaces, instead of solving an SVD from scratch at each step of joint subspace construction. The execution time of the proposed algorithm is slightly higher than the PCA-con approach for CESC, OV, and BRCA data sets. The lower execution time of PCA-con is achieved due to direct PCA on the large integrated data. However, the results using external evaluation indices show that such naive integration methods fail to capture the true cluster structure of the



(a) CESC

(b) LGG

(c) OV

(d) BRCA

Figure 3.8: Kaplan-Meier survival plots for proposed subtypes of CESC, LGG, OV, and BRCA data sets.

data. For model fitting, iCluster and LRAcluster use expectation maximization algorithm, while JIVE uses alternate optimization. These iterative algorithms have slow convergence on the high-dimensional multimodal data sets. This leads to huge execution time and poor scalability of these algorithms.

3.3.5 Robustness and Stability Analysis

To assess the robustness of the clusters identified by the proposed method to small perturbation in the data set, a bootstrap approach is undertaken. For each data set, 1000 bootstrap samples are generated by sampling with replacement from the original data set. Two stage pipeline of the proposed joint subspace construction and subsequent clustering is then performed on each bootstrap sample. The quality of clustering of the bootstrap samples is assessed using Davies-Bouldin (DB) index [48], which is an internal cluster evaluation metric. For each bootstrap sample, 1000 different permutations of the cluster labels are obtained and the DB index is computed for each permuted labelling. Depending on the mean and standard deviation of the DB index obtained over different permutations, the Z-score and p-value of the observed DB score are evaluated. The distribution of p-values obtained from the bootstrap samples is given in Figure 3.9 for the four data sets. The distributions show that majority of the p-values lie in the range of 1e-14 and 1e-06 for the BRCA data set, while for the OV and LGG data sets, the range is in between 1e-10 and 1e-05. These p-value ranges imply that the observed DB scores for the BRCA data set deviate more from the DB scores obtained under random permutation of cluster labels compared to the CESC and OV data sets. On the other hand, for the CESC data set, more than 90% of the p-values lie in between 1e-06 and 1e-04, indicating lowest deviation from random clustering compared to other three data sets. However, for all three data sets, more than 99% of the p-values are less than 0.001, which implies that the clusters in the bootstrap samples show significant deviation from randomly assigned clusters. Thus, clusters identified in all four data sets are robust against random perturbation of the data sets.

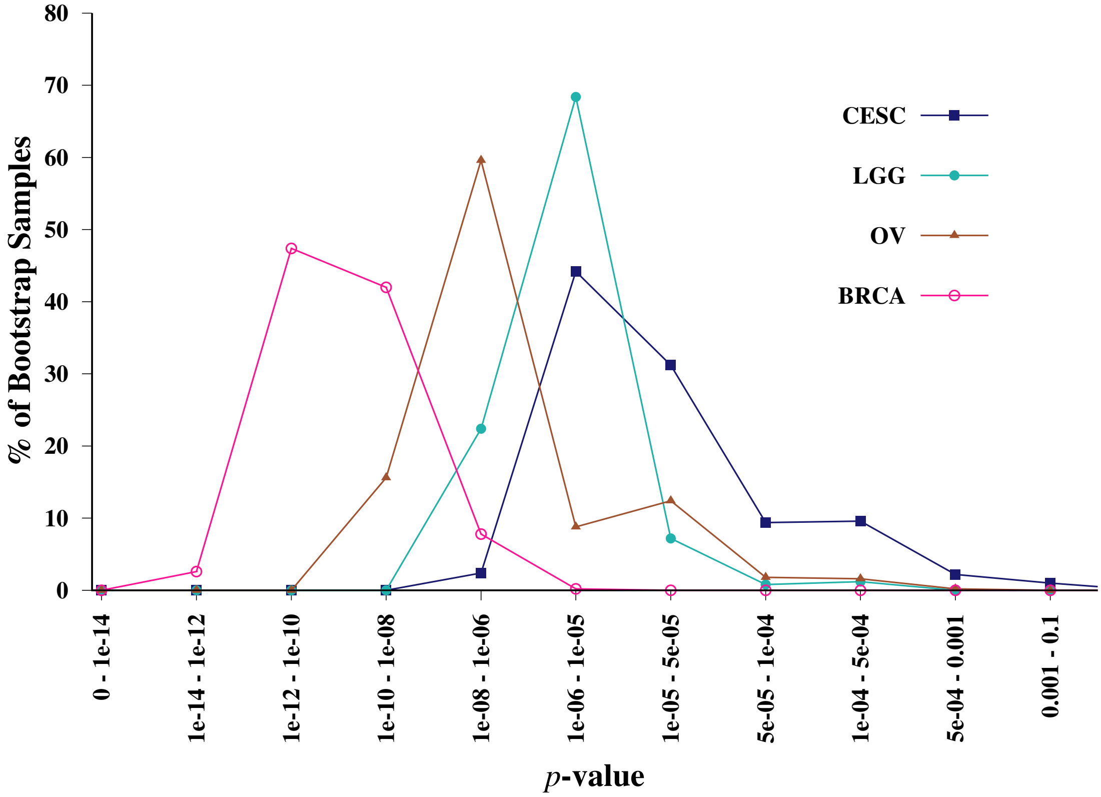


Figure 3.9: Distribution of p-values obtained from robustness analysis on different data.

For any clustering method, it is necessary to analyze whether the patterns identified by cluster analysis are necessarily meaningful or not. Stability means that a meaningful valid cluster should not disappear if the data set is changed in a “non-essential" way. Statistically, this means that data sets drawn from the same underlying distribution should give rise to more or less the same clustering. Hennig [92] proposed a method for cluster-wise stability analysis where clusters found in a data set are treated as “true" clusters and several bootstrap replications of the original data set are generated. For each true cluster, the most similar cluster in the bootstrap samples is identified using Jaccard similarity coefficient [84]. For a cluster, a summary statistic like mean Jaccard coefficient over the bootstrap samples is a measure of its stability. Jaccard coefficient value lies in between 0 and 1, and a higher value is indicative of better stability. Hennig [92] also suggested other summary statistics for stability assessment, namely, number of dissolutions and number of good recoveries. Dissolution refers to those cases for which the Jaccard coefficient value is less than 0.5, and the cluster is said to be “dissolved" or lost in the bootstrap sample. Lower value of the number of dissolutions for a cluster indicates that it represents a meaningful pattern which is not easily lost in the perturbed bootstrap samples. Number of good recoveries measures the number of times the Jaccard coefficient value is greater than 0.75 in the bootstrap samples. It indicates how well a cluster can be recovered from the perturbed bootstrap samples.

Table 3.7: Stability Analysis of Each Cluster

Different Data Sets	Different Subtypes	Mean Jaccard Coefficient	No. of Dissolutions	No. of Good Recoveries
CESC	Subtype1	0.7479388	295	669
	Subtype2	0.7845754	151	692
	Subtype3	0.5590394	356	215
LGG	Subtype1	0.5396389	690	130
	Subtype2	0.4145083	860	103
	Subtype3	0.7359946	12	230
OV	Subtype1	0.3325994	685	0
	Subtype2	0.4312147	137	0
	Subtype3	0.6631375	11	352
	Subtype4	0.5888738	8	144
BRCA	Subtype1	0.7839652	45	734
	Subtype2	0.9477903	0	982
	Subtype3	0.8666537	7	913
	Subtype4	0.8300830	7	911

To assess the stability of the clusters identified by the proposed approach, these summary statistics over 1000 bootstrap samples are reported in Table 3.7. The results in Table 3.7 show that all the identified subtypes of CESC and BRCA show high stability (mean Jaccard coefficient > 0.5). The subtypes of BRCA data set have maximum stability amongst all the data sets and subtype 2 shows the highest stability value of 0.9477903 among all the identified subtypes. The dissolution and recovery values in Table 3.7 also indicate that subtypes 2, 3, and 4 of BRCA data set are stable meaningful patterns which are dissolved in less than 10 bootstrap samples and could be recovered successfully in more than 900 times out of the 1000 bootstrap samples. Subtypes 1 and 2 of CESC and subtype 3 of LGG also have very high stability values of 0.7479388, 0.7845754, and 0.7359946, respectively. Subtypes 1 and 2 of LGG data set have poor stability and also fewer recoveries compared to the number of dissolutions. This is also evident from their close and intersecting survival curves in Figure 3.8(b). Subtypes 1 and 2 of OV have poor stability, but the other two subtypes 3 and 4 have moderate stability values. In 9 out of 14 cases, the identified clusters have higher recoveries compared to dissolutions. This implies that most of the identified subtypes indicate stable and meaningful patterns that are less likely to be dissolved when the data set is subjected to small perturbations.

3.4 Conclusion

The chapter presents a new algorithm for the extraction of a low-rank joint subspace from the high-dimensional multimodal data sets. The algorithm uses hypothesis testing to estimate efficiently the rank of each individual modality by separating its signal or structural component from the noise component. In order to address the major challenge of appropriate modality selection during data integration, two modality evaluation measures are proposed. One evaluates the relevance of a modality in terms of the quality of cluster structure embedded within it, while other measures the amount of shared information contained within the modalities. The modalities with highest relevance and maximum shared information are selected for integration. Moreover, intersection between two subspaces is considered to extract only the residual cluster information of different modalities, while removing the noisy components. Extensive experimental results show that the proposed method of rank estimation, modality selection, and joint subspace construction provides better clustering performance as compared to several existing integrative clustering approaches on several real-life multimodal omics data sets. The results also show that the subtypes identified by clustering on the extracted joint subspace have close resemblance with the previously established TCGA subtypes and have statistically significant difference in survival profiles. Finally, robustness analysis demonstrates that the identified subtypes indicate stable and meaningful patterns that are robust against small perturbation in the data set.

One of the major problems in multi-view data analysis is the high dimensional nature of the modalities. It makes sample clustering computationally expensive. In this regard, a novel algorithm is proposed in Chapter 4 to construct a low-rank joint subspace of integrated data from the low-rank subspaces of individual modalities. The problem of incrementally updating the singular value decomposition of a data matrix is formulated for the multimodal data framework.

Chapter 4

Selective Update of Relevant Eigenspaces for Integrative Clustering of Multi-View Data

4.1 Introduction

Integrative genomic data analysis refers to the design of algorithms to combine, infer, and analyze data from multiple genomic modalities like gene expression, DNA methylation, copy number variation, etc. Data from a single modality reflects biological patterns and variations within a specific molecular level. Integrative analysis allows modeling of intrinsic patterns of the individual modalities or views, and also captures correlated patterns across multiple modalities. One major objective of integrative genomic data analysis is cancer subtype discovery. Cancer subtyping provides insight into disease pathogenesis and design of personalized therapies. Data driven subtype discovery is most popularly achieved by clustering data from one or more genomic modalities [90]. Several integrative clustering algorithms exist for cancer subtyping [93, 140, 141, 156, 192, 214, 234, 243, 278]. A brief survey of existing integrative clustering algorithms is provided in Chapter 2. Clustering multimodal genomic data has mainly three major challenges.

1. The main challenge is the appropriate selection of modalities those provide relevant and shared subtype information, over modalities that provide noisy and inconsistent information.

2. Another challenge is handling the highly heterogeneous nature, in terms of scale, unit, and variance, of different genomic modalities.

3. The third challenge is that due to the high dimensional nature of the genomic modalities, the feature space becomes geometrically sparse; and most of the clustering methods become computationally expensive and prone to degrade their performance [46].

The existing integrative clustering approaches, as mentioned in Chapter 3, do not address all these challenges together. In general, direct integrative clustering approaches concatenate data matrices obtained from multiple modalities into a single matrix, which is used to get the joint clusters. However, the curse of dimensionality gets amplified due to the concatenation of several modalities. Most of the existing integrative clustering approaches assume that all the available modalities provide homogeneous and consistent cluster information; and thus consider all of them for integrative clustering. However, some modalities may provide disparate or even worse information [278]. Due to the presence of such noisy modalities, naive integration of information from all the available modalities can degrade the final cluster structure. During data integration, relevant modalities with shared cluster information should be chosen, instead of considering noisy and inconsistent ones. Therefore, one of the important problems in multimodal data clustering is how to select a subset of relevant modalities.

Another major challenge in clustering high-dimensional data is how to extract a lower dimensional subspace that best preserves the underlying cluster structure. PCA is an extensively used dimensionality reduction method for large-scale genomic data sets [5, 6]. It extracts the principal subspace that maximizes the variance along the projected axes and also minimizes the reconstruction error for any given rank. The principal subspace can be effectively represented using eigenspaces, which are widely used in various pattern recognition and image processing applications [32, 157, 184, 185]. Eigenspaces can be computed using SVD which has high computational complexity. This motivates the use of eigenspace update algorithms to prevent re-computation of eigenspace from scratch every time new observations are added to the data set. Such strategies include incremental update [32, 85] where eigenspace is updated on addition of every new observation, and block update [21, 29, 86] where update occurs on addition of new sets of observations. However, these algorithms have been proposed for a framework where data sets are incrementally updated with new observations. Eigenspace update model for multimodal data sets, where new modalities are being added for the same set of samples, has not been proposed to the best of our knowledge.

In this regard, this chapter introduces a novel algorithm, termed as SURE (Selective Update of Relevant Eigenspaces), to construct a low-rank joint subspace of the integrated data. The joint subspace is constructed from the low-rank subspaces of the individual modalities, such that it preserves best the underlying cluster structure. A theoretical formulation for updating the eigenspace is introduced for multimodal data sets, where new modalities are added for the same set of samples. The formulation enables efficient construction of the joint subspace compared to performing PCA on the concatenated data matrix. Moreover, the algorithm evaluates the quality of each modality before integrating it into the joint subspace. This allows the proposed algorithm to select most relevant modalities with maximum shared information, and hence addresses the problem of modality selection. Some new quantitative indices are proposed to measure theoretically the gap between the joint subspace extracted by the proposed SURE algorithm and the principal subspace obtained by PCA on the concatenated data. Finally, clustering is performed on the extracted joint subspace to identify the tumor subtypes. The efficiency of clustering by the proposed algorithm is extensively studied and compared with existing integrative clustering approaches on real-life multimodal cancer data sets. Some of the results of this chapter are reported in [112].

The rest of the chapter is organized as follows: Section 4.2 describes the basics of the SVD based eigenspace model of a data set. Section 4.3 presents the proposed multimodal clustering algorithm based on the updation of relevant eigenspaces, while Section 4.4 introduces some quantitative indices proposed in order to theoretically measure the gap between the full-rank eigenspace and the approximate eigenspace extracted by the proposed algorithm. Section 4.5 presents the experimental results on different multimodal cancer data sets and comparative performance analysis with existing approaches. Concluding remarks are provided in Section 4.6.

4.2 SVD Eigenspace Model

The basic assumption of the eigenspace model is that the data follows a multivariate Gaussian distribution. Under this assumption, the eigenspace model of the data set refers to the statistical description of a set of n observations in d-dimensional space in the form of a hyper-ellipsoid [85]. The hyper-ellipsoid is centered at the mean of the observations, and its axes point in directions where spread of the observations is maximized, subject to orthogonality. The hyper-ellipsoid is flat in the directions where the spread is negligible. This indicates a lower dimensional embedding of the hyper-ellipsoid considering only the top few axes along which the spread is significantly high. Eigenspace models can be computed either by eigenvalue decomposition of the covariance matrix of the data or by SVD of the mean centered data matrix itself. As $n << d$ for omics data, computation of large $d \times d$ covariance matrix needs intensive space and time. Also, multicollinearity of omic features often leads to a singular covariance matrix. Hence, the SVD eigenspace model is used in this work.

Let $X \in \Re^{n \times d}$ be a data matrix of n observations or samples, each having d features, and $rank(X) = r$. As stated previously in Section 3.2.1, the SVD of the mean-centered data matrix X is given by

$$X - \mathbf{1}\mu(X)^T = U(X)\Sigma(X)V(X)^T, \tag{4.1}$$

where $\mu(X) \in \Re^d$ is the mean of the data, A^T denotes the transpose of a matrix A, and $\mathbf{1}$ denotes a column vector of length n of all ones. The matrix $U(X)$ contains the r left singular vectors of X in its columns, which gives the r-dimensional principal subspace projection of the n samples of X. $\Sigma(X)$ is a diagonal matrix with entries $diag\{\sigma_1, \ldots, \sigma_i, \ldots, \sigma_r\}$, where $\sigma_1 \geqslant \ldots \geqslant \sigma_i \geqslant \ldots \geqslant \sigma_r > 0$. The σ_i's are the singular values of X, which give the spread of the projections along singular vectors in $U(X)$. The matrix $V(X)$ contains the r right singular vectors of X in its columns, which are the loadings of the d variables of X corresponding to the projections in $U(X)$. The principal components of X are obtained by multiplying the projections in $U(X)$ with the corresponding spread values in $\Sigma(X)$, given by $Y = U(X)\Sigma(X)$. The SVD eigenspace of X is given by a four-tuple as follows [86]:

$$\Psi(X) = \langle \mu(X), U(X), \Sigma(X), V(X) \rangle. \tag{4.2}$$

The SVD eigenspace model defined above in (4.2) differs from the principal subspace model defined in (3.4) of Chapter 3 in the sense that the SVD eigenspace contains two additional terms: the data mean $\mu(X)$ and the right singular subspace $V(X)$. This is because, only the left subspace $U(X)$ and singular values $\Sigma(X)$ are sufficient to extract the principal of a data matrix. The other two components are not required. Moreover, in Chapter 3 the joint subspace is constructed by simply concatenating the U and Σ components from the relevant modalities. However, this chapter focuses on entirely reconstructing the SVD of the integrated data from SVDs of individual modalities which requires contribution from all four components, U, Σ, V, and μ. Hence, the SVD eigenspace is defined as a four tuple as in (4.2).

Zha *et al.* [271] showed that the continuous relaxation of the discrete cluster membership indicators in k-means clustering problem is given by the top k principal components. So, the rank k truncated eigenspace, containing only top k singular vectors and corresponding singular values, sufficiently represents the cluster information of X. The noisy information, embedded in remaining $(n-k)$ singular triplets, gets eliminated from the truncated eigenspace. So, the rank r of the eigenspace is considered to be k in the current work.

4.3 SURE: Proposed Method

This section presents the proposed SURE algorithm to construct a low-rank joint subspace of the integrated data. Prior to describing the proposed algorithm, theoretical formulation for the eigenspace update problem is described next.

4.3.1 Eigenspace Updation

Let $X_1, \ldots, X_m, \ldots, X_M$, where $X_m \in \Re^{n \times d_m}$, be M different modalities or views of a multimodal data set, all measured on the same set of n samples. The M data matrices can be concatenated to form an integrated data matrix as follows:

$$\mathbf{X} = \begin{bmatrix} X_1 & \ldots & X_m & \ldots & X_M \end{bmatrix}. \tag{4.3}$$

The eigenspace of $\mathbf{X}$ gives a low-rank representation of the integrated data matrix $\mathbf{X}$. However, computation of eigenspace of $\mathbf{X}$ from scratch involves solving the SVD of $\mathbf{X}$, which is computationally expensive due to the large size of $\mathbf{X}$. A theoretical formulation is developed next for updating the SVD of a multimodal data set. This formulation allows construction of the eigenspace of $\mathbf{X}$ from the low-rank eigenspaces of the individual modalities.

Let the rank k eigenspace for modality X_m be given by

$$\Psi(X_m) = \langle \mu(X_m), U(X_m), \Sigma(X_m), V(X_m) \rangle. \tag{4.4}$$

Let the data matrix, formed by column-wise concatenation of m modalities, be given by

$$\widetilde{\mathbf{X}}_m = \begin{bmatrix} X_1 & X_2 & \ldots & X_m \end{bmatrix}.$$

The eigenspace of $\mathbf{X}$ is constructed sequentially in M steps by constructing the eigenspace of $\widetilde{\mathbf{X}}_m$ at each step for $m = 1, \ldots, M$. Let the eigenspace of $\widetilde{\mathbf{X}}_m$ obtained at m-th step be given by

$$\Psi(\widetilde{\mathbf{X}}_m) = \langle \mu(\widetilde{\mathbf{X}}_m), U(\widetilde{\mathbf{X}}_m), \Sigma(\widetilde{\mathbf{X}}_m), V(\widetilde{\mathbf{X}}_m) \rangle. \tag{4.5}$$

At $(m+1)$-th step, the matrix $\widetilde{\mathbf{X}}_{m+1}$ is formed by concatenation of the matrix $\widetilde{\mathbf{X}}_m$ of m-th step and the $(m+1)$-th modality X_{m+1}, given by

$$\widetilde{\mathbf{X}}_{m+1} = \begin{bmatrix} \widetilde{\mathbf{X}}_m & X_{m+1} \end{bmatrix}. \tag{4.6}$$

Let the eigenspace of $\widetilde{\mathbf{X}}_{m+1}$ be given by

$$\Psi(\widetilde{\mathbf{X}}_{m+1}) \quad = \quad \langle \mu(\widetilde{\mathbf{X}}_{m+1}), U(\widetilde{\mathbf{X}}_{m+1}), \Sigma(\widetilde{\mathbf{X}}_{m+1}), V(\widetilde{\mathbf{X}}_{m+1}) \rangle. \tag{4.7}$$

The idea of eigenspace construction is as follows: At $(m+1)$-th step, the eigenspace $\Psi(\widetilde{\mathbf{X}}_{m+1})$ is not constructed by solving the SVD of data matrix $\widetilde{\mathbf{X}}_{m+1}$ from scratch, rather it is constructed from the eigenspace $\Psi(\widetilde{\mathbf{X}}_m)$ obtained at m-th step and the eigenspace $\Psi(X_{m+1})$ of modality X_{m+1}. The initial eigenspace is given by

$$\Psi(\widetilde{\mathbf{X}}_1) = \Psi(X_1).$$

Let $\oplus$ be the operator denoting the addition of two eigenspaces. The eigenspace at $(m+1)$-th step is given by

$$\Psi(\widetilde{\mathbf{X}}_{m+1}) = \Psi(\widetilde{\mathbf{X}}_m) \oplus \Psi(X_{m+1}). \tag{4.8}$$

For the integrated data matrix $\mathbf{X}$, the final eigenspace is iteratively obtained as follows:

$$\begin{aligned}
&\text{Step } 1 && : \Psi(\widetilde{\mathbf{X}}_1) = \Psi(X_1) \\
&\text{Step } 2 && : \Psi(\widetilde{\mathbf{X}}_2) = \Psi(\widetilde{\mathbf{X}}_1) \oplus \Psi(X_2) \\
& && \quad \cdots \\
&\text{Step } (m+1) && : \Psi(\widetilde{\mathbf{X}}_{m+1}) = \Psi(\widetilde{\mathbf{X}}_m) \oplus \Psi(X_{m+1}) \\
& && \quad \cdots \\
&\text{Step } M && : \Psi(\widetilde{\mathbf{X}}_M) = \Psi(\widetilde{\mathbf{X}}_{M-1}) \oplus \Psi(X_M) = \Psi(\mathbf{X}).
\end{aligned}$$

The operator $\oplus$ in (4.8), for the construction of components of the new eigenspace $\Psi(\widetilde{\mathbf{X}}_{m+1})$, is described next.

The relation between a data matrix X and the components of its eigenspace $\Psi(X)$ is obtained by SVD using (4.1). Applying (4.1) to data matrices $\widetilde{\mathbf{X}}_m$ and X_{m+1}, the following relations are obtained:

$$\begin{aligned}
\widetilde{\mathbf{X}}_m - \mathbf{1}\mu(\widetilde{\mathbf{X}}_m)^T &= U(\widetilde{\mathbf{X}}_m)\Sigma(\widetilde{\mathbf{X}}_m)V(\widetilde{\mathbf{X}}_m)^T; \\
X_{m+1} - \mathbf{1}\mu(X_{m+1})^T &= U(X_{m+1})\Sigma(X_{m+1})V(X_{m+1})^T.
\end{aligned} \tag{4.9}$$

The matrix $\widetilde{\mathbf{X}}_{m+1}$ is constructed by column-wise concatenation of X_{m+1} to $\widetilde{\mathbf{X}}_m$. So, the mean component $\mu(\widetilde{\mathbf{X}}_{m+1})$ of $\Psi(\widetilde{\mathbf{X}}_{m+1})$ is also obtained directly by column-wise concatenation of mean vectors $\mu(\widetilde{\mathbf{X}}_m)$ and $\mu(X_{m+1})$, that is,

$$\mu(\widetilde{\mathbf{X}}_{m+1}) = \Big[\mu(\widetilde{\mathbf{X}}_m) \quad \mu(X_{m+1})\Big]. \tag{4.10}$$

The left singular subspace $U(\widetilde{\mathbf{X}}_{m+1})$ consists of unit vectors corresponding to the principal subspace projection of the data in $\widetilde{\mathbf{X}}_{m+1}$. The matrix $\widetilde{\mathbf{X}}_{m+1}$ has $\widetilde{\mathbf{X}}_m$ and X_{m+1} as its constituent block matrices. Therefore, the left subspace of $\widetilde{\mathbf{X}}_{m+1}$ must be constructed in such a way that it contains the information of projection of data from the m modalities in $\widetilde{\mathbf{X}}_m$ and also the projection information from $(m+1)$-th modality X_{m+1}. So, both $U(\widetilde{\mathbf{X}}_m)$ and $U(X_{m+1})$ must be subspaces of $U(\widetilde{\mathbf{X}}_{m+1})$. The subspace $U(\widetilde{\mathbf{X}}_{m+1})$ can be obtained by constructing a basis sufficient to span both the subspaces $U(\widetilde{\mathbf{X}}_m)$ and $U(X_{m+1})$. $U(\widetilde{\mathbf{X}}_m)$ is itself a basis for the left subspace of $\widetilde{\mathbf{X}}_m$. Let Γ be the basis for the subspace lying orthogonal to left subspace spanned by $U(\widetilde{\mathbf{X}}_m)$. Therefore, a sufficient basis for $U(\widetilde{\mathbf{X}}_{m+1})$ can be formed by augmenting the basis $U(\widetilde{\mathbf{X}}_m)$ with basis Γ of the orthogonal space.

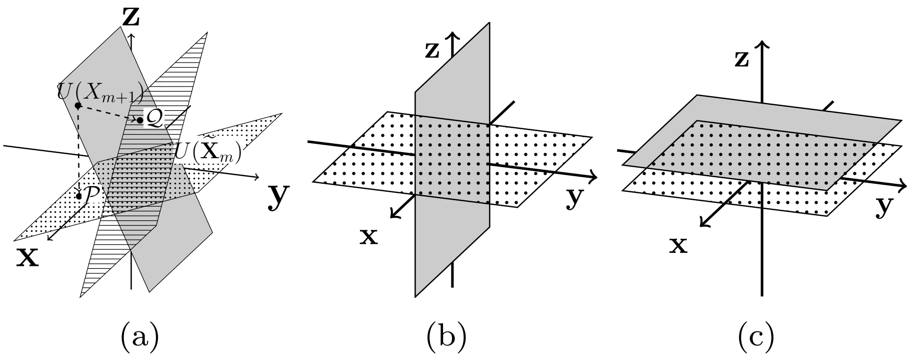


Figure 4.1: (a) Projected and residual components of subspace $U(X_{m+1})$ with respect to $U(\widetilde{\mathbf{X}}_m)$; (b) Intersection between $U(X_{m+1})$ and $U(\widetilde{\mathbf{X}}_m)$ is empty; (c) $U(X_{m+1})$ is a subspace of $U(\widetilde{\mathbf{X}}_m)$.

In Figure 4.1(a), the gray plane represents the subspace $U(X_{m+1})$ and the dotted plane represents the subspace $U(\widetilde{\mathbf{X}}_m)$. The basis Γ has to span the subspace orthogonal to the dotted plane $U(\widetilde{\mathbf{X}}_m)$. It is constructed by projecting $U(X_{m+1})$ to $U(\widetilde{\mathbf{X}}_m)$ and then obtaining orthogonal bases for the residual matrix. The projection is given by

$$\mathcal{I} = U(\widetilde{\mathbf{X}}_m)^T U(X_{m+1}). \tag{4.11}$$

The component of $U(X_{m+1})$, lying in the subspace spanned by $U(\widetilde{\mathbf{X}}_m)$, is obtained by multiplying the projection $\mathcal{I}$ with the corresponding basis $U(\widetilde{\mathbf{X}}_m)$. The projected component $\mathcal{P}$ is given by

$$\mathcal{P} = U(\widetilde{\mathbf{X}}_m)\mathcal{I}. \tag{4.12}$$

Finally, the residual component $\mathcal{Q}$ is obtained by subtracting the projected component $\mathcal{P}$ from $U(X_{m+1})$ itself, given by,

$$\mathcal{Q} = U(X_{m+1}) - \mathcal{P}. \tag{4.13}$$

The residual $\mathcal{Q}$ lies in the subspace orthogonal to the one spanned by $U(\widetilde{\mathbf{X}}_m)$. In Figure 4.1(a), $\mathcal{P}$ denotes the projection of the gray plane $U(X_{m+1})$ onto the dotted plane $U(\widetilde{\mathbf{X}}_m)$ and the stripped plane denotes the residual component $\mathcal{Q}$, which is orthogonal to the dotted plane $U(\widetilde{\mathbf{X}}_m)$. An orthonormal basis Γ for the residual component can be obtained by Gram-Schmidt orthogonalization of $\mathcal{Q}$. However, if the intersection between the subspaces $U(\widetilde{\mathbf{X}}_m)$ and $U(X_{m+1})$ is non-empty, the rank of the residual space reduces. So, the rank of the intersection space is evaluated in order to choose the right number of basis vectors required for the residual space. This can be evaluated using the following theorem.

Theorem 4.1. *Let $\mathcal{A}$ and $\mathcal{B}$ be two subspaces of $\Re^n$. Let columns of matrices $A \in \Re^{n \times r_1}$ and $B \in \Re^{n \times r_2}$ be orthonormal bases for the subspaces $\mathcal{A}$ and $\mathcal{B}$, respectively. Let SVD of $A^T B$ be $U \Sigma V^T$, where $\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_r)$ and $\sigma_1 \geqslant \sigma_2 \geqslant ... \geqslant \sigma_r$. Then, the dimension of the intersection subspace $\mathcal{A} \cap \mathcal{B}$ is ω iff $\sigma_1 = \sigma_2 = ... = \sigma_\omega = 1 \geqslant \sigma_{\omega+1}$* [16].

The above theorem states that the number of singular values of $A^T B$, which are equal to 1, gives the dimension of the intersection subspace of $\mathcal{A}$ and $\mathcal{B}$. For subspaces $U(\widetilde{\mathbf{X}}_m)$ and $U(X_{m+1})$, the matrices themselves form orthonormal bases. Therefore, according to Theorem 4.1, the number of singular values of the matrix $\mathcal{I}$ of (4.11), those are equal to 1, gives the dimension of the intersection space. Let t be the number of such singular values of $\mathcal{I}$ that are equal to 1. Then, the dimension of the residual space is $(r - t)$, where r is the dimension of the subspace $U(X_{m+1})$. Let $\mathcal{G}$ be the orthonormal basis obtained from Gram-Schmidt orthogonalization of $\mathcal{Q}$. If the rank of residual space is $(r - t)$, then exactly t column vectors of $\mathcal{G}$ would have norm zero. Finding the rank t of the intersection space through SVD of $\mathcal{I}$ has complexity of $\mathcal{O}(r^3)$. Alternatively, t can be computed by finding the number of vectors in $\mathcal{G}$ having norm zero. For $t > 0$, $(r - t)$ non-zero vectors of $\mathcal{G}$ are used to form Γ, which spans the residual space. Following two special cases arise while considering the intersection between the subspaces.

- **Case 1 - Intersection between two subspaces $U(\widetilde{\mathbf{X}}_m)$ and $U(X_{m+1})$ is empty**: This case arises when $U(X_{m+1})$ lies entirely in the subspace orthogonal to $U(\widetilde{\mathbf{X}}_m)$ as shown in Figure 4.1(b). Therefore, when $U(X_{m+1})$ is projected onto $U(\widetilde{\mathbf{X}}_m)$, the projection magnitudes are all zeros, that is, $\mathcal{I} = \mathbf{0}$, where $\mathbf{0}$ denotes a zero matrix of appropriate dimension. Hence, the projected component $\mathcal{P}$ in (4.12) is also $\mathbf{0}$. So, the residual $\mathcal{Q}$ in (4.13) is the whole subspace $U(X_{m+1})$, that is, $\mathcal{Q} = U(X_{m+1})$, which is itself an orthonormal basis. Therefore, the basis for the residual space is $\Gamma = U(X_{m+1})$.
- **Case 2 - Subspace $U(X_{m+1})$ is a subspace of $U(\widetilde{\mathbf{X}}_m)$**: This case arises when $U(X_{m+1})$ is itself a subspace of $U(\widetilde{\mathbf{X}}_m)$, as shown in Figure 4.1(c) where the subspaces are parallel to each other. This implies that all the column vectors of $U(X_{m+1})$ can be expressed as a linear combination of those in $U(\widetilde{\mathbf{X}}_m)$. So, the projected component $\mathcal{P}$ in (4.12) is $U(X_{m+1})$ itself, and the residual $\mathcal{Q}$ in (4.13) is $\mathbf{0}$. Since the residual space is empty, the basis $U(\widetilde{\mathbf{X}}_m)$ is sufficient to span both the subspaces $U(\widetilde{\mathbf{X}}_m)$ and $U(X_{m+1})$. Therefore, $\Gamma = \varnothing$.

After constructing the appropriate basis Γ for residual space, it is appended to the basis $U(\widetilde{\mathbf{X}}_m)$. Thus, $\begin{bmatrix} U(\widetilde{\mathbf{X}}_m) & \Gamma \end{bmatrix}$ spans both the subspaces $U(\widetilde{\mathbf{X}}_m)$ and $U(X_{m+1})$. This basis differs from the required basis $U(\widetilde{\mathbf{X}}_{m+1})$ by a rotation $\mathcal{R}(\widetilde{\mathbf{X}}_{m+1})$. Hence, $U(\widetilde{\mathbf{X}}_{m+1})$ is obtained as follows:

$$U(\widetilde{\mathbf{X}}_{m+1}) = \begin{bmatrix} U(\widetilde{\mathbf{X}}_m) & \Gamma \end{bmatrix} \mathcal{R}(\widetilde{\mathbf{X}}_{m+1}), \tag{4.14}$$

where $\mathcal{R}(\widetilde{\mathbf{X}}_{m+1})$ is an orthonormal rotation matrix. The $\Sigma(\widetilde{\mathbf{X}}_{m+1})$ and $V(\widetilde{\mathbf{X}}_{m+1})$ components of the eigenspace of $\widetilde{\mathbf{X}}_{m+1}$ and the rotation matrix $\mathcal{R}(\widetilde{\mathbf{X}}_{m+1})$ are computed as follows. The SVD of $\widetilde{\mathbf{X}}_{m+1}$ gives the following relation:

$$\widetilde{\mathbf{X}}_{m+1} - \mathbf{1}\mu(\widetilde{\mathbf{X}}_{m+1})^T = U(\widetilde{\mathbf{X}}_{m+1})\Sigma(\widetilde{\mathbf{X}}_{m+1})V(\widetilde{\mathbf{X}}_{m+1})^T. \tag{4.15}$$

Substituting $U(\widetilde{\mathbf{X}}_{m+1})$ from (4.14) in (4.15), we get

$$\widetilde{\mathbf{X}}_{m+1} - \mathbf{1}\mu(\widetilde{\mathbf{X}}_{m+1})^T = \begin{bmatrix} U(\widetilde{\mathbf{X}}_m) & \Gamma \end{bmatrix} \mathcal{R}(\widetilde{\mathbf{X}}_{m+1})\Sigma(\widetilde{\mathbf{X}}_{m+1})V(\widetilde{\mathbf{X}}_{m+1})^T; \tag{4.16}$$

$$\Rightarrow \quad \mathcal{R}(\widetilde{\mathbf{X}}_{m+1})\Sigma(\widetilde{\mathbf{X}}_{m+1})V(\widetilde{\mathbf{X}}_{m+1})^T = \begin{bmatrix} U(\widetilde{\mathbf{X}}_m) & \Gamma \end{bmatrix}^T \left(\widetilde{\mathbf{X}}_{m+1} - \mathbf{1}\mu(\widetilde{\mathbf{X}}_{m+1})^T\right) \tag{4.17}$$

as $U(\widetilde{\mathbf{X}}_m)$ and Γ are orthonormal matrices, and $\begin{bmatrix} U(\widetilde{\mathbf{X}}_m) & \Gamma \end{bmatrix}^T \begin{bmatrix} U(\widetilde{\mathbf{X}}_m) & \Gamma \end{bmatrix} = \mathbf{I}_s$, where $\mathbf{I}_s$ is the identity matrix of order s.

Substituting the values of $\widetilde{\mathbf{X}}_{m+1}$ and $\mu(\widetilde{\mathbf{X}}_{m+1})$ from (4.6) and (4.10), respectively, in (4.17), we get

$$\begin{aligned} &\mathcal{R}(\widetilde{\mathbf{X}}_{m+1})\Sigma(\widetilde{\mathbf{X}}_{m+1})V(\widetilde{\mathbf{X}}_{m+1})^T \\ &= \begin{bmatrix} U(\widetilde{\mathbf{X}}_m) & \Gamma \end{bmatrix}^T \begin{bmatrix} \widetilde{\mathbf{X}}_m & X_{m+1} \end{bmatrix} - \mathbf{1} \begin{bmatrix} \mu(\widetilde{\mathbf{X}}_m)^T & \mu(X_{m+1})^T \end{bmatrix} \\ &= \begin{bmatrix} U(\widetilde{\mathbf{X}}_m) & \Gamma \end{bmatrix}^T \begin{bmatrix} \widetilde{\mathbf{X}}_m - \mathbf{1}\mu(\widetilde{\mathbf{X}}_m)^T & X_{m+1} - \mathbf{1}\mu(X_{m+1})^T \end{bmatrix}. \end{aligned} \tag{4.18}$$

Using (4.9) in (4.18), we get

$$\mathcal{R}(\widetilde{\mathbf{X}}_{m+1})\Sigma(\widetilde{\mathbf{X}}_{m+1})V(\widetilde{\mathbf{X}}_{m+1})^T = \begin{bmatrix} \mathcal{M}_{11}\Sigma(\widetilde{\mathbf{X}}_m)V(\widetilde{\mathbf{X}}_m)^T & \mathcal{M}_{12}\Sigma(X_{m+1})V(X_{m+1})^T \\ \mathcal{M}_{21}\Sigma(\widetilde{\mathbf{X}}_m)V(\widetilde{\mathbf{X}}_m)^T & \mathcal{M}_{22}\Sigma(X_{m+1})V(X_{m+1})^T \end{bmatrix}; \tag{4.19}$$

$$\text{where } \mathcal{M}_{11} = U(\widetilde{\mathbf{X}}_m)^T U(\widetilde{\mathbf{X}}_m) = \mathbf{I}_k; \quad \mathcal{M}_{21} = \Gamma^T U(\widetilde{\mathbf{X}}_m) = \mathbf{0};$$
$$\mathcal{M}_{12} = U(\widetilde{\mathbf{X}}_m)^T U(X_{m+1}) = \mathcal{I}; \quad \mathcal{M}_{22} = \Gamma^T U(X_{m+1}).$$

Substituting the values of $\mathcal{M}_{ij}, \forall i, j = 1, 2$ in (4.19), we get

$$\mathcal{R}(\widetilde{\mathbf{X}}_{m+1})\Sigma(\widetilde{\mathbf{X}}_{m+1})V(\widetilde{\mathbf{X}}_{m+1})^T = \begin{bmatrix} \mathbf{I}_k\Sigma(\widetilde{\mathbf{X}}_m)V(\widetilde{\mathbf{X}}_m)^T & \mathcal{I}\Sigma(X_{m+1})V(X_{m+1})^T \\ \mathbf{0} & \mathcal{M}_{22}\Sigma(X_{m+1})V(X_{m+1})^T \end{bmatrix}. \tag{4.20}$$

Solving the SVD problem for the matrix of (4.20), the components $\mathcal{R}(\widetilde{\mathbf{X}}_{m+1})$, $\Sigma(\widetilde{\mathbf{X}}_{m+1})$, and $V(\widetilde{\mathbf{X}}_{m+1})$ are obtained. The left subspace $U(\widetilde{\mathbf{X}}_{m+1})$ is obtained by substituting the value of $\mathcal{R}(\widetilde{\mathbf{X}}_{m+1})$ in (4.14). Finally, the matrices $U(\widetilde{\mathbf{X}}_{m+1})$ and $V(\widetilde{\mathbf{X}}_{m+1})$ are truncated to store only the top k singular vectors and $\Sigma(\widetilde{\mathbf{X}}_{m+1})$ is truncated to store the corresponding k largest singular values in the eigenspace of $\widetilde{\mathbf{X}}_{m+1}$.

For **Case 1**, where intersection between two left subspaces is empty, substituting the values $\mathcal{I} = \mathbf{0}$ and $\Gamma = U(X_{m+1})$ in the SVD of (4.20), we get

$$\mathcal{R}(\widetilde{\mathbf{X}}_{m+1})\Sigma(\widetilde{\mathbf{X}}_{m+1})V(\widetilde{\mathbf{X}}_{m+1})^T = \begin{bmatrix} \mathbf{I}_k\Sigma(\widetilde{\mathbf{X}}_m)V(\widetilde{\mathbf{X}}_m)^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_k\Sigma(X_{m+1})V(X_{m+1})^T \end{bmatrix}. \tag{4.21}$$

The SVD of (4.21) is a block-diagonal SVD problem whose solution is given by

$$\mathcal{R}(\widetilde{\mathbf{X}}_{m+1}) = \mathbf{I}_{2k}; \qquad \Sigma(\widetilde{\mathbf{X}}_{m+1}) = \begin{bmatrix} \Sigma(\widetilde{\mathbf{X}}_m) & \mathbf{0} \\ \mathbf{0} & \Sigma(X_{m+1}) \end{bmatrix};$$
$$V(\widetilde{\mathbf{X}}_{m+1})^T = \begin{bmatrix} V(\widetilde{\mathbf{X}}_m)^T & \mathbf{0} \\ \mathbf{0} & V(X_{m+1})^T \end{bmatrix}.$$

Substituting $\mathcal{R}(\widetilde{\mathbf{X}}_{m+1})$ in (4.14),

$$U(\widetilde{\mathbf{X}}_{m+1}) = [U(\widetilde{\mathbf{X}}_m)\ U(X_{m+1})].$$

This signifies that for non-intersecting subspaces $U(\widetilde{\mathbf{X}}_m)$ and $U(X_{m+1})$, the bases for the joint left and right singular subspaces are formed by the union of the individual bases and has rank $2k$. In the context of integrative clustering, this implies that the cluster structure reflected in modality X_{m+1} is completely disparate with respect to cluster structure embedded in joint modality $\widetilde{\mathbf{X}}_m$. So, incorporation of modality X_{m+1} can introduce totally inconsistent cluster information into the joint cluster structure embedded in the eigenspace of $\widetilde{\mathbf{X}}_m$. Therefore, careful evaluation of a modality is necessary before updating it into the joint eigenspace.

4.3.2 Evaluation of Individual Modality

This section introduces two modality evaluation measures, namely, relevance and concordance. While relevance assesses the quality of cluster information provided by each modality, the concordance measures the amount of cluster information shared between two modalities. Let $X_i \in \Re^{n\times d_i}$ and $X_j \in \Re^{n\times d_j}$ be two modalities of a multimodal data set whose rank k eigenspaces are given by

$$\Psi(X_i) = \langle \mu(X_i), U(X_i), \Sigma(X_i), V(X_i)\rangle; \tag{4.22}$$
$$\Psi(X_j) = \langle \mu(X_j), U(X_j), \Sigma(X_j), V(X_j)\rangle. \tag{4.23}$$

4.3.2.1 Relevance

The relevance of a modality is defined in terms of the compactness of the cluster structure embedded in its eigenspace. The compactness is evaluated in the left subspace, which contains principal subspace projection of the samples. The relevance measure is independent of the difference in scale, unit, and variance of the modalities, as the left subspace of each modality contains k unit vectors. The compactness of cluster structure of a modality X_i is given by the percentage of variance explained (PVE) by a partition of its left subspace $U(X_i)$. Let $\mathcal{C}^i = \{C_1^i, \ldots, C_j^i, \ldots, C_k^i\}$ be a partition of the left subspace $U(X_i)$ into k clusters.

The PVE in $U(X_i)$ by partition $\mathcal{C}^i$ is given by the ratio of between-cluster variance in $\mathcal{C}^i$ to the total variance of $U(X_i)$. The total variance is the total sum-of-squared distance of each sample from its mean, given by

$$\mathrm{T}(U(X_i)) = \sum_{p=1}^{n} ||x_p^i - \bar{x}^i||^2 \tag{4.24}$$

where $\bar{x}^i$ is the mean of $U(X_i)$. Since $U(X_i)$ contains principal subspace projection of data in X_i, the projection values in $U(X_i)$ also have zero mean. Hence, $\bar{x}^i = 0$. Moreover, the columns of $U(X_i)$ are orthonormal to each other, therefore,

$$\mathrm{T}(U(X_i)) = \sum_{p=1}^{n} ||x_p^i||^2 = ||U(X_i)||_F^2 \quad = trace(U(X_i)^T U(X_i)) = trace(\mathbf{I}_k) = k, \tag{4.25}$$

where $||A||_F^2$ denotes the Frobenius norm of matrix A. The within-cluster variance of partition $\mathcal{C}^i$ is the sum-of-squared distance of each data point from its cluster centroid, given by

$$\mathrm{W}_{\mathcal{C}^i}(U(X_i)) = \sum_{j=1}^{k} \sum_{x_p^i \in C_j^i} ||x_p^i - m_j||^2 \tag{4.26}$$

where m_j is the centroid of cluster C_j^i. The between-cluster variance in $\mathcal{C}^i$ is obtained by subtracting the within-cluster variance in $\mathcal{C}^i$ from the total variance of $U(X_i)$. Thus, the PVE in $U(X_i)$ by the partition $\mathcal{C}^i$ is given by

$$\mathrm{PVE}(U(X_i)) = \frac{\mathrm{T}(U(X_i)) - \mathrm{W}_{\mathcal{C}^i}(U(X_i))}{\mathrm{T}(U(X_i))}. \tag{4.27}$$

The relevance of a modality X_i is given by

$$\mathrm{Rel}(X_i) = \mathrm{PVE}(U(X_i)). \tag{4.28}$$

The relevance measure gives a value in between 0 and 1 with higher value indicating better cluster structure. So, the modality X_i has higher relevance than modality X_j if $\mathrm{PVE}(U(X_i)) > \mathrm{PVE}(U(X_j))$. The relevance measure gives an ordering of the modalities, based on the quality of their cluster structures.

4.3.2.2 Concordance

The construction of joint eigenspace begins with the most relevant modality, having the best inherent cluster structure. Updating this eigenspace with a modality having very discordant cluster structure may degrade the final cluster solution. Therefore, a concordance measure, based on normalized mutual information (NMI) [68] between the cluster assignments of two modalities, is used to capture the joint cluster information shared between two modalities. Let $\mathcal{C}^i$ and $\mathcal{C}^j$ be k-partitions of the subspaces $U(X_i)$ and $U(X_j)$, respectively. The concordance $\mathbb{C}$ between X_i and X_j is given by the NMI between the cluster solutions $\mathcal{C}^i$ and $\mathcal{C}^j$

$$\mathbb{C}\,(X_i, X_j) = \text{NMI}(\mathcal{C}^i, \mathcal{C}^j). \tag{4.29}$$

NMI is defined as follows:

$$\text{NMI}(\mathcal{C}^i, \mathcal{C}^j) = \frac{2\,\mathbb{I}\left(\mathcal{C}^i, \mathcal{C}^j\right)}{[\mathbb{H}(\mathcal{C}^i) + \mathbb{H}(\mathcal{C}^j)]}; \tag{4.30}$$

where $\mathbb{H}(\mathcal{C}^i)$ is the entropy of $\mathcal{C}^i$ and $\mathbb{I}\left(\mathcal{C}^i, \mathcal{C}^j\right)$ is the mutual information between $\mathcal{C}^i$ and $\mathcal{C}^j$, which are as follows:

$$\begin{aligned}
\mathbb{H}\left(\mathcal{C}^i\right) &= -\sum_{p=1}^{k} Pr(C_p^i)\log Pr(C_p^i);\\
\mathbb{I}\left(\mathcal{C}^i, \mathcal{C}^j\right) &= \sum_{p=1}^{k}\sum_{q=1}^{k} Pr(C_p^i \cap C_q^j)\log\left[\frac{Pr(C_p^i \cap C_q^j)}{Pr(C_p^i)Pr(C_q^j)}\right];
\end{aligned}$$

where $Pr(S)$ denotes the probability of the set S. The value of concordance $\mathbb{C}$ lies in the range $[0, 1]$, with larger value being indicative of more shared information between two modalities. While selecting a modality, the average concordance between a candidate modality and all the previously integrated ones is computed. A candidate modality is selected for update only if its average concordance is beyond some threshold τ.

4.3.3 Proposed Algorithm

The relevance and concordance measures together help to select relevant modalities during data integration. The main steps of the proposed SURE algorithm are reported next. Let $X_1, \ldots, X_m, \ldots, X_M$, where $X_m \in \Re^{n \times d_m}$, be M modalities, $\mathcal{S}$ is the set of selected modalities and initially $\mathcal{S} = \varnothing$. The SURE algorithm constructs the joint eigenspace $\Psi(\widetilde{\mathbf{X}}_M)$ is given in Algorithm 4.1.

After the construction of $\Psi(\widetilde{\mathbf{X}}_M)$, the principal components of the integrated data are obtained by

$$\mathbf{Y} = U(\widetilde{\mathbf{X}}_M)\Sigma(\widetilde{\mathbf{X}}_M).$$

Finally, the rows of $(n \times k)$ matrix $\mathbf{Y}$ are clustered using k-means algorithm to obtain the cancer subtypes.

Algorithm 4.1 Proposed Algorithm: **SURE**

```
 1: for $m \leftarrow 1$ to $M$ do in parallel
 2:     Compute eigenspace: $\Psi(X_m)$ using SVD of $X_m$.
 3:     Perform $k$-means on the left subspace $U(X_m)$ of $X_m$.
 4:     Compute relevance: $\text{Rel}(X_m)$ using (4.28) .
 5: end for
 6: Compute pairwise concordance $\mathbb{C}\left(X_i, X_j\right), \forall i \neq j$.
 7: $X_\pi \leftarrow$ modality with maximum relevance.
 8: $m \leftarrow 1$; $\mathcal{S} = \{X_\pi\}$.
 9: $\widetilde{\mathbf{X}}_1 = X_\pi$; Initial eigenspace: $\Psi(\widetilde{\mathbf{X}}_1) \leftarrow \Psi(X_\pi)$.
10: for $m \leftarrow 1$ to $(M-1)$ do
11:     for each $X_j$ not added to joint eigenspace $\Psi(\widetilde{\mathbf{X}}_m)$ do
12:         Compute average concordance of $X_j$ with previously integrated modalities:
            $\bar{\mathbb{C}}(X_j) = 1/|\mathcal{S}| \sum_{\omega \in \mathcal{S}} \mathbb{C}(X_\omega, X_j)$.
13:     end for
14:     $X_l \leftarrow X_j$ with maximum average concordance.
15:     if $\bar{\mathbb{C}}(X_l) \geqslant \tau$, then
16:         Update $\Psi(\widetilde{\mathbf{X}}_{m+1}) = \Psi(\widetilde{\mathbf{X}}_m) \oplus \Psi(X_l)$ as follows:
17:         $\widetilde{\mathbf{X}}_{m+1} = \begin{bmatrix} \widetilde{\mathbf{X}}_m & X_l \end{bmatrix}$.
18:         $m \leftarrow m+1$; $\mathcal{S} = \mathcal{S} \cup \{X_l\}$.
19:         Compute $\mu(\widetilde{\mathbf{X}}_{m+1})$ using (4.10).
20:         Compute $\mathcal{I}$, $\mathcal{P}$, and $\mathcal{Q}$ using (4.11), (4.12), and (4.13), respectively.
21:         $\mathcal{G} \leftarrow$ Gram-Schmidt orthogonalization of $\mathcal{Q}$.
22:         $t \leftarrow$ number of columns of $\mathcal{G}$ having norm zero.
23:         $\Gamma \leftarrow$ first $(k-t)$ basis vectors of $\mathcal{G}$.
24:         Compute $\mathcal{R}(\widetilde{\mathbf{X}}_{m+1})$, $\Sigma(\widetilde{\mathbf{X}}_{m+1})$, and $V(\widetilde{\mathbf{X}}_{m+1})$ using SVD of (4.20).
25:         Compute $U(\widetilde{\mathbf{X}}_{m+1})$ from (4.14).
26:         Truncate the matrices $U(\widetilde{\mathbf{X}}_{m+1})$, $\Sigma(\widetilde{\mathbf{X}}_{m+1})$, and $V(\widetilde{\mathbf{X}}_{m+1})$ at rank $k$.
27:         $\Psi(\widetilde{\mathbf{X}}_{m+1}) = \langle \mu(\widetilde{\mathbf{X}}_{m+1}), U(\widetilde{\mathbf{X}}_{m+1}), \Sigma(\widetilde{\mathbf{X}}_{m+1}), V(\widetilde{\mathbf{X}}_{m+1}) \rangle$.
28:     else
29:         break
30: end for
31: $\Psi(\widetilde{\mathbf{X}}_M) \leftarrow \Psi(\widetilde{\mathbf{X}}_m)$.
```

4.3.4 Compuational Complexity

In the proposed algorithm, for each modality $X_m \in \Re^{n \times d_m}$, a SVD problem of size $(n \times d_m)$ is solved in step 2. Let $d_{max} = \max\{d_m\}$ and $d = \sum\limits_{m=1}^{M} d_m$. The SVD problems on the individual modalities are independent of each other and can be computed parallelly for all the modalities. This time complexity is bounded by the time required for the largest modality, that is, $\mathcal{O}(\min\{nd_{max}^2, n^2 d_{max}\}) = \mathcal{O}(n^2 d_{max})$, assuming $n < d_{max}$ due to the high dimension low sample size nature of the data sets. Similarly, performing k-means on the left subspace $U(X_m)$ of X_m and computation of its relevance $\text{Rel}(X_m)$ from the clustering solution, in steps 3 and 4 can be done for all the modalities in parallel. The

k-means clustering on $(n \times k)$ matrix $U(X_m)$ has time complexity of $\mathcal{O}(t_{max} nk^2)$, where t_{max} is the maximum number of iterations the k-means algorithm runs and $k << n$. Computation of $\text{Rel}(X_m)$ takes $\mathcal{O}(n)$ time, owing to the computation of within-cluster variance in $U(X_m)$. Thus, for M modalities, the time complexity of steps 1-5 is bounded by $\big(\mathcal{O}\left(M\left(n^2 d_{max} + t_{max} nk^2 + n\right)\right) =\big)$ $\mathcal{O}(Mn^2 d_{max})$, considering sequential construction of eigenspaces for different modalities.

After computation of individual eigenspaces in steps 1-5, concordance $\mathbb{C}$ between every pair of modalities is computed in step 6. This involves computation of normalized mutual information which takes $\mathcal{O}(k^2)$ time. Step 7 has time complexity of $\mathcal{O}\left(M\right)$ to find the modality with maximum relevance. Steps 8 and 9 are assignments operations which take $\mathcal{O}(1)$ time. For the remaining modalities, the loop in step 10 can execute at most $(M - 1)$ times. On m-th execution of the loop, there are $(M - m)$ candidate modalities for the eigenspace update. For each candidate modality, its average concordance $\bar{\mathbb{C}}$ with the formerly updated ones is computed in step 12. This has a complexity of $\mathcal{O}\left(m\right)$. For $(M - m)$ candidate modalities, the total complexity of steps 11-13 is $\mathcal{O}\left(m(M - m)\right)$. The one with maximum average concordance is chosen in $\mathcal{O}\left(M - m\right)$ time. If its average concordance $\bar{\mathbb{C}}$ is greater than threshold τ then the eigenspace is updated in steps 16-27.

During eigenspace update, steps 17-19 consist of concatenation and union operations which take at most $\mathcal{O}(d_{max})$ time. Step 20 takes $\mathcal{O}(nk^2)$ time to compute the matrices $\mathcal{I}$, $\mathcal{P}$, and $\mathcal{Q}$. The Gram-Schmidt orthogonalization in step 21 has complexity of $\mathcal{O}(nk^2)$ for $(n \times k)$ matrix $\mathcal{Q}$. To find t in step 22, the norm of the columns of $\mathcal{Q}$ is computed, which takes $\mathcal{O}(nk)$ time. Step 24 requires solving the SVD problem of (4.20) of the main article, which is of size at most $(2k \times d)$ and has time complexity of $\mathcal{O}(k^2 d)$. $U(\widetilde{\mathbf{X}}_{m+1})$ in step 25 computed in $\mathcal{O}(nk^2)$ time. Steps 26 and 27 have constant complexity of $\mathcal{O}(1)$. Hence, the total complexity of steps 16-27 for updating the eigenspace is $\big(\mathcal{O}(d_{max} + nk^2 + nk + k^2 d + nk^2) =\big)$ $\mathcal{O}(k^2 d)$. Therefore, time complexity of updating the eigenspace in m-th iteration of the loop in step 10 is $\big(\mathcal{O}(m(M - m) + k^2 d) =\big)$ $\mathcal{O}(k^2 d)$. Step 10 is executed at most $(M - 1)$ times which gives a total complexity of $\mathcal{O}(Mk^2 d)$. The overall computational complexity of the proposed SURE algorithm is $\big(\mathcal{O}(Mn^2 d_{max} + Mk^2 d) =\big)$ $\mathcal{O}(Mn^2 d_{max})$, assuming $M, k << n < d_{max}$. Thus the time complexity is bounded by that of individual eigenspace construction in steps 1-5.

4.4 Accuracy of Eigenspace Construction

This section introduces some quantitative indices to measure the gap between “full-rank" eigenspace of the integrated data and its approximate eigenspace constructed by the proposed SURE algorithm. Let $\mathbf{X}$ be the integrated data given by (4.3). The full-rank eigenspace of $\mathbf{X}$ contains the full-rank information of all its component modalities and constructed by the SVD of $\mathbf{X}$ using (4.1). Let its rank r eigenspace of $\mathbf{X}$ be given by

$$\Psi(\mathbf{X}) = \langle \mu(\mathbf{X}), U(\mathbf{X}^r), \Sigma(\mathbf{X}^r), V(\mathbf{X}^r) \rangle. \tag{4.31}$$

The superscript r denotes that r largest singular values and corresponding singular vectors are considered in the eigenspace. This full-rank eigenspace representation is also same as the principal subspace extracted by PCA on the integrated data $\mathbf{X}$. Let $\Psi(\widetilde{\mathbf{X}})$ be the approximate rank r eigenspace of $\mathbf{X}$ obtained by the proposed SURE algorithm, that is,

$$\Psi(\widetilde{\mathbf{X}}) = \bigoplus_{m=1}^{M} \Psi(X_m),$$

where $\Psi(X_m)$ is the rank r eigenspace for modality X_m. It is further assumed that all the M modalities are used during the eigenspace update. Let $\Psi(\widetilde{\mathbf{X}})$ be given by

$$\Psi(\widetilde{\mathbf{X}}) = \langle \mu(\mathbf{X}), U(\widetilde{\mathbf{X}}^r), \Sigma(\widetilde{\mathbf{X}}^r), V(\widetilde{\mathbf{X}}^r) \rangle. \tag{4.32}$$

Here, $\Psi(\widetilde{\mathbf{X}})$ is an approximate eigenspace of $\mathbf{X}$ as it is constructed from truncated rank r individual eigenspaces. The truncation errors, inherent in individual eigenspaces, get propagated onto joint eigenspace during the updating process. This results in a gap between full-rank eigenspace $\Psi(\mathbf{X})$ and approximate eigenspace $\Psi(\widetilde{\mathbf{X}})$. However, as r increases, the truncation errors in the individual eigenspaces reduce and the gap decreases. So, the gap between two eigenspaces can be computed for different values of rank r. For any $r' > r$, an eigenspace of rank r' has more singular values and vectors in its Σ, U, and V components than an eigenspace of rank r. So, for different values of r, the gap is always measured between fixed number of singular values and vectors of two eigenspaces.

4.4.1 Error Bound on Principal Sines

The gap between left and right subspaces can be measured using the principal angles between subspaces (PABS) [16]. PABS generalizes the concept of angle between two lines to a set of angles between two subspaces, defined next.

Definition 4.1. *Let $\mathcal{A}$ and $\mathcal{B}$ be two subspaces of $\Re^n$ of dimension r_1 and r_2, respectively. Let $t = min(r_1, r_2)$. The principal angles between subspaces $\mathcal{A}$ and $\mathcal{B}$ are given by a sequence of t angles, $\Theta(\mathcal{A}, \mathcal{B}) = [\theta_1, \ldots, \theta_j, \ldots, \theta_t]$, where $0 \leqslant \theta_1 \leqslant \ldots \leqslant \theta_t \leqslant \pi/2$. The angle θ_j is defined by*

$$\theta_j = \underset{a \in \mathcal{A}}{max} \; \underset{b \in \mathcal{B}}{max} \arccos\left(|a^T b|\right);$$

subject to $||a|| = ||b|| = 1$, $a_i^T a = 0$, $b_i^T b = 0$, for $i = 1, 2, ..., j-1$ [16].

The principal sines $\sin(\theta_j)'s$ of the angles can be computed using singular values as follows.

Theorem 4.2. *Let the columns of matrices $A \in \Re^{n \times r_1}$ and $B \in \Re^{n \times r_2}$ be orthonormal bases for subspaces $\mathcal{A}$ and $\mathcal{B}$, respectively. Let $\begin{bmatrix} A & A^{\perp} \end{bmatrix}$ be a unitary matrix such that the columns of $A^{\perp}$ span the subspace orthogonal to $\mathcal{A}$. Also, let the singular values of $(A^{\perp})^T B$ be given by the elements of the diagonal matrix $\Xi = diag\left(\nu_1, \ldots, \nu_t\right)$, where $\nu_1 \geqslant \ldots \geqslant \nu_j \geqslant \ldots \geqslant \nu_t$. The principal sine $\sin(\theta_{t+1-j}) = \nu_j$* [106, 116].

Thus, the principal sines between subspaces $\mathcal{A}$ and $\mathcal{B}$ are given by the singular values of $(A^{\perp})^T B$. The principal sines can be used to define a notion of difference between two subspaces.

Definition 4.2. *Let $\mathcal{A}$ and $\mathcal{B}$ be two subspaces of $\Re^n$. Let the diagonal matrix Ξ contains the singular values of $(A^{\perp})^T B$ as in Theorem 4.2. The measure of difference between two subspaces $\mathcal{A}$ and $\mathcal{B}$ is defined by* $\sin\Theta(\mathcal{A},\mathcal{B}) \overset{\text{def}}{=} \Xi$ [202].

The squared Frobenius norm of a matrix, denoted by $\| \, . \, \|_F^2$, is the sum of squares of its singular values. So, using Theorem 4.2 and Definition 4.2, we get

$$\| \sin\Theta(\mathcal{A},\mathcal{B}) \|_F^2 = \| \; \Xi \; \|_F^2 = \sum_{j=1}^{t} \nu_j^2 = \sum_{j=1}^{t} \sin^2\left(\theta_{t+1-j}\right). \tag{4.33}$$

Hence, (4.33) implies that the sum of squares of the principal sines between two subspaces $\mathcal{A}$ and $\mathcal{B}$ is given by $\| \sin\Theta(\mathcal{A},\mathcal{B}) \|_F^2$.

The gaps between two left subspaces $U(\mathbf{X}^r)$ and $U(\widetilde{\mathbf{X}}^r)$ and two right subspaces $V(\mathbf{X}^r)$ and $V(\widetilde{\mathbf{X}}^r)$ are computed using the sum of squares of the principal sines between the two sets of subspaces. The matrices $U(\mathbf{X}^r)$ and $U(\widetilde{\mathbf{X}}^r)$ are themselves orthonormal bases of rank r for the corresponding left subspaces. Let the principal angles between subspaces $U(\mathbf{X}^r)$ and $U(\widetilde{\mathbf{X}}^r)$ be given by $\theta_1, \ldots, \theta_r$ and the singular values of $U(\mathbf{X}^{r\perp})^T U(\widetilde{\mathbf{X}}^r)$ be given by $\gamma_1, \ldots, \gamma_r$, arranged in decreasing order, where columns of $U(\mathbf{X}^{r\perp})$ span the subspace orthogonal to one spanned by $U(\mathbf{X}^r)$. Then, following Theorem 4.2 and Definition 4.1, the sum of squared principal sines between two left subspaces $U(\mathbf{X}^r)$ and $U(\widetilde{\mathbf{X}}^r)$ is given by

$$\| \sin\Theta(U(\mathbf{X}^r), U(\widetilde{\mathbf{X}}^r)) \|_F^2 = \sum_{i=1}^{r} \gamma_i^2 = \sum_{i=1}^{r} \sin^2\left(\theta_{r+1-i}\right).$$

Similarly, for two right subspaces $V(\mathbf{X}^r)$ and $V(\widetilde{\mathbf{X}}^r)$, let the principal angles between them be given by $\phi_1, \ldots, \phi_r$ and the singular values of $V(\mathbf{X}^{r\perp})^T V(\widetilde{\mathbf{X}}^r)$ be given by $\omega_1, \ldots, \omega_r$, arranged in decreasing order, where columns of $V(\mathbf{X}^{r\perp})$ span the subspace orthogonal to $V(\mathbf{X}^r)$. Then, sum of squared principal sines between two right subspaces is given by

$$\| \sin\Theta(V(\mathbf{X}^r), V(\widetilde{\mathbf{X}}^r)) \|_F^2 = \sum_{j=1}^{r} \omega_j^2 = \sum_{j=1}^{r} \sin^2\left(\phi_{r+1-j}\right).$$

The cumulative gap between full-rank and approximate pairs of left and right subspaces is given by the root mean squared principal sines between them, which is given by

$$\mathit{Gap}\Theta\left(\mathbf{X}^r, \widetilde{\mathbf{X}}^r\right) = \left[\frac{1}{2r}\left\{\| \sin\Theta(U(\mathbf{X}^r), U(\widetilde{\mathbf{X}}^r)) \|_F^2 + \| \sin\Theta(V(\mathbf{X}^r), V(\widetilde{\mathbf{X}}^r)) \|_F^2\right\}\right]^{\frac{1}{2}}. \tag{4.34}$$

Since the principal angles $\theta_i^{\prime}s$ and $\phi_j^{\prime}s$ lie in $[0, \pi/2]$, $\sin^2\theta_i^{\prime}s$ and $\sin^2\phi_j^{\prime}s$ lie in $[0, 1]$ and $\mathit{Gap}\Theta$ also lies in $[0, 1]$. If the approximate left and right subspaces $U(\widetilde{\mathbf{X}}^r)$ and $V(\widetilde{\mathbf{X}}^r)$ are close approximations of the full-rank ones, then $\theta_i^{\prime}s$ and $\phi_j^{\prime}s$ are close to 0. This implies that a value of $\mathit{Gap}\Theta$ close to 0 indicates a better approximation.

Next, upper bound on the value of $\mathit{Gap}\Theta\left(\mathbf{X}^r, \widetilde{\mathbf{X}}^r\right)$ is derived as a function of rank r of the singular subspaces. Without loss of generality, let us assume that the individual modalities X_m's are mean centered and have dimension $(n \times d_m)$, where $n \leqslant d_m$. The SVD of a modality X_m can be partitioned as:

$$\begin{aligned} X_m &= U(X_m)\Sigma(X_m)V(X_m)^T \\ &= \begin{bmatrix} U(X_m^r) & U(X_m^{r\perp}) \end{bmatrix} \begin{bmatrix} \Sigma(X_m^r) & \mathbf{0} \\ \mathbf{0} & \Sigma(X_m^{r\perp}) \end{bmatrix} \begin{bmatrix} V(X_m^r)^T \\ V(X_m^{r\perp})^T \end{bmatrix} \\ &= U(X_m^r)\Sigma(X_m^r)V(X_m^r)^T + U(X_m^{r\perp})\Sigma(X_m^{r\perp})V(X_m^{r\perp})^T \\ &= X_m^r + X_m^{r\perp}, \end{aligned} \tag{4.35}$$

where $\Sigma(X_m^r) = diag(\lambda_m^1, \ldots, \lambda_m^r)$ consists of r largest singular values of X_m, and $U(X_m^r)$ and $V(X_m^r)$ contain the corresponding r left and right singular vectors in their columns, respectively. Similarly, $\Sigma(X_m^{r\perp})$ contains the remaining $(n-r)$ singular values $\lambda_m^{r+1}, \ldots, \lambda_m^n$, while $U(X_m^{r\perp})$ and $V(X_m^{r\perp})$ contain the corresponding singular vectors. Thus, X_m^r is the rank r approximation of X_m using the r largest singular triplets, and $X_m^{r\perp}$ is the approximation using the remaining $(n-r)$ singular triplets. Using (4.35), the integrated data matrix $\mathbf{X}$ in (4.3) can be decomposed as

$$\begin{aligned} \mathbf{X} &= \begin{bmatrix} X_1 & \ldots & X_m & \ldots & X_M \end{bmatrix} \\ &= \begin{bmatrix} (X_1^r + X_1^{r\perp}) & \ldots & (X_m^r + X_m^{r\perp}) & \ldots & (X_M^r + X_M^{r\perp}) \end{bmatrix} \\ &= \begin{bmatrix} X_1^r & \ldots & X_M^r \end{bmatrix} + \begin{bmatrix} X_1^{r\perp} & \ldots & X_M^{r\perp} \end{bmatrix} \\ &= \mathbf{X}^r + \mathbf{X}^{r\perp}. \end{aligned} \tag{4.36}$$

Thus, $\mathbf{X}$ is the full-rank integrated data and $\mathbf{X}^r$ is its approximation using rank r approximations of the individual modalities. The SVD of $\mathbf{X}$ is used to obtain the full-rank eigenspace $\Psi(\mathbf{X})$ in (4.31). On the other hand, the proposed algorithm constructs the approximate eigenspace $\Psi(\widetilde{\mathbf{X}})$ for data matrix $\mathbf{X}^r$ by iteratively updating the rank r eigenspaces of the individual modalities $\Psi(X_m)$'s. Let the SVD of $\mathbf{X}$ be partitioned as

$$\begin{aligned} \mathbf{X} &= U(\mathbf{X})\Sigma(\mathbf{X})V(\mathbf{X})^T \\ &= \begin{bmatrix} U(\mathbf{X}^r) & U(\mathbf{X}^{r\perp}) \end{bmatrix} \begin{bmatrix} \Sigma(\mathbf{X}^r) & \mathbf{0} \\ \mathbf{0} & \Sigma(\mathbf{X}^{r\perp}) \end{bmatrix} \begin{bmatrix} V(\mathbf{X}^r)^T \\ V(\mathbf{X}^{r\perp})^T \end{bmatrix} \end{aligned} \tag{4.37}$$

and the SVD of $\mathbf{X}^r$ obtained by eigenspace update be partitioned as

$$\begin{aligned} \mathbf{X}^r &= U(\widetilde{\mathbf{X}})\Sigma(\widetilde{\mathbf{X}})V(\widetilde{\mathbf{X}})^T \\ &= \begin{bmatrix} U(\widetilde{\mathbf{X}}^r) & U(\widetilde{\mathbf{X}}^{r\perp}) \end{bmatrix} \begin{bmatrix} \Sigma(\widetilde{\mathbf{X}}^r) & \mathbf{0} \\ \mathbf{0} & \Sigma(\widetilde{\mathbf{X}}^{r\perp}) \end{bmatrix} \begin{bmatrix} V(\widetilde{\mathbf{X}}^r)^T \\ V(\widetilde{\mathbf{X}}^{r\perp})^T \end{bmatrix} \end{aligned} \tag{4.38}$$

where $U(\mathbf{X}^r), U(\widetilde{\mathbf{X}}^r) \in \Re^{n \times r}$, $V(\mathbf{X}^r), V(\widetilde{\mathbf{X}}^r) \in \Re^{d \times r}$, and

$$\begin{aligned} &\Sigma(\mathbf{X}^r) = diag(\sigma_1, \ldots, \sigma_r),\ \Sigma(\mathbf{X}^{r\perp}) = diag(\sigma_{r+1}, \ldots, \sigma_n), \\ &\Sigma(\widetilde{\mathbf{X}}^r) = diag(\widetilde{\sigma}_1, \ldots, \widetilde{\sigma}_r),\ \Sigma(\widetilde{\mathbf{X}}^{r\perp}) = diag(\widetilde{\sigma}_{r+1}, \ldots, \widetilde{\sigma}_n). \end{aligned}$$

According to (4.36), $\mathbf{X} = \mathbf{X}^r + \mathbf{X}^{r\perp}$, therefore, using matrix perturbation theory [202], the integrated data matrix $\mathbf{X}$ can be viewed as a perturbation of its rank r approximation $\mathbf{X}^r$ due to the presence of error component $\mathbf{X}^{r\perp}$. Next, Wedin's $\sin\Theta$ *theorem* [242] can be used to bound the principal angles between the rank r left and right singular subspaces of a matrix and its perturbation. Let the residuals of left and right subspaces be

$$\begin{aligned} \mathbb{R}_L &= \mathbf{X}^r V(\mathbf{X}^r) - U(\mathbf{X}^r)\Sigma(\mathbf{X}^r); \\ \text{and} \quad \mathbb{R}_R &= (\mathbf{X}^r)^T U(\mathbf{X}^r) - V(\mathbf{X}^r)\Sigma(\mathbf{X}^r). \end{aligned}$$

Let δ be defined as

$$\delta \overset{\text{def}}{=} \min \left\{ \min_{1\leqslant i\leqslant r, 1\leqslant j\leqslant (n-r)} |\sigma_i - \widetilde{\sigma}_{r+j}|, \min_{1\leqslant i\leqslant r} \sigma_i \right\}.$$

Wedin's $\sin\Theta$ theorem states that if $\delta > 0$, then

$$\sqrt{\| \sin\Theta(U(\mathbf{X}^r), U(\widetilde{\mathbf{X}}^r)) \|_F^2 + \| \sin\Theta(V(\mathbf{X}^r), V(\widetilde{\mathbf{X}}^r)) \|_F^2} \leqslant \frac{\sqrt{\| \mathbb{R}_L \|_F^2 + \| \mathbb{R}_R \|_F^2}}{\delta}.$$

$$\text{So,} \quad Gap\Theta\left(\mathbf{X}^r, \widetilde{\mathbf{X}}^r\right) \leqslant \frac{\sqrt{\| \mathbb{R}_L \|_F^2 + \| \mathbb{R}_R \|_F^2}}{\sqrt{2r}\delta}. \tag{4.39}$$

The above relation states that the cumulative sum of squares of the principal sines between the full-rank and approximate left and right subspaces is bounded in terms of the Frobenius norm of the residual matrices $\mathbb{R}_L$ and $\mathbb{R}_R$, and the minimum difference between full-rank and approximate sets of singular values, δ.

As the value of rank r approaches the full rank n, the residual component $\mathbf{X}^{r\perp} \to 0$ and $\mathbf{X}^r \to \mathbf{X}$. Similarly, the components $U(\mathbf{X}^r)$, $\Sigma(\mathbf{X}^r)$, and $V(\mathbf{X}^r)$ also tend towards $U(\mathbf{X})$, $\Sigma(\mathbf{X})$, and $V(\mathbf{X})$, respectively. So,

$$\begin{aligned} \lim_{r\to n} \mathbb{R}_L &= \lim_{r\to n} \mathbf{X}^r V(\mathbf{X}^r) - U(\mathbf{X}^r)\Sigma(\mathbf{X}^r) \\ &= \lim_{r\to n} \mathbf{X} V(\mathbf{X}) - U(\mathbf{X})\Sigma(\mathbf{X}) \\ &= \lim_{r\to n} U(\mathbf{X})\Sigma(\mathbf{X})V(\mathbf{X})^T V(\mathbf{X}) - U(\mathbf{X})\Sigma(\mathbf{X}) = 0. \end{aligned}$$

Similarly, $\lim_{r\to n} \mathbb{R}_R = 0$. Substituting the limiting values of $\mathbb{R}_L$ and $\mathbb{R}_R$ in (4.39), we get

$$\lim_{r\to n} Gap\Theta\left(\mathbf{X}^r, \widetilde{\mathbf{X}}^r\right) = 0.$$

This implies that as the approximation rank r approaches the full rank n, the principal angles between full-rank and approximate pairs of left and right subspaces reduce to 0.

4.4.2 Accuracy of Singular Triplets

This subsection introduces two more quantitative indices to evaluate the difference between U, Σ, and V components of full-rank and approximate eigenspaces.

4.4.2.1 Mean Relative Difference of Singular Values

For any rank r, both $\Sigma(\mathbf{X}^r)$ and $\Sigma(\widetilde{\mathbf{X}}^r)$ consist of r largest singular values. The relative difference between the singular values in $\Sigma(\mathbf{X}^r)$ and $\Sigma(\widetilde{\mathbf{X}}^r)$, with respect to the singular values of $\Sigma(\mathbf{X}^r)$, is given by a sequence $\mathcal{H} = [\lambda_1, \ldots, \lambda_i, \ldots, \lambda_r]$, where

$$\lambda_i = \frac{\Sigma(\mathbf{X}^r)_i - \Sigma(\widetilde{\mathbf{X}}^r)_i}{\Sigma(\mathbf{X}^r)_i}; \tag{4.40}$$

$\Sigma(.)_i$ is the i-th largest singular value of the respective eigenspace. The singular values capture the spread of the data along the principal axes. The maximum spread, captured by the singular values in $\Sigma(\widetilde{\mathbf{X}}^r)$, is bounded by spread captured by the top r components of the individual eigenspaces. This is much less than the actual spread of samples in $\mathbf{X}$, which is reflected in $\Sigma(\mathbf{X}^r)$. Hence, $\Sigma(\mathbf{X}^r)_i \geqslant \Sigma(\widetilde{\mathbf{X}}^r)_i$, so the value of λ_i lies in $[0, 1]$. A value of λ_i close to 0 indicates less difference between the i-th component of the two eigenspaces. A cumulative measure of the gap between $\Sigma(\mathbf{X}^r)$ and $\Sigma(\widetilde{\mathbf{X}}^r)$ is given by the mean of first h values of $\mathcal{H}$ as follows:

$$\text{DiffSV}\left(\mathbf{X}^r, \widetilde{\mathbf{X}}^r\right) = \frac{1}{h}\sum_{i=1}^{h} \lambda_i. \tag{4.41}$$

The value of DiffSV also lies in $[0, 1]$, with a value closer to 0 indicating a better approximation.

4.4.2.2 Relative Dimension of Intersection Space

Let us assume that r' is the dimension of the space lying in the intersection of two left subspaces $U(\mathbf{X}^r)$ and $U(\widetilde{\mathbf{X}}^r)$. According to Theorem 4.1, reported in Section 4.3.1, r' is the number of singular values of $U(\mathbf{X}^r)^T U(\widetilde{\mathbf{X}}^r)$ having value 1. The relative dimension of intersection space between two left subspaces is defined as the ratio of the dimension of intersection space and that of the left subspace $U(\mathbf{X}^r)$, which is as follows:

$$\text{DimIS}\left(\mathbf{X}^r, \widetilde{\mathbf{X}}^r\right) = \frac{r'}{r}; \tag{4.42}$$

where $r' \leqslant r$. So, the value of DimIS lies in $[0, 1]$. If the overlap between two left subspaces is high, the dimension of the intersection subspace r' is close to r. Thus, the value of DimIS close to 1 indicates lower gap between the two left subspaces. Similarly, DimIS between two right subspaces $V(\mathbf{X}^r)$ and $V(\widetilde{\mathbf{X}}^r)$ can be calculated using the number of singular values of $V(\mathbf{X}^r)^T V(\widetilde{\mathbf{X}}^r)$ having value 1.

4.5 Experimental Results and Discussion

The proposed SURE algorithm is used to extract a low-rank joint subspace of the integrated data. The clustering performance of the extracted subspace is studied and compared with several existing integrative clustering approaches. The approaches compared are Bayesian consensus clustering (BCC) [140], cluster of cluster analysis (COCA) [93], PCA on naively concatenated data (PCA-Con) [6], joint and individual variance explained (JIVE) [141], A-JIVE [63], iCluster [192], LRAcluster [243], and NormS [111] (proposed in Chapter 3). The performance of JIVE is reported considering both permutation test (JIVE-Perm) and Bayesian information criteria (JIVE-BIC) for rank selection. The experimental setup for the existing approaches is followed same as that of Chapter 3. It is also described in the supplementary material of [111]. The source code of the proposed SURE algorithm, written in R language, is available at `https://github.com/Aparajita-K/SURE`.

To evaluate the performance of different clustering algorithms, six external cluster evaluation indices, namely, accuracy, normalized mutual information (NMI), adjusted Rand index (ARI), F-measure, Rand index, and purity, are used, which compare the identified subtypes with the established subtypes. The indices are described in Appendix B. For all six indices, a value close to one indicates that the identified subtypes have close resemblance with the previously established ones. Two other performance measures, namely, p-value of Cox log-rank test [96] and p-value of Peto & Peto's modification of the Gehan-Wilcoxon test [172], are also considered to evaluate the significance of the differences in survival profiles of the identified subtypes.

Multimodal omics data for seven types of cancers, namely, cervical carcinoma (CESC), glioblastoma multiforme (GBM), lower grade glioma (LGG), lung carcinoma (LUNG) and kidney carcinoma (KIDNEY), ovarian carcinoma (OV), and breast invasive carcinoma (BRCA), are obtained from TCGA (`http://cancergenome.nih.gov/`), having 124, 168, 267, 671, 737, 334, and 398 samples, respectively. By comprehensive integrated analysis, TCGA Research Network has identified three molecular subtypes of both CESC [218] and LGG [217], and four subtypes of OV [215] and BRCA [214]. Four subtypes of GBM were identified by Veerhak *et al.* [228]. The samples of LUNG and KIDNEY data sets are divided into two and three subtypes, respectively, based on the tissue of origin. The CESC, LGG, KIDNEY, and LUNG data sets have four different modalities, namely, gene expression (RNA), DNA methylation (mDNA), miRNA expression (miRNA), and reverse phase protein array expression (RPPA), while the GBM data set has three modalities, namely, RNA, miRNA, and copy number variation (CNV). A brief description of these data sets is provided in Appendix A.

4.5.1 Optimum Value of Concordance Threshold

The threshold parameter τ of the proposed SURE algorithm (in step 15 of Algorithm 4.1) decides whether the a remaining individual eigenspaces will be considered for updating the current joint eigenspace. At each iteration of joint eigenspace construction, the modality having maximum average concordance $\bar{\mathtt{C}}$, with respect to pre-selected modalities, is taken into consideration. The joint eigenspace is updated only if the value of $\bar{\mathtt{C}}$ is beyond some threshold τ. This threshold prevents modalities having low concordance or shared information with the previously updated ones from being integrated into the joint eigenspace.

Given M modalities, different subsets of modalities get selected for different values of threshold τ. For each data set, the value of τ is varied in the range $[0, 0.95]$ at an interval of 0.05. For each value of threshold τ, the PVE by a k partition of the final joint subspace is evaluated, which is denoted by PVE_τ. The optimum value τ^* for each data set is chosen using the following relation:

$$\tau^* = \arg\max_{\tau} \; \{\text{PVE}_\tau\}. \tag{4.43}$$

It is worth noting that the upper bound for varying τ is 0.95 instead of 1.00. For $\tau = 1.00$, a candidate modality has to have full concordance or agreement in cluster structure with all the previously integrated ones. For real-life omics data sets, this is highly unlikely, and hence no candidate modality will ever get selected for updating the eigenspace. So, for $\tau = 1.00$, a unimodal solution, consisting of only the most relevant modality, will be considered always. As integration of multiple modalities can capture the biological variations across multiple genomic levels, the threshold τ is upper bounded at 0.95 in order to prefer multiple modalities.

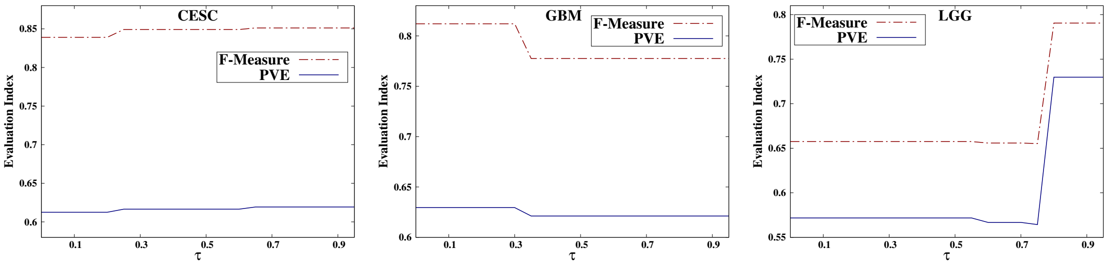
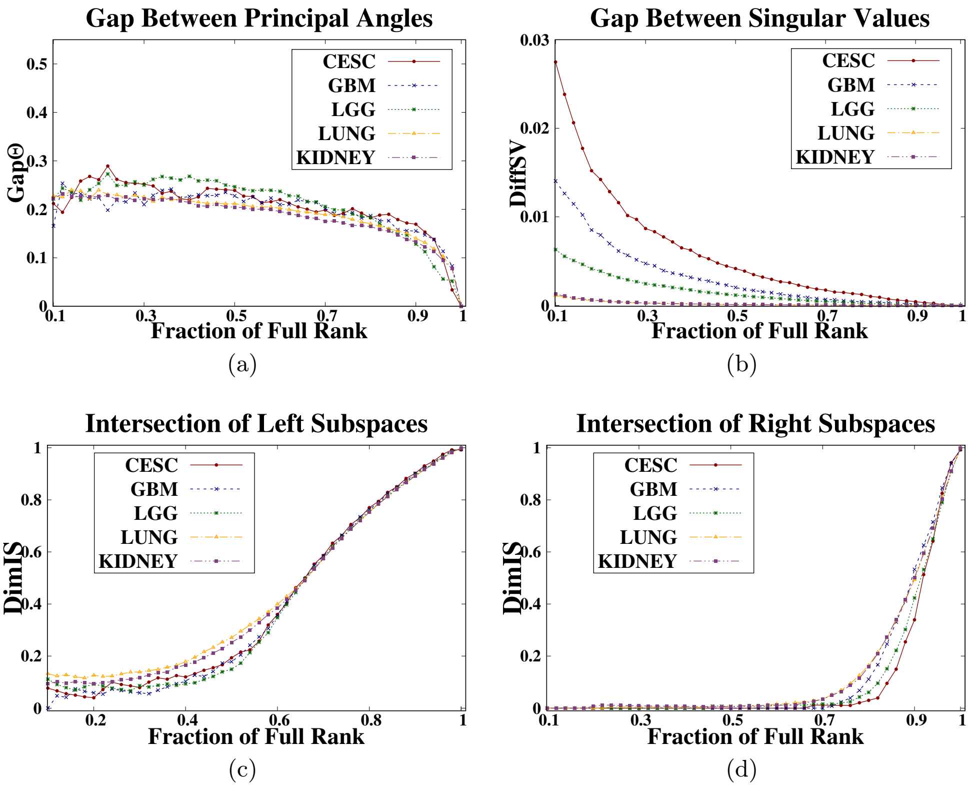


Figure 4.2: Variation of PVE and F-measure for different values of threshold τ for CESC, GBM, and LGG data sets.

Figure 4.2 shows the variation of F-measure and PVE for different values of τ for CESC, GBM, and LGG data sets, as examples. From Figure 4.2, it is seen that the values of F-measure and PVE vary in a similar fashion with the change in τ. The PVE is calculated based on the generated clusters, while the F-measure is computed based on the ground truth subtype information. Since these two indices are found to vary similarly, the optimal value of τ inferred from PVE also gives the optimal value of F-measure, thus giving good clustering performance. For each data set, the best value of F-measure, obtained from all possible values of threshold τ, is compared with that obtained for optimal threshold τ^*. For all data sets, the best F-measure is exactly same with the F-measure corresponding to τ^*.

4.5.2 Accuracy of Subspace Representation

The proposed SURE algorithm constructs the joint subspace of the integrated data from individual principal subspaces, using a eigenspace update approach. The extracted joint subspace is an approximation of the principal subspace extracted by PCA on the integrated data matrix. Three quantitative indices, namely, $Gap\Theta$, DiffSV, and DimIS, are proposed in Section 4.4 to evaluate the gap between full-rank and approximate eigenspaces.



(a)

(b)

(c)

(d)

Figure 4.3: Different quantitative indices for the evaluation of gap between true and approximate eigenspaces.

To observe the variation in gap between these two eigenspaces with the increase in rank parameter r, the three proposed indices are evaluated for different values of rank r. Due to the high dimension and low sample size nature of the data sets, the full rank of the integrated data matrix is always bounded by the number of samples. So, for each data set, the indices are evaluated for different fractions of the full rank of the integrated data. The value of the h parameter for the DiffSV is set to be 10, which implies that the gap between singular values is measured between the top 10 components of the two eigenspaces. The variation of these quantitative indices, with increase in rank, is shown in Figure 4.3 for different data sets.

While Figure 4.3(a) shows the root mean squared principal sines between the left and right subspaces of full-rank and approximate eigenspaces, Figure 4.3(b) shows the difference between their singular values. Figure 4.3(b) shows that the difference between singular values monotonically decreases to 0 with the increase in rank, for all data sets. Figure 4.3(a) shows that the difference between the singular subspaces, in terms of their principal sines, also converges to 0. However, the change in variation in case of singular subspaces is not monotonically decreasing as of singular values in Figure 4.3(b). For some of the smaller values of rank r, the difference also increases between two consecutive values. This is due to the fact that, for a given value of r, there can be infinitely many rank r subspaces of an n-dimensional vector space. For smaller values of r, the rank r singular subspaces of individual modalities can be very different from each other due to the large number of possibilities. Consequently, the approximate singular subspace, constructed from these individual subspaces, tends to vary a lot from the full-rank subspace. However, as r approaches the full rank n, the number of possible subspaces reduces and the difference between them converges to 0.

Figure 4.3(c) shows that the intersection between two left subspaces increases gradually and uniformly with the increase in rank r. But, for the right subspaces, as seen in Figure 4.3(d), intersection continues to remain almost 0 for all data sets, until the rank considered for eigenspace is more than 70% of the full rank. This implies that there is more gap between the pair of right subspaces compared to that of left ones. This difference in gap arises because the right subspaces consist of loadings from different sets of variables in different modalities, while the left subspaces consist of the projections of same set of samples across all the modalities. The disjointness of variables in the right subspaces leads to larger gap between the pair of right subspaces.

4.5.3 Execution Efficiency of SURE

One major advantage of the proposed algorithm is that it extracts the principal subspace of the integrated data matrix by iteratively updating the principal subspaces of the individual modalities, and its time complexity is $\mathcal{O}(n^2 d_{max})$. On the other hand, the time complexity of performing PCA on the integrated data matrix using eigenvalue decomposition (EVD) of the covariance matrix is $\mathcal{O}(d^3)$, while that using SVD of mean-centered data matrix is $\mathcal{O}(n^2 d)$, where $n << d_{max} << d$. This makes the proposed algorithm particularly efficient for PCA based dimensionality reduction of large multimodal data sets. Figure 4.4 compares the execution time of the proposed SURE algorithm with that for extracting the principal components using EVD and SVD for LGG, LUNG, and KIDNEY data sets. The RNA and mDNA modalities have large number of features such as 20,502 and 25,978, respectively. The variation in execution time for extracting top k principal components using these three algorithms is observed by gradually increasing the number of features from RNA and mDNA modalities. The plots in Figure 4.4(a) - (c) show that the execution time of PCA computed using EVD increases quadratically with respect the proposed SURE approach. This is because PCA using EVD takes $\mathcal{O}(d^3)$ time which is significantly higher compared to $\mathcal{O}(n^2 d_{max})$. Figure 4.4(d) - (f) show that the execution time of PCA using SVD as well as of the proposed SURE algorithm increases linearly with increase in number of features. However, SURE takes significantly lesser time to extract the principal components as compared to PCA using SVD, especially for large data sets like LUNG and KIDNEY with 671 and 757 samples, respectively.

4.5.4 Importance of Data Integration and Modality Selection

To establish the importance of data integration, the clustering performance on top k principal components of individual modalities is compared with that of the rank k joint subspace extracted by the proposed algorithm. There can be a total of $(2^M - M - 1)$ possible combinations of two or more modalities from M modalities. Each multimodal combination gives a different clustering solution. Therefore, the clustering performance of the top k principal components of each multimodal combination is evaluated using Silhouette index [179].

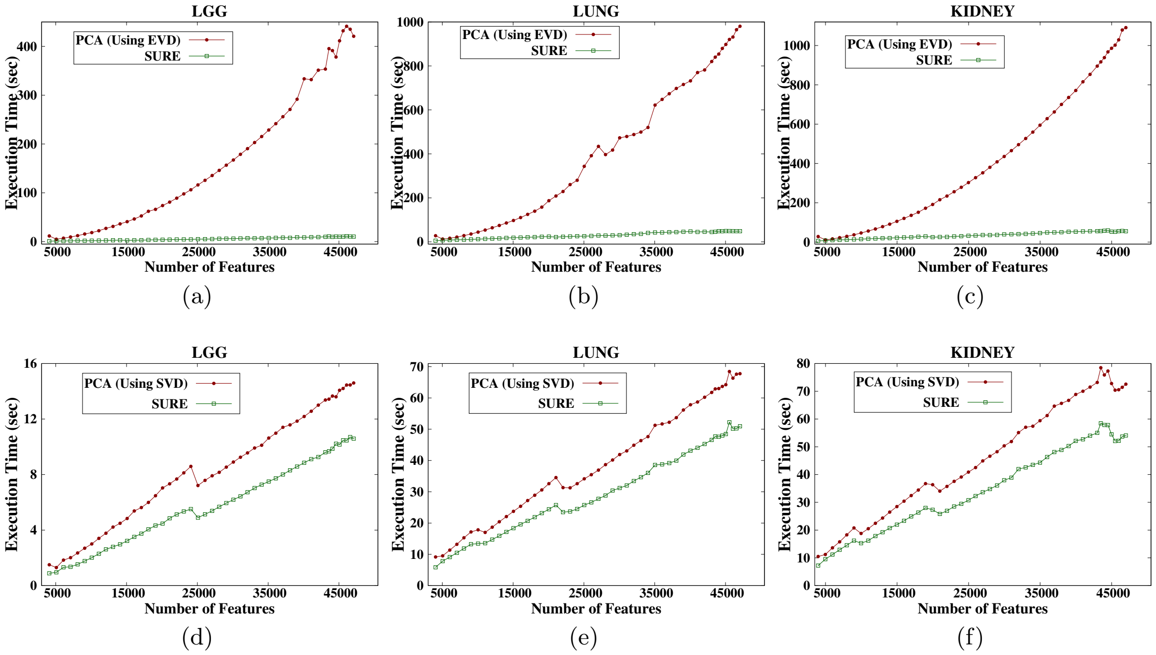


Figure 4.4: Comparison of execution time for PCA computed using EVD (top row) and SVD (bottom row) and the proposed SURE approach on LGG, LUNG, and KIDNEY data sets.

The best combination is chosen to be the one with maximum value of Silhouette index. To evaluate the strength of the proposed SURE algorithm in selecting appropriate subset of modalities, its performance is compared with that of PCA on the best combination of modalities, henceforth termed as PCA_Combine. The comparative performance of the individual modalities, the best multimodal combination, and the proposed SURE approach is reported in Table 4.1 for CESC, GBM, LGG, LUNG, and KIDNEY data sets, as examples.

Table 4.1 shows that the joint subspace extracted by the SURE algorithm gives better performance compared to all the unimodal solutions for four data sets, namely, CESC, GBM, LUNG, and KIDNEY, in terms of four external evaluation indices. This establishes the significance of integrative analysis over unimodal analysis. For LGG data set, the mDNA gives the best performance among all possible unimodal and multimodal combinations. The SURE algorithm also efficiently chooses only mDNA to construct the final eigenspace. For GBM and LUNG data sets, the modalities selected by SURE algorithm are same as the best combination of modalities obtained for PCA. The combination differs for CESC and KIDNEY data sets, however, the performance of SURE is always better as compared to PCA_Combine. This is due to the fact that the individual eigenspaces in the proposed algorithm are truncated at rank k, thus filtering out the noisy information present in them. The joint subspace constructed from these informative truncated eigenspaces preserves better cluster structure compared to PCA_Combine that considers the complete information of each eigenspace. The results in Table 4.1 also show that the performance of SURE is atleast as good as that of PCA on best combination of modalities for all data sets. This establishes that the proposed SURE approach is able to select the

Table 4.1: Comparative Performance Analysis of Individual Modalities, PCA Combinations, and SURE

	Modality/ Algorithm	Accuracy	NMI	ARI	F-Measure	Rand	Purity
CESC	mDNA	0.5241935	0.2420431	0.1175554	0.5453798	0.5819565	0.5806452
	RNA	0.8467742	0.6242327	0.6168352	0.8310850	0.8164175	0.8467742
	miRNA	0.5564516	0.1589298	0.1512301	0.5697384	0.6087071	0.5887097
	RPPA	0.5000000	0.0803494	0.0917321	0.5166786	0.5847102	0.5322581
	PCA_subset	0.8145161	0.5868124	0.5579264	0.7882956	0.7844217	0.8145161
	SURE	**0.8629032**	**0.6461946**	**0.6507274**	**0.8512028**	**0.833989**	**0.8629032**
	Best PCA subset:			RNA, miRNA, RPPA			
	Subset selected by SURE:			RNA, miRNA			
GBM	RNA	0.7619048	0.5636125	0.4870354	0.7775749	0.8029655	0.7619048
	miRNA	0.6071429	0.3636915	0.3329748	0.6408620	0.7343171	0.6547619
	CNV	0.4166667	0.1207564	0.1061846	0.4678243	0.5688623	0.4464286
	PCA_subset	0.7916667	0.5729951	0.5441936	0.8072570	0.8244226	0.7916667
	SURE	**0.797619**	**0.5815764**	**0.5588514**	**0.8120413**	**0.8300542**	**0.7976190**
	Best PCA subset:			RNA, miRNA, CNV			
	Subset selected by SURE:			RNA, miRNA, CNV			
LGG	mDNA	**0.7940075**	**0.5335888**	**0.4668931**	**0.7904750**	**0.7465292**	**0.7940075**
	RNA	0.659176	0.2782794	0.2558892	0.6600498	0.6461660	0.6591760
	miRNA	0.4007491	0.0318103	0.0251035	0.4425295	0.5499986	0.5018727
	RPPA	0.5767790	0.1808821	0.1435186	0.5820448	0.5910563	0.5767790
	PCA_subset	0.6554307	0.3414426	0.2968495	0.6576214	0.6572893	0.6554307
	SURE	**0.7940075**	**0.5335888**	**0.4668931**	**0.7904750**	**0.7465292**	**0.7940075**
	Best PCA subset:			mDNA, RNA, miRNA			
	Subset selected by SURE:			mDNA			
LUNG	mDNA	0.8077496	0.2949746	0.3778454	0.8065767	0.6889561	0.8077496
	RNA	0.9344262	0.6580123	0.7545235	0.9342231	0.8772694	0.9344262
	miRNA	0.8226528	0.3547613	0.4155187	0.8222429	0.7077741	0.8226528
	RPPA	0.5305514	0.0005234	0.0011007	0.6089374	0.5011233	0.5365127
	PCA_subset	0.9388972	0.6773549	0.7701654	0.9386955	0.8850902	0.9388972
	SURE	**0.9418778**	**0.6878184**	**0.7806842**	**0.9417093**	**0.8903486**	**0.9418778**
	Best PCA subset:			mDNA, RNA, miRNA			
	Subset selected by SURE:			mDNA, RNA, miRNA			
KIDNEY	mDNA	0.6716418	0.3889985	0.3406900	0.7217190	0.6741896	0.8317503
	RNA	0.9457259	0.7483180	0.8308028	0.9462649	0.9156687	0.9457259
	miRNA	0.8493894	0.4923203	0.6068730	0.8573787	0.8044216	0.8493894
	RPPA	0.4261872	0.0020027	0.0061810	0.4639016	0.5078609	0.6241520
	PCA_subset	0.9511533	0.7670505	0.8489024	0.9516854	0.9246800	0.9511533
	SURE	**0.9525102**	**0.7726162**	**0.8534490**	**0.9530685**	**0.9269512**	**0.9525102**
	Best PCA subset:			mDNA, RNA, miRNA, RPPA			
	Subset selected by SURE:			mDNA, RNA, miRNA			

best subset of modalities among all possible $(2^M - 1)$ combinations.

The results corresponding to survival analysis show that the subtypes identified by SURE algorithm have statistically significant difference in survival profiles, considering 5% significance level of both log rank and generalized Wilcoxon tests, for the LGG, and KIDNEY data sets. For different data sets, different combinations of modalities achieve the lowest p-values in survival analysis. However, their performance with respect to external indices is considerably poor as compared to SURE. In brief, the relevance and concordance measures of the proposed algorithm appropriately select the best subset of modalities and the eigenspace update approach efficiently integrates their cluster information. In effect, the subtypes identified by SURE have closest resemblance with the previously established cancer subtypes.

4.5.5 Importance of Relevance

The proposed algorithm first evaluates the relevance of each modality based on compactness of the cluster structure embedded within its left subspace. The relevance measure provides a linear ordering of the modalities, and the process of integration starts with the most relevant one. To establish the importance of relevance based ordering in data integration, the performance of clustering is studied for three other cases where the process of integration is initiated with the second, third, and fourth most relevant modalities, keeping all other components of the algorithm fixed. For different initiating modalities, different subset of modalities are selected during the construction of joint subspace, giving rise to different clustering solutions. The starting modality for other three cases, their corresponding subset of selected modalities and their comparative performance with the proposed approach are reported in Table 4.2 for different data sets.

The results in Table 4.2 show that for the LGG data set, only the proposed relevance ordering gives the best performance, while for other orderings the performance is degraded drastically. For the other data sets, however, one or more orderings have the same performance as that of the proposed algorithm. This is due to the presence of the concordance measure and the value of threshold τ selected for each of those orderings. For example, for the CESC data set, if the process starts with RNA, miRNA has the highest concordance and the remaining modalities have concordance below the optimal threshold τ selected for CESC. Again, starting with miRNA, only RNA has the highest concordance that exceeds the optimal threshold. Hence, same subsets of modalities are selected for both the cases of CESC, giving rise to identical clustering performance. Similar cases occur for both GBM and KIDNEY data sets. For LUNG data set, for each different ordering, all four modalities get selected without degrading final clustering performance. However, the proposed ordering gives the best performance with smaller subset of modalities. So, the performance of the proposed relevance based ordering is atleast as best as the other orderings.

4.5.6 Significance of Concordance

At each iteration of eigenspace update, the proposed algorithm considers the modality having maximum average concordance $\overline{\mathtt{C}}$ or shared information with respect to the previously updated ones. However, if the value of $\overline{\mathtt{C}}$ is below the optimal threshold of τ, then it is not updated with the current joint eigenspace. To assess the significance of the concordance measure for modality selection, all the modalities are naively integrated based on their

Table 4.2: Importance of Relevance Based Ordering of Views

	Integration starts with	Starting view	Relevance of view	Selected views (in order)	External evaluation index			
					NMI	ARI	F-Measure	Purity
CESC	2^{nd} best	mDNA	0.47007	{mDNA}	0.24204	0.11755	0.54537	0.58064
	3^{rd} best	RPPA	0.45550	{RPPA, mDNA, miRNA, RNA}	0.67509	0.63330	0.83902	0.85483
	4^{th} best	miRNA	0.44951	{miRNA, RNA}	**0.64619**	**0.65072**	**0.85120**	**0.86290**
	SURE	RNA	0.47533	{RNA, miRNA}	**0.64619**	**0.65072**	**0.85120**	**0.86290**
GBM	2^{nd} best	CNV	0.48196	{CNV}	0.12075	0.10618	0.46782	0.44642
	3^{rd} best	miRNA	0.43332	{miRNA, RNA, CNV}	**0.58157**	**0.55885**	**0.81204**	**0.79761**
	SURE	RNA	0.50859	{RNA, miRNA, CNV}	**0.58157**	**0.55885**	**0.81204**	**0.79761**
LGG	2^{nd} best	RNA	0.44798	{RNA, RPPA, mDNA, miRNA}	0.34387	0.30313	0.65748	0.66666
	3^{rd} best	RPPA	0.43591	{RPPA, RNA, mDNA, miRNA}	0.34387	0.30313	0.65748	0.66666
	4^{th} best	miRNA	0.42871	{miRNA}	0.03181	0.02510	0.44252	0.50187
	SURE	mDNA	0.50396	{mDNA}	**0.53358**	**0.46689**	**0.79047**	**0.79400**
LUNG	2^{nd} best	mDNA	0.35353	{mDNA, RNA, miRNA}	**0.68781**	**0.78068**	**0.94170**	**0.94187**
	3^{rd} best	miRNA	0.35334	{miRNA, RNA, mDNA }	**0.68781**	**0.78068**	**0.94170**	**0.94187**
	4^{th} best	RPPA	0.31047	{RPPA, RNA, miRNA, mDNA}	**0.68781**	**0.78068**	**0.94170**	**0.94187**
	SURE	RNA	0.43179	{RNA, miRNA, mDNA}	**0.68781**	**0.78068**	**0.94170**	**0.94187**
KIDNEY	2^{nd} best	miRNA	0.53257	{miRNA, RNA, mDNA}	**0.77261**	**0.85344**	**0.95306**	**0.95251**
	3^{rd} best	mDNA	0.50915	{mDNA, RNA}	0.76805	0.84888	0.95172	0.95115
	4^{th} best	RPPA	0.39006	{RPPA, mDNA, RNA, miRNA}	**0.77261**	**0.85344**	**0.95306**	**0.95251**
	SURE	RNA	0.58383	{RNA, miRNA, mDNA}	**0.77261**	**0.85344**	**0.95306**	**0.95251**

relevance ordering, and the clustering performance of the resulting subspace is studied. The comparative performance of this relevance-based subspace (without concordance $\bar{\mathbb{C}}$) and the proposed SURE algorithm is reported in Table 4.3. The results in Table 4.3 show that for CESC and LGG data sets, selection of a subset of modalities gives better performance compared to the naive integration of all modalities. For GBM, there are only three modalities and the proposed algorithm selects all of them. So, the performance on GBM is identical with or without concordance. For LUNG and KIDNEY data sets, the proposed algorithm selects only three modalities out of four using the concordance measure. However, the results in Table 4.2 show that for these data sets, selection of all four modalities does not degrade the clustering performance. But, the concordance measure for modality selection gives better performance with smaller subset of modalities compared to relevance alone.

Table 4.3: Importance of Concordance in SURE

Data Set	Algorithm Settings	External Evaluation Index					
		Accuracy	NMI	ARI	F-Measure	Rand	Purity
CESC	Without $\bar{\mathbb{C}}$	0.8548387	0.6750978	0.6333073	0.8390298	0.8237608	0.8548387
	SURE	0.8629032	**0.6461946**	**0.6507274**	**0.8512028**	**0.833989**	**0.8629032**
GBM	Without $\bar{\mathbb{C}}$	**0.797619**	**0.5815764**	**0.5588514**	**0.8120413**	**0.8300542**	**0.797619**
	SURE	**0.797619**	**0.5815764**	**0.5588514**	**0.8120413**	**0.8300542**	**0.797619**
LGG	Without $\bar{\mathbb{C}}$	0.6666667	0.3438738	0.3031312	0.6574834	0.6616823	0.6666667
	SURE	**0.7940075**	**0.5335888**	**0.4668931**	**0.7904750**	**0.7465292**	**0.7940075**
LUNG	Without $\bar{\mathbb{C}}$	**0.9418778**	**0.6878184**	**0.7806842**	**0.9417093**	**0.8903486**	**0.9418778**
	SURE	**0.9418778**	**0.6878184**	**0.7806842**	**0.9417093**	**0.8903486**	**0.9418778**
KIDNEY	Without $\bar{\mathbb{C}}$	**0.9525102**	**0.7726162**	**0.8534490**	**0.9530685**	**0.9269512**	**0.9525102**
	SURE	**0.9525102**	**0.7726162**	**0.8534490**	**0.9530685**	**0.9269512**	**0.9525102**

4.5.7 Performance Analysis of Different Algorithms

Finally, the performance of the proposed SURE algorithm is compared with that of seven existing integrative clustering approaches, namely, BCC [140], COCA [93], JIVE [141], A-JIVE [63], iCluster [192], LRAcluster [243], PCA-Con [6], and NormS [111] (proposed in Chapter 3). Comparative results with respect to six external indices are reported in Tables 4.4 and 4.5, while survival analysis and execution times are reported in Table 4.7. The results in Tables 4.4 and 4.5 show that the SURE approach performs better than all the existing approaches with respect to most of the external indices, on four data sets, namely, GBM, LGG, LUNG, and OV. For LGG data set, the performance of SURE algorithm is significantly better compared to all the existing algorithms, except NormS. The better performance is attributed to the efficient selection of relevant modalities only during joint subspace construction, which is also applicable in case of NormS algorithm. For KIDNEY data set, LRAcluster gives the best performance. However, the performance of the SURE on KIDNEY data set, considering only three modalities, is almost close to the best results. The JIVE, A-JIVE, iCluster, LRAcluster, and PCA-Con are low-rank based approaches. The results in Tables 4.4 and 4.5 show that the joint subspace extracted by the proposed algorithm preserves better cluster structure compared to the ones extracted by these existing low-rank based approaches. This is because the proposed algorithm first truncates the individual eigenspaces at rank k, and then considers only the cluster information of top k singular triplets for further integration; thus filtering out the inherent noise present in the $(n-k)$ remaining components. The existing low-rank based approaches, however, consider cluster as well as noisy information of all the modalities; thus giving poor cluster structure in the extracted subspace.

For GBM data, BIC based JIVE algorithm estimates the rank of joint structure to be 0, which implies that the four different modalities do not share any correlated information among them. On the other hand, for LGG and KIDNEY data, the joint rank estimated by JIVE is the same using both BIC and permutation tests. However, the overall performance differs due to difference in rank of the individual modalities estimated by these two criteria. The survival analysis results of Table 4.6 show that the subtypes identified by all the

Table 4.4: Comparative Performance Analysis of SURE and Existing Approaches

	Different Algorithms	Rank of Subspace	External Evaluation Index					
			Accuracy	NMI	ARI	F-Measure	RAND	Purity
CESC	COCA	-	0.6693548	0.4172592	0.3677157	0.6870510	0.6971282	0.6774194
	BCC	-	0.6895161	0.2854917	0.3144526	0.6795619	0.6687779	0.6935484
	JIVE-Perm	24	0.7177419	0.4425848	0.3860367	0.7097880	0.7164962	0.7177419
	JIVE-BIC	4	0.8064516	0.5296325	0.5229385	0.8011385	0.7791765	0.8064516
	A-JIVE	48	0.6500000	0.3700238	0.3355826	0.6511586	0.6857724	0.6814516
	iCluster	2	0.5483871	0.1737526	0.1017765	0.5568753	0.5731707	0.5645161
	LRAcluster	1	0.8145161	0.5176602	0.5384740	0.8123256	0.7867821	0.8145161
	PCA-con	3	0.8548387	0.6750978	0.6333073	0.8390298	0.8237608	0.8548387
	NormS	6	**0.8870968**	**0.6854921**	**0.7004411**	**0.8801172**	**0.8587726**	**0.8870968**
	SURE	3	0.8629032	0.6461946	0.6507274	0.8512028	0.8339890	0.8629032
GBM	COCA	-	0.6863095	0.3682423	0.3367219	0.6771487	0.7251354	0.6863095
	BCC	-	0.4113095	0.1273042	0.0578081	0.4363617	0.5873253	0.4511905
	JIVE-Perm	12	0.6666667	0.3802445	0.3664034	0.6909252	0.7566296	0.6726190
	A-JIVE	36	0.6940476	0.4829471	0.4580430	0.7211722	0.7907898	0.7208333
	iCluster	3	0.7678571	0.5441494	0.5298306	0.7850480	0.8182207	0.7678571
	LRAcluster	3	0.7678571	0.5421434	0.5201970	0.7894569	0.8152267	0.7678571
	PCA-Con	4	0.7916667	0.5729951	0.5441936	0.8072570	0.8244226	0.7916667
	NormS	13	0.6964286	0.4610496	0.4593267	0.7190554	0.7931993	0.7023810
	SURE	4	**0.7976190**	**0.5815764**	**0.5588514**	**0.8120413**	**0.8300542**	**0.7976190**
LGG	COCA	-	0.6591760	0.2772248	0.2533847	0.6608123	0.6454901	0.6591760
	BCC	-	0.6340824	0.2737596	0.248606	0.63111660	0.6382755	0.6355805
	JIVE-Perm	8	0.5617978	0.2299551	0.1606599	0.5757978	0.6056715	0.5730337
	JIVE-BIC	8	0.6741573	0.3441747	0.3050874	0.6679019	0.6642730	0.6741573
	A-JIVE	48	0.7168539	0.4267241	0.3376560	0.7172792	0.6869055	0.7168539
	iCluster	2	0.4382022	0.1379678	0.0996867	0.5187438	0.5821858	0.5355805
	LRAcluster	2	0.4719101	0.1240057	0.1030798	0.5137382	0.5831714	0.5280899
	PCA-con	3	0.6666667	0.3438738	0.3031312	0.6574834	0.6616823	0.6666667
	NormS	14	**0.7940075**	0.5325030	0.4649223	**0.7916535**	**0.7465292**	**0.7940075**
	SURE	3	**0.7940075**	**0.5335888**	**0.4668931**	0.790475	**0.7465292**	**0.7940075**
LUNG	COCA	-	0.9284650	0.6287671	0.7339231	0.9283705	0.8669662	0.9284650
	BCC	-	0.9372578	0.6648076	0.7645295	0.9371445	0.8822697	0.9372578
	JIVE-Perm	8	0.9269747	0.6333526	0.7288041	0.9266709	0.8644127	0.9269747
	JIVE-BIC	8	0.9388972	0.6883994	0.7701592	0.9385860	0.8850902	0.9388972
	A-JIVE	32	0.9478390	0.7192028	0.8019299	0.9476450	0.9009720	0.9478390
	iCluster	1	0.6333830	0.0627751	0.0696293	0.6299231	0.5348889	0.6333830
	LRAcluster	1	0.9344262	0.6535038	0.7545277	0.9342966	0.8772694	0.9344262
	PCA-Con	2	0.9388972	0.6773549	0.7701654	0.9386955	0.8850902	0.9388972
	NormS	27	0.9359165	0.6650183	0.7597192	0.9357050	0.8798674	0.9359165
	SURE	2	**0.9418778**	**0.6878184**	**0.7806842**	**0.9417093**	**0.8903486**	**0.9418778**

algorithms for LGG and KIDNEY data have significantly different survival profiles. On the other hand, for the CESC and LUNG data sets, most of the algorithms fail to give statistically significant results at 5% significance level.

Comparing the execution time of different algorithms in Table 4.6, it is seen that SURE

Table 4.5: Comparative Performance Analysis of SURE and Existing Approaches

	Different Algorithms	Rank of Subspace	External Evaluation Index					
			Accuracy	NMI	ARI	F-Measure	RAND	Purity
KIDNEY	COCA	-	0.9408280	0.7493140	0.8393954	0.9477422	0.9199568	0.9470828
	BCC	-	0.9122117	0.6783448	0.7299573	0.9139998	0.8657292	0.9122117
	JIVE-Perm	12	0.9308005	0.6955325	0.7786981	0.9300085	0.8893944	0.9308005
	JIVE-BIC	12	0.9253731	0.6777835	0.7724587	0.9250073	0.8863305	0.9253731
	A-JIVE	48	0.9582090	0.7902576	0.8695284	0.9585611	0.9349404	0.9582090
	iCluster	2	0.6065129	0.2547010	0.1717458	0.6514716	0.5842023	0.6811398
	LRAcluster	2	**0.9538670**	**0.7862018**	**0.8579391**	**0.9545717**	**0.9292298**	**0.9538670**
	PCA-Con	3	0.9511533	0.7670505	0.8489024	0.9516854	0.9246800	0.9511533
	NormS	35	0.9525102	0.7726162	0.8534490	0.9530685	0.9269512	0.9525102
	SURE	3	0.9525102	0.7726162	0.8534490	0.9530685	0.9269512	0.9525102
OV	COCA	-	0.5943114	0.3131466	0.2810761	0.6068513	0.7039183	0.5943114
	BCC	-	0.4610778	0.1567582	0.1254690	0.4755846	0.6268706	0.4622754
	JIVE-Perm	32	0.5718563	0.2629523	0.2027605	0.5653910	0.6885005	0.5718563
	A-JIVE	64	0.5191617	0.2124862	0.1981556	0.5111353	0.6942997	0.5221557
	iCluster	3	0.5089820	0.2249889	0.2005886	0.4808256	0.6916078	0.5119760
	LRAcluster	2	0.6287425	0.3745173	0.2999204	0.6384046	0.7322472	0.6287425
	PCA-con	4	0.6946108	0.4424701	0.4068449	0.6868295	0.7734621	0.6946108
	NormS	10	0.6976048	0.4504552	0.4142200	0.6910392	0.7766269	0.6976048
	SURE	4	**0.7215569**	**0.4680312**	**0.4372574**	**0.7148805**	**0.7857258**	**0.7215569**
BRCA	COCA	-	0.7434673	0.5002408	0.4864778	0.7457304	0.7905295	0.7434673
	BCC	-	0.6251256	0.3169187	0.3049874	0.6242493	0.7055783	0.6334171
	JIVE-Perm	12	0.6859296	0.4287142	0.3772649	0.6889363	0.7464906	0.6859296
	JIVE-BIC	4	0.6608040	0.4372675	0.3603942	0.6678438	0.7286432	0.6608040
	A-JIVE	64	0.6140704	0.4482479	0.3710317	0.6707575	0.7363682	0.6841709
	iCluster	3	0.7638191	0.5176193	0.4745746	0.7658865	0.7842867	0.7638191
	LRAcluster	2	0.7110553	0.4368520	0.4035040	0.7101385	0.7521740	0.7110553
	PCA-con	4	0.7587940	**0.5506612**	0.5038795	0.7601317	0.7984380	0.7587940
	NormS	11	**0.7688442**	0.5437267	**0.5090183**	**0.7699789**	**0.7999063**	**0.7688442**
	SURE	4	0.7663317	0.5528011	0.5104814	0.7683344	0.8010455	0.7663317

has the minimum execution time compared to all existing algorithms on three larger data sets, namely, LGG, LUNG, and KIDNEY, having 267, 671, and 737 samples, respectively. For two smaller data sets, namely, CESC and GBM having 124 and 168 samples, respectively, PCA-Con achieves the minimum execution time. Comparing the execution time of SURE with the state-of-the-art low-rank approaches such as iCluster, JIVE, A-JIVE, and LRAcluster in Table 4.6, it is evident that the SURE extracts the low-rank subspace in significantly lower time as compared to all these approaches for five data sets. Hence, the proposed algorithm is computationally more efficient compared to all the existing approaches considered in this work.

4.5.8 Survival Analysis

Clinical information of the samples, retrieved from the RTCGA.clinical package [117], is used to analyze the survival profiles of the subtypes identified by the proposed SURE

Table 4.6: Survival p-values and Execution Times of Proposed and Existing Approaches

Different Algorithms		Survival Analysis (p-value)		Time (in sec)		Survival Analysis (p-value)		Time (in sec)
		Log-Rank	Wilcoxon			Log-Rank	Wilcoxon	
COCA	**CESC**	5.563e-02	3.126e-02	6.01	**LGG**	1.166e-04	2.805e-05	18.61
BCC		5.318e-01	4.572e-01	10.33		3.721e-06	3.434e-07	12.78
JIVE-Perm		**4.074e-02**	**2.479e-02**	575.95		3.736e-04	1.310e-04	622.28
JIVE-BIC		8.295e-02	8.341e-02	69.08		3.156e-08	**5.134e-10**	1636.94
A-JIVE		3.463e-01	2.469e-01	251.77		3.784e-07	1.922e-08	462.65
iCluster		1.448e-01	1.212e-01	1054.89		4.201e-03	7.864e-03	1241.97
LRAcluster		2.404e-01	2.418e-01	9.29		9.278e-02	1.682e-01	25.09
PCA-con		1.243e-01	9.175e-02	**0.23**		**3.144e-08**	9.196e-10	1.61
NormS		1.352e-01	1.064e-01	1.09		2.473e-07	6.000e-09	1.05
SURE		1.370e-01	1.079e-01	0.321		2.125e-07	5.901e-09	**0.87**
COCA	**BRCA**	1.159e-02	7.210e-03	26.90	**OV**	6.042e-02	2.699e-01	25.40
BCC		5.174e-01	5.433e-01	17.94		2.333e-01	3.957e-01	40.38
JIVE-Perm		**1.137e-02**	1.435e-02	934.13		**7.982e-03**	**8.471e-03**	1491.21
JIVE-BIC		5.314e-01	4.693e-01	734.10		-	-	-
A-JIVE		2.358e-01	2.206e-01	761.76		1.825e-01	2.489e-01	557.67
iCluster		1.409e-02	**4.282e-03**	511.87		5.831e-01	6.338e-01	2076.36
LRAcluster		1.513e-01	2.320e-01	23.53		1.583e-01	2.305e-01	15.35
PCA-con		2.765e-02	2.047e-02	**1.06**		7.744e-02	2.583e-01	**1.07**
NormS		6.887e-02	5.397e-02	1.47		4.296e-02	1.516e-01	1.72
SURE		4.845e-02	3.442e-02	6.32		3.872e-02	1.650e-01	4.62
COCA	**LUNG**	4.796e-01	7.407e-01	45.49	**KIDNEY**	1.650e-04	5.035e-04	69.50
BCC		3.970e-01	5.827e-01	1156.40		5.607e-03	1.705e-02	1404.64
JIVE-Perm		3.691e-01	5.893e-01	10814.40		3.087e-07	6.640e-07	5510.53
JIVE-BIC		**6.254e-02**	**1.145e-01**	14510.26		3.298e-07	8.795e-07	23494.11
A-JIVE		2.185e-01	4.442e-01	1348.23		1.265e-06	3.515e-06	1235.29
iCluster		1.523e-01	3.375e-01	2666.65		**4.408e-11**	**4.711e-10**	2152.22
LRAcluster		6.971e-01	9.892e-01	358.09		4.131e-04	1.485e-03	386.15
PCA-con		4.292e-01	6.561e-01	5.61		1.725e-04	6.006e-04	7.85
NormS		4.702e-01	7.274e-01	**3.85**		1.709e-04	5,946e-04	**2.53**
SURE		5.102e-01	7.778e-01	5.26		1.709e-04	5.946e-04	6.19

algorithm on different data sets. The survival profiles of the subtypes are compared using Kaplan-Meier survival plots, median survival times, survival probability of the samples within a subtype after two, five, and seven years of diagnosis of the disease, and log-rank test p-value from pairwise comparison of subtypes. Median survival time is a statistic that refers to how long patients are expected to survive with a disease. It is the time expressed in months or years, when half of the patients in a group of patients diagnosed with the disease are still alive. It gives an approximate indication of the survival as well as the prognosis of a group of patients with the disease. The median survival time for a disease subtype is given by the time period where the Kaplan-Meier curve for the subtype crosses the survival probability of 0.5, and it is not available for subtypes whose survival curves end before the survival probability of 0.5 due to low sample count or presence of censored samples. The total number of deaths in each subtype, the number of samples at risk and the number of events of death at two, five, and seven years of diagnosis are also observed

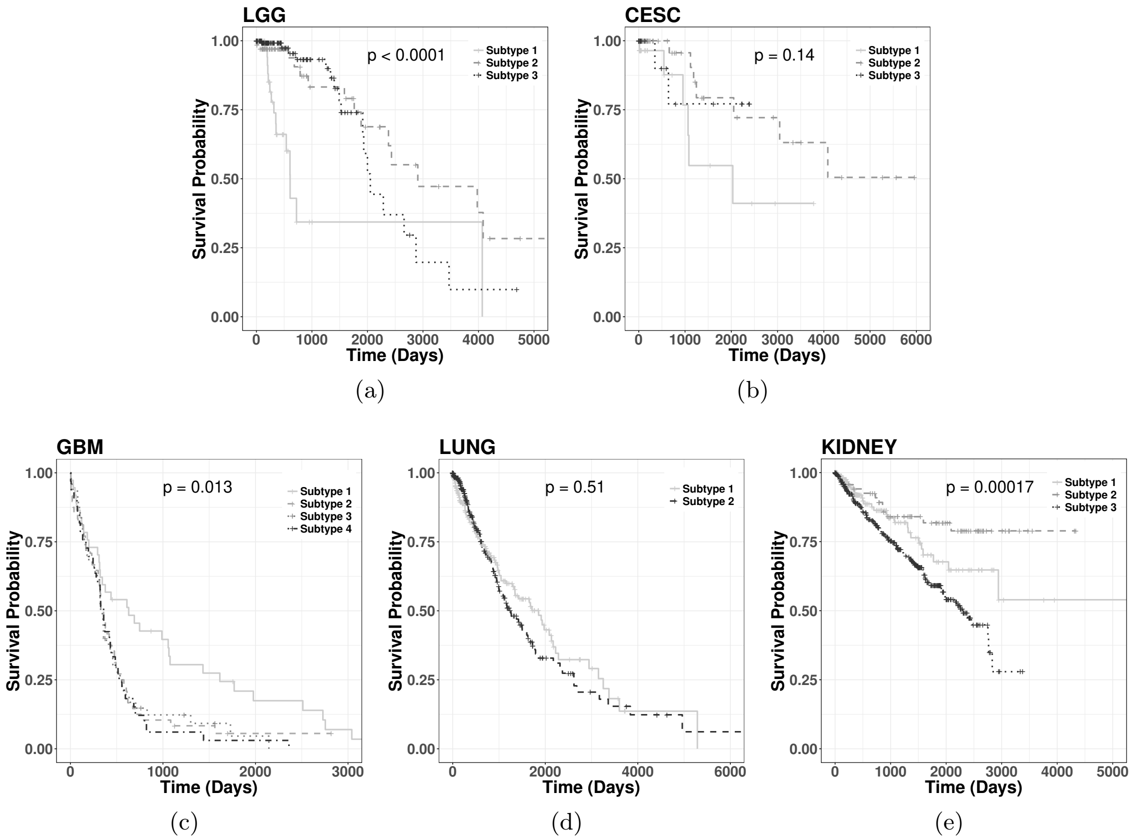


Figure 4.5: Kaplan-Meier survival plots for subtypes identified by SURE on different data.

to study the prognosis of respective cancer with time. The survival results are reported in Figure 4.5 and Table 4.7.

The Kaplan-Meier plot for the subtypes of LGG data set is given in Figure 4.5(a). The p-values for the log-rank test and the generalized Wilcoxon test are $2.125e{-}07$ and $5.901e{-}09$, respectively. These p-values show that there is a statistically significant difference in survival profiles of the subtypes of LGG, identified by the SURE algorithm. Table 4.7 shows that subtype 2 and subtype 3 have median survival times of 7.96 and 5.62 years, respectively. Hence, subtype 2 and Subtype 3 have much better prognosis than subtype 1 which has survival time of 1.66 years. The survival risk is also very high for subtype 1, as the number of death is 15 out of 51 samples and the survival probability is only 0.343 after two years of diagnosis. The p-value from pairwise log-rank test comparing subtypes 1 and 2 is $5.117e-05$, comparing subtypes 1 and 3 is $5.915e-06$, while the p-value between subtypes 2 and 3 is 0.32947. Thus, the difference between survival profiles of subtypes 1 and 2 and subtypes 1 and 3 are statistically significant, while the difference is not statistically significant between subtypes 2 and 3. Both the subtypes 2 and 3 have similar survival probabilities at two and five years of diagnosis. However, the survival probability for subtype 3 is 0.370 which is very low compared to subtype 2 having probability 0.551 after seven years of diagnosis of cancer.

The survival plot for the CESC data is given in Figure 4.5(b). Figure 4.5(b) and Table

Table 4.7: Survival Analysis for Subtypes Identified by SURE on Different Data Sets

	Different Subtypes	No. of Samples	Total No. Of Deaths	Median Survival Time (Years)	Time (Years)	No. of Risks	No. of Events Of Death	Survival Probability
LGG	Subtype 1	51	15	1.66	2	3	14	0.343
					5	1	0	0.343
					7	1	0	0.343
	Subtype 2	73	14	7.96	2	28	4	0.906
					5	15	4	0.741
					7	8	3	0.551
	Subtype 3	143	17	5.62	2	43	4	0.933
					5	10	5	0.740
					7	5	5	0.370
CESC	Subtype 1	33	6	5.57	2	8	2	0.877
					5	4	3	0.548
					7	2	1	0.411
	Subtype 2	70	7	NA	2	21	1	0.957
					5	11	3	0.794
					7	9	1	0.721
	Subtype 3	21	2	NA	2	6	2	0.771
					5	4	0	0.771
GBM	Subtype 1	37	34	1.726	2	16	20	0.455
					5	6	8	0.209
					7	4	2	0.139
	Subtype 2	48	45	0.944	2	6	42	0.125
					5	1	3	0.055
					7	1	0	0.055
	Subtype 3	50	46	0.921	2	7	42	0.147
					5	1	3	0.046
	Subtype 4	33	33	0.984	2	4	29	0.1212
					5	1	3	0.0303
LUNG	Subtype 1	285	86	5.08	2	92	50	0.717
					5	31	21	0.501
					7	13	9	0.323
	Subtype 2	363	105	3.45	2	98	56	0.703
					5	22	39	0.329
					7	13	3	0.274
KIDNEY	Subtype 1	214	28	NA	2	73	16	0.864
					5	27	10	0.677
					7	13	1	0.648
	Subtype 2	74	12	NA	2	55	6	0.909
					5	35	5	0.818
					7	18	1	0.789
	Subtype 3	445	140	6.3	2	263	70	0.811
					5	91	55	0.591
					7	14	12	0.449

4.7 show that the median survival time is not reached for subtypes 2 and 3, while for subtype 1, the median survival time is 5.57 years. Moreover, subtypes 2 and 3 have 7 and 2 deaths out of 70 and 21 samples, respectively. On the other hand, subtype 1 has 3 death cases out of 33 samples. The survival probability after seven years of diagnosis is only 0.411 for subtype 1, while the probabilities are 0.721 and 0.771 for subtypes 2 and 3, respectively. These results show that subtypes 2 and 3 have better prognosis compared to subtype 1. The pairwise log-rank test p-values for subtypes 1 and 2 is 0.04712, that for subtypes 1 and 3 is 0.29749, and that for subtypes 2 and 3 is 0.78188. The difference in survival profiles is statistically significant only for subtypes 1 and 2 and is not significant for other pairs.

Table 4.7 reports the survival analysis results for the GBM data set and the Kaplan-Meier plot for the GBM subtypes identified by the proposed SURE approach is given in Figure 4.5(c). For the GBM data set, the overall log-rank p-value is 0.0137, which shows that the subtypes have significant difference in their survival profiles. The median survival times for subtypes 1, 2, 3, and 4 are 1.726, 0.944, 0.921, and 0.984, years respectively. Comparative results from survival analysis of other data sets in Table 4.7 show that the GBM subtypes have significantly poor prognosis compared to subtypes of other cancers. Moreover, across all the subtypes, the number of deaths is very close to the total number of samples. Death rate is most severe for subtype 4, where death occurs for all the 33 samples of the subtype. The p-values for pairwise log-rank test for subtypes 1 and 2 is 0.01413, that for subtypes 1 and 3 is 0.00743, and that for subtypes 1 and 4 is 0.00290. The pairwise survival difference between subtype 1 and the other subtypes is statistically significant. On the other hand, the pairwise log-rank test p-values for subtypes 2 and 3 is 0.95869, for subtypes 2 and 4 is 0.71164, and that for subtypes 3 and 4 is 0.86016, which show no significant difference among survival profiles of subtypes 2, 3, and 4.

The Kaplan-Meier plot and survival analysis results for the LUNG data set are given in Figure 4.5(d) and Table 4.7, respectively. The median survival time for subtype 1 is 5.08 years, while for subtype 2 the median survival time is worse, that is, 3.45 years. The log-rank p-value for survival difference is 0.51, which does not show statistical significance. However, the survival probabilities for subtype 1 and subtype 2 after five years of diagnosis are 0.501 and 0.329, respectively, and after seven years of diagnosis, the survival probabilities are 0.323 and 0.274, respectively. This shows increased survival risk for subtype 2 compared to subtype 1.

For the KIDNEY data set, the survival curves are plotted in Fig. Figure 4.5(e) and the results are reported in Table 4.7. In the KIDNEY data set, for both the subtypes 1 and 2, the survival curves end before the median survival probability of 0.5. Moreover, the survival probabilities for subtypes 1 and 2 after seven years of diagnosis are 0.648 and 0.789, respectively, while for subtype 3, this probability drops to 0.449. This indicates that subtypes 1 and 2 have better prognosis than the subtype 3 which has a median survival time of 6.3 years. The p-value from pairwise log-rank test comparing subtypes 1 and 2 is 0.124566, comparing subtypes 1 and 3 is 0.01657646, and for subtypes 2 and 3 is 0.0001816. The p-values are statistically significant when compared between subtypes 3 and 1 and between subtypes 3 and 2. The overall log-rank p-value is 0.00017 when the profiles of all the three subtypes are compared together, which is statistically significant.

4.6 Conclusion

The chapter presents a novel algorithm to extract a low-rank joint subspace of the high dimensional multimodal data. The sample clustering is performed on the extracted subspace to find the subtypes of respective cancer. The problem of updating the SVD of a data matrix is formulated for multimodal data, where new modalities are added for the same set of samples. The theoretical formulation introduced here enables the proposed SURE algorithm to extract the principal components in lesser time compared to performing PCA on the concatenated data. Some new quantitative indices are proposed to evaluate theoretically the gap between joint subspace extracted by the proposed algorithm and the principal subspace extracted by PCA. Theoretical analysis also shows that the extracted subspace converges to the full-rank subspace extracted by PCA, as the rank approaches full rank of the integrated data. Unlike the existing integrative clustering approaches, the proposed approach considers that each modality may not provide relevant and consistent information about the true subtypes; hence, it evaluates the quality of each modality before integration. The evaluation measures and eigenspace update based approach allow the proposed algorithm to efficiently select only relevant modalities, discarding the noisy and inconsistent ones. The effectiveness of the proposed algorithm for cancer subtype identification has been studied and compared with existing integrative clustering approaches on several real-life multimodal cancer data sets. The experimental results show that the proposed algorithm performs better than unimodal and multimodal approaches in identification of cancer subtypes.

One of the important approaches of handling data heterogeneity in multimodal data clustering is modeling each modality using a separate similarity graph. Information from the multiple graphs is integrated by combining them into a unified graph. A major challenge here is how to preserve cluster information while removing noise from individual graphs. In this regard, Chapter 5 introduces a novel algorithm that integrates noise-free approximations of multiple similarity graphs.

Chapter 5

Approximate Graph Laplacians for Multi-View Data Clustering

5.1 Introduction

Advancement in information acquisition technologies has made multimodal data ubiquitous in numerous real-world application domains like social networking [78], image processing [54,127], 3D modeling [171], cancer biology [199], to name a few. Whole-genome sequencing project has given rise to a wide variety of "omics" data, which include genomic, epigenomic, transcriptomic, and proteomic data. The system-level insight, provided by different omics data, has led to numerous scientific discoveries and clinical applications over the past decade [89]. Cancer subtype identification has emerged out to be a major clinical application of multi-omics study. It can provide deeper understanding of disease pathogenesis and design of targeted therapies. While each type of omic data reflects the characteristic traits of a specific molecular level, integrative analysis of multi-omics data, which considers the biological variations across multiple molecular levels, can reveal novel cancer subtypes.

Multi-view clustering is the primary tool for identification of disease subtypes from multi-omics data [31, 102]. A brief survey on different multi-view clustering algorithms is reported in Chapter 2. The main challenge is how to integrate information appropriately, obtained from different modalities. Naive integration of different modalities with varying scales may give inconsistent results. Another challenge is to handle efficiently the 'high dimension-low sample size' nature of the individual data sets, which degrades the signal-to-noise ratio in the data and makes clustering computationally expensive.

In multi-omics data, different modalities vary immensely in terms of unit and scale. For instance, RNA sequence based gene expression data consists of RPM (reads per million) values having six-orders of magnitude, while DNA methylation data consists of β values which lie in [0, 1]. So, concatenation of features from these heterogeneous modalities would reflect only the properties of features having high variance. In order to capture the inherent properties of different modalities, it is essential to model the variations within each modality separately and then integrate them using a common platform. One widely used approach is to model each individual modality using a separate similarity graph. The individual similarity graphs are constructed in such a way that their vertices represent the samples, while their edges are weighted by the pairwise affinities between the samples of the respective modalities. The challenge is then how to integrate information efficiently from multiple similarity graphs. This comes under the paradigm of graph based multi-view learning [120,134,142,247,256,286,294], where the main objective is to learn a unified graph that is sufficiently "close" to all the graphs in some sense. In most multi-view learning algorithms, spectral clustering [147,152,230] is performed on the similarity graph corresponding to the unified view to identify the clusters of a given data set. The spectral clustering uses spectrum of the graph Laplacian [45] to identify the clusters in a data set. It has been shown in [230] that the relaxed solution to the k cluster indicators of a data set is given by the eigenvectors corresponding to the k smallest eigenvalues of its graph Laplacian. Hence, spectral clustering algorithms perform simple k-means on the k smallest eigenvectors of the graph Laplacian. However, it also implies that only a few eigenvectors of the Laplacian contain the cluster discriminatory information of the data set. The remaining eigenvectors may not necessarily encode cluster information and may reflect background noise. As a consequence, a major drawback of these multi-view algorithms is that both similarity graphs and their Laplacians, constructed from different views, inherently contain noisy information. This unwanted noise of the individual views may get propagated into the unified view during integration. This can degrade the quality of the cluster structure inferred from the unified view. Therefore, it is essential to prevent the noise in the individual views from being propagated into the unified view.

In this regard, the chapter presents a novel algorithm, termed as CoALa (Convex-combination of Approximate Laplacians), which integrates noise-free approximations of multiple similarity graphs. The proposed method models each modality using a separate similarity graph, as different modalities are highly heterogeneous in nature and are measured in different scales. The noise in each individual graph is eliminated by approximating it using the most informative eigenpairs of its Laplacian which contain cluster information. The approximate Laplacians are then integrated and a low-rank subspace is constructed that best preserves the overall cluster information of multiple graphs. The graphs are integrated using a convex combination, where they are weighted according to the quality of their inherent cluster structure. Hence, noisy graphs have lower impact on the final subspace compared to the ones with good cluster structure. However, the approximate subspace constructed by the proposed method differs from the full-rank subspace that integrates information from all the eigenpairs of each Laplacian. The matrix perturbation theory is used to theoretically upper bound the difference between the full-rank and approximate subspaces, as a function of the approximation rank. It is shown, both theoretically and experimentally, that the approximate subspace converges to the full-rank one as the rank of approximation approaches to the full-rank of the individual Laplacians. Finally, the efficacy of clustering in the approximate subspace is extensively studied and compared with different existing integrative clustering approaches, on several real-life multi-omics cancer data sets. The results on benchmark data sets from other domains like image processing and social networks are also provided to establish the generality of the proposed approach. Some of the results of this chapter are reported in [113].

The rest of this chapter is organized as follows: Section 5.2 introduces the basics of graph Laplacian and its properties, while Section 5.3 presents the proposed graph based algorithm for multi-view data clustering. Section 5.4 presents theoretical upper bounds on the difference between full-rank and approximate subspaces. Experimental results and comparison with existing approaches on multi-omics cancer and benchmark data sets are presented in Section 5.5. Section 5.6 concludes the chapter.

5.2 Basics of Graph Laplacian

Given a set of samples or objects $X = \{x_1, \ldots, x_i, \ldots, x_n\}$, and a similarity matrix $W = [w(i,j)]_{n \times n}$, where $x_i \in \Re^d$ and $w(i,j) = w(j,i) \geqslant 0$ is the similarity between objects x_i and x_j, the intuitive goal of clustering is to partition the objects into several groups such that objects in the same group are similar to each other, while those in different groups are dissimilar. The problem of clustering can also be approached from a graph theoretic point of view, where the data set X can be represented as an undirected similarity graph $G = (V, E)$ having vertex set $V = \{v_1, \ldots, v_i, \ldots, v_n\}$, where each vertex v_i represents the object x_i, and the edge between vertices v_i and v_j is weighted by the similarity $w(i,j)$. The degree $\widetilde{d}_i$ of vertex v_i is given by $\widetilde{d}_i = \sum\limits_{j=1}^{n} w(i,j)$, and the degree matrix D is given by the diagonal matrix

$$D = diag(\widetilde{d}_1, \ldots, \widetilde{d}_i, \ldots, \widetilde{d}_n). \tag{5.1}$$

Given the number of clusters k, clustering can be viewed as partitioning the graph G into k subgraphs such that edges between different subgraphs have lower weights, while edges within a subgraph have higher weights. For a subset of vertices $A \subset V$, let its complement $\bar{A}$ be given by $\bar{A} = V \backslash A$. A measure of size of subset A can be given by $vol(A) = \sum\limits_{v_i \in A} \widetilde{d}_i$. For two not necessarily disjoint subsets $A, B \subset V$, let

$$\mathbb{C}(A, B) = \sum_{v_i \in A, v_j \in B} w(i,j). \tag{5.2}$$

For a subset A of vertices, $\mathbb{C}(A, \bar{A})$ gives the weight of the cut that separates the vertices in A from the rest of vertices in G. So, given the number of subsets k, the graph partitioning problem finds a partition $A_1, \ldots, A_k$ of V such that it minimizes the cut weight $\mathbb{C}(A_i, \bar{A}_i)$ for each A_i. However, minimizing only $\mathbb{C}(A_i, \bar{A}_i)$ can lead to singleton subsets A_i's. In clustering, it is desirable to achieve clusters with reasonably large set of points. So, minimizing $\frac{\mathbb{C}(A_i, \bar{A}_i)}{vol(A_i)}$, instead of $\mathbb{C}(A_i, \bar{A}_i)$, would constrain each subset A_i to be fairly large. The most common optimization problem in this regard is the normalized cut or $Ncut$ [194], defined as

$$\begin{aligned} \underset{A_1, \ldots, A_k}{\text{minimize}} \quad & Ncut(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{\mathbb{C}(A_i, \bar{A}_i)}{vol(A_i)} \\ \text{such that} \quad & A_i \cap A_j = \varnothing \text{ and } \bigcup_{i=1}^{k} A_i = V. \end{aligned} \tag{5.3}$$

However, the above optimization problem is NP-hard [231]. The spectral clustering [230] provides a computationally tractable solution to this Ncut problem. It analyzes the spectrum or eigenspace of graph Laplacian to find the solution [158]. The graph Laplacian and several its variants are described next.

Let $G = (V, E)$ be a graph with similarity matrix W and degree matrix D as given by (5.1). The matrix $(D - W)$ is called the Laplacian of graph G [158], and the normalized Laplacian of G is given by [45]

$$\mathcal{L} = D^{-1/2}(D - W)D^{-1/2} = I - D^{-1/2}WD^{-1/2}, \tag{5.4}$$

where I is identity matrix of appropriate order. Two important properties of normalized Laplacian are as follows [45]:

Property 5.1. $\mathcal{L}$ *is symmetric and positive semi-definite.*

Property 5.2. *The eigenvalues of* $\mathcal{L}$ *lie in* $[0, 2]$.

Let the k clusters in a data set X be represented by the indicator matrix

$$\mathcal{E} = [e_1 \dots e_j \dots e_k] \in \Re^{n \times k}, \tag{5.5}$$

where e_j is the indicator vector in $\Re^n$ for the j-th cluster, that is, $e_j \in \{0, 1\}^n$, such that e_j has a nonzero component only for the points in the j-th cluster. Let the r largest eigenvectors of a matrix correspond to its r largest eigenvalues. It is shown in [230] that if the constraint on the cluster indicators e_j's is relaxed such that $e_j \in [0, 1]$, then the real-valued solution to the indicators $e_1, \dots, e_k$ is given by the k smallest eigenvectors of the normalized Laplacian $\mathcal{L}$. The normalized spectral clustering algorithm by Ng *et al.* [162] is described in Algorithm 5.1. The spectral clustering algorithm [162, 194] first computes the graph Laplacian and then k-means clustering is performed on its k smallest eigenvectors. The main advantage of spectral clustering is that it transforms the representations of the objects $\{x_i\}$ from their original space to an indicator subspace where the cluster characteristics are more prominent. As the cluster properties are enhanced in this new subspace, even simple clustering algorithms, such as k-means, have no difficulty in distinguishing the clusters.

Algorithm 5.1 Normalized Spectral Clustering [162]

```
Input: Similarity matrix $W$, number of clusters $k$.
Output: Clusters $A_1, \dots, A_k$.
Construct degree matrix $D$ and normalized Laplacian $\mathcal{L}$ as in (5.1) and (5.4), respectively.
Find eigenvectors $U = [u_1 \dots u_k]$ corresponding to $k$ smallest eigenvalues of $\mathcal{L}$.
Normalize the rows of $U$, i.e. $U = diag(UU^T)^{-\frac{1}{2}}U$.
Perform clustering on the rows of $U$ using $k$-means algorithm.
Return clusters $A_1, \dots, A_k$ from $k$-means clustering.
```

In a Laplacian matrix, the necessary cluster information is embedded in its k smallest eigenvectors. However, based on Eckart-Young theorem [56], the best low-rank approximation of a symmetric matrix can be constructed from its few largest eigenpairs. So, the best low-rank approximation of a Laplacian matrix primarily encodes noise, rather than cluster information. In the proposed work, the final subspace of a multimodal data set is constructed from low-rank approximations of individual graph Laplacians. So, in order to reflect the cluster information in the low-rank approximations, the shifted Laplacian [51] is used, which is defined as

$$L = 2I - \mathcal{L} = I + D^{-1/2} W D^{-1/2}. \tag{5.6}$$

The following property of shifted Laplacian makes it feasible to reflect the cluster information in its best low-rank approximation.

Property 5.3. *If* (λ, v) *is an eigenvalue-eigenvector pair of normalized Laplacian* $\mathcal{L}$*, then* $(2 - \lambda, v)$ *is an eigenpair of shifted Laplacian* L [51].

Property 5.3 implies that the k smallest eigenvalues and eigenvectors of normalized Laplacian $\mathcal{L}$ correspond to the k largest eigenvalues and eigenvectors of shifted Laplacian L. Therefore, the relaxed solution to the cluster indicators $e_1, \ldots, e_k$ in (5.5) is given by the k largest eigenvectors of L. So, the best rank k approximation of L also encodes its cluster information. As the eigenvalues of $\mathcal{L}$ lie in $[0, 2]$, the eigenvalues of L also lie in $[0, 2]$. Moreover, L is symmetric and positive semi-definite [51].

5.3 CoALa: Proposed Method

This section presents a novel algorithm to extract a low-rank joint subspace from multiple graph Laplacians. Some analytical formulations, required for subspace construction, are reported next, prior to describing the proposed algorithm.

5.3.1 Convex Combination of Graph Laplacians

Let a multimodal data set, consisting of M modalities or views, be given by $X_1, \ldots, X_m, \ldots, X_M$. Each modality $X_m \in \Re^{n \times d_m}$ represents the observations for same set of n samples from the m-th data source. Let X_m be encoded by the similarity graph G_m having similarity matrix W_m and degree matrix D_m. The shifted Laplacian for modality X_m is given by

$$L_m = I + D_m^{-1/2} W_m D_m^{-1/2}. \tag{5.7}$$

Let the eigen-decomposition of L_m be given by

$$L_m = U_m \Sigma_m U_m^T, \tag{5.8}$$

where $U_m = [u_1^m, \ldots, u_n^m] \in \Re^{n \times n}$ contains the eigenvectors of L_m in its columns, B^T denotes the transpose of B, and $\Sigma_m = diag(\lambda_1^m, \ldots, \lambda_n^m)$, where $2 \geqslant \lambda_1^m \geqslant \ldots \geqslant \lambda_n^m \geqslant 0$. For a given rank r, the eigen-decomposition of shifted Laplacian L_m in (5.8) can be partitioned as follows:

$$\begin{aligned} L_m &= U_m \Sigma_m U_m^T \\ &= \begin{bmatrix} U_m^r & U_m^{r\perp} \end{bmatrix} \begin{bmatrix} \Sigma_m^r & \mathbf{0} \\ \mathbf{0} & \Sigma_m^{r\perp} \end{bmatrix} \begin{bmatrix} U_m^r & U_m^{r\perp} \end{bmatrix}^T \\ &= U_m^r \Sigma_m^r (U_m^r)^T + U_m^{r\perp} \Sigma_m^{r\perp} (U_m^{r\perp})^T \\ &= L_m^r + L_m^{r\perp}, \end{aligned} \tag{5.9}$$

where $\mathbf{0}$ denotes a matrix of all zeros of appropriate order, $\Sigma_m^r = diag(\lambda_1^m, \ldots, \lambda_r^m)$ consists of the r largest eigenvalues and U_m^r contains the corresponding r eigenvectors in its columns. Similarly, $\Sigma_m^{r\perp}$ and $U_m^{r\perp}$ contain the remaining $(n-r)$ eigenvalues $\lambda_{r+1}^m, \ldots, \lambda_n^m$ and eigenvectors, respectively. Thus, L_m^r is the rank r approximation of L_m using the r largest eigenpairs, and $L_m^{r\perp}$ is the approximation using the remaining $(n-r)$ eigenpairs. Given the number of clusters k, the properties of shifted Laplacian imply that the relaxed solution to the cluster indicators is given by the k largest eigenvectors of L_m. Therefore, for each modality X_m, a rank r eigenspace representation is constructed, where $k \leqslant r << n$, which encodes the cluster information of its shifted Laplacian L_m. Choosing the rank r to be greater than k allows extra information from each Laplacian at the initial stage.

The rank r eigenspace of shifted Laplacian L_m for modality X_m is defined by a two-tuple:

$$\Psi(L_m^r) = \langle U_m^r, \Sigma_m^r \rangle. \tag{5.10}$$

The individual graph Laplacians contain the cluster information of their respective modalities. Multiple modalities are integrated using a convex combination $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m, \ldots, \alpha_M]$ of individual shifted Laplacians, defined by

$$\mathbf{L} = \sum_{m=1}^{M} \alpha_m L_m, \text{ such that } \alpha_m \geqslant 0 \text{ and } \sum_{m=1}^{M} \alpha_m = 1. \tag{5.11}$$

The matrix $\mathbf{L}$ is called the joint shifted Laplacian and it has the following properties.

Property 5.4. $\mathbf{L}$ *is symmetric and positive semi-definite.*

Proof. Each shifted Laplacian L_m is symmetric for $m = 1, 2, ..., M$. So,

$$\mathbf{L}^T = \left(\sum_{m=1}^{M} \alpha_m L_m \right)^T = \sum_{m=1}^{M} \alpha_m L_m^T = \sum_{m=1}^{M} \alpha_m L_m = \mathbf{L}.$$

Therefore, $\mathbf{L}$ is symmetric. By Property 5.3, each L_m is positive semi-definite, so, for any vector $a \in \Re^n$, $a^T L_m a \geqslant 0$. Therefore,

$$a^T \mathbf{L} a = a^T \left(\sum_{m=1}^{M} \alpha_m L_m \right) a = \sum_{m=1}^{M} \alpha_m \left(a^T L_m a \right) \geqslant 0,$$

as $\alpha_m \geqslant 0$. Therefore, $\mathbf{L}$ is positive semi-definite. ■

Property 5.5. **L** *has* n *eigenvalues* $\gamma_1 \geqslant \ldots \geqslant \gamma_i \geqslant \ldots \geqslant \gamma_n$, *where* $\gamma_i \in [0, 2]$.

Proof. By Property 5.3, the eigenvalues of each individual shifted Laplacian L_m lie in $[0, 2]$ for $m = 1, 2, \ldots, M$. So, the maximum eigenvalue of L_m and $\alpha_m L_m$ satisfy $\lambda_1^m \leqslant 2$ and $\alpha_m \lambda_1^m \leqslant 2\alpha_m$, respectively. Since each Laplacian L_m is a real symmetric matrix, it is also Hermitian as it is equal to its own conjugate transpose. Now, **L** is the sum of M Hermitian matrices. So, using Weyl's inequality [202], which bounds the eigenvalues of the sum of two Hermitian matrices, we get

$$\gamma_1 \leqslant \sum_{m=1}^{M} \alpha_m \lambda_1^m \leqslant \sum_{m=1}^{M} 2\alpha_m = 2. \tag{5.12}$$

L is positive semi-definite, so all of its eigenvalues $\gamma_i \geqslant 0$. Therefore, $\gamma_i \in [0, 2]$. ■

Hence, the joint shifted Laplacian **L** has similar properties as individual shifted Laplacians L_m's have. In rest of the chapter, the term joint Laplacian is used to refer to the joint shifted Laplacian.

5.3.2 Construction of Joint Eigenspace

This subsection describes the construction of eigenspace of the joint Laplacian from low-rank eigenspaces of individual shifted Laplacians. Let eigen-decomposition of **L** be given by

$$\mathbf{L} = \mathbf{Z}\Gamma\mathbf{Z}^T, \tag{5.13}$$

where **Z** consists of the eigenvectors of **L** in its columns and $\Gamma = diag(\gamma_1, \ldots, \gamma_n)$ is the diagonal matrix of eigenvalues arranged in descending order of magnitude. The "full-rank" eigenspace of **L** is given by the two-tuple

$$\Psi\left(\mathbf{L}^r\right) = \langle \mathbf{Z}^r, \Gamma^r \rangle, \tag{5.14}$$

where $\Gamma^r = diag(\gamma_1, \ldots, \gamma_r)$ and $\mathbf{Z}^r$ contains the eigenvectors corresponding to the eigenvalues in Γ^r. The term "full-rank" is used to imply that in **L**, the complete information of all the eigenpairs of each Laplacian is considered during convex combination. The superscript r in $\Psi\left(\mathbf{L}^r\right)$ indicates that the eigenspace has rank r. The "approximate" joint Laplacian is defined as

$$\mathbf{L}^{r*} = \sum_{m=1}^{M} \alpha_m L_m^r. \tag{5.15}$$

Thus, $\mathbf{L}^{r*}$ is the convex combination of best rank r approximation of individual shifted Laplacians. For each shifted Laplacian L_m, instead of storing its complete eigen-decomposition, only the r largest eigenpairs are stored in its eigenspace $\Psi(L_m^r)$. Given these eigenspaces $\Psi(L_m^r)$s, the proposed method aims at construction of the rank r eigenspace $\Psi\left(\mathbf{L}^{r*}\right)$, of the approximate joint Laplacian $\mathbf{L}^{r*}$. The main advantage of this construction is that it finds the joint eigenspace from the r largest eigenpairs of individual Laplacians. The cluster information of individual modalities is expected to embed in the k largest eigenpairs of their respective shifted Laplacians. Hence, storing $r \geqslant k$ eigenpairs allows for some extra information from each Laplacian as well as gets rid of the noisy information in the $(n-r)$ eigenpairs. Thus, the approximate eigenspace $\Psi\left(\mathbf{L}^{r*}\right)$, constructed from the r largest eigenpairs, is expected to preserve better cluster information compared to the full-rank eigenspace $\Psi\left(\mathbf{L}^{r}\right)$.

One straight forward approach for the construction of eigenspace of $\mathbf{L}^{r*}$ is to first solve the eigen-decomposition of the individual L_m's, reconstruct the L_m^r's from the top r eigenpairs of respective L_m's, combine the reconstructed L_m^r's using the convex combination and then perform another eigen-decomposition on the combination $\mathbf{L}^{r*}$. This requires solving a total of $(M+1)$ eigen-decompositions of size $(n \times n)$. However, in the proposed method, the eigenspaces $\Psi(L_m^r)$'s of the individual Laplacians inorder are used to construct a smaller eigenvalue problem of size $(Mr \times Mr)$ whose solution is used to get the required eigenspace $\Psi\left(\mathbf{L}^{r*}\right)$. So, it requires solving M eigen-decompositions of size $(n \times n)$ and one of size $(Mr \times Mr)$, where $Mr << n$. This makes the proposed approach computationally more efficient.

The block decomposition of L_m in (5.9) gives us that $L_m^r = U_m^r \Sigma_m^r (U_m^r)^T$. So,

$$\mathbf{L}^{r*} = \sum_{m=1}^{M} \alpha_m L_m^r = \sum_{m=1}^{M} \alpha_m U_m^r \Sigma_m^r (U_m^r)^T. \tag{5.16}$$

The expansion of $\mathbf{L}^{r*}$ in (5.16) implies that the subspace spanned by its columns is same as the one spanned by the union of the columns of U_m^r for $m = 1, \ldots, M$. Let that subspace be given by

$$\mathcal{J}^r = span\left(\bigcup_{m=1}^{M} \mathcal{C}(U_m^r)\right), \tag{5.17}$$

where $\mathcal{C}(B)$ denotes the column space of matrix B. To compute the eigenspace of $\mathbf{L}^{r*}$, the first step is to construct a sufficient basis that spans the subspace $\mathcal{J}^r$. Since $\mathcal{J}^r$ is the union of M subspaces, its basis is constructed iteratively in M steps. At step 1, the initial basis $\mathbf{U}_1$ is given by

$$\mathbf{U}_1 = U_1^r, \tag{5.18}$$

which spans the subspace $\mathcal{C}(U_1^r)$. At step m, let the union of m subspaces be given by the subspace

$$\mathcal{J}_m^r = span\left(\bigcup_{j=1}^{m} \mathcal{C}(U_j^r)\right) \tag{5.19}$$

and let its orthonormal basis be given by $\mathbf{U}_m \in \Re^{n \times r}$. Given the basis $\mathbf{U}_m$ obtained at step m, and the basis U_{m+1}^r for L_{m+1}^r, the basis $\mathbf{U}_{m+1}$ at step $(m+1)$ is constructed as follows.

The basis $\mathbf{U}_{m+1}$ has to span both the subspaces $\mathcal{J}_m^r$ and $\mathcal{C}(U_{m+1}^r)$. The column vectors of $\mathbf{U}_m$ themselves form a basis for the subspace $\mathcal{J}_m^r$. Therefore, a sufficient basis for the subspace $\mathcal{J}_{m+1}^r$ can be constructed by appending a basis Υ_{m+1} that spans the subspace orthogonal to $\mathcal{J}_m^r$. The construction of basis Υ_{m+1} begins by computing the residue of each basis vector in U_{m+1}^r with respect to the basis $\mathbf{U}_m$. To compute the residues, each vector in U^r_{m+1} is projected on each of the basis vectors in $\mathbf{U}_m$. In matrix notation, this is given by

$$S_{m+1} = \mathbf{U}_m^T U^r_{m+1}. \tag{5.20}$$

The matrix S_{m+1} gives the magnitude of projection of the columns of U^r_{m+1} onto the orthonormal basis $\mathbf{U}_m$. The projected component P_{m+1} of U^r_{m+1}, lying in the subspace $\mathcal{J}^r_m$, is obtained by multiplying the projection magnitudes in S_{m+1} by the corresponding basis vectors in $\mathbf{U}_m$, given by

$$P_{m+1} = \mathbf{U}_m S_{m+1}. \tag{5.21}$$

The residual component Q_{m+1} of U^r_{m+1} is obtained by subtracting projected component P_{m+1} from itself, given by

$$Q_{m+1} = U^r_{m+1} - P_{m+1}. \tag{5.22}$$

An orthogonal basis Υ_{m+1} for the residual space, spanned by columns of Q_{m+1}, can be obtained by Gram-Schmidt orthogonalization of Q_{m+1}. The basis Υ_{m+1} spans the subspace orthogonal to $\mathcal{J}^r_m$. Therefore, a sufficient basis for the subspace $\mathcal{J}^r_{m+1}$ is obtained by appending Υ_{m+1} to $\mathbf{U}_m$, given by

$$\mathbf{U}_{m+1} = \begin{bmatrix}\mathbf{U}_m & \Upsilon_{m+1}\end{bmatrix}. \tag{5.23}$$

Let $\Upsilon_1 = \mathbf{U}_1$. After M steps, the basis $\mathbf{U}_M$, for the subspace $\mathcal{J}^r$ in (5.17), is given by

$$\mathbf{U}_M = \begin{bmatrix}\Upsilon_1 & \Upsilon_2 & \dots & \Upsilon_M\end{bmatrix}. \tag{5.24}$$

Let the eigen-decomposition of $\mathbf{L}^{r*}$ be given by

$$\mathbf{L}^{r*} = \mathbf{V}\Pi\mathbf{V}^T, \tag{5.25}$$

where $\mathbf{V} \in \Re^{n\times n}$ contains the eigenvectors of $\mathbf{L}^{r*}$ in its columns, and $\Pi = diag(\pi_1, \dots, \pi_n)$ contains the eigenvalues arranged in descending order. The eigenvectors in $\mathbf{V}$ span the column space of $\mathbf{L}^{r*}$, which from (5.17) is the subspace $\mathcal{J}^r$. $\mathbf{U}_M$ is also a basis for $\mathcal{J}^r$. These two bases $\mathbf{V}$ and $\mathbf{U}_M$ span the same subspace $\mathcal{J}^r$ and they differ by a rotation. So,

$$\mathbf{V} = \mathbf{U}_M\mathbf{R}, \tag{5.26}$$

where $\mathbf{R}$ is an orthogonal rotation matrix. The eigenvalues Π in (5.25) and the rotation matrix $\mathbf{R}$ in (5.26) are obtained as follows.

$$\mathbf{L}^{r*} = \sum_{m=1}^{M} \alpha_m U^r_m \Sigma^r_m (U^r_m)^T, \qquad \text{[from (5.16)]}$$

$$\Rightarrow \mathbf{V}\Pi\mathbf{V}^T = \sum_{m=1}^{M} \alpha_m U^r_m \Sigma^r_m (U^r_m)^T, \qquad \text{[from (5.25)]}$$

$$\Rightarrow (\mathbf{U}_M\mathbf{R})\Pi(\mathbf{U}_M\mathbf{R})^T = \sum_{m=1}^{M} \alpha_m U^r_m \Sigma^r_m (U^r_m)^T, \text{[from (5.26)]}$$

$$\Rightarrow \mathbf{R}\Pi\mathbf{R}^T = \mathbf{U}_M^T \left(\sum_{m=1}^{M} \alpha_m U_m^r \Sigma_m^r (U_m^r)^T \right) \mathbf{U}_M,$$

$$\Rightarrow \mathbf{R}\Pi\mathbf{R}^T = \sum_{m=1}^{M} \alpha_m \mathbf{U}_M^T U_m^r \Sigma_m^r (U_m^r)^T \mathbf{U}_M,$$

$$\Rightarrow \mathbf{R}\Pi\mathbf{R}^T = \sum_{m=1}^{M} \alpha_m \begin{bmatrix} \Upsilon_1^T \\ \vdots \\ \Upsilon_M^T \end{bmatrix} U_m^r \Sigma_m^r (U_m^r)^T \begin{bmatrix} \Upsilon_1 & \ldots & \Upsilon_M \end{bmatrix},$$

$$\Rightarrow \mathbf{R}\Pi\mathbf{R}^T = \sum_{m=1}^{M} \alpha_m H_m, \tag{5.27}$$

where $H_m \in \Re^{(Mr \times Mr)}$ is given by

$$H_m = [\Upsilon_1 \ldots \Upsilon_M]^T U_m^r \Sigma_m^r (U_m^r)^T [\Upsilon_1 \ldots \Upsilon_M]. \tag{5.28}$$

While constructing the basis $\mathbf{U}_M$, the Υ_p's are appended iteratively such that whenever $p > m$, Υ_p is orthogonal to U_m^r and $\Upsilon_p^T U_m^r = 0$. Thus, the matrix H_m can be partitioned into M^2 blocks, each of size $(r \times r)$, and the (i, j)-th block of H_m is given by

$$H_m(i,j) = \begin{cases} \Upsilon_i^T U_m^r \Sigma_m^r (U_m^r)^T \Upsilon_j & \text{if } \ i \leqslant m \text{ and } \ j \leqslant m, \\ 0 & \text{if } \ i > m \text{ or } \ j > m. \end{cases}$$

$$\text{Let} \quad \mathbf{H} = \sum_{m=1}^{M} \alpha_m H_m; \qquad \Rightarrow \mathbf{H} = \mathbf{R}\Pi\mathbf{R}^T. \tag{5.29}$$

This implies that solving the eigen-decomposition of the $(Mr \times Mr)$ matrix $\mathbf{H}$, the eigenvalues Π of $\mathbf{L}^{r*}$ and the rotation matrix $\mathbf{R}$ are obtained. Then, $\mathbf{R}$ is substituted in (5.26) to get the eigenvectors of $\mathbf{L}^{r*}$ in columns of $\mathbf{V}$. The rank r eigenspace of $\mathbf{L}^{r*}$ is then given by the two-tuple

$$\Psi\left(\mathbf{L}^{r*}\right) = \langle \mathbf{V}^r, \Pi^r \rangle, \tag{5.30}$$

where $\Pi^r = diag(\pi_1, \ldots, \pi_r)$ consists of the r largest eigenvalues of Π arranged in descending order, and $\mathbf{V}^r$ contains the corresponding r eigenvectors in its columns.

5.3.3 Proposed Algorithm

Given similarity matrices $W_1, \ldots, W_M$ corresponding to M modalities $X_1, \ldots, X_M$, convex combination vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_M]$ and rank r, the proposed algorithm, termed as CoALa, extracts a rank r eigenspace for the approximate joint Laplacian $\mathbf{L}^{r*}$. For each modality X_m, the proposed algorithm first computes the eigen-decomposition of its shifted Laplacian L_m and then stores the $r \geqslant k$ largest eigenpairs in its eigenspace. Next, it iteratively computes the basis $\mathbf{U}_M$ and the eigen-decomposition of the new eigenvalue problem $\mathbf{H}$. The eigenvalues of $\mathbf{L}^{r*}$ are given by the eigenvalues of $\mathbf{H}$, while the eigenvectors of $\mathbf{H}$ are used to rotate the basis $\mathbf{U}_M$ and get the eigenvectors of $\mathbf{L}^{r*}$. Finally, k-means clustering is performed on the k largest eigenvectors of $\mathbf{L}^{r*}$ to get the clusters of the multimodal data set. The proposed algorithm is described in Algorithm 5.2.

Algorithm 5.2 Proposed Algorithm: **CoALa**

```
Input: Similarity matrices $W_1, \ldots, W_M$, combination vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_M]$, number
    of clusters $k$, and rank $r \geqslant k$.
Output: Clusters $A_1, \ldots, A_k$.
for $m \leftarrow 1$ to $M$ do
    Construct degree matrix $D_m$ and shifted normalized Laplacian $L_m$ as in (5.1) and
    (6.6), respectively.
    Compute the eigen-decomposition of $L_m$.
    Store the $r$ largest eigenvalues in $\Sigma_m^r$ and corresponding eigenvectors in $U_m^r$ in the
    rank $r$ eigenspace of $X_m$.
end for
Compute initial basis $\mathbf{U}_1 \leftarrow U_1^r$.
for $m \leftarrow 1$ to $M - 1$ do
    Compute $S_{m+1}$, projected component $P_{m+1}$, and residual component $Q_{m+1}$ according
    to (5.20), (5.21), and (5.22), respectively.
    $\Upsilon_{m+1} \leftarrow$ Gram-Schmidt orthogonalization of $Q_{m+1}$.
    Update basis $\mathbf{U}_{m+1} \leftarrow \left[\mathbf{U}_m \quad \Upsilon_{m+1}\right]$.
end for
For each modality $X_m$, compute $H_m$ as in (5.28).
Compute the new eigenvalue problem $\mathbf{H}$ as in (5.29).
Solve the eigen-decomposition of $\mathbf{H}$ to get $\mathbf{R}$ and $\Pi$.
Compute eigenvectors $\mathbf{V} \leftarrow \mathbf{U}_M\mathbf{R}$.
Compute joint eigenspace $\Psi\left(\mathbf{L}^{r*}\right) \leftarrow \langle \mathbf{V}^r, \Pi^r \rangle$ as in (5.30).
Find $k$ largest eigenvectors $\mathbf{V}^k = [v_1 \ldots v_k]$.
Perform clustering on the rows of $\mathbf{V}^k$ using $k$-means algorithm.
Return clusters $A_1, \ldots, A_k$ from $k$-means clustering.
```

In the normalized spectral clustering by Ng *et al.* [162], the eigenvectors are row normalized (step 3 of Algorithm 5.1) before clustering. The advantage of this additional normalization has been shown for the ideal case where the similarity is zero between points belonging to different clusters and strictly positive between points in the same clusters. In such a situation, the eigenvalue 0 has multiplicity k, and the eigenvectors are given by the columns of $D^{\frac{1}{2}}\mathcal{E}$, where $\mathcal{E}$ is the ideal cluster indicator matrix as in (5.5). By normalizing each row by its norm, the eigenvector matrix coincides with the indicator matrix $\mathcal{E}$, and the points become trivial to cluster. Ng *et al.* [162] have also shown that when the similarity matrix is “close" to the ideal case, properly normalized rows tend to tightly cluster around an orthonormal basis. However, in real-life data sets, the clusters are generally not well-separated due to the high dimension and heterogeneous nature of different modalities. As a result, the similarity matrices deviate far from the ideal block diagonal ones. So, additional row normalization may lead to undesirable scaling which is not advantageous for the subsequent k-means clustering step. Therefore, row normalization is not recommended in the proposed algorithm.

5.3.4 Computational Complexity

In the proposed algorithm, the first step is to compute the eigenspace of each modality X_m. Given the similarity matrix W_m for modality X_m, its degree matrix D_m and shifted Laplacian L_m are computed in step 2 in $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$ time, respectively. Then, the eigen-decomposition of L_m is computed in step 3 which takes $\mathcal{O}(n^3)$ time for the $(n \times n)$ matrix. Therefore, for M modalities, the total complexity of initial eigenspace construction is $\mathcal{O}(Mn^3)$. Next, the basis $\mathbf{U}_M$ is constructed in M steps. At each step of basis construction, the matrices S_{m+1}, P_{m+1}, and Q_{m+1} are computed in step 8 of the algorithm. It takes $\mathcal{O}(nr^2)$ time. The Gram-Schmidt orthogonalization in step 9 also has complexity of $\mathcal{O}(nr^2)$ for $(n \times r)$ matrix Q_{m+1}. The total complexity of basis construction in steps 7-11 is $\mathcal{O}(nr^2)$. The new eigenvalue problem $\mathbf{H}$ of size $(Mr \times Mr)$ is formulated in steps 12-13, which takes $\mathcal{O}(M^3r^3)$ time, owing to matrix multiplications. The subsequent eigen-decomposition of $\mathbf{H}$ in step 14 also takes $\mathcal{O}(M^3r^3)$ time. The rotation of $\mathbf{U}_M$ in step 15 has complexity of $\mathcal{O}(nr^2)$. Finally, after the construction of joint eigenspace $\Psi\left(\mathbf{L}^{r*}\right)$, k-means clustering is performed on $(n \times k)$ matrix $\mathbf{V}^k$ which has time complexity of $\mathcal{O}(t_{max}nk^2)$, where t_{max} is the maximum number of iterations the k-means algorithm runs.

Hence, the overall computational complexity of the proposed CoALa algorithm, to extract the joint eigenspace and perform spectral clustering on a multimodal data set, is $(\mathcal{O}(Mn^3 + nr^2 + M^3r^3 + nr^2 + t_{max}nk^2) =)\mathcal{O}(Mn^3)$, assuming $M, r, k << n$. It implies that the overall complexity of the proposed algorithm is dominated by the individual eigenspace construction of initial stage.

5.3.5 Choice of Convex Combination

The convex combination vector $\boldsymbol{\alpha}$ determines the weight of the influence of each Laplacian on the final eigenspace. According to Fiedler's theory of spectral graph partitioning [66], the algebraic connectivity or the Fiedler value of a graph G is the second minimum eigenvalue of the Laplacian of G. The Fiedler value represents the weight of the minimum cut that partitions the corresponding graph into two subgraphs. Moreover, by Property 5.3, the lower the eigenvalue or cut-weight of the normalized Laplacian $\mathcal{L}$, the higher is the corresponding eigenvalue of its shifted Laplacian L. The smallest eigenvalue of $\mathcal{L}$ is 0 which corresponds to largest eigenvalue, λ_1, of L which is 2, and the second largest eigenvalue, λ_2, reflects how high is the separability of graph G. The corresponding eigenvector u_2, known as the Fiedler vector, can be used to partition the vertices of G [200]. For example, if the Fiedler vector is $u_2 = (u_{21}, ..., u_{2j}, ..., u_{2n})$, spectral partitioning finds a splitting value s such that the objects with $u_{2j} \leqslant s$ belong to a set, while that with $u_{2j} > s$ belong to other. Several popular choices for s have been proposed, or the standard 2-means algorithm can also be applied on u_2 to obtain a 2-partition. Once a 2-partition is obtained, Silhouette index [179] can internally assess the quality of the partition. Silhouette index lies between [-1, 1] and higher value indicates a better partition. A modality with good inherent cluster information is expected to have a higher Fiedler value as well as higher Silhouette index on the Fiedler vector. Let $\mathcal{S}(u_2^m)$ denote the Silhoutte index computed based on a 2-partition of the Fiedler vector corresponding to the m-th modality. Thus, a measure of "relevance" of a modality X_m is defined as

$$\boldsymbol{\chi}_m = \frac{1}{4}\lambda_2^m \left[\mathcal{S}(u_2^m) + 1\right] \tag{5.31}$$

where λ_2^m is the second largest eigenvalue of shifted Laplacian L_m of X_m and u_2^m is the corresponding eigenvector. The term $(\mathcal{S}(u_2^m) + 1)$ lies in $[0, 2]$, while the value of λ_2^m can be at most 2. The factor 1/4 acts as a normalizing factor which upper bounds the value of $\boldsymbol{\chi}$ to 1. Hence, the value of relevance measure $\boldsymbol{\chi}$ lies in $[0, 1]$. Higher value of $\boldsymbol{\chi}_m$ implies higher relevance and better cluster structure. Hence, $\boldsymbol{\chi}$ can be used to obtain a linear ordering of the modalities $X_1, \ldots, X_M$. Let $X_{(1)}, \ldots, X_{(m)}, \ldots, X_{(M)}$ be the ordering of $X_1, \ldots, X_m, \ldots, X_M$ based on decreasing value of relevance $\boldsymbol{\chi}$. In the convex combination vector $\boldsymbol{\alpha}$, the component $\alpha_{(m)}$ corresponding to the weighting factor of modality $X_{(m)}$ is given by

$$\alpha_{(m)} = \boldsymbol{\chi}_{(m)}\beta^{-m}, \text{ where } \beta > 1. \tag{5.32}$$

This implies that based on the index of $X_{(m)}$ in the ordering $X_{(1)}, \ldots, X_{(M)}$, the relevance value of $X_{(m)}$ is damped by a factor of β^m and then used as its contribution in the convex combination $\boldsymbol{\alpha}$. Thus, in $\boldsymbol{\alpha}$, the most relevant modality has contribution of $\frac{\boldsymbol{\chi}_{(1)}}{\beta}$, while the second most relevant one contributes $\frac{\boldsymbol{\chi}_{(2)}}{\beta^2}$, and so on. This assignment of $\boldsymbol{\alpha}$ upweights modalities with better cluster structure, while dampens the effect of irrelevant ones those having poor structure.

5.4 Quality of Eigenspace Approximation

The proposed algorithm constructs the eigenspace $\Psi\left(\mathbf{L}^{r*}\right)$ from a convex combination of rank r approximations of the individual Laplacians L_m's. This eigenspace differs from the full-rank eigenspace $\Psi\left(\mathbf{L}^{r}\right)$, which is the convex combination of complete or full rank information of the individual Laplacians. In real-life multimodal data sets, the individual modalities inherently contain noisy information. The approximation approach prevents propagation of noise from the individual modalities into the final approximate eigenspace $\Psi\left(\mathbf{L}^{r*}\right)$. As a consequence, the approximate subspace is expected to preserve better cluster structure compared to the full-rank one. However, in the ideal case, where the clusters in the individual modalities are well-separated, the approximation approach may loose some important information. So, the difference between the two eigenspaces $\Psi\left(\mathbf{L}^{r}\right)$ and $\Psi\left(\mathbf{L}^{r*}\right)$ is evaluated as a function of the approximation rank r, and can be quantified in terms of their eigenvalues and eigenvectors. The difference between the eigenvalues can be measured directly in terms of their magnitude, while the difference between the eigenvectors is measured in terms of difference between the subspaces spanned by the two sets of eigenvectors. Similar to Chapter 4, the principal angles between subspaces (PABS) [16, 70] is also used here to measure the difference between two subspaces. The PABS is a generalization of the concept of angle between two vectors to a set of angles between two subspaces. The principal angles between two subspace $\mathcal{A}$ and $\mathcal{B}$ and their corresponding principal sines, denoted by $\sin\Theta(\mathcal{A}, \mathcal{B})$, are defined in Definition 4.1 and Definition 4.2 of Chapter 4.

In order to bound the difference between the eigenvectors of two eigenspaces $\Psi\left(\mathbf{L}^{r}\right)$

and $\Psi(\mathbf{L}^{r*})$, the theory of *perturbation of invariant subspaces* [202] and Davis Kahan theorem [49] are used. The eigenvalues and eigenvectors of the full-rank eigenspace $\Psi(\mathbf{L}^r)$ are given by $\Gamma^r = diag(\gamma_1, \ldots, \gamma_r)$ and $\mathbf{Z}^r$, respectively, as in (5.14), where $\gamma_r \neq \gamma_{r+1}$, while those for the approximate eigenspace $\Psi(\mathbf{L}^{r*})$ are given by $\Pi^r = diag(\pi_1, \ldots, \pi_r)$ and $\mathbf{V}^r$, respectively, as in (5.30). The columns of $\mathbf{Z}^r$ span the full-rank subspace formed by the convex combination of full rank L_m's, while those of $\mathbf{V}^r$ span the approximate subspace formed by rank r approximation of L_m's. The difference between the subspaces spanned by the column vectors of $\mathbf{Z}^r$ and $\mathbf{V}^r$ is given by the following theorem.

Theorem 5.1. *For any unitarily invariant norm* $\| \,.\, \|$*, the following bound holds on the principal angles between the subspaces defined by* $\mathcal{C}(\mathbf{Z}^r)$ *and* $\mathcal{C}(\mathbf{V}^r)$*:*

$$\|\sin\Theta\left(\mathcal{C}(\mathbf{Z}^r), \mathcal{C}(\mathbf{V}^r)\right)\| \leqslant \frac{\left\|\left(\sum_{m=1}^{M} \alpha_m L_m^{r\perp}\right)\mathbf{V}^r\right\|}{\left(\pi_r - \pi_{r+1} - \sum_{m=1}^{M} \alpha_m \lambda_{r+1}^m\right)}, \tag{5.33}$$

assuming $\pi_r > \pi_{r+1} + \sum_{m=1}^{M} \alpha_m \lambda_{r+1}^m$.

Proof. The matrices $\mathbf{Z}$ and Γ contain the eigenpairs of $\mathbf{L}$. For the given r, let $\mathbf{Z}$ and Γ be partitioned as

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}^r & \mathbf{Z}^{r\perp} \end{bmatrix} \quad \text{and} \quad \Gamma = \begin{bmatrix} \Gamma^r & \mathbf{0} \\ \mathbf{0} & \Gamma^{r\perp} \end{bmatrix}. \tag{5.34}$$

Since $\mathbf{Z}^r$ and $\mathbf{Z}^{r\perp}$ contain eigenvectors of $\mathbf{L}$, so,

$$\mathbf{L}\mathbf{Z}^r = \mathbf{Z}^r\Gamma^r \subset \mathcal{C}(\mathbf{Z}^r). \tag{5.35}$$

This implies that the transformation of any vector $v \in \mathcal{C}(\mathbf{Z}^r)$ lies in $\mathcal{C}(\mathbf{Z}^r)$ itself. So, $\mathbf{Z}^r$ spans an invariant subspace of the matrix $\mathbf{L}$ [202]. Similarly,

$$\mathbf{L}\mathbf{Z}^{r\perp} = \mathbf{Z}^{r\perp}\Gamma^{r\perp} \subset \mathcal{C}(\mathbf{Z}^{r\perp}). \tag{5.36}$$

So, $\mathbf{Z}^{r\perp}$ also spans an invariant subspace of $\mathbf{L}$. Moreover, the columns of $\mathbf{Z}^{r\perp}$ span the subspace orthogonal to the one spanned by the columns of $\mathbf{Z}^r$. Now, let

$$\mathbf{B}_1 = (\mathbf{Z}^r)^T\mathbf{L}\mathbf{Z}^r = \Gamma^r \text{ and } \ \mathbf{B}_2 = (\mathbf{Z}^{r\perp})^T\mathbf{L}\mathbf{Z}^{r\perp} = \Gamma^{r\perp}. \tag{5.37}$$

According to the theory of invariant subspaces [202], $\mathbf{B}_1$ and $\mathbf{B}_2$ are called the representation of $\mathbf{L}$ with respect to the bases $\mathbf{Z}^r$ and $\mathbf{Z}^{r\perp}$, respectively. The matrix $\mathbf{B}_1 = \Gamma^r$ contains eigenvalues $\gamma_1, \ldots, \gamma_r$, while $\mathbf{B}_2 = \Gamma^{r\perp}$ contains eigenvalues $\gamma_{r+1}, \ldots, \gamma_n$. Let $\Omega(B)$ denote the set of eigenvalues of a matrix B. Under the assumption that $\gamma_r \neq \gamma_{r+1}$, we have

$$\Omega(\mathbf{B}_1) \cap \Omega(\mathbf{B}_2) = \varnothing. \tag{5.38}$$

It follows from (5.38) that the eigenvalues of $\mathbf{B}_1$ and $\mathbf{B}_2$ are non-intersecting. So, $\mathbf{Z}^r$ spans a simple invariant subspace of $\mathbf{L}$ with its complementary subspace being spanned by $\mathbf{Z}^{r\perp}$. Also, $\begin{bmatrix}\mathbf{Z}^r & \mathbf{Z}^{r\perp}\end{bmatrix}$ is unitary and $\mathbf{L}$ can be decomposed as

$$\mathbf{L} = \mathbf{Z}^r\mathbf{B}_1(\mathbf{Z}^r)^T + \mathbf{Z}^{r\perp}\mathbf{B}_2(\mathbf{Z}^{r\perp})^T. \tag{5.39}$$

The decomposition in (5.39) is called the spectral resolution of $\mathbf{L}$ along $\mathbf{Z}^r$ and $\mathbf{Z}^{r\perp}$. Now, let $\mathbf{L}$ be written as

$$\begin{aligned}
& & \mathbf{L} &= \sum_{m=1}^{M} \alpha_m L_m = \sum_{m=1}^{M} \alpha_m \left(L_m^r + L_m^{r\perp}\right), \\
\Rightarrow & & \mathbf{L} &= \sum_{m=1}^{M} \alpha_m L_m^r + \sum_{m=1}^{M} \alpha_m L_m^{r\perp}, \\
\Rightarrow & & \mathbf{L} &= \mathbf{L}^{r*} + \mathbf{L}^{r\perp*}, \text{ where } \mathbf{L}^{r\perp*} = \sum_{m=1}^{M} \alpha_m L_m^{r\perp}. && (5.40)
\end{aligned}$$

Let the eigenvectors and eigenvalues of $\mathbf{L}^{r*}$ be partitioned as

$$\mathbf{V} = \begin{bmatrix}\mathbf{V}^r & \mathbf{V}^{r\perp}\end{bmatrix} \quad \text{and} \quad \Pi = \begin{bmatrix}\Pi^r & \mathbf{0} \\ \mathbf{0} & \Pi^{r\perp}\end{bmatrix}. \tag{5.41}$$

Since $\mathbf{V}^r$ contains eigenvectors of $\mathbf{L}^{r*}$, so

$$\mathbf{L}^{r*}\mathbf{V}^r = \mathbf{V}^r\Pi^r \subset \mathcal{C}(\mathbf{V}^r). \tag{5.42}$$

This implies that $\mathbf{V}^r$ spans an invariant subspace of $\mathbf{L}^{r*}$ and Π^r is a Hermitian matrix of order r which gives the representation of $\mathbf{L}^{r*}$ with respect to the basis $\mathbf{V}^r$. According to (5.40), $\mathbf{L}$ can be written as the sum of $\mathbf{L}^{r*}$ and a perturbation $\mathbf{L}^{r\perp*}$. The perturbation theory [202] analyzes how near is the perturbed subspace $\mathcal{C}(\mathbf{V}^r)$ to an invariant subspace $\mathcal{C}(\mathbf{Z}^r)$ of $\mathbf{L}$, in terms of the perturbation matrix $\mathbf{L}^{r\perp*}$. So, the residual $\mathcal{R}$ of the matrix $\mathbf{L}$, with respect to a perturbed basis $\mathbf{V}^r$ and the Hermitian matrix Π^r, is given by

$$\begin{aligned}
\mathcal{R} &= \mathbf{L}\mathbf{V}^r - \mathbf{V}^r\Pi^r \\
&= \left(\mathbf{L}^{r*} + \sum_{m=1}^{M} \alpha_m L_m^{r\perp}\right)\mathbf{V}^r - \mathbf{V}^r\Pi^r && \text{[from (5.40)]}
\end{aligned}$$

$$
\begin{aligned}
&= \mathbf{L}^{r*}\mathbf{V}^r + \left(\sum_{m=1}^{M} \alpha_m L_m^{r\perp}\right)\mathbf{V}^r - \mathbf{V}^r\Pi^r \\
&= \mathbf{V}^r\Pi^r + \left(\sum_{m=1}^{M} \alpha_m L_m^{r\perp}\right)\mathbf{V}^r - \mathbf{V}^r\Pi^r \\
&= \left(\sum_{m=1}^{M} \alpha_m L_m^{r\perp}\right)\mathbf{V}^r. \qquad (5.43)
\end{aligned}
$$

The matrices Π^r and $\mathbf{B}_2 = \Gamma^{r\perp}$ consist of the eigenvalues of the perturbed subspace $\mathcal{C}(\mathbf{V}^r)$ and the complementary invariant subspace $\mathcal{C}(\mathbf{Z}^{r\perp})$, respectively. According to the Davis-Kahan theorem [49], the bound on the difference between an invariant subspace $\mathcal{C}(\mathbf{Z}^r)$ and its perturbation $\mathcal{C}(\mathbf{V}^r)$ holds only if the eigenvalues of the perturbed subspace and the complementary invariant subspace are non-intersecting. So, the range in which the eigenvalues of Π^r and $\mathbf{B}_2$ lie are derived.

The matrix Π contains the eigenvalues of $\mathbf{L}^{r*}$ given by $\Pi = diag(\pi_1, \ldots, \pi_r, \pi_{r+1}, \ldots, \pi_n)$ which can be partitioned into Π^r and $\Pi^{r\perp}$ as in (5.41). So, the eigenvalues of Π^r satisfy

$$\Omega(\Pi^r) \in [\pi_r, \pi_1]. \qquad (5.44)$$

The range of the eigenvalues of $\mathbf{B}_2$ is derived next. Since each L_m is a real symmetric matrix, its low-rank approximations L_m^r and $L_m^{r\perp}$ are also real symmetric matrices. So, each L_m^r and $L_m^{r\perp}$ have the Hermitian property and $\mathbf{L}^{r\perp*}$ is the sum of M Hermitian matrices according to (5.40). The eigenvalues of $L_m^{r\perp}$ lie in $[\lambda_{r+1}^m, \lambda_n^m]$, and those of $\alpha_m L_m^{r\perp}$ lie in $[\alpha_m\lambda_{r+1}^m, \alpha_m\lambda_n^m]$. Applying Weyl's inequality [202] for the eigenvalues of sum of Hermitian matrices to $\mathbf{L}^{r\perp*}$, we get

$$\Omega(\mathbf{L}^{r\perp*}) \in \left[\sum_{m=1}^{M} \alpha_m\lambda_{r+1}^m, \sum_{m=1}^{M} \alpha_m\lambda_n^m\right]. \qquad (5.45)$$

The eigenvalues of $\mathbf{L}$ lie in $[\gamma_n, \gamma_1]$, while those of $\mathbf{L}^{r*}$ lie in $[\pi_n, \pi_1]$. The range of eigenvalues of $\mathbf{L}^{r\perp*}$ is given by (5.45). Again, $\mathbf{L}$ $(= \mathbf{L}^{r*} + \mathbf{L}^{r\perp*})$ is the sum of two Hermitian matrices $\mathbf{L}^{r*}$ and $\mathbf{L}^{r\perp*}$. So, using Weyl's inequality, the eigenvalues of $\mathbf{L}$ satisfy

$$\pi_j + \sum_{m=1}^{M} \alpha_m\lambda_n^m \leqslant \gamma_j \leqslant \pi_j + \sum_{m=1}^{M} \alpha_m\lambda_{r+1}^m, \qquad (5.46)$$

for $j = 1, ..., n$. As stated previously, $\mathbf{B}_2 = \Gamma^{r\perp}$ consists of eigenvalues $\gamma_{r+1}, \ldots, \gamma_n$ of $\mathbf{L}$.

Thus, the maximum eigenvalue of $\mathbf{B}_2$ is γ_{r+1}, which using (5.46) is bounded by

$$\gamma_{r+1} \leqslant \pi_{r+1} + \sum_{m=1}^{M} \alpha_m \lambda_{r+1}^m. \tag{5.47}$$

According to (5.44), the minimum eigenvalue of Π^r is π_r. Let δ be the minimum of the separation between the eigenvalues of Π^r and $\mathbf{B}_2$, which is given by

$$\begin{aligned} \delta &= \min\{\Omega(\Pi^r)\} - \max\{\Omega(\mathbf{B}_2)\} \\ &= \pi_r - \pi_{r+1} - \sum_{m=1}^{M} \alpha_m \lambda_{r+1}^m > 0. \end{aligned} \tag{5.48}$$

$$\text{So, } \pi_r - \delta = \pi_{r+1} + \sum_{m=1}^{M} \alpha_m \lambda_{r+1}^m. \tag{5.49}$$

From (5.47) and (5.49), we get $\gamma_{r+1} \leqslant (\pi_r - \delta)$. Moreover, as $\gamma_n \leqslant \gamma_{r+1}$, $\gamma_n \leqslant (\pi_r - \delta)$. Also, $(\pi_1 + \delta) \geqslant (\pi_r - \delta)$, as $\pi_1 \geqslant \pi_r$. This implies that the eigenvalues of $\mathbf{B}_2$, that is, $\gamma_{r+1}, \ldots, \gamma_n$ satisfy

$$\Omega(\mathbf{B}_2) \in \mathbb{R} \backslash [\pi_r - \delta, \pi_1 + \delta]. \tag{5.50}$$

The constraints in (5.44) and (5.50) imply that the eigenvalues of Π^r are included in the interval $[\pi_r, \pi_1]$, while those of $\mathbf{B}_2$ are excluded from the interval $[\pi_r - \delta, \pi_1 + \delta]$, where $\delta > 0$. So, for an invariant subspace $\mathcal{C}(\mathbf{Z}^r)$, the eigenvalues of its complementary subspace $\mathcal{C}(\mathbf{Z}^{r\perp})$ and those of its perturbed subspace $\mathcal{C}(\mathbf{V}^r)$ are non-intersecting. Finally, according to the Davis-Kahan theorem [49] which bounds the difference between an invariant subspace and its perturbation, for any unitarily invariant norm $\| \, . \, \|$,

$$\|\sin\Theta\left(\mathcal{C}(\mathbf{Z}^r), \mathcal{C}(\mathbf{V}^r)\right)\| \leqslant \frac{\| \, \mathcal{R} \, \|}{\delta}. \tag{5.51}$$

Substituting the value of $\mathcal{R}$ and δ from (5.43) and (5.48), respectively, in (5.51), we get

$$\|\sin\Theta\left(\mathcal{C}(\mathbf{Z}^r), \mathcal{C}(\mathbf{V}^r)\right)\| \leqslant \frac{\left\| \left(\sum\limits_{m=1}^{M} \alpha_m L_m^{r\perp} \right) \mathbf{V}^r \right\|}{\left(\pi_r - \pi_{r+1} - \sum\limits_{m=1}^{M} \alpha_m \lambda_{r+1}^m \right)} \tag{5.52}$$

This concludes the proof. ■

The above theorem holds for any set of M symmetric positive semi-definite matrices and their convex combination.

Corollary 5.1. *Let $tr(B)$ denote the trace of matrix B. Then,*

$$\|\sin\Theta\left(\mathcal{C}(\mathbf{Z}^r),\mathcal{C}(\mathbf{V}^r)\right)\|_F^2 \leqslant \frac{tr\left((\mathbf{V}^r)^T\left(\sum\limits_{m=1}^{M}\alpha_m L_m^{r\perp}\right)^2\mathbf{V}^r\right)}{\left(\pi_r-\pi_{r+1}-\sum\limits_{m=1}^{M}\alpha_m\lambda_{r+1}^m\right)}. \tag{5.53}$$

Proof. The Frobenius norm of a matrix B, given by $\|B\|_F = \sqrt{tr(B^TB)}$, is an unitarily invariant norm. The squared Frobenius norm of $\mathcal{R}$ in (5.43) is given by

$$\|\mathcal{R}\|_F^2 = tr\left((\mathbf{V}^r)^T\left(\sum_{m=1}^{M}\alpha_m L_m^{r\perp}\right)^2\mathbf{V}^r\right). \tag{5.54}$$

The Davis-Kahan theorem holds for any unitarily invariant norm. So, substituting the value of δ and the Frobenius norm of $\mathcal{R}$ in (5.51), the required bound in (5.53) is obtained. ■

For a given value of r, $\|\sin\Theta\left(\mathcal{C}(\mathbf{Z}^r),\mathcal{C}(\mathbf{V}^r)\right)\|_F^2$ measures the difference between the full-rank and approximate subspaces, in terms of the sum of squares of r principal sines between them. To make the differences comparable across different values of r, the mean squared principal sine is considered, which is given by

$$\Phi^r = \frac{1}{r}\|\sin\Theta\left(\mathcal{C}(\mathbf{Z}^r),\mathcal{C}(\mathbf{V}^r)\right)\|_F^2 \leqslant \frac{tr\left((\mathbf{V}^r)^T\left(\sum\limits_{m=1}^{M}\alpha_m L_m^{r\perp}\right)^2\mathbf{V}^r\right)}{r\left(\pi_r-\pi_{r+1}-\sum\limits_{m=1}^{M}\alpha_m\lambda_{r+1}^m\right)}. \tag{5.55}$$

The matrix $L_m^{r\perp}$ denotes the approximation of L_m using eigenpairs $(r+1)$ to n. As r approaches the full rank n, the approximation of L_m using the remaining $(n-r)$ eigenpairs approaches to 0, that is, $L_m^{r\perp}\rightarrow 0$. Hence,

$$\lim_{r\rightarrow n}\sum_{m=1}^{M}\alpha_m L_m^{r\perp} = 0. \tag{5.56}$$

Taking limits in (5.55) and then substituting the value of (5.56) in the right hand side of (5.55), we get

$$\lim_{r\rightarrow n}\Phi^r = 0. \tag{5.57}$$

This implies that, as the rank r approaches to the full rank of the individual L_m, the difference between the full-rank and approximate subspace converges to 0, that is, the approximate subspace converges to the full-rank subspace.

The eigenvalues of $\mathbf{L}^r$ and $\mathbf{L}^{r*}$ are given by the elements of the diagonal matrices Γ and Π, respectively. The bound on the difference between the eigenvalues is given as follows.

Theorem 5.2. *The eigenvalues of* $\mathbf{L}$ *and* $\mathbf{L}^{r*}$ *satisfy the following bound:*

$$\sum_{j=1}^{n} (\gamma_j - \pi_j)^2 \leqslant \sum_{j=r+1}^{n} \sum_{m=1}^{M} \alpha_m (\lambda_j^m)^2. \tag{5.58}$$

Proof. The decomposition of $\mathbf{L}$ in (5.40) gives $\mathbf{L} = \mathbf{L}^{r*} + \mathbf{L}^{r\perp*}$. Both $\mathbf{L}^{r*}$ and $\mathbf{L}^{r\perp*}$ are low-rank approximations of the real-symmetric matrix $\mathbf{L}$ using its eigenpairs. So, $\mathbf{L}^{r*}$ and $\mathbf{L}^{r\perp*}$ are also real and symmetric. The eigenvalues of $\mathbf{L}^{r*}$ are given by $\pi_1, \ldots, \pi_n$, while those of $\mathbf{L}^{r\perp*}$ are given by

$$\sum_{m=1}^{M} \alpha_m \lambda_{r+1}^m, \ldots, \sum_{m=1}^{M} \alpha_m \lambda_n^m, \tag{5.59}$$

according to (5.45). $\mathbf{L}$ is the sum of two real-symmetric matrices and has eigenvalues $\gamma_1, \ldots, \gamma_n$. The squared Frobenius norm of $\mathbf{L}^{r\perp*}$, given by the sum of squares of its eigenvalues, is

$$\left\|\mathbf{L}^{r\perp*}\right\|_F^2 = \sum_{j=r+1}^{n} \sum_{m=1}^{M} \alpha_m (\lambda_j^m)^2. \tag{5.60}$$

According to the Weilandt-Hoffman theorem [76], the sum of squares of the difference between the eigenvalues of $\mathbf{L}$ and $\mathbf{L}^{r*}$ is bounded by the squared Frobenius norm of the residual $\mathbf{L}^{r\perp*}$. Therefore,

$$\sum_{j=1}^{n} (\gamma_j - \pi_j)^2 \leqslant \sum_{j=r+1}^{n} \sum_{m=1}^{M} \alpha_m (\lambda_j^m)^2. \tag{5.61}$$

This proves the bound on the eigenvalues. ■

Following analysis establishes that the difference between the eigenvalues of $\mathbf{L}$ and $\mathbf{L}^{r*}$ approaches to 0 as the rank r approaches to the full rank of $\mathbf{L}$. Let

$$\Delta^r = \frac{1}{n} tr\{(\Gamma - \Pi)^2\} = \frac{1}{n} \sum_{j=1}^{n} (\gamma_j - \pi_j)^2. \tag{5.62}$$

According to (5.58),

$$\Delta^r \leqslant \frac{1}{n} \sum_{j=r+1}^{n} \sum_{m=1}^{M} \alpha_m (\lambda_j^m)^2. \tag{5.63}$$

So, Δ^r bounds the squared sum of the difference between the eigenvalues of $\mathbf{L}$ and $\mathbf{L}^{r*}$. For $m = 1, \ldots, M$, each L_m is a positive semi-definite matrix with n eigenvalues $\lambda_1^m \geqslant \ldots \geqslant \lambda_n^m \geqslant 0$. As the value of r approaches n, the eigenvalue λ_r^m approaches the smallest eigenvalue λ_n^m. Moreover, as there are only n eigenvalues, the value of λ_j^m is 0 for any $j > r$

when r tends to n. Therefore,

$$\lim_{r\to n} \Delta^r = \lim_{r\to n} \frac{1}{n} \sum_{j=r+1}^{n} \sum_{m=1}^{M} \alpha_m (\lambda_j^m)^2 = 0. \tag{5.64}$$

The limits in (5.57) and (5.64) imply that as the approximation rank r approaches to the full rank, the approximate eigenspace $\Psi\left(\mathbf{L}^{r*}\right)$ converges to the full-rank one $\Psi\left(\mathbf{L}^{r}\right)$, in terms of both eigenvectors and eigenvalues.

5.5 Experimental Results and Discussion

The performance of the proposed CoALa algorithm is compared with that of ten existing integrative clustering approaches, namely, cluster of cluster analysis (COCA) [93], LR-Acluster [243], joint and individual variance explained (JIVE) [141], angle-based JIVE (A-JIVE) [63], iCluster [192], principal component analysis (PCA) on the concatenated data (PCA-con) [6], similarity network fusion (SNF) [234], normality based low-rank subspace (NormS) [111] (proposed in Chapter 3), and selective update of relevant eigenspaces (SURE) [112] (proposed in Chapter 4). The experimental setup for the existing approaches is followed same as that of Chapter 3. The performance of the JIVE algorithm, from this chapter onwards, is reported corresponding to permutation test based rank estimation (JIVE-Perm), as that is the default choice as mentioned in [141]. The clustering performance corresponding to Bayesian information criteria based rank estimation (JIVE-BIC) is reported in Tables 4.4 and 4.5 of Chapter 4. These tables also show that amongst the two consensus clustering approaches, namely, COCA and Bayesian consensus clustering (BCC) [140], COCA has better clustering performance in majority of the data sets. Hence, the performance of COCA is reported from this chapter onwards. The results of the BCC algorithm are available in Tables 4.4 and 4.5 of the last chapter. The R implementation of the proposed algorithm is available at `https://github.com/Aparajita-K/CoALa`.

The performance of different algorithms is evaluated using six external cluster evaluation indices, namely, accuracy, normalized mutual information (NMI), adjusted Rand index (ARI), F-measure, Rand index, and purity, which compare the identified clusters with the clinically established cancer subtypes and the ground truth class information for the benchmark data sets. Experimental results corresponding to Jaccard and Dice coefficients are reported in [113]. For the low-rank based approaches, where clustering is performed in a subspace, four internal cluster validity indices, namely, Silhouette, Dunn, Davies-Bouldin (DB), and Xie-Beni indices are used to evaluate the compactness and separability of the clusters in the extracted subspace. The evaluation indices are described in Appendix B.

5.5.1 Description of Data Sets

In this work, the clustering performance is extensively studied on eight real-life cancer data sets, obtained from The Cancer Genome Atlas (TCGA) (`https://cancergenome.nih.gov/`). The data sets considered here are, namely, colorectal carcinoma (CRC), lower grade glioma (LGG), stomach adenocarcinoma (STAD), breast adenocarcinoma (BRCA), ovarian carcinoma (OV), cervical carcinoma (CESC), lung carcinoma (LUNG), and kidney carcinoma (KIDNEY). The CRC has two subtypes: colon and rectum carcinoma, depending on their site of origin. The LUNG and KIDNEY cancers have two and three histological subtypes, respectively, based on the tissue of origin. For the other cancers, TCGA research network has identified three subtypes in LGG [217] and CESC [218], and four subtypes in STAD [216], BRCA [214], and OV [215], by comprehensive integrated analysis. The CRC, LGG, STAD, BRCA, CESC, OV, LUNG, and KIDNEY data sets have 464, 267, 242, 398, 124, 334, 671, and 737 samples, respectively. For each of these data sets, four different omic modalities are considered, namely, DNA methylation (mDNA), gene expression (RNA), microRNA expression (miRNA), and reverse phase protein array expression (RPPA). The pairwise similarity $w_m(i,j)$ between samples x_i and x_j of the modality X_m is computed using the Gaussian similarity kernel

$$w_m(i,j) = \exp\left\{-\frac{\rho_m^2(x_i,x_j)}{2\sigma_m^2}\right\}, \tag{5.65}$$

where $\rho_m(x_i,x_j)$ denotes the Euclidean distance between samples x_i and x_j in X_m and σ_m is the standard deviation of the Gaussian kernel. The value of σ_m is empirically set to be half of the maximum pairwise distance between any two points of the modality. Choice of this similarity function results in a completely connected graph for each modality.

Seven other data sets from different application domains like social networks and general images are also employed in this study to compare the clustering performance of the proposed and existing algorithms. Among the social network data sets, Football, Politics-UK, and Rugby are Twitter data sets which consist of social connection information among Twitter users, while CORA is a citation network data set of machine learning papers. Each Twitter data set has a heterogeneous collection of nine network and content-based modalities, namely, follows, followed-by, mentions, mentioned-by, retweets, retweeted-by, lists500, tweets500, and listmerged500. The CORA data set consists of two modalities, one represents content information and the other represents inbound/outbound citation relation. The cosine similarity is used to compute the pairwise similarities between the samples in the social network data set. Among the general image data sets, ORL is a face clustering data set and Digits is handwritten numeral identification data set, while Caltech7 is an object recognition data set. The modalities of Digits, ORL, and Caltech7 data sets are constructed from different types features extracted from the sample images. The Gaussian similarity kernel described above is used to construct the similarity matrices for the image data sets. A brief description of the omics and benchmark data sets and pre-processing steps is provided in Appendix A.

5.5.2 Optimum Value of Rank

For each multi-view data set having M views and k clusters, the proposed algorithm selects r eigenpairs from each of the M individual Laplacians and constructs a joint eigenspace of rank rM. Similar to the existing spectral clustering algorithms [162, 194], the proposed CoALa algorithm also performs k-means clustering on k eigenvectors of the final eigenspace. Since the clustering is performed in a k-dimensional subspace, the rank r of the individual Laplacians should be $r \geqslant \lceil k/M \rceil$. To find out the optimal value of rank r, the Silhouette index [179] is used. It lies between $[-1, 1]$ and a higher value implies better clustering. In

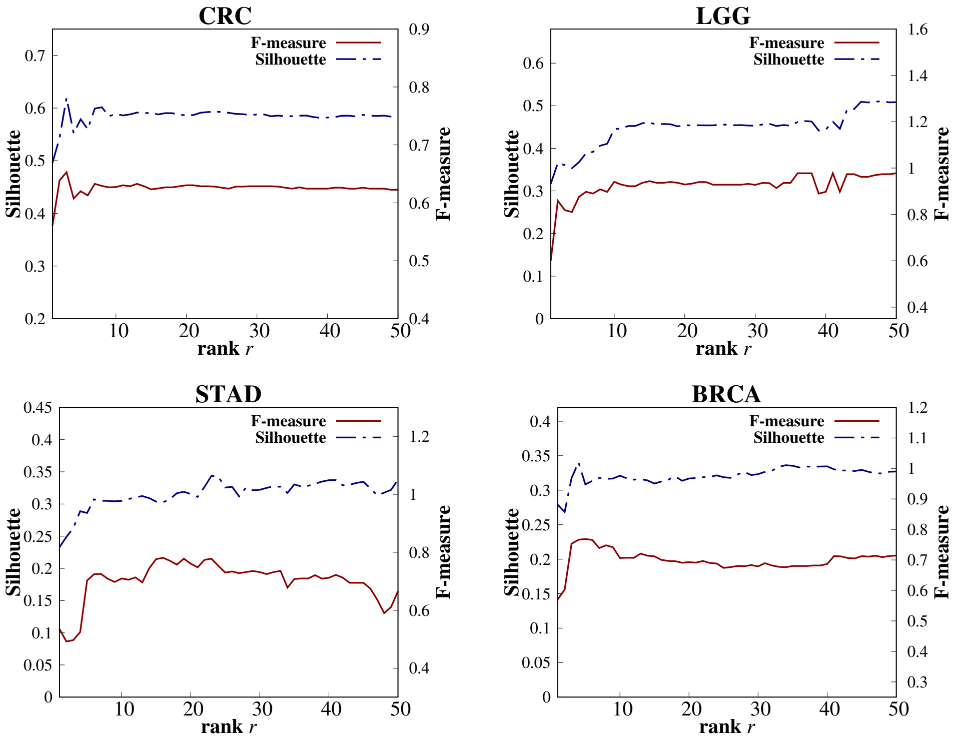


Figure 5.1: Variation of Silhouette index and F-measure for different values of rank parameter r on omics data sets.

order to choose the rank parameter, the value of r is varied from $\lceil k/M \rceil$ to 50 and for each value of r, the Silhouette index $\mathcal{S}(r)$ is evaluated for clustering on the k largest eigenvectors of the final eigenspace. The optimal value of r, that is $r^\star$, is obtained using the following relation:

$$r^\star = \arg\max_r \{\mathcal{S}(r)\}. \tag{5.66}$$

The variation of both Silhouette index and F-measure with respect to the rank r is shown in Figure 5.1 for different omics data sets and in Figure 5.2 for the benchmark data sets. The plots in Figures 5.1 and 5.2 show that the values of Silhouette index and F-measure vary in a similar fashion. The Silhouette index is an internal cluster validity measure computed based on the generated clusters, while F-measure is an external index which compares the generated clusters with the ground truth class information. Since these two indices are found to vary similarly, the optimum value of Silhouette index would also produce the optimum value of F-measure for the same parameter configuration. Using this criterion, the optimal values of rank for CRC, LGG, STAD, BRCA, CESC, OV, LUNG, and KIDNEY data sets are 3, 48, 23, 4, 2, 20, 4, and 46, respectively, while for the benchmark data sets Football, Politics-UK, Rugby, and Digits are 22, 45, 7, and 6, respectively. It is also observed that for BRCA, CRC, Football, Politics-UK, and Digits data sets, the F-measure corresponding to $r^\star$ coincides with the best value of F-measure obtained over different values of rank r. The similarly varying curves of Silhouette and F-measure in

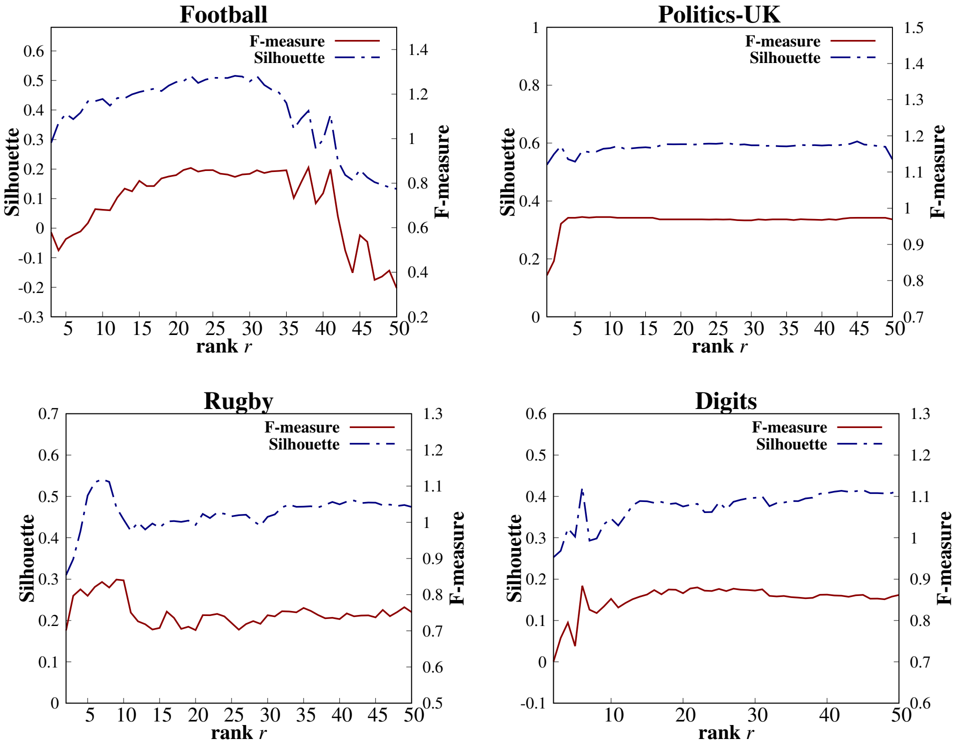
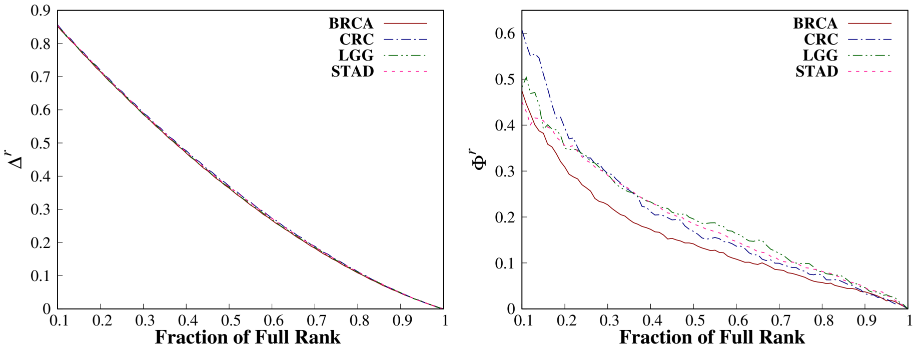


Figure 5.2: Variation of Silhouette index and F-measure for different values of rank parameter r on benchmark data sets.

Figures 5.1 and 5.2 justify the use of Silhouette index to find out the optimal rank.

5.5.3 Difference Between Eigenspaces

The proposed method constructs an eigenspace from low-rank approximations of individual graph Laplacians. This eigenspace is an approximation of the full-rank eigenspace which considers the complete or full rank information of all the Laplacians. As defined in Section 5.4, for a given rank r, the difference between the full-rank and approximate eigenspaces, in terms of its eigenvalues and eigenvectors, is given by Δ^r and Φ^r, respectively. Here, the variation in the difference between these two eigenspaces is observed with the increase in rank r. For each omic data set, Δ^r and Φ^r are computed for different fractions of the full rank of that data set. The variation in the values of Δ^r and Φ^r, with the increase in rank r, is shown in Figures 5.3(a) and 5.3(b), respectively, for different data sets. Figure 5.3(a) shows that the difference between eigenvalues of the two eigenspaces monotonically decreases to 0 with the increase in rank, for all the data sets. Figure 5.3(b), on the other hand, shows that the difference between the subspaces, spanned by the eigenvectors of the two eigenspaces, also converges to 0 as the value of rank r approaches the full rank of the data set. However, the change in variation in case of eigenvectors is not monotonically decreasing as in the case of eigenvalues. For some of the smaller values of rank r, the difference also increases between two consecutive values. This is due to the fact that for a given value of r, there can be infinitely possible rank r subspaces of an n dimensional



(a) Difference in Eigenvalues

(b) Difference in Eigenvectors

Figure 5.3: Variation of difference between full-rank and approximate eigenspaces with respect to rank r.

vector space. For small values of r, the rank r subspaces of individual modalities can be very different from each other due to the large number of possibilities. Consequently, the approximate subspace constructed from these subspaces tends to vary a lot from the full-rank subspace. Hence, the variation in the difference between the full-rank and approximate sets of eigenvectors fluctuates for small values of rank r. However, as r approaches the full-rank, the number of possible subspaces reduces and the difference between the eigenvectors monotonically decreases to 0.

5.5.4 Effectiveness of Proposed CoALa Algorithm

This subsection illustrates the significance of different aspects of the proposed algorithm such as integration of multiple modalities over individual ones, use of approximate Laplacians as opposed to full-rank ones, choice of the convex combination α, and so on, for four omics data sets: CRC, LGG, STAD, and BRCA, and four benchmark data sets: Football, Politics-UK, Rugby, and Digits, as examples.

5.5.4.1 Importance of Data Integration

The proposed CoALa algorithm performs clustering on the k largest eigenvectors of the approximate eigenspace constructed by integrating multiple low-rank Laplacians. To establish the importance of this integration, the performance of the proposed algorithm is compared with spectral clustering on the individual modalities in Tables 5.1, 5.2, and 5.3.

5.5.4.1.1 Omics Data Sets The results in Table 5.1 show that the proposed algorithm performs better than all four individual modalities for CRC, LGG, and STAD data sets, in terms of all external indices, except for the purity measure on the CRC data set. The performance is equal for the purity measure on the CRC data set across all modalities. Since the highest value of the F-measure on CRC data set is obtained for the proposed algorithm, it identifies the smaller cluster better than all the individual Laplacians. For the BRCA data set, RNA outperforms the proposed algorithm, albeit by a very small margin. Among the individual modalities, mDNA gives the best performance for CRC, LGG, and

Table 5.1: Comparative Performance Analysis of Spectral Clustering on Individual Modalities and Proposed Approach on Omics Data Sets

Data Set	Modalities→	mDNA	RNA	miRNA	RPPA	CoALa
CRC	Accuracy	0.5043103	0.5107759	0.5409483	0.5301724	**0.6400862**
	NMI	0.0231494	0.0023715	0.0131094	0.0004871	**0.0185660**
	ARI	-0.018933	-0.001914	0.0041107	-0.004704	**0.0548748**
	F-measure	0.5849894	0.5397796	0.5673758	0.5741394	**0.6529565**
	Rand	0.4989573	0.4991528	0.5022809	0.5007448	**0.5382531**
	Purity	**0.7370690**	**0.7370690**	**0.7370690**	**0.7370690**	**0.7370690**
LGG	Accuracy	0.8352060	0.5917603	0.4307116	0.3970037	**0.9737828**
	NMI	0.5734568	0.2176187	0.0498676	0.0254500	**0.8689965**
	ARI	0.5567870	0.1801875	0.0510240	0.0238319	**0.9199392**
	F-measure	0.8269248	0.5875701	0.4717221	0.4326018	**0.9737835**
	Rand	0.7861508	0.6149925	0.5593760	0.5476050	**0.9622089**
	Purity	0.8352060	0.5917603	0.5318352	0.5280899	**0.9737828**
STAD	Accuracy	0.5413223	0.4793388	0.3719008	0.4173554	**0.768595**
	NMI	0.2282198	0.1779419	0.0771419	0.0831100	**0.510726**
	ARI	0.1927570	0.1047749	0.0514998	0.0460928	**0.4559866**
	F-measure	0.5469686	0.4781377	0.3998266	0.4469459	**0.7778227**
	Rand	0.6509722	0.6239155	0.5989164	0.5883543	**0.7661946**
	Purity	0.5867769	0.5495868	0.4917355	0.4917355	**0.7685950**
BRCA	Accuracy	0.5804020	**0.7688442**	0.4623116	0.4798995	0.7613065
	NMI	0.3408150	0.5277072	0.1947561	0.3140984	**0.5281849**
	ARI	0.3047769	**0.5130244**	0.1663564	0.2359641	0.4874579
	F-measure	0.5982526	**0.7690661**	0.5105008	0.5630781	0.7660191
	Rand	0.7193018	**0.7995519**	0.6455071	0.6689493	0.7922357
	Purity	0.6532663	**0.7688442**	0.5703518	0.5879397	0.7613065

STAD data sets. For LGG and STAD data sets, the performance of the proposed CoALa algorithm is significantly higher than that of their best modality, mDNA.

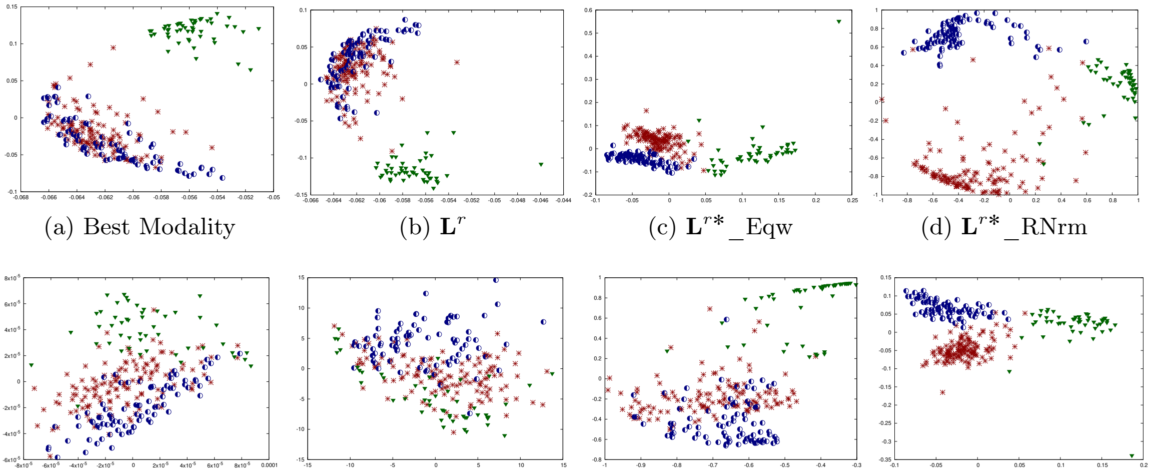
The scatter plots of the first two dimensions for the best modality, mDNA, and the proposed CoALa algorithm are given in Figures 5.4 and 5.5 for LGG and STAD data sets, respectively. The objects in Figures 5.4 and 5.5 are colored according to the previously established TCGA subtypes of LGG [217] and STAD [216]. For the LGG data set, Figure 5.4(a) shows that in the two-dimensional Laplacian subspace of mDNA, one of the subtypes is compact and well-separated while the other two intermingled amongst each other. On the other hand, Figure 5.4(h) for LGG shows that in the proposed subspace all the three clusters are compact and separated from each other. For STAD, Figure 5.5(a) shows that a major part of the two-dimensional subspace consists of points randomly scattered from all the four clusters. However, Figure 5.5(h) shows that although the clusters lack well separability, the proposed subspace can be partitioned into regions where most of the data points belong to a single cluster. The scatter plots for the remaining data sets are provided in the supplementary material. The distinct omic modalities together cover a wide spectrum of biological information and the results in Table 5.1 show that integration of multiple modalities leads to better identification of the disease subtypes compared to unimodal analysis.

Table 5.2: Comparative Performance of Spectral Clustering on Individual Modalities and Proposed Approach on Twitter Data Sets

Modalities→		Followed-By	Follows	Mentioned-By	Mentions	Retweeted-By	Retweets	Tweets500	ListMerged500	Lists500	CoALa
Accuracy	**Football**	0.7419354	0.6584677	0.6673387	0.6737903	0.5427419	0.4834677	0.1895161	0.6649193	0.6241935	**0.8500000**
NMI		0.7910368	0.6899672	0.7399262	0.7432407	0.6169596	0.5489673	0.2404490	0.7199201	0.7123953	**0.8625365**
ARI		0.5814725	0.3702998	0.4422042	0.4531746	0.2413006	0.1239701	0.0244924	0.4588554	0.3908937	**0.7278994**
F-measure		0.7747023	0.7042013	0.7241344	0.7109046	0.5537196	0.5202768	0.2022110	0.7232265	0.6606393	**0.8683491**
Rand		0.9472965	0.9197825	0.9356405	0.9384256	0.8593378	0.7958926	0.7691328	0.9322776	0.9147218	**0.9739682**
Purity		0.7282258	0.6766129	0.7362903	0.7092741	0.5447580	0.5008064	0.2072580	0.6931451	0.6399193	**0.8584677**
Accuracy	**Politics-UK**	0.8902148	0.9140811	0.8310262	0.7878281	0.7248209	0.8377088	0.5011933	0.7016706	0.7517900	**0.9665871**
NMI		0.8382287	0.7343181	0.5463468	0.4936030	0.4158612	0.5270268	0.1308606	0.6834820	0.7278979	**0.9434825**
ARI		0.8375676	0.7843860	0.6325506	0.4746045	0.4046677	0.5552591	0.1496182	0.6300209	0.6998277	**0.9633130**
F-measure		0.9175316	0.8836935	0.8660595	0.7619363	0.8346957	0.7991772	0.5804394	0.8635673	0.8464556	**0.9736129**
Rand		0.9196880	0.8728323	0.8429422	0.7114181	0.7991423	0.7510534	0.6330178	0.8562195	0.8346941	**0.9826084**
Purity		0.9713604	0.9021479	0.8778042	0.7823389	0.8477326	0.8138425	0.6658711	0.9021480	0.8782816	**0.9785203**
Accuracy	**Rugby**	0.6679156	0.6797423	0.5955503	0.6004683	0.7121779	0.6860655	0.3621779	0.3223653	0.6499999	**0.8305621**
NMI		0.6395692	0.6301835	0.6135364	0.6059998	0.6151681	0.5863386	0.2623035	0.2283989	0.5733881	**0.7093834**
ARI		0.4204121	0.4827426	0.3977919	0.3758964	0.5461666	0.5022541	0.1373920	0.0130416	0.4969606	**0.6627701**
F-measure		0.7113898	0.6643790	0.6873041	0.6705410	0.7078636	0.6856623	0.3737361	0.3460789	0.7426962	**0.8349647**
Rand		0.8609769	0.8580120	0.8562299	0.8482375	0.8560331	0.8406967	0.7177268	0.5223523	0.8672685	**0.9067597**
Purity		0.8474238	0.8435597	0.8274004	0.8121780	0.7915691	0.7816159	0.4871194	0.4566745	0.7796253	**0.8606557**

Table 5.3: Comparative Performance of Spectral Clustering on Individual Modalities and Proposed Approach on Digits Data Set

Data Set	Modalities→	Fac	Fou	Kar	Mor	Pix	Zer	CoALa
Digits	Accuracy	0.5614000	0.7096000	0.6638000	0.5109000	0.6520000	0.5350000	**0.8835000**
	NMI	0.6192075	0.6443707	0.6407076	0.5361673	0.6385617	0.4766979	**0.7981981**
	ARI	0.4731459	0.5416071	0.5383434	0.3723571	0.5216976	0.3286005	**0.7645096**
	F-measure	0.6451628	0.7209662	0.7022988	0.5651531	0.6829546	0.5545294	**0.8839913**
	Rand	0.8994301	0.9173923	0.9156842	0.8655854	0.9108559	0.8757654	**0.9576618**
	Purity	0.6223000	0.7100000	0.7027000	0.5414000	0.6890000	0.5350500	**0.8835000**



(a) Best Modality (b) $\mathbf{L}^r$ (c) $\mathbf{L}^{r*}$_Eqw (d) $\mathbf{L}^{r*}$_RNrm

(e) JIVE (f) iCluster (g) SNF (h) $\mathbf{L}^{r*}$_Damp (CoALa)

Figure 5.4: Scatter plots using first two components of different low-rank based approaches on LGG data set.

5.5.4.1.2 Benchmark Data Sets Three Twitter data sets, namely, Football, Politics-UK, and Rugby have nine different modalities, while the image data set, Digits has six. Tables 5.2 and 5.3 compare the performance of clustering on the k largest eigenvectors of the individual shifted Laplacians with that of the proposed approximate subspace for the Twitter and Digits data set, respectively. From the results of Tables 5.2 and 5.3, it is evident that the proposed CoALa algorithm consistently and significantly outperforms all the individual modalites across all four benchmark data sets. For the Digits data set, Table 5.3 shows that all six modalities have significantly lower performance than the proposed approach. In brief, for all the benchmark data sets, integration of multiple modalities always beats the performance of individual Laplacians by a wide margin.

5.5.4.2 Importance of the choice of Convex Combination

In order to establish the effectiveness of the proposed weighting factor (termed as $\mathbf{L}^{r*}$_Damp) described in Section 5.3.5, the clustering performance of the resulting subspace obtained using $\mathbf{L}^{r*}$_Damp is compared with that of the one where all the modalities are equally weighted (termed as $\mathbf{L}^{r*}$_Eqw). The damping factor β in (5.32) is empirically set to 1.25 for all data sets.

5.5.4.2.1 Omics Data Sets The scatter plots for the first two components of $\mathbf{L}^{r*}$_Eqw and $\mathbf{L}^{r*}$_Damp (CoALa) subspaces are given in Figures 5.4 and 5.5 for LGG and STAD data sets, respectively. For LGG, Figure 5.4(c) for $\mathbf{L}^{r*}$_Eqw shows that two of the three clusters are highly compact, however, they also lack inter-cluster separability. In case of the proposed $\mathbf{L}^{r*}$_Damp subspace, in Figure 5.4(h), these two clusters have lower compactness but are well-separated from each other. For STAD, scatter plots for $\mathbf{L}^{r*}$_Eqw and $\mathbf{L}^{r*}$_Damp (CoALa) in Figures 5.5(c) and 5.5(h), respectively, are of similar nature, although $\mathbf{L}^{r*}$_Eqw shows slightly better inter-cluster separability compared to $\mathbf{L}^{r*}$_Damp.

Table 5.4: Comparative Performance Analysis of Equally and Damped Weighted Combination on Omics Data

Index	Data Set	$\mathbf{L}^{r*}$_Eqw	$\mathbf{L}^{r*}$_Damp	Data Set	$\mathbf{L}^{r*}$_Eqw	$\mathbf{L}^{r*}$_Damp
Accuracy	**CRC**	0.6163793	**0.6400862**	**LGG**	0.9625468	**0.9737828**
NMI		0.0084103	**0.0185660**		0.8509861	**0.8689965**
ARI		0.0317469	**0.0548748**		0.8806075	**0.9199392**
F-measure		0.6309431	**0.6529565**		0.9625844	**0.9737835**
Rand		0.5260669	**0.5382531**		0.9437921	**0.9622089**
Purity		**0.7370690**	**0.7370690**		0.9625468	**0.9737828**
Accuracy	**STAD**	**0.7727273**	0.768595	**BRCA**	0.6733668	**0.7613065**
NMI		**0.5150229**	0.510726		0.4531777	**0.5281849**
ARI		**0.4639222**	0.4559866		0.3964856	**0.4874579**
F-measure		**0.7788198**	0.7778227		0.6834253	**0.7660191**
Rand		**0.7703782**	0.7661946		0.7523132	**0.7922357**
Purity		**0.7727273**	0.7685950		0.6783920	**0.7613065**

Table 5.5: Comparative Performance Analysis of Equally and Damped Weighted Combination on Benchmark Data Sets

Index	Data Set	$\mathbf{L}^{r*}$_Eqw	$\mathbf{L}^{r*}$_Damp	Data Set	$\mathbf{L}^{r*}$_Eqw	$\mathbf{L}^{r*}$_Damp
Accuracy	**Football**	**0.8758064**	0.8500000	**Politics-UK**	**0.9785203**	0.9665871
NMI		**0.8908789**	0.8625365		0.9345135	**0.9434825**
ARI		**0.7841728**	0.7278994		**0.9637864**	0.9633130
F-measure		**0.8848290**	0.8683491		0.9735519	**0.9736129**
Rand		**0.9760741**	0.9739682		**0.9828368**	0.9826084
Purity		**0.8778225**	0.8584677		**0.9785203**	**0.9785203**
Accuracy	**Rugby**	0.8196721	**0.8305621**	**Digits**	0.8170000	**0.8835000**
NMI		0.7038184	**0.7093834**		0.8288566	**0.7981981**
ARI		0.6454866	**0.6627701**		0.7616410	**0.7645096**
F-measure		0.8288040	**0.8349647**		0.8746977	**0.8839913**
Rand		0.8972515	**0.9067597**		0.9564677	**0.9576618**
Purity		**0.8621780**	0.8606557		0.8310000	**0.8835000**

The quantitative results for this comparison are reported in Table 5.4, which show that for CRC, LGG, and BRCA data sets, the damping strategy $\mathbf{L}^{r*}$_Damp performs better than $\mathbf{L}^{r*}$_Eqw, in terms of all external indices. Only for the STAD data set, weighting all the modalities equally gives slightly better performance. This is also evident from the increased inter-cluster separability in Figure 5.5(c) compared to Figure 5.5(h). However, the results in Table 5.4 show that assigning maximum weightage to the most relevant modality and gradually damping it by a factor β, based on its relevance, preserves better cluster information in majority of the cases.

5.5.4.2.2 Benchmark Data Sets The comparative results for the benchmark data sets are reported in Table 5.5. It can be observed from Table 5.4 that for a majority of omics data sets, damped weighting of modalities based on relevance outperforms the equally weighted one. On the contrary, the results in Table 5.5 shows that the equally weighted

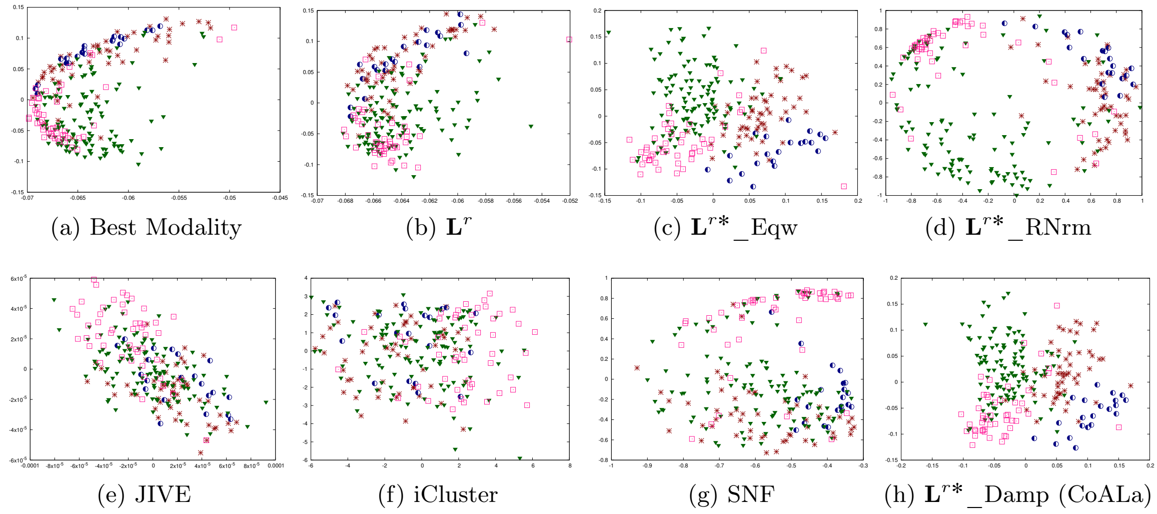
Table 5.6: Comparative Performance Analysis of Full-Rank and Approximate Subspaces of Omics Data

Index	Data Set	$\mathbf{L}^r$	CoALa ($\mathbf{L}^{r*}$)	Data Set	$\mathbf{L}^r$	CoALa ($\mathbf{L}^{r*}$)
Accuracy	**CRC**	0.5301724	**0.6400862**	**LGG**	0.6441948	**0.9737828**
NMI		0.0134459	**0.0185660**		0.3597365	**0.8689965**
ARI		-0.025277	**0.0548748**		0.2844081	**0.9199392**
F-measure		0.6052757	**0.6529565**		0.6577440	**0.9737835**
Rand		0.5007448	**0.5382531**		0.6524739	**0.9622089**
Purity		**0.7370690**	**0.7370690**		0.6441948	**0.9737828**
Accuracy	**STAD**	**0.5619835**	0.768595	**BRCA**	0.5477387	**0.7613065**
NMI		**0.2605140**	0.510726		0.4315712	**0.5281849**
ARI		**0.2248704**	0.4559866		0.3507615	**0.4874579**
F-measure		0.6158419	**0.7778227**		0.6197007	**0.7660191**
Rand		0.6706560	**0.7661946**		0.7403390	**0.7922357**
Purity		0.6157025	**0.768595**		0.7185930	**0.7613065**

strategy gives better performance than the damped one on the Football and Politics-UK data sets. One possible explanation is that most of the component modalities of the Twitter data sets are similar to each other and have close performances. For instance, 'follows' and 'followed-by', both are network based modalities where 'follows' captures the outgoing links from the nodes, while 'followed-by' captures the incoming links to the nodes. Other pairs of modalities like 'mentions' and 'mentioned-by', and 'retweets' and 'retweeted-by' are also very similar to each other. In the damped weighting introduced in Section 5.3.5, slight differences in the relevance values of these similar modalities would dampen the effect of the one with lower relevance by a factor of β. This leads to degraded cluster structure in eigenspace of the joint Laplacian for two Twitter data sets when using the damped weighted strategy. For Rugby and Digits data sets, damped weighted strategy $\mathbf{L}^{r*}$_Damp has better performance compared to equally weight $\mathbf{L}^{r*}$_Eqw one for majority of the external indices.

5.5.4.3 Importance of Noise-Free Approximation

The proposed eigenspace is an approximate one, as it is constructed from de-noised approximations of the individual eigenspaces. This approximate eigenspace is expected to preserve better cluster structure compared to the full-rank eigenspace constructed from the complete set of eigenpairs of the individual Laplacians. In order to establish this, the performance of clustering on the k largest eigenvectors of the full-rank eigenspace $\mathbf{L}^r$ is compared with that of the approximate eigenspace $\mathbf{L}^{r*}$ (CoALa) in Table 5.6. From the results of Table 5.6, it can be observed that the proposed CoALa algorithm outperforms the full-rank subspace $\mathbf{L}^r$ for all the data sets. The performance is significantly better for BRCA, LGG, and STAD data sets. The full-rank information of individual Laplacians in $\mathbf{L}^r$ inherently contains the noisy information of the $(n - r)$ smallest eigenvectors of each Laplacian. However, in the proposed algorithm, each individual Laplacian is truncated at rank r, to contain mostly the cluster discriminatory information, where $r << n$. So, the approximate eigenspace automatically eliminates the noise present in the $(n - r)$ remaining eigenvectors. The results of Table 5.6 show that this truncated de-noised Laplacians



(a) Best Modality (b) $\mathbf{L}^r$ (c) $\mathbf{L}^{r*}$_Eqw (d) $\mathbf{L}^{r*}$_RNrm

(e) JIVE (f) iCluster (g) SNF (h) $\mathbf{L}^{r*}$_Damp (CoALa)

Figure 5.5: Scatter plots using first two components of different low-rank based approaches on STAD data set.

preserve better cluster structure in the resulting eigenspace compared to the full-rank one. The scatter plots for the full-rank subspaces of LGG and STAD data sets are given in Figures 5.4(b) and 5.5(b), respectively. For LGG, Figures 5.4(b) shows that only one cluster is well-separated. On the other hand, data points from the other two clusters of LGG and all the four clusters of STAD in Figure 5.5(b) are cluttered amongst each other exhibiting poor separability. The optimal rank, $r^\star$, for LGG and STAD data sets are 48, and 39, respectively, while their full-ranks are 267 and 242, respectively. The scatter plots for rank $r^\star$ approximation in Figures 5.4(h) and 5.5(h) show that filtering out the noise in the remaining 219 and 203 eigen-pairs of the individual Laplacians preserves significantly better cluster structure for these data sets.

5.5.4.4 Advantage of Averting Row-normalization

In normalized spectral clustering (Algorithm 5.1), row-normalization tends to shift the objects in the projected subspace in such a way that they cluster tightly around an orthogonal basis. This is primarily justified when the objects lie close to the ideal case where the clusters are infinitely apart [162]. However, row-normalization may not necessarily give better performance on real-life data sets. The two-dimensional scatter plots for the row-normalized subspaces of LGG and STAD data sets are given in Figures 5.4(d) and 5.5(d), respectively. For both data sets, as expected, row-normalization pushes objects from different clusters further away from the origin in different directions of the subspace, which increases the inter-cluster separability. However, points lying in the boundaries of different clusters are not necessarily pushed away and are projected around the origin of the subspaces, which in turn reduces the compactness of the clusters. When the number of boundary points is relatively large, row-normalization tends to give degraded performance. To study this quantitatively, the clustering performance of the row-normalized subspace (termed as $\mathbf{L}^{r*}$_RNrm) is compared with that of not normalized one in Table 5.7. The re-

Table 5.7: Effect of Row-normalization on Different Subspaces on Omics Data

Index	Data Set	$\mathbf{L}^{r}$*_RNrm	CoALa	Data Set	$\mathbf{L}^{r}$*_RNrm	CoALa
Accuracy	**CRC**	0.5991379	**0.6400862**	**LGG**	0.8951311	**0.9737828**
NMI		0.0056913	**0.0185660**		0.6857991	**0.8689965**
ARI		0.0220924	**0.0548748**		0.7359314	**0.9199392**
F-measure		0.6169586	**0.6529565**		0.9010565	**0.9737835**
Rand		0.5186192	**0.5382531**		0.8771367	**0.9622089**
Purity		**0.7370690**	**0.7370690**		0.8951311	**0.9737828**
Accuracy	**STAD**	0.7355372	**0.768595**	**BRCA**	0.6859296	**0.7613065**
NMI		0.4582469	**0.5107260**		0.4806899	**0.5281849**
ARI		0.4012421	**0.4559866**		0.4012943	**0.4874579**
F-measure		0.7389739	**0.7778227**		0.6946324	**0.7660191**
Rand		0.7474024	**0.7661946**		0.7588193	**0.7922357**
Purity		0.7355372	**0.7685950**		0.6859296	**0.7613065**

Table 5.8: Effect of Row-Normalization on Benchmark Data Sets

Index	Data Set	$\mathbf{L}^{r}$*_RNrm	CoALa	Data Set	$\mathbf{L}^{r}$*_RNrm	CoALa
Accuracy	**Football**	**0.8669354**	0.8500000	**Politics-UK**	0.9465394	**0.9665871**
NMI		**0.8856647**	0.8625365		0.8414139	**0.9434825**
ARI		**0.7775054**	0.7278994		0.9075232	**0.9633130**
F-measure		0.8679092	**0.8683491**		0.9571452	**0.9736129**
Rand		**0.9785490**	0.9739682		0.9746971	**0.9826084**
Purity		**0.8911290**	0.8584677		0.9715990	**0.9785203**
Accuracy	**Rugby**	0.6224824	**0.8305621**	**Digits**	0.8600000	**0.8835000**
NMI		0.6527087	**0.7093834**		**0.8528484**	0.7981981
ARI		0.3970370	**0.6627701**		**0.7970693**	0.7645096
F-measure		0.6737320	**0.8349647**		0.8629902	**0.8839913**
Rand		0.8606810	**0.9067597**		**0.9629410**	0.9576618
Purity		0.8545667	**0.8606557**		0.8565000	**0.8835000**

sults reported in Table 5.7 show that for all four data sets, the proposed subspace performs better than its row-normalized counterpart $\mathbf{L}^{r}$*_RNrm.

Table 5.8 compares the performance of the proposed approximate subspace with and without the row-normalization step for the benchmark data. Table 5.8 shows that for Politics-UK and Rugby data sets, avoiding row-normalization give better performance with respect to all the external indices. On the other hand, for Football and Digits data set, majority of the external indices gives better performance with row-normalization. Scatter plots for the first two dimensions of $\mathbf{L}^{r}$*_RNrm and the proposed CoALa algorithm are given in Figures 5.6 and 5.7 for the Politics-UK and the Digits data set, respectively.

5.5.5 Comparative Performance Analysis on Multi-Omics Data Sets

The performance of the proposed algorithm is compared with that of the existing ones, in Tables 5.9 and 5.10 in terms of the external cluster evaluation indices. The COCA and BCC algorithms are consensus clustering based approaches, while the other existing

Table 5.9: Comparative Performance Analysis of CoALa and Existing Approaches Based on External Indices on Omics Data Sets

	Different Algorithms	Rank of Subspace	External Evaluation Index					
			Accuracy	NMI	ARI	F-Measure	RAND	Purity
CRC	COCA	-	0.5323276	0.0120929	0.0007663	0.5586055	0.5010706	**0.7370690**
	BCC	-	0.5745690	0.0070894	0.0074889	0.5973067	0.5158300	**0.7370690**
	JIVE(Perm)	16	0.6034483	0.0071359	0.0256478	0.6210774	0.5203694	**0.7370690**
	A-JIVE	32	0.6034483	0.0064720	0.0246106	0.6206032	0.5203694	**0.7370690**
	iCluster	1	0.6163793	0.0069992	0.0293081	0.6298050	0.5260669	**0.7370690**
	LRAcluster	1	0.5129310	0.0030437	-0.001822	0.5410661	0.4992552	**0.7370690**
	PCA-Con	2	0.5366379	0.0057828	0.0036971	0.5641984	0.5016106	**0.7370690**
	SNF	2	0.5991379	0.0069730	0.0240692	0.6178576	0.5186192	**0.7370690**
	NormS	16	0.6206897	0.0093881	0.0347351	0.6345375	0.5281150	**0.7370690**
	SURE	2	0.5107759	0.0027977	-0.002148	0.5416716	0.4991528	**0.7370690**
	CoALa	2	**0.6400862**	**0.0185660**	**0.0548748**	**0.6529565**	**0.5382531**	**0.7370690**
LGG	COCA	-	0.6591760	0.2772248	0.2533847	0.6608123	0.6454901	0.6591760
	BCC	-	0.6340824	0.2737596	0.248606	0.63111660	0.6382755	0.6355805
	JIVE	8	0.5617978	0.2299551	0.1606599	0.5757978	0.6056715	0.5730337
	A-JIVE	48	0.7168539	0.4267241	0.3376560	0.7172792	0.6869055	0.7168539
	iCluster	2	0.4382022	0.1379678	0.0996867	0.5187438	0.5821858	0.5355805
	LRAcluster	2	0.4719101	0.1240057	0.1030798	0.5137382	0.5831714	0.5280899
	PCA-con	3	0.6666667	0.3438738	0.3031312	0.6574834	0.6616823	0.6666667
	SNF	3	0.8689139	0.6253254	0.6331662	0.8720595	0.8268142	0.8689139
	NormS	14	0.7940075	0.5325030	0.4649223	0.7916535	0.7465292	0.7940075
	SURE	3	0.7940075	0.5335888	0.4668931	0.7904750	0.7465292	0.7940075
	CoALa	4	**0.9737828**	**0.8689965**	**0.9199392**	**0.9737835**	**0.9622089**	**0.9737828**
STAD	COCA	-	0.4450413	0.1309746	0.0740987	0.4558087	0.5981242	0.5173554
	BCC	-	0.5392562	0.1500351	0.1421471	0.5520075	0.6081204	0.5673554
	JIVE(Perm)	8	0.4049587	0.1288122	0.0657955	0.4487487	0.5981619	0.5165289
	A-JIVE	64	0.4148760	0.1234864	0.0763413	0.4458621	0.6086142	0.5227273
	iCluster	3	0.3512397	0.0650589	0.0288255	0.3832114	0.5855423	0.4917355
	LRAcluster	1	0.4256198	0.1259879	0.0912460	0.4746753	0.6122218	0.5619835
	PCA-Con	2	0.6900826	0.3654109	0.3204142	0.6959782	0.7110524	0.6900826
	SNF	2	0.5661157	0.3216270	0.2694201	0.6333622	0.6945235	0.6363636
	NormS	27	0.5702479	0.1805281	0.1625013	0.5770884	0.6435993	0.5950413
	SURE	2	0.6983471	0.3511439	0.3445607	0.7056674	0.7216145	0.6983471
	CoALa	4	**0.7685950**	**0.5107260**	**0.4559866**	**0.7778227**	**0.7661946**	**0.768595**
BRCA	COCA	-	0.7434673	0.5002408	0.4864778	0.7457304	0.7905295	0.7434673
	BCC	-	0.6251256	0.3169187	0.3049874	0.6242493	0.7055783	0.6334171
	JIVE	12	0.6859296	0.4287142	0.3772649	0.6889363	0.7464906	0.6859296
	A-JIVE	64	0.6140704	0.4482479	0.3710317	0.6707575	0.7363682	0.6841709
	iCluster	3	0.7638191	0.5176193	0.4745746	0.7658865	0.7842867	0.7638191
	LRAcluster	2	0.7110553	0.4368520	0.4035040	0.7101385	0.7521740	0.7110553
	PCA-con	4	0.7587940	**0.5506612**	0.5038795	0.7601317	0.7984380	0.7587940
	SNF	4	0.6783920	0.4558955	0.4111794	0.6865447	0.7602370	0.6959799
	NormS	11	**0.7688442**	0.5437267	**0.5090183**	**0.7699789**	**0.7999063**	**0.7688442**
	SURE	4	0.7663317	0.5528011	0.5104814	0.7683344	0.8010455	0.7663317
	CoALa	4	0.7613065	0.5281849	0.4874579	0.7660191	0.7922357	0.7613065

Table 5.10: Comparative Performance Analysis of CoALa and Existing Approaches Based on External Indices on Omics Data Sets

	Different Algorithms	Rank of Subspace	External Evaluation Index					
			Accuracy	NMI	ARI	F-Measure	RAND	Purity
KIDNEY	COCA	-	0.9408280	0.7493140	0.8393954	0.9477422	0.9199568	0.9470828
	BCC	-	0.9122117	0.6783448	0.7299573	0.9139998	0.8657292	0.9122117
	JIVE(Perm)	12	0.9308005	0.6955325	0.7786981	0.9300085	0.8893944	0.9308005
	A-JIVE	48	0.9582090	0.7902576	0.8695284	0.9585611	0.9349404	0.9582090
	iCluster	2	0.6065129	0.2547010	0.1717458	0.6514716	0.5842023	0.6811398
	LRAcluster	2	**0.9538670**	**0.7862018**	**0.8579391**	**0.9545717**	**0.9292298**	**0.9538670**
	PCA-Con	3	0.9511533	0.7670505	0.8489024	0.9516854	0.9246800	0.9511533
	SNF	3	0.9579376	0.7946083	0.8796762	0.9590236	0.9400330	0.9579376
	NormS	35	0.9525102	0.7726162	0.8534490	0.9530685	0.9269512	0.9525102
	SURE	3	0.9525102	0.7726162	0.8534490	0.9530685	0.9269512	0.9525102
	CoALa	3	0.9294437	0.6987468	0.7786424	0.9285111	0.8893207	0.9294437
OV	COCA	-	0.5943114	0.3131466	0.2810761	0.6068513	0.7039183	0.5943114
	BCC	-	0.4610778	0.1567582	0.1254690	0.4755846	0.6268706	0.4622754
	JIVE	32	0.5718563	0.2629523	0.2027605	0.5653910	0.6885005	0.5718563
	A-JIVE	64	0.5191617	0.2124862	0.1981556	0.5111353	0.6942997	0.5221557
	iCluster	3	0.5089820	0.2249889	0.2005886	0.4808256	0.6916078	0.5119760
	LRAcluster	2	0.6287425	0.3745173	0.2999204	0.6384046	0.7322472	0.6287425
	PCA-con	4	0.6946108	0.4424701	0.4068449	0.6868295	0.7734621	0.6946108
	SNF	4	0.5269461	0.2753886	0.2058407	0.5642052	0.6557695	0.5389222
	NormS	10	0.6976048	0.4504552	0.4142200	0.6910392	0.7766269	0.6976048
	SURE	4	**0.7215569**	**0.4680312**	**0.4372574**	**0.7148805**	**0.7857258**	**0.7215569**
	CoALa	4	0.6736527	0.3381426	0.3199015	0.6700606	0.7379295	0.6736527
LUNG	COCA	-	0.9284650	0.6287671	0.7339231	0.9283705	0.8669662	0.9284650
	BCC	-	0.9372578	0.6648076	0.7645295	0.9371445	0.8822697	0.9372578
	JIVE(Perm)	8	0.9269747	0.6333526	0.7288041	0.9266709	0.8644127	0.9269747
	A-JIVE	32	0.9478390	0.7192028	0.8019299	0.9476450	0.9009720	0.9478390
	iCluster	1	0.6333830	0.0627751	0.0696293	0.6299231	0.5348889	0.6333830
	LRAcluster	1	0.9344262	0.6535038	0.7545277	0.9342966	0.8772694	0.9344262
	PCA-Con	2	0.9388972	0.6773549	0.7701654	0.9386955	0.8850902	0.9388972
	SNF	2	0.9493294	0.7152672	0.8072916	0.9492292	0.9036502	0.9493294
	NormS	27	0.9359165	0.6650183	0.7597192	0.9357050	0.8798674	0.9359165
	SURE	2	**0.9418778**	**0.6878184**	**0.7806842**	**0.9417093**	**0.8903486**	**0.9418778**
	CoALa	2	0.9403875	0.6970004	0.7754083	0.9400693	0.8877149	0.9403875
CESC	COCA	-	0.6693548	0.4172592	0.3677157	0.6870510	0.6971282	0.6774194
	BCC	-	0.6895161	0.2854917	0.3144526	0.6795619	0.6687779	0.6935484
	JIVE(Perm)	24	0.7177419	0.4425848	0.3860367	0.7097880	0.7164962	0.7177419
	A-JIVE	48	0.6500000	0.3700238	0.3355826	0.6511586	0.6857724	0.6814516
	iCluster	2	0.5483871	0.1737526	0.1017765	0.5568753	0.5731707	0.5645161
	LRAcluster	1	0.8145161	0.5176602	0.5384740	0.8123256	0.7867821	0.8145161
	PCA-con	3	0.8548387	0.6750978	0.6333073	0.8390298	0.8237608	0.8548387
	SNF	3	0.6693548	0.4927941	0.4239905	0.7073802	0.7043011	0.6935484
	NormS	6	**0.8870968**	**0.6854921**	**0.7004411**	**0.8801172**	**0.8587726**	**0.8870968**
	SURE	3	0.8629032	0.6461946	0.6507274	0.8512028	0.8339890	0.8629032
	CoALa	3	0.8225806	0.5479227	0.5637070	0.8139970	0.7951744	0.8225806

algorithms are subspace based approaches for which the optimal rank of the clustering subspace is reported in Tables 5.9 and 5.10. The optimal ranks are selected using the selection criteria suggested by the authors for the respective approaches. The results in Table 5.9 and 5.10 show that the proposed algorithm performs better than all the existing approaches for CRC, LGG, and STAD data sets in terms of the external indices, except for the purity measure on the CRC data set. However, F-measure and other external indices indicate that the proposed algorithm identifies the smaller sized cluster better than the existing ones. For the rest of the omics data sets, the algorithms proposed in the two previous chapters, namely, NormS (Chapter 3) and SURE (Chapter 4) are among the best performing algorithms in terms of external indices, while the proposed algorithm achieves third or fourth best performance. The iCluster algorithm has comparable performance for BRCA and CRC data sets, however, its degraded performance in the remaining data sets is due to the poor selection of its optimal lasso penalty parameter from the high-dimensional parameter space.

Due to the heterogeneous nature of the individual modalities, LRAcluster models each modality using a separate probability distribution having its own set of parameters. The proposed algorithm handles data heterogeneity by considering separate similarity matrices for separate modalities. Moreover, the modalities are integrated using their shifted Laplacians whose elements always lie in $[0, 2]$ as opposed to the raw data format. So, the difference in unit and scale of the individual modalities does not affect the final eigenspace. Similar to the proposed algorithm, the SNF approach also uses spectral clustering on a unified similarity graph to identify the clusters. However, in terms of the external indices, the proposed algorithm outperforms SNF on all data sets, except KIDNEY and LUNG data sets. In SNF, the unified graph is iteratively made similar to the individual graphs. This can often lead to propagation of unwanted information from noisy graphs into the final unified one. On the other hand, the proposed algorithm amplifies the effect of the most relevant graph, as well as dampens the effect of the irrelevant ones in the convex combination. Moreover, truncation of individual Laplacians at rank $r << n$ helps in propagating mostly cluster discriminatory information into the final subspace and automatically filters out the noise. These two aspects of the proposed CoALa algorithm are primarily responsible for its significantly better performance, especially for the LGG and STAD data sets.

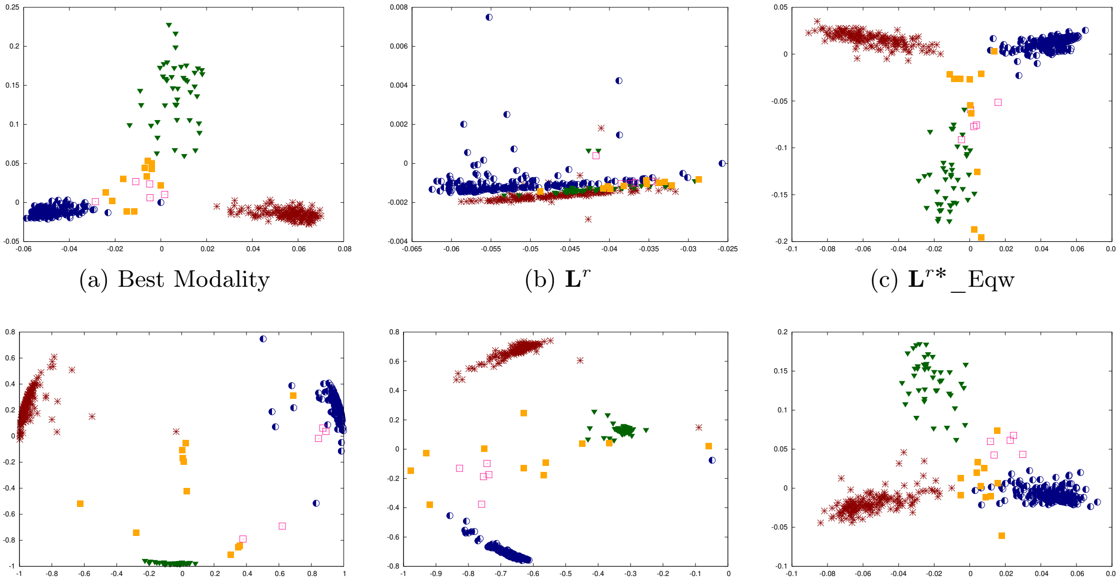
Different low-rank based approaches extract subspaces of different ranks. Tables 5.9 and 5.10 show that the ranks vary from 1 to as high as 64. The comparison of cluster compactness and separability in these subspaces of varying dimensions is not reasonable. So, the goodness of clustering is evaluated using internal cluster validity indices by performing k-means clustering on the first two dimensions of each subspace. This makes the internal evaluation results comparable and also easy to visualize. Four internal cluster evaluation measures, namely, Silhouette and Dunn, which are maximization based indices, and Davies-Bouldin (DB) and Xie-Beni, which are minimization based, are used. The internal cluster evaluation results are reported in Table 5.11 for four omics data sets, namely, CRC, LGG, STAD, and BRCA, as examples. The results show that the proposed algorithm has best performance for Silhouette, DB, and Xie-Beni indices for LGG data set and the second best for Silhouette and Dunn indices for BRCA data set. The SNF has best performance for two or more internal indices for CRC, STAD, and, BRCA data sets. This implies that on these three data sets, the cluster structure reflected in the first two dimensions of SNF more are compact and well-separated compared to the proposed and

Table 5.11: Comparative Performance Analysis of CoALa and Existing Approaches Based Internal Indices and Execution Time on Omics Data Sets

	Different Algorithms	Internal Evaluation Index				Time (in sec)
		Silhouette	Dunn	DB	Xie-Beni	
CRC	JIVE(Perm)	0.4199826	0.0120740	0.8821177	348.50660	3098.75
	A-JIVE	0.5016133	0.0043986	0.6872426	2314.1590	946.18
	iCluster	0.6229586	**0.3317529**	0.5770987	**0.3629792**	337.51
	LRAcluster	0.4337712	0.0160840	0.8751325	202.80470	104.12
	PCA-Con	0.3417350	0.0190144	1.1650270	155.15390	2.62
	SNF	**0.7834208**	0.0549104	**0.2980235**	17.069770	9.66
	NormS	0.3640602	0.0185685	1.0995640	116.69010	**1.45**
	SURE	0.3732788	0.0050670	1.0689000	1622.504	5.41
	CoALa	0.3483722	0.0179209	1.1021510	115.98920	32.77
LGG	JIVE(Perm)	0.4138221	**0.0355064**	0.8684623	51.054660	665.82
	A-JIVE	0.3375023	0.0241153	0.9444459	87.842080	364.43
	iCluster	0.3952103	0.0252834	0.9330074	93.144060	3230.52
	LRAcluster	0.3921144	0.0344110	0.8593495	43.233820	37.71
	PCA-Con	0.4624043	0.0322859	0.7439401	58.96720	1.08
	SNF	0.4441981	0.0149314	0.7388554	318.54730	1.33
	NormS	0.4305583	0.0218683	0.8441603	175.06670	**0.96**
	SURE	0.3709216	0.0378629	1.0097820	59.552690	3.71
	CoALa	**0.6273401**	0.0287595	**0.4905286**	**12.563470**	17.02
STAD	JIVE(Perm)	0.3618677	0.0257650	0.9526717	84.992880	734.70
	A-JIVE	0.3365825	0.0203049	0.9617136	101.30600	302.98
	iCluster	0.3790058	0.0357959	0.9584001	54.286930	1138.88
	LRAcluster	0.4015128	0.0304117	0.7928001	40.097030	49.36
	PCA-Con	0.3862858	0.0182291	0.8355266	227.84070	1.02
	SNF	**0.4477905**	**0.0596324**	**0.7872797**	**19.297210**	1.14
	NormS	0.3395181	0.0181344	0.9157146	181.9440	**0.80**
	SURE	0.2679056	0.0613939	1.2882420	17.74966	3.67
	CoALa	0.4102003	0.0325467	0.8490579	58.722830	13.79
BRCA	JIVE(Perm)	0.4429883	0.0134063	0.7463430	277.15980	866.00
	A-JIVE	0.3148863	0.0142913	0.9765342	187.56970	686.85
	iCluster	0.4400869	0.0258263	0.7819524	77.708790	511.87
	LRAcluster	0.4300455	**0.0369472**	0.8211325	**43.223840**	88.32
	PCA-Con	0.4232505	0.0241363	0.8269517	86.81890	**0.93**
	SNF	**0.5005988**	0.0189055	**0.6814998**	112.0742	1.91
	NormS	0.4218991	0.0090550	0.8069696	504.69590	1.47
	SURE	0.3408657	0.0544491	1.0923470	19.350910	6.32
	CoALa	0.4478377	0.0253506	0.7873740	81.048340	14.36

other existing algorithms. The scatter plots for the first two dimensions of some low-rank based approaches are given in Figures 5.4(e)-(g) and in Figures 5.5(e)-(g), respectively, for LGG and STAD data sets. The data points are labeled in different colors based on the previously established TCGA subtypes. Although SNF has the best performance for all the internal indices for STAD data set, the scatter plot of SNF for LGG, in Figures 5.4(g), shows that the compact and well-separated clusters do not necessarily conform with the clinically established TCGA labellings. In brief, out of 16 cases of internal evaluation, reported in Table 5.11, the proposed CoALa algorithm ranks among the top three in 8 cases.

The execution times reported in Table 5.11 show that the proposed CoALa algorithm is computationally much faster than the consensus based COCA approach and other low-rank approaches like LRAcluster, JIVE, A-JIVE, and iCluster. However, PCA-con, SNF, NormS, and SURE have lower execution time compared to the proposed algorithm across all the data sets. For model fitting, iCluster uses expectation maximization algorithm, while JIVE uses alternate optimization. These iterative algorithms have slow convergence on the high-dimensional multimodal data sets. This leads to huge execution time and poor scalability of these algorithms as seen in Table 5.11. PCA-con achieves the lowest execution time on CRC and STAD data sets, as it performs SVD on the concatenated data only once. On the other hand, NormS achieves the same on LGG and STAD data sets. NormS achieves this computational advantage by simply concatenating relevant principal components from different modalities, at the cost of constructing a relatively much higher dimensional subspace. However, the external evaluation indices show that such naive concatenation in PCA-con and NormS often fails to capture the true cluster structure of the multimodal data.



(a) Best Modality

(b) $\mathbf{L}^r$

(c) $\mathbf{L}^{r*}$_Eqw

(d) $\mathbf{L}^{r*}$_RNrm

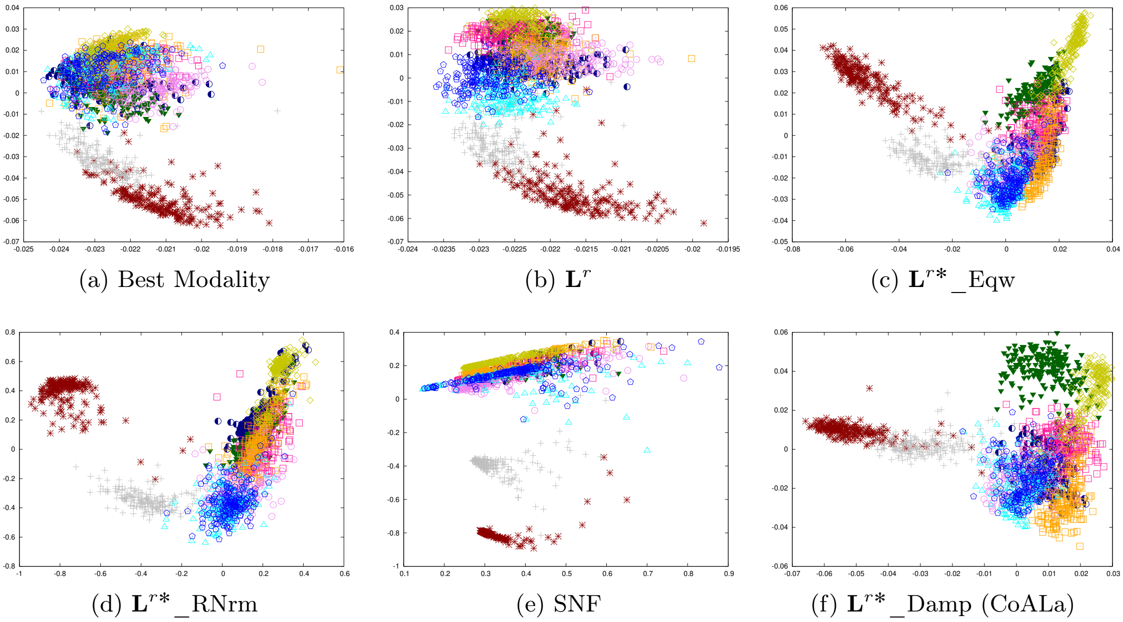
(e) SNF

(f) $\mathbf{L}^{r*}$_Damp (CoALa)

Figure 5.6: Scatter plots using first two components of different low-rank based subspaces for Politics-UK data set.

5.5.6 Comparative Performance Analysis on Benchmark Data Sets

Finally, the performance of different algorithms is studied on seven benchmark multimodal data sets, namely, Football, Politics-UK, Rugby, Digits, ORL, Caltech7, and CORA. Among them, Football, Politics-UK, Rugby, and CORA are social network data sets, while



(a) Best Modality

(b) $\mathbf{L}^r$

(c) $\mathbf{L}^{r*}$_Eqw

(d) $\mathbf{L}^{r*}$_RNrm

(e) SNF

(f) $\mathbf{L}^{r*}$_Damp (CoALa)

Figure 5.7: Scatter plots using first two components of different low-rank based subspaces for Digits data set.

Digits, ORL, and Caltech7 are general image data sets. For the social network data sets, most of the component modalities have graph based representation. However, apart from SNF, all other existing algorithms require feature based representations of the component modalities, so their performance could not be evaluated on social network data sets. The comparative performance of the best modality (in terms of external indices), the full-rank subspace $\mathbf{L}^r$, SNF, and the proposed CoALa algorithm are reported in Tables 5.12 and 5.13 for different data sets. The scatter plots for the first two dimensions of the corresponding subspaces are given in Figures 5.6 and 5.7 respectively, for Politics-UK and Digits data sets. The convex combination $\boldsymbol{\alpha}$ and the optimal rank $r^\star$ are assigned as described previously in Sections 5.3.5 and 5.5.2, respectively.

The comparative results of Tables 5.12 and 5.13 show that the proposed algorithm has the best performance in terms of majority of the external indices for all four social network data sets, namely, Football, Politics-UK, Rugby, and CORA, and two image data sets, namely, ORL and Caltech7. The SNF algorithm has the second best performance on the three Twitter data sets and the best modality always outperforms the full-rank subspace $\mathbf{L}^r$. For the Digits data set, SNF outperforms the proposed algorithm in four external indices. The proposed algorithm has the second best performance and is followed by the full-rank subspace $\mathbf{L}^r$. The Football data set has been recently been used for the performance evaluation of latent multi-view subspace clustering (LMSC) [275] algorithm. LMSC has two formulations, namely, linear (lLMSC) and generalized (gLMSC). For the Football data set, the aggregate F-measure values for lLMSC and gLMSC are 0.7082 and 0.7940, respectively, while aggregate Rand index are 0.9714 and 0.9797, respectively, while F-measure and Rand index for CoALa are 0.8852 and 0.9780, respectively, which show that CoALa outperforms both lLMSC and gLMSC in terms of F-measure. In terms of Rand

Table 5.12: Comparative Performance Analysis on Benchmark Data Sets: Football, Politics-UK, Rugby, Digits

	Measure	Data Set	Best View	$\mathbf{L}^r$	SNF	CoALa
	Subspace Rank	**Football** [n = 248; k = 20; M = 9]	20	20	20	20
External	Accuracy		0.7757224	0.6564516	0.8145161	**0.8500000**
	NMI		0.7910368	0.7748572	**0.8829152**	0.8625365
	ARI		0.5814725	0.3777853	**0.7458860**	0.7278994
	F-measure		0.7747023	0.6616297	0.8431825	**0.8683491**
	Rand		0.9472965	0.8843737	0.9735862	**0.9739682**
	Purity		0.7282258	0.6572580	0.8266129	**0.8584677**
Internal	Silhouette		**0.5565601**	0.4392812	0.4750064	0.5170209
	Dunn		0.0122200	0.0304905	0.0496361	**0.0506094**
	DB		**0.4087806**	0.5388078	0.6463104	0.5318746
	Xie-Beni		181.35320	36.629720	16.878340	**15.47080**
	Time (in sec)		**0.68**	1.13	1.05	1.34
	Subspace Rank	**Politics-UK** [n = 419; k = 5; M = 9]	5	5	5	5
External	Accuracy		0.8902148	0.7591885	**0.9737470**	0.9665871
	NMI		0.8382287	0.6777684	0.9194125	**0.9434825**
	ARI		0.8375676	0.7205330	0.9608391	**0.9633130**
	F-measure		0.9175316	0.8192186	0.9701235	**0.9736129**
	Rand		0.9196880	0.8603076	0.9814665	**0.9826084**
	Purity		0.9713604	0.8591885	0.9761337	**0.9785203**
Internal	Silhouette		**0.7877163**	0.5531584	0.7599383	0.6165161
	Dunn		0.0691656	0.0082616	0.0121941	**0.0216676**
	DB		0.5042173	**0.4124179**	0.4971371	0.6299340
	Xie-Beni		**4.1253610**	544.13860	66.892230	68.551380
	Time (in sec)		**0.95**	1.86	3.83	3.68
	Subspace Rank	**Rugby** [n = 854; k = 15; M = 9]	15	15	15	15
External	Accuracy		0.7121779	0.7283372	0.7611241	**0.8305621**
	NMI		0.6151681	0.6526318	0.6768068	**0.7093834**
	ARI		0.5461666	0.6416748	0.5485665	**0.6627701**
	F-measure		0.7426962	0.6845209	0.7778990	**0.8349647**
	Rand		0.8672685	0.8578210	0.8818113	**0.9067597**
	Purity		0.7796253	0.6803279	0.8454333	**0.8606557**
Internal	Silhouette		**0.5444214**	0.5195532	0.4713082	0.4123312
	Dunn		0.0012972	0.0085216	0.0051843	**0.0086649**
	DB		**0.4727219**	0.4603219	0.5856659	0.7474256
	Xie-Beni		780.66640	**212.29020**	827.27610	328.6280
	Time (in sec)		**4.77**	7.21	22.94	27.42
	Subspace Rank	**Digits** [n = 2000; k = 10; M = 6]	10	10	10	10
External	Accuracy		0.7096000	0.8500000	**0.8835000**	**0.8835000**
	NMI		0.6443707	0.7951372	**0.8904675**	0.7981981
	ARI		0.5416071	0.7267166	**0.8435742**	0.7645096
	F-measure		0.7209662	0.8481826	**0.8932872**	0.8839913
	Rand		0.9173923	0.9503602	**0.9715983**	0.9576618
	Purity		0.7100000	0.8500000	**0.8835000**	**0.8835000**
Internal	Silhouette		0.4860050	**0.5265748**	0.4452352	0.4269673
	Dunn		0.0050409	0.0064673	0.0031041	**0.0071841**
	DB		0.5722576	**0.5331665**	0.8063785	0.7470644
	Xie-Beni		1275.5800	830.76950	1166.0330	**659.67560**
	Time (in sec)		**80.71**	135.65	189.03	154.57

Table 5.13: Comparative Performance Analysis on Benchmark Data Sets: ORL, Caltech7, CORA

	Measure	Data Set	Best View	$\mathbf{L}^r$	SNF	CoALa
	Subspace Rank	**ORL** [$n = 400$; $k = 40$; $M = 3$]	40	40	40	40
External	Accuracy		0.7167500	0.7155000	0.6907500	**0.7715000**
	NMI		0.8800011	0.8804695	0.8616789	**0.8980924**
	ARI		0.6232466	0.6225230	0.6054544	**0.6932679**
	F-measure		0.7643606	0.7609769	0.7257119	**0.7962088**
	Rand		0.9807581	0.9803784	0.9804474	**0.9850088**
	Purity		0.7740000	0.7722500	0.7450000	**0.8090000**
Internal	Silhouette		0.3613618	0.3631277	**0.5106249**	0.2952385
	Dunn		0.1980555	0.1887661	0.0642730	**0.2695878**
	DB		1.2051272	1.2094668	**1.0163360**	1.4134120
	Xie-Beni		2.4075381	2.7465589	102.50140	**1.978858**
	Time (in sec)		**12.93**	62.32	16.36	13.54
	Subspace Rank	**Caltech7** [$n = 1474$; $k = 7$; $M = 6$]	7	7	7	7
External	Accuracy		0.5468114	0.4789688	0.5440299	**0.5685210**
	NMI		0.3222844	0.4730254	**0.5676032**	0.5650165
	ARI		0.3202251	0.3476278	0.4126422	**0.4397484**
	F-measure		0.6273082	0.6023481	0.6363390	**0.6689529**
	Rand		0.7038184	0.7242808	0.7482112	**0.7583422**
	Purity		0.7761194	0.8127544	0.8516282	**0.8548168**
Internal	Silhouette		0.2790141	0.325057	**0.5682432**	0.3631100
	Dunn		0.0207478	**0.040731**	0.0357273	0.0361922
	DB		1.1931470	1.082995	**0.8238205**	0.9804257
	Xie-Beni		77.491480	**27.67097**	83.675470	44.50850
	Time (in sec)		**12.18**	20.85	26.76	21.64
	Subspace Rank	**CORA** [$n = 2708$; $k = 7$; $M = 2$]	7	7	7	7
External	Accuracy		0.4090472	0.4237444	0.5450517	**0.5896233**
	NMI		0.2276971	0.2739455	0.3829834	**0.4364573**
	ARI		0.0676929	0.1337145	0.2941402	**0.3256322**
	F-measure		0.3924244	0.4335790	**0.5957978**	0.5844190
	Rand		0.4164793	0.5395755	**0.7936103**	0.7460456
	Purity		0.4333456	0.4362629	0.6012186	**0.6206425**
Internal	Silhouette		**0.6649553**	0.3931360	0.3874047	0.3951749
	Dunn		0.0006289	0.0027002	**0.0371434**	0.0111024
	DB		0.8543087	**0.7333716**	0.9988864	0.8347002
	Xie-Beni		172.76125	77.745998	**74.81975**	116.997500
	Time (in sec)		27.97	**7.12**	14.36	11.49

index, performance of LMSC and CoALa are competitive. Also, the Digits data set has been used for the evaluation of multiple kernel learning based late fusion incomplete multi-view clustering (LF-IMVC) [138] algorithm and spectral clustering based Wang *et. al*'s algorithm [234]. The aggregate purity and normalized mutual information (NMI) values for Digits data set for LF-IMVC are 0.7980 and 0.6899, respectively, while for Wang *et. al*'s algorithm NMI achieved is 0.785. For CoALa, aggregate purity and NMI obtained are 0.8835 and 0.797659, respectively. The results imply that CoALa outperforms both these algorithms on Digits data set.

In terms of internal cluster evaluation indices, the results in Tables 5.12 and 5.13 show that out of 28 cases, the proposed algorithm achieves best performance in nine cases, while the second best in ten cases. For the Twitter data sets, the best modality achieves superior performance for majority of the internal indices. The execution times reported in Tables 5.12 and 5.13 indicate that the proposed method is computationally more efficient compared to SNF for five out of seven data sets. Although for the omics data sets in Table 5.11, SNF needs lower execution time compared to CoALa, CoALa demonstrates higher computational efficiency compared to SNF for the benchmark data sets with larger number of samples or component modalities.

5.6 Conclusion

This chapter presents a novel algorithm, for the integration of multiple similarity graphs, that prevents the noise of the individual graphs from being propagated into the unified one. The proposed method first approximates each graph using the most informative eigenpairs of its Laplacian which contains its cluster information. Thus, the noise in the individual graphs is not reflected in their approximations. These de-noised approximations are then integrated for the construction of a low-rank subspace that best preserves the overall cluster structure of multiple graphs. However, this approximate subspace differs from the full-rank one which integrates information of all the eigenpairs of each Laplacian. Using the concept of matrix perturbation, theoretical bounds are derived as a function of the approximation rank, inorder to precisely evaluate how far the approximate subspace deviates from the full-rank one. The clusters in the data set are identified by performing k-means clustering on the approximate de-noised subspace. The effectiveness of the proposed approximation based approach is established by showing that the approximate subspace encodes better cluster structure compared to the full-rank one. The clustering performance of the approximate subspace is compared with that of existing integrative clustering approaches on multiple real-life cancer data sets as well as on several benchmark data sets from varying application domains. Experimental results show that the clusters identified by the proposed approach have closest resemblance with the clinically established cancer subtypes and also with the ground-truth class information, when compared with individual modalities as well as existing algorithms.

The meaningful patterns embedded in high-dimensional multi-view data sets typically tend to have a much more compact representation that often lies close to a low-dimensional manifold. Identification of hidden structures in such data mainly depends on the proper modeling of the geometry of low-dimensional manifolds. In this regard, Chapter 6 presents a manifold optimization based integrative clustering algorithm for multi-view data. The optimization is performed alternatively over k-means and Stiefel manifolds. The Stiefel manifold helps to model the non-linearities and differential clusters within the individual views, while k-means manifold tries to elucidate the best-fit joint cluster structure of the data.

Chapter 6

Multi-Manifold Optimization for Multi-View Subspace Clustering

6.1 Introduction

Multi-view clustering explores the consistency and complementary properties of different views to improve clustering performance. It has been extensively used over the last decade [264,288,289] in various applications like face detection [241], action recognition [168], social networking [78,275], information retrieval [236], cancer biology [113,192,199], to name just a few. The observations in different views can convey similar or even differential information. In multi-view clustering paradigm, these views are expected to agree upon an underlying global cluster structure [141, 199]. Therefore, during data integration, it is essential to capture the inherent (dis)similarities in the individual views as well as elucidate the global cluster structure reflected across different views. Identification of cancer subtypes, from multiple omic data types like gene expression, DNA methylation, and protein expression, is one of the important application areas of multi-view clustering. The integrative multi-omic study can provide a comprehensive view of cancer mechanisms, and complement the diagnosis and therapeutic choices.

Different components of real-world systems, for instance, genes, micro RNAs, and other biomolecules in aggressive diseases like cancers, often share non-linear relationships [163]. These non-linearities tend to generate observations that lie on or close to a low-dimensional manifold. Identification of hidden structures and patterns in data crucially depends on modeling the geometry of the low-dimensional manifolds. Several popular machine learning approaches like principal component analysis, independent component analysis, and matrix approximation, can be given a geometric framework and modeled as an optimization problem whose natural domain is a manifold [3]. For example, the ubiquitous eigenvalue problem, imposed with norm equality constraints, results in a spherical search space which is an embedded submanifold of the original vector space. In manifold optimization framework, subspaces simply become single points on the manifold and search algorithms do not have to rely on the Euclidean vector space assumption of the search space. Two important submanifolds of the Euclidean space are Stiefel and k-means manifolds. While Stiefel manifold is used to model the geometry of an algorithm with orthonormality constraints [57],

k-means manifold generalizes spectral clustering over manifolds [28]. The current work judiciously integrates the merits of these two manifolds to develop a multi-view clustering algorithm.

Manifold optimization has been used in contemporary applications such as face recognition [180], computer vision [154], objection detection [30], and social networking [290], to identify non-linear patterns in data. Previous efforts have also resulted in tools that combine manifold learning and gene expression analysis to uncover non-linear structures among gene networks [163]. Ding *et al.* [53] identified breast cancer subtypes by merging linear subspaces on a Grassmanian manifold. A brief survey on manifold based multi-view clustering approaches is reported in Section 2.2.8 of Chapter 2. Nevertheless, discovering the structure of the manifold from a set of data points, sampled from the manifold possibly with noise, still remains a challenging problem. This is also unsupervised in nature. The problem gets aggravated in the presence of multiple views. Although different views of the same data set are expected to conform to the same underlying manifold structure, even subtle behavioral differences can give rise to different non-linear manifolds corresponding to different views. To identify meaningful clusters, it is not only essential to model the individual non-linearities, but also to identify the common structures conveyed by different manifolds.

In this regard, this chapter presents a novel manifold optimization algorithm, termed as MiMIC (Multi-Manifold Integrative Clustering), to perform multi-view data clustering. The proposed algorithm extracts a manifold representation for each view, which is intended to capture the individual non-linearities. It also constructs a joint graph Laplacian that contains the de-noised cluster information of the individual views. A joint optimization objective is proposed, comprising of a clustering component and a disagreement minimization component, to look into the consistent cluster information in the individual views with respect to the joint one. While the clustering component attempts to identify the joint cluster structure, the other component minimizes the disagreement between the manifold representation of the joint and individual views. The proposed joint objective is optimized over the k-means manifold for the clustering component, and Stiefel manifold for the disagreement component. During optimization, a gradient based movement is performed separately on the individual manifold corresponding to each view, so that the inherent individual non-linearity is preserved while looking for common cluster information. This multi-manifold approach is expected to model the individual differential cluster information, as well as infer the best-fit global cluster structure of the data set. The convergence analysis of the proposed algorithm is theoretically established. Asymptotic convergence bound theoretically quantify how fast the sequence of iterates generated by the proposed algorithm converges to an optimal solution, if exists. Moreover, the bound is used to make inference regarding the separability of clusters present in the data set. The efficacy of the proposed algorithm is studied and compared with that of existing approaches on several synthetic and benchmark multi-view data sets. The algorithm is also applied for cancer patient stratification using multi-omics data sets. Some of the results of this chapter are reported in [114].

The rest of the chapter is organized as follows: Section 6.2 outlines the basic principles of manifold based data clustering. Section 6.3 presents the proposed multi-view clustering algorithm based on alternating optimization over multiple non-linear manifolds. In Section 6.4 the asymptotic convergence bound of the proposed algorithm is derived in order to theoretically quantify how fast the algorithm converges to a local minima. Case studies on different multi-view benchmark data sets and multi-omics cancer data sets, along with a comparative performance analysis with existing approaches, are presented in Section 6.5. Concluding remarks are provided in Section 6.6.

6.2 Basics of Manifold Based Clustering

Clustering aims at partitioning a finite set of n samples $\{x_i\}_{i=1}^{n}$ into multiple subsets such that a dissimilarity based cost is minimized. Assuming that the samples lie in the d-dimensional Euclidean space and the number of subsets is k, the cost function for the k-means clustering problem is the sum of squared distance of each sample from the centroid of the cluster it is assigned to. An equivalant formulation of the k-means objective is given by [28]

$$\begin{aligned} &\min_{U \in \Re^{n \times k}} - tr(U^T W U) \\ &\text{such that} \;\; U \geqslant 0; \; U^T U = \mathbf{I}_k; \; U U^T \mathbf{1} = \mathbf{1}, \end{aligned} \tag{6.1}$$

where tr denotes the trace of a matrix, U^T denotes the transpose of matrix U, $\mathbf{I}_k$ denotes the identity matrix of order k, $\mathbf{1}$ denotes a column vector of all ones, U denotes the real-valued relaxation of the discrete cluster indicator matrix, and $W = [w(i,j)]$ is an $(n \times n)$ symmetric positive semi-definite Gram matrix or “affinity" matrix. The affinity matrix W can be replaced by the normalized affinity matrix $D^{-1/2} W D^{-1/2}$ [162], where D is the degree matrix given by $D = diag(\bar{d}_1, \ldots, \bar{d}_i, \ldots, \bar{d}_n)$ with $\bar{d}_i = \sum_{j=1}^{n} w(i,j)$. Replacing the affinity matrix W in (6.1) by the normalized affinity $D^{-1/2} W D^{-1/2}$ and adding the constant identity matrix $\mathbf{I}_n$, the objective of (6.1) becomes the minimization of $tr\left(U^T(\mathbf{I}_n - D^{-1/2} W D^{-1/2})U\right)$. The matrix $\mathcal{L} = (\mathbf{I}_n - D^{-1/2} W D^{-1/2})$ is known as the normalized graph Laplacian corresponding to the affinity matrix W. Let the eigenvectors corresponding to the k smallest eigenvalues of a matrix be referred to as the k smallest eigenvectors in rest of the chapter. The minimization of $tr(U^T \mathcal{L} U)$, subject to the orthogonality constraint $U^T U = \mathbf{I}_k$, is actually the spectral clustering objective [230], [45] for which the optimal U is given by the k smallest eigenvectors of $\mathcal{L}$. The k smallest eigenvectors of $\mathcal{L}$ thus contain the cluster information of $\mathcal{L}$. However, the best rank k approximation of $\mathcal{L}$ is obtained using the k largest eigenvectors and their corresponding eigenvalues. In order to merge the best low-rank approximation of $\mathcal{L}$ with the cluster information contained in it, the shifted normalized Laplacian [113], [51] is used instead of normalized Laplacian $\mathcal{L}$, which is defined as

$$L = 2\mathbf{I}_n - \mathcal{L} = \mathbf{I}_n + D^{-1/2} W D^{-1/2}. \tag{6.2}$$

The k smallest eigenvectors of the normalized Laplacian $\mathcal{L}$ correspond to the k largest eigenvectors of shifted normalized Laplacian L [113], [51]. So, the minimization of $tr(U^T \mathcal{L} U)$ becomes the maximization of $tr(U^T L U)$ in terms of the shifted normalized Laplacian. In this chapter, however, a gradient descent based approach is developed, for which the minimization objective in terms of the shifted normalized Laplacian becomes

$$\begin{aligned} &\min_{U \in \Re^{n \times k}} - tr(U^T L U) \\ &\text{such that } \; U \geqslant 0; \; U^T U = \mathbf{I}_k; \; U U^T \mathbf{1} = \mathbf{1}. \end{aligned} \tag{6.3}$$

A relaxation of (6.3) is to include the non-negativity constraint $U \geqslant 0$ as a penalty in the objective, which is given by

$$\begin{aligned} &\min_{U \in \Re^{n \times k}} - tr(U^T L U) + \xi \parallel U_- \parallel_F^2 \\ &\text{such that } \; U^T U = \mathbf{I}_k; \; U U^T \mathbf{1} = \mathbf{1}, \end{aligned} \tag{6.4}$$

where U_- denotes the negative entries of U, $\parallel . \parallel_F$ denotes the Frobenius norm of a matrix, and ξ is a non-negative parameter. The constraint set in (6.4) is given by

$$\mathsf{Km}(n,k) := \{U \in \Re^{n \times k} : U^T U = \mathbf{I}_k, U U^T \mathbf{1} = \mathbf{1}\}, \tag{6.5}$$

which is a submanifold of the Euclidean space $\Re^{n \times k}$, and known as the k-means manifold [28]. Thus, the NP-hard k-means clustering objective can be relaxed to the constrained optimization in (6.4), where the constraint set is a manifold. The problem now falls under the elegant theory of manifold optimization [3], which allows us to model the problem as the following unconstrained optimization problem

$$\min_{U \in \mathsf{Km}(n,k)} - tr(U^T L U) + \xi \parallel U_- \parallel_F^2$$

over the manifold $\mathsf{Km}(n,k)$. In the current work, a manifold optimization based algorithm is designed to efficiently integrate cluster information from different views of a multi-view data set. In rest of the chapter, the term ‘Laplacian’ refers to the shifted normalized Laplacian L as defined in (6.2), unless stated otherwise.

6.3 MiMIC: Proposed Method

Given a set of n samples or objects $\{x_i\}_{i=1}^n$, a multi-view data set, consisting of M views, is given by M matrices $X_1, \ldots, X_m, \ldots, X_M$. Each view $X_m \in \Re^{n \times d_m}$ represents the observations for a common set of n samples from the m-th data source. Let X_m be encoded by the similarity graph G_m having similarity matrix $W_m = [w_m(i,j)]_{n \times n}$, where $w_m(i,j) = w_m(j,i) \geqslant 0$ is the similarity between objects x_i and x_j in the m-th view X_m. The degree matrix D_m, corresponding to affinity matrix W_m, is given by the diagonal matrix $D_m = diag(\bar{d}_1^m, \ldots, \bar{d}_i^m, \ldots, \bar{d}_n^m)$, where $\bar{d}_i^m = \sum_{j=1}^{n} w_m(i,j)$. The shifted normalized Laplacian L_m for the corresponding view X_m is given by

$$L_m = \mathbf{I}_n + D_m^{-1/2} W_m D_m^{-1/2}. \tag{6.6}$$

Each X_m is expected to provide a different viewpoint for understanding the true nature of the data set. A truly integrative approach should be able to leverage the cluster information in different views to uncover the structure of the data set.

6.3.1 Multi-View Integration

An efficient way of integrating information from multiple views is to consider a convex combination of the corresponding graph Laplacians L_m's [199], where the views are weighted according to the quality of their cluster information. In [113], it has also been shown that the “approximate" graph Laplacian L_m^r, constructed from the $r(\geqslant k)$ largest eigenpairs of L_m, encodes better cluster information compared to that of the “full-rank" Laplacian L_m. The approximate Laplacian L_m^r is inherently free from the noise embedded in the $(n-r)$ smallest eigenpairs of L_m. Let the approximate joint Laplacian be given by

$$\mathbf{L}_{\text{Joint}}^r = \sum_{m=1}^{M} \alpha_m L_m^r, \text{ such that } \alpha_m \geqslant 0 \text{ and } \sum_{m=1}^{M} \alpha_m = 1. \tag{6.7}$$

The joint Laplacian $\mathbf{L}_{\text{Joint}}^r$ encodes the de-noised cluster information of all the views. Thus, the relaxed k-means objective of (6.4) can be optimized using $\mathbf{L}_{\text{Joint}}^r$. Note that the optimization of (6.4) using $\mathbf{L}_{\text{Joint}}^r$ would produce an $(n \times k)$ cluster indicator matrix, say U_{Joint}. However, for real-life data set, an indicator subspace of rank r $(\geqslant k)$ is generally considered inorder to retain more cluster information from the Laplacian. The relaxed clustering objective corresponding to $\mathbf{L}_{\text{Joint}}^r$ is given by

$$\begin{aligned} &\min_{U_{\text{Joint}} \in \Re^{n \times r}} -\frac{1}{2} tr(U_{\text{Joint}}^T \mathbf{L}_{\text{Joint}}^r U_{\text{Joint}}) + \frac{\xi}{2} \parallel U_{\text{Joint}-} \parallel_F^2 \\ &\text{such that } \; U_{\text{Joint}}^T U_{\text{Joint}} = \mathbf{I}_r; \; U_{\text{Joint}} U_{\text{Joint}}^T \mathbf{1} = \mathbf{1}, \end{aligned} \tag{6.8}$$

where $U_{\text{Joint}-}$ denotes the negative entries of U_{Joint}. The matrix U_{Joint} can alternatively be thought of as a low-rank orthonormal subspace representation of the joint cluster information in $\mathbf{L}_{\text{Joint}}^r$. Under this new representation of U_{Joint}, the pairwise similarities between the samples can be computed using their inner product in U_{Joint}, given by

$$S_{\text{Joint}} = U_{\text{Joint}} U_{\text{Joint}}^T \in \Re^{n \times n}.$$

Similarly, for each view X_j, for $j \in \{1, \ldots, M\}$, let $U_j \in \Re^{n \times r}$ denote its rank r orthonormal subspace representation, such that $U_j^T U_j = \mathbf{I}_r$. The pairwise similarity matrix for X_j using subspace U_j is given by $S_j = U_j U_j^T$.

Different views of a multi-view data set are expected to convey similar information. Therefore, during integration, it is intended to reduce the disagreement between the joint and individual views. The disagreement between view X_j and the joint view is given by

$$\mathfrak{D}\left(U_{\text{Joint}}, U_j\right) = \left\| S_{\text{Joint}} - S_j \right\|_F^2.$$

Substituting the values of S_{Joint} and S_j, we get

$$\begin{aligned}
&\mathfrak{D}(U_{\text{Joint}}, U_j) = \left\| U_{\text{Joint}} U_{\text{Joint}}^T - U_j U_j^T \right\|_F^2 \\
&= tr\left(U_{\text{Joint}}^T U_{\text{Joint}}\right) + tr\left(U_j^T U_j\right) - 2tr\left(U_{\text{Joint}} U_{\text{Joint}}^T U_j U_j^T\right) \\
&= 2r - 2\ tr\left(U_{\text{Joint}} U_{\text{Joint}}^T U_j U_j^T\right).
\end{aligned}$$

Hence, disagreement minimization reduces to the minimization of $-\ tr\left(U_{\text{Joint}} U_{\text{Joint}}^T U_j U_j^T\right)$. For each view X_j, the aim is to find an orthonormal subspace U_j that optimizes the spectral clustering objective $tr(U_j^T L_j^r U_j)$ as well as minimizes its disagreement with the joint view. Here also, the approximate Laplacian L_j^r is used because of its de-noising properties as mentioned in [113]. Hence, in the proposed approach, the integrative clustering objective is given by $\boldsymbol{f}\left(U_{\text{Joint}}, U_1, \ldots, U_j, \ldots, U_M\right) =$

$$\begin{aligned}
&-\frac{1}{2r} tr\left(U_{\text{Joint}}^T \mathbf{L}_{\text{Joint}}^r U_{\text{Joint}}\right) + \frac{\xi}{2r} \parallel U_{\text{Joint}-} \parallel_F^2 \\
&-\frac{1}{2rM} \sum_{j=1}^{M} \left[tr\left(U_{\text{Joint}} U_{\text{Joint}}^T U_j U_j^T\right) + tr(U_j^T L_j^r U_j)\right].
\end{aligned} \tag{6.9}$$

The Laplacians $\mathbf{L}_{\text{Joint}}$ and L_j's have maximum eigenvalue of 2 for the corresponding eigenvector $\mathbf{1}$. In the ideal case where all the individual graph Laplacians have identical r disconnected components, $\mathbf{L}_{\text{Joint}}$ and L_j's have eigenvalue 2 with multiplicity r. Then, the $\mathbf{L}_{\text{Joint}}$ and L_j based trace terms of (6.9) reduce to $2r$, the squared norm term reduces to 0, while the disagreement based trace term reduces to r. So, the final optimization problem is given by

$$\begin{aligned}
&\min_{U_{\text{Joint}},\, U_j \in \Re^{n \times r}} \boldsymbol{f}\left(U_{\text{Joint}}, U_1, \ldots, U_j, \ldots, U_M\right) \qquad (6.10) \\
&\text{such that } \ U_{\text{Joint}}^T U_{\text{Joint}} = \mathbf{I}_r, U_{\text{Joint}} U_{\text{Joint}}^T \mathbf{1} = \mathbf{1},\ U_j^T U_j = \mathbf{I}_r.
\end{aligned}$$

The above-mentioned constrained optimization problem can be solved by formulating an unconstrained optimization problem over the Euclidean space $\Re^{n \times r}$ using Lagrange multipliers. In such case, the second constraint, that is, $U_{\text{Joint}} U_{\text{Joint}}^T \mathbf{1} = \mathbf{1}$, imposes a row sum to 1 criterion on U_{Joint} and introduces a set of n Lagrange multipliers, while the orthonormality constraint on the subspaces U_{Joint} and U_j's introduces $r(r+1)/2$ Lagrange multipliers each. These add up to a total of $\left(n + (M+1)\frac{r(r+1)}{2}\right)$ Lagrange multipliers. Instead of solving a large set of partial derivatives for those multipliers in the Euclidean space, the problem is mapped to an unconstrained optimization problem over manifolds. Moreover, manifold optimization has the advantage of capturing low-dimensional non-linear manifold structure of the high-dimensional views.

The constraints on U_{Joint} indicate that U_{Joint} must belong to $\mathsf{Km}(n, r)$, the k-means manifold of rank r, given by (6.5). On the other hand, the orthonormality constraint on U_j implies that U_j must be an element of the Stiefel manifold [3] of rank r, which is given by

$$\mathsf{St}(n, r) := \{U \in \Re^{n \times r} : U^T U = \mathbf{I}_r\}. \tag{6.11}$$

Thus, the constrained optimization problem of (6.10) boils down to the optimization of $\boldsymbol{f}$ over two different types of manifolds. The unconstrained multi-manifold optimization problem is, therefore, given by

$$\min_{\substack{U_{\text{Joint}} \in \mathsf{Km}(n,r) \\ U_j \in \mathsf{St}(n,r)}} \boldsymbol{f}\left(U_{\text{Joint}}, U_1, \ldots, U_j, \ldots, U_M\right). \tag{6.12}$$

A line-search based multi-manifold optimization algorithm, for the proposed objective function of (6.12), is described next.

6.3.2 Manifold Optimization Based Solution

The solution space for U_{Joint}, in the optimization problem of (6.12), is the k-means manifold $\mathsf{Km}(n,r)$, while for each U_j, it is the Stiefel manifold $\mathsf{St}(n,r)$. The parameters n and r are kept fixed for both of these manifolds, and are dropped for notational simplicity. The k-means and Stiefel manifolds are, henceforth, referred to as Km and St, respectively, both having parameters (n,r). Both k-means and Stiefel manifolds are non-linear submanifolds of the Euclidean space $\Re^{n\times r}$, which are not necessarily endowed with a vector space structure. Consequently, the standard gradient descent, where the iterates are obtained based on vector operations, cannot be applied on these manifolds. Line-search generalizes the concept of gradient descent on non-linear manifolds [3]. It implements following three steps iteratively until convergence: (i) project the gradient of the objective function onto the tangent space of the manifold; (ii) move along the direction of negative gradient in the tangent space; and (iii) project the point obtained in step (ii) back to the manifold. The optimization objective $\boldsymbol{f}$ in (6.9) is a continuously differentiable scalar field over both the manifolds. Given initializations $U_{\text{Joint}}^{(0)}$ and $U_j^{(0)}$'s, for $j \in \{1,\ldots,M\}$, the line-search optimization of $\boldsymbol{f}$ over multiple manifolds proceeds as follows.

6.3.2.1 Optimization of U_{Joint}

Given $U_{\text{Joint}}^{(t)}$ obtained at iteration t, and a set of fixed U_j's, for $j \in \{1,\ldots,M\}$, let

$$\mathbb{U} = \sum_{j=1}^{M} U_j U_j^T.$$

Substituting the value of $\mathbb{U}$ in (6.9), the direction of negative gradient of $\boldsymbol{f}$ at $U_{\text{Joint}}^{(t)}$ is given by

$$\begin{aligned} -\nabla_{U_{\text{Joint}}^{(t)}} \boldsymbol{f} &= -\nabla_{U_{\text{Joint}}^{(t)}} \left[-\frac{1}{2} tr\left(U_{\text{Joint}}^T (\mathbf{L}_{\text{Joint}}^r + \mathbb{U}) U_{\text{Joint}}\right) + \frac{\xi}{2} \parallel U_{\text{Joint}-}^{(t)} \parallel_F^2 \right] \\ &= \left(\mathbf{L}_{\text{Joint}}^r + \mathbb{U}\right) U_{\text{Joint}}^{(t)} - \xi U_{\text{Joint}-}^{(t)} = \boldsymbol{Q}_{\text{Joint}}^{(t)} \text{ (say)}. \end{aligned} \tag{6.13}$$

Let the tangent space of the k-means manifold rooted at the current iterate $U_{\text{Joint}}^{(t)} \in \mathsf{Km}$ be denoted by $T_{U_{\text{Joint}}^{(t)}} \mathsf{Km}$. Unlike the non-linear manifold Km, its tangent space is a vector space [3]. This makes the movement along the tangent space feasible using vector addition and scalar multiplication. The first step of line-search is to project the negative gradient $\boldsymbol{Q}^{(t)}_{\text{Joint}}$ of (6.13) onto the tangent space. This is done using the projection operator $\mathbf{\Pi}$ [57]. Let $\mathbf{\Pi}_{T_Y\mathsf{Km}}(W)$ denote the projection of $W \in \Re^{n\times r}$ onto the tangent space of Km rooted at Y. In the present case, the root of the tangent is the current iterate $U^{(t)}_{\text{Joint}}$, while the point to be projected is the negative gradient $\boldsymbol{Q}^{(t)}_{\text{Joint}}$. At iteration t, this projection is given by [28]

$$\mathbf{\Pi}_{T_{U^{(t)}_{\text{Joint}}}\mathsf{Km}}\left(\boldsymbol{Q}^{(t)}_{\text{Joint}}\right) = \boldsymbol{Q}^{(t)}_{\text{Joint}} - 2U^{(t)}_{\text{Joint}}\Omega - (\mathbf{z}\mathbf{1}^T + \mathbf{1}\mathbf{z}^T)U^{(t)}_{\text{Joint}} = \boldsymbol{Z}^{(t)}_{\text{Joint}} \quad \text{(say)}, \tag{6.14}$$

where $\mathbf{z} = \dfrac{1}{n}\boldsymbol{Q}^{(t)}_{\text{Joint}}\left(U^{(t)}_{\text{Joint}}\right)^T\mathbf{1}$ and

$$\Omega = \frac{1}{4}\left(\left(\boldsymbol{Q}^{(t)}_{\text{Joint}}\right)^T U^{(t)}_{\text{Joint}} + \left(U^{(t)}_{\text{Joint}}\right)^T\boldsymbol{Q}^{(t)}_{\text{Joint}} - 2\left(U^{(t)}_{\text{Joint}}\right)^T(\mathbf{z}\mathbf{1}^T + \mathbf{1}\mathbf{z}^T)U^{(t)}_{\text{Joint}}\right).$$

In Figure 6.1, the curved surface is used to denote the k-means manifold, while the plane denotes its tangent space. The root of the tangent space is the current iterate $U^{(t)}_{\text{Joint}}$, denoted by the point lying on both the tangent plane and the manifold. The vector moving out of the tangent plane points towards the negative gradient direction, while its projection lies on the tangent plane. Given the tangent vector $\boldsymbol{Z}^{(t)}_{\text{Joint}}$ of (6.14) and step length $\eta_{\mathsf{K}} > 0$, the next step is to move in the direction of $\boldsymbol{Z}^{(t)}_{\text{Joint}}$ within the tangent space and then project the obtained point from the tangent space $T_{U^{(t)}_{\text{Joint}}}\mathsf{Km}$ back to the manifold Km. This is achieved using the retraction operator R [3]. Given a manifold $\mathcal{M}$, a point $y \in \mathcal{M}$, and $\xi \in T_y\mathcal{M}$, the retraction $\mathsf{R}_y(\xi)$ has two steps: (i) move along ξ to get the point $y + \xi$ in

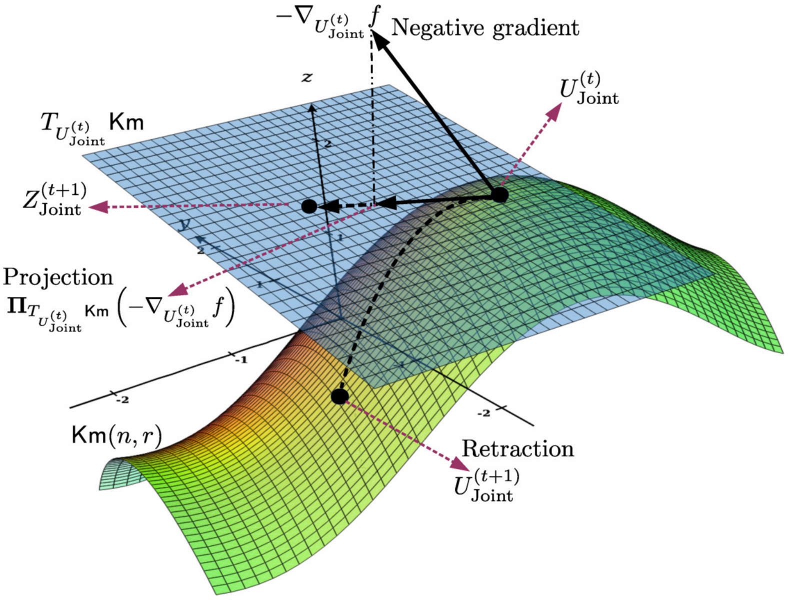

$-\nabla_{U^{(t)}_{\text{Joint}}} f$ Negative gradient
z
$U^{(t)}_{\text{Joint}}$
$T_{U^{(t)}_{\text{Joint}}}\mathsf{Km}$
$Z^{(t+1)}_{\text{Joint}}$
y
Projection
$\mathbf{\Pi}_{T_{U^{(t)}_{\text{Joint}}}\mathsf{Km}}\left(-\nabla_{U^{(t)}_{\text{Joint}}} f\right)$
-1
-2
-2
$\mathsf{Km}(n, r)$
Retraction
$U^{(t+1)}_{\text{Joint}}$

Figure 6.1: Optimization of U_{Joint} over k-means manifold.

the linear embedding space; and (ii) “project" the point $y + \xi$ to the manifold $\mathcal{M}$. For the current problem, retraction is performed on the tangent vector $\boldsymbol{Z}_{\text{Joint}}^{(t)}$ at point $U_{\text{Joint}}^{(t)}$. Retraction on Km is performed as follows. Let

$$\boldsymbol{Z}_{\text{Joint}}^{(t+1)} = U_{\text{Joint}}^{(t)} + \eta_{\mathsf{K}} \boldsymbol{Z}_{\text{Joint}}^{(t)}$$

be the point obtained by moving along $\boldsymbol{Z}_{\text{Joint}}^{(t)}$ in the tangent space. Since the tangent space is a vector space, $\boldsymbol{Z}_{\text{Joint}}^{(t+1)}$ belongs to the tangent space itself. The next step is to project $\boldsymbol{Z}_{\text{Joint}}^{(t+1)}$ from the tangent space to the manifold Km. Retraction on Km involves the matrix exponential operation [159]. A projection $\mathsf{P}(x)$ is called retractive projection of x if $\mathsf{P} : x \to y$ is a retraction. Let $\mathsf{PKm}_Y(Z)$ denote the retractive projection of Z from the tangent space $T_Y\mathsf{Km}$ rooted at Y back to Km. Here, the retractive projection is performed on $\boldsymbol{Z}_{\text{Joint}}^{(t+1)}$. This is given by [28]

$$\begin{aligned} \mathsf{PKm}_{U_{\text{Joint}}^{(t)}}\left(\boldsymbol{Z}_{\text{Joint}}^{(t+1)}\right) &= \exp(B)\ \exp(Q')\ U_{\text{Joint}}^{(t)}, \qquad (6.15)\\ \text{where} \quad Q &= \left(U_{\text{Joint}}^{(t)}\right)^T \boldsymbol{Z}_{\text{Joint}}^{(t+1)} \in \Re^{r\times r};\\ Q' &= U_{\text{Joint}}^{(t)}\ Q\ \left(U_{\text{Joint}}^{(t)}\right)^T \in \Re^{n\times n}; \ \ \text{and}\\ B &= \boldsymbol{Z}_{\text{Joint}}^{(t+1)}\left(U_{\text{Joint}}^{(t)}\right)^T - U_{\text{Joint}}^{(t)}\left(\boldsymbol{Z}_{\text{Joint}}^{(t+1)}\right)^T - 2Q' \in \Re^{n\times n}. \end{aligned}$$

Finally, the retracted point in (6.15) is the U_{Joint} obtained at iteration $(t+1)$, that is,

$$U_{\text{Joint}}^{(t+1)} = \mathsf{PKm}_{U_{\text{Joint}}^{(t)}}\left(\boldsymbol{Z}_{\text{Joint}}^{(t+1)}\right).$$

Theorem 6.1. *$U_{\text{Joint}}^{(t+1)}$ belongs to the k-means manifold.*

Proof. For $U_{\text{Joint}}^{(t+1)}$ to belong to k-means manifold, denoted by Km, it must satisfy its properties given in (6.5) of the main chapter. So, $U_{\text{Joint}}^{(t+1)}$ must have orthonormal columns:

$$\begin{aligned} \left(U_{\text{Joint}}^{(t+1)}\right)^T U_{\text{Joint}}^{(t+1)} &= \left(\exp(B)\exp(Q')U_{\text{Joint}}^{(t)}\right)^T\left(\exp(B)\exp(Q')U_{\text{Joint}}^{(t)}\right) \qquad \text{(from (6.15))}\\ &= \left(U_{\text{Joint}}^{(t)}\right)^T \exp(Q')^T\exp(B)^T\exp(B)\exp(Q')U_{\text{Joint}}^{(t)}\\ &= \left(U_{\text{Joint}}^{(t)}\right)^T \exp(-Q')\exp(-B)\exp(B)\exp(Q')U_{\text{Joint}}^{(t)}\\ &= \left(U_{\text{Joint}}^{(t)}\right)^T U_{\text{Joint}}^{(t)} = \mathbf{I}_r. \end{aligned}$$

It can be shown that $U_{\text{Joint}}^{(t)}\left(U_{\text{Joint}}^{(t)}\right)^T$ commutes with $\exp(Q')$ [28] (see Lemma 6.1 for details). Hence,

$$\exp(Q')U_{\text{Joint}}^{(t)}\left(U_{\text{Joint}}^{(t)}\right)^T = U_{\text{Joint}}^{(t)}\left(U_{\text{Joint}}^{(t)}\right)^T\exp(Q').$$

Also, $\exp(B)\mathbf{1} = \exp(-B)\mathbf{1} = \mathbf{1}$. So,

$$\begin{aligned} U_{\text{Joint}}^{(t+1)} \left(U_{\text{Joint}}^{(t+1)}\right)^T \mathbf{1} &= \left(\exp(B)\exp(Q')U_{\text{Joint}}^{(t)}\right)\left(\exp(B)\exp(Q')U_{\text{Joint}}^{(t)}\right)^T \mathbf{1} \\ &= \exp(B)\exp(Q')U_{\text{Joint}}^{(t)} \left(U_{\text{Joint}}^{(t)}\right)^T \exp(Q')^T \exp(B)^T \mathbf{1} \\ &= \exp(B)\exp(Q')U_{\text{Joint}}^{(t)} \left(U_{\text{Joint}}^{(t)}\right)^T \exp(-Q')\exp(-B)\mathbf{1} \\ &= \exp(B)U_{\text{Joint}}^{(t)} \left(U_{\text{Joint}}^{(t)}\right)^T \exp(Q')\exp(-Q')\exp(-B)\mathbf{1} \\ &= \exp(B)U_{\text{Joint}}^{(t)} \left(U_{\text{Joint}}^{(t)}\right)^T \mathbf{1} = \exp(B)\mathbf{1} = \mathbf{1}. \end{aligned}$$

Thus, the next iterate $U_{\text{Joint}}^{(t+1)}$ satisfies both the properties of Km, and therefore, belongs to it. ■

In Theorem 6.1, the commutative property of $U_{\text{Joint}}^{(t)}\left(U_{\text{Joint}}^{(t)}\right)^T$ and $\exp(Q')$ is used to prove that $U_{\text{Joint}}^{(t+1)}$ belongs to the k-means manifold. The following lemma proves the commutative property [28].

Lemma 6.1. $U_{\text{Joint}}^{(t)}\left(U_{\text{Joint}}^{(t)}\right)^T$ *commutes with* $\exp(Q')$.

Proof. The t-th iterate of U_{Joint} belongs to the k-means manifold. So, from the properties of k-means manifold (defined in (6.5)), it satisfies that

$$\left(U_{\text{Joint}}^{(t)}\right)^T U_{\text{Joint}}^{(t)} = \mathbf{I}_r. \tag{6.16}$$

From (6.15), we have

$$Q' = U_{\text{Joint}}^{(t)}\, Q\, \left(U_{\text{Joint}}^{(t)}\right)^T \in \Re^{n\times n}, \tag{6.17}$$

where $Q \in \Re^{r\times r}$. The exponential of Q' is given by [159]

$$\exp(Q') = \mathbf{I}_n + Q' + \frac{Q'^2}{2!} + \frac{Q'^3}{3!} + \ldots = \sum_{j=0}^{\infty} \frac{Q'^j}{j!}.$$

Now,

$$\begin{aligned} Q'\, U_{\text{Joint}}^{(t)}\left(U_{\text{Joint}}^{(t)}\right)^T &= U_{\text{Joint}}^{(t)} Q (U_{\text{Joint}}^{(t)})^T U_{\text{Joint}}^{(t)} \left(U_{\text{Joint}}^{(t)}\right)^T && \text{(from (6.17))} \\ &= U_{\text{Joint}}^{(t)} (U_{\text{Joint}}^{(t)})^T U_{\text{Joint}}^{(t)} Q \left(U_{\text{Joint}}^{(t)}\right)^T && \text{(from (6.16))} \\ &= U_{\text{Joint}}^{(t)}\left(U_{\text{Joint}}^{(t)}\right)^T Q'. && (6.18) \end{aligned}$$

Therefore,

$$\begin{aligned}\exp(Q')\, U^{(t)}_{\text{Joint}} \left(U^{(t)}_{\text{Joint}}\right)^T &= \left(\mathbf{I}_n + Q' + \frac{Q'^2}{2!} + \frac{Q'^3}{3!} + \dots\right) U^{(t)}_{\text{Joint}} \left(U^{(t)}_{\text{Joint}}\right)^T \\ &\quad \text{(applying (6.18) repetatively)} \\ &= U^{(t)}_{\text{Joint}} \left(U^{(t)}_{\text{Joint}}\right)^T \left(\mathbf{I}_n + Q' + \frac{Q'^2}{2!} + \frac{Q'^3}{3!} + \dots\right) \\ &= U^{(t)}_{\text{Joint}} \left(U^{(t)}_{\text{Joint}}\right)^T \exp(Q').\end{aligned}$$

Hence, $U^{(t)}_{\text{Joint}} \left(U^{(t)}_{\text{Joint}}\right)^T$ commutes with $\exp(Q')$. ■

The algorithm for a single update of U_{Joint} is given in Algorithm 6.1. Figure 6.1 shows the diagrammatic representation of the gradient computation, tangent space projection, and retraction operation on the k-means manifold. As shown in Figure 6.1, the point $\boldsymbol{Z}^{(t+1)}_{\text{Joint}}$ obtained by moving along the tangent plane lies on the tangent plane itself, while the retracted point $U^{(t+1)}_{\text{Joint}}$ lies only on the curved surface (manifold). The variable U_{Joint} in the objective function $\boldsymbol{f}$ is optimized over the k-means manifold. There are M other variables U_j's, each corresponding to one of the views. The solution space for U_j, for $j \in \{1, \dots, M\}$, is the Stiefel manifold St.

Algorithm 6.1 Optimize_k-means

```
▷ Optimization of $U_{\text{Joint}}$ over $k$-means manifold
Input: Joint Laplacian $\mathbf{L}^r_{\text{Joint}}$, subspaces $U_j$ for $j = 1, ..., M$, $U^{(t)}_{\text{Joint}}$ of iteration $t$, step
    length $\eta_{\mathsf{K}} > 0$, $\xi > 0$.
Output: $U^{(t+1)}_{\text{Joint}}$.
Compute negative gradient $\boldsymbol{Q}^{(t)}_{\text{Joint}} \leftarrow \left[-\nabla_{U^{(t)}_{\text{Joint}}} \boldsymbol{f}\right]$ by (6.13).
Project negative gradient onto tangent space:
    $\boldsymbol{Z}^{(t)}_{\text{Joint}} \leftarrow \boldsymbol{\Pi}_{T_{U^{(t)}_{\text{Joint}}}\mathsf{Km}} \left(\boldsymbol{Q}^{(t)}_{\text{Joint}}\right)$ using (6.14).
$\boldsymbol{Z}^{(t+1)}_{\text{Joint}} \leftarrow U^{(t)}_{\text{Joint}} + \eta_{\mathsf{K}} \boldsymbol{Z}^{(t)}_{\text{Joint}}$.
Find retractive projection $\mathsf{PKm}_{U^{(t)}_{\text{Joint}}} \left(\boldsymbol{Z}^{(t+1)}_{\text{Joint}}\right)$ using (6.15).
$U^{(t+1)}_{\text{Joint}} \leftarrow \mathsf{PKm}_{U^{(t)}_{\text{Joint}}} \left(\boldsymbol{Z}^{(t+1)}_{\text{Joint}}\right)$.
Return $U^{(t+1)}_{\text{Joint}}$.
```

6.3.2.2 Optimization of U_j

Let $U^{(t)}_j$ denote the U_j obtained at iteration t. For a specific $j \in \{1, \dots, M\}$, considering U_{Joint} and all other U_i's to be fixed for $i \in \{1, \dots, M\}$ and $i \neq j$, the direction of negative gradient of $\boldsymbol{f}$ at $U_j^{(t)}$ is given by

$$-\nabla_{U_j^{(t)}}\boldsymbol{f} = -\nabla_{U_j^{(t)}}\Big[-\frac{1}{2}tr\left(\left(U_j^{(t)}\right)^T\left(U_{\text{Joint}}U_{\text{Joint}}^T + L_j^r\right)U_j^{(t)}\right)\Big],$$
$$\Rightarrow -\nabla_{U_j^{(t)}}\boldsymbol{f} = \left(U_{\text{Joint}}U_{\text{Joint}}^T + L_j^r\right)U_j^{(t)} = \boldsymbol{Q}_j^{(t)} \text{ (say).} \tag{6.19}$$

To optimize U_j, first the negative gradient direction $\boldsymbol{Q}_j^{(t)}$ of (6.19) is projected onto the tangent space $T_{U_j^{(t)}}\mathsf{St}$ of Stiefel manifold at $U_j^{(t)}$. The operator $\boldsymbol{\Pi}$ for projecting $\boldsymbol{Q}_j^{(t)}$ onto the tangent space of St rooted at the current iterate $U_j^{(t)}$ is given by [57]

$$\begin{aligned}\Pi_{T_{U_j^{(t)}}\mathsf{St}}\left(\boldsymbol{Q}_j^{(t)}\right) &= \boldsymbol{Q}_j^{(t)} - \frac{1}{2}U_j^{(t)}\left((U_j^{(t)})^T\boldsymbol{Q}_j^{(t)} + (\boldsymbol{Q}_j^{(t)})^TU_j^{(t)}\right)\\ &= \left(\mathbf{I}_n - U_j^{(t)}(U_j^{(t)})^T\right)\boldsymbol{Q}_j^{(t)} = \boldsymbol{Z}_j^{(t)} \ \ \text{(say).}\end{aligned} \tag{6.20}$$

Given the step length $\eta_{\mathsf{S}} > 0$ and the tangent vector $\boldsymbol{Z}_j^{(t)}$, the current iterate $U_j^{(t)}$ is moved in the direction of the tangent $\boldsymbol{Z}_j^{(t)}$ to obtain

$$\boldsymbol{Z}_j^{(t+1)} = U_j^{(t)} + \eta_{\mathsf{S}}\boldsymbol{Z}_j^{(t)}.$$

The point $\boldsymbol{Z}_j^{(t+1)}$ which lies in the tangent space is retracted back to the manifold St to obtain the next iterate $U_j^{(t+1)}$. Retraction on St is performed using the singular value decomposition (SVD) of $\boldsymbol{Z}_j^{(t+1)}$ [2]. Let the SVD of $\boldsymbol{Z}_j^{(t+1)}$ be given by

$$\boldsymbol{Z}_j^{(t+1)} = E_j^{(t+1)}\,\Xi_j^{(t+1)}\,\left(V_j^{(t+1)}\right)^T,$$

where $E_j^{(t+1)}$ and $V_j^{(t+1)}$ contain the left and right singular vectors of $\boldsymbol{Z}_j^{(t+1)}$ in their columns, respectively, and $\Xi_{(j)}^{(t+1)}$ is a diagonal matrix containing the singular values stored in non-increasing order. Following the notation of (6.15), the retractive projection of $\boldsymbol{Z}_j^{(t+1)}$ onto St is given by

$$\mathsf{PSt}_{U_j^{(t)}}\left(\boldsymbol{Z}_j^{(t+1)}\right) = E_j^{(t+1)}\,\left(V_j^{(t+1)}\right)^T. \tag{6.21}$$

The retracted point in (6.21) is the next iterate of U_j, that is,

$$U_j^{(t+1)} = \mathsf{PSt}_{U_j^{(t)}}\left(\boldsymbol{Z}_j^{(t+1)}\right).$$

Theorem 6.2. *$U_j^{(t+1)}$ belongs to the Stiefel manifold.*

Proof. For $U_j^{(t+1)}$ to belong to the Stiefel manifold, it must satisfy its properties given by (6.11), that is, it must have orthonormal columns. The matrices $E_j^{(t+1)}$ and $V_j^{(t+1)}$, given by (6.21), contain the left and right singular vectors of $\boldsymbol{Z}_j^{(t+1)}$, respectively, which have onrthonormal columns. Therefore,

$$\left(U_j^{(t+1)}\right)^T U_j^{(t+1)} = V_j^{(t+1)} \left(E_j^{(t+1)}\right)^T E_j^{(t+1)} \left(V_j^{(t+1)}\right)^T = \mathbf{I}_r.$$

Thus, the next iterate of U_j belongs to the Stiefel manifold. ■

The algorithm for a single update of U_j is given in Algorithm 6.2. The optimization in U_j is performed for each of the M views separately, considering U_{Joint} and all U_i's for $i \in \{1, \ldots, M\}$ and $i \neq j$ to be fixed.

Algorithm 6.2 Optimize_Stiefel

```
▻ Optimization of $U_j$ over Stiefel manifold
Input: Laplacian $L_j^r$, subspace $U_{\text{Joint}}$, $U_j^{(t)}$ of iteration $t$, step length $\eta_{\mathsf{S}}$.
Output: $U_j^{(t+1)}$.
Compute negative gradient $\boldsymbol{Q}_j^{(t)} \leftarrow -\nabla_{U_j^{(t)}} \boldsymbol{f}$ using (6.19).
Project negative gradient onto tangent space:
    $\boldsymbol{Z}_j^{(t)} \leftarrow \Pi_{T_{U_j^{(t)}}\mathsf{St}}\left(\boldsymbol{Q}_j^{(t)}\right)$ using (6.20).
$\boldsymbol{Z}_j^{(t+1)} \leftarrow U_j^{(t)} + \eta_{\mathsf{S}} \boldsymbol{Z}_j^{(t)}$.
Find retractive projection $\mathsf{PSt}_{U_j^{(t)}}\left(\boldsymbol{Z}_j^{(t+1)}\right)$ using (6.21).
$U_j^{(t+1)} \leftarrow \mathsf{PSt}_{U_j^{(t)}}\left(\boldsymbol{Z}_j^{(t+1)}\right)$.
Return $U_j^{(t+1)}$.
```

6.3.3 Proposed Algorithm

Given M affinity matrices $W_1, \ldots, W_M$, corresponding to M views $X_1, \ldots, X_M$ and a fixed rank r, the proposed method extracts a low-rank joint subspace representation U_{Joint} that best preserves the cluster structure of a multi-view data set. The clusters embedded within the data set are identified by performing k-means clustering on the first k columns of U_{Joint}. The convex combination $\boldsymbol{\alpha}$ of (6.7) assigns the importance of the individual graphs during data integration. In the proposed algorithm, the weights α_m's are assigned according to the quality of cluster structure reflected in the individual views, which is determined by the eigenvalues and eigenvectors of their corresponding Laplacians (see Section 5.3.5 of Chapter 5).

6.3.3.1 Choice of Initial Iterates

For each view X_m, where $m \in \{1, \ldots, M\}$, spectral clustering, in terms of its shifted normalized Laplacian L_m, solves the following optimization problem [162], [45]:

$$\min_{U_m \in \Re^{n \times k}} - tr\left(U_m^T L_m U_m\right) \text{ such that } U_m^T U_m = \mathbf{I}_k, \tag{6.22}$$

where k is the number of clusters. The solution to (6.22) is given by the k largest eigenvectors of L_m. The rank r spectral clustering solution is chosen as the initial iterate for the subspaces corresponding to joint and individual views. Let the eigen-decomposition of the graph Laplacians, corresponding to joint and individual views, be given by

$$\mathbf{L}_{\text{Joint}} = U_{\text{Joint}}\, \Sigma_{\text{Joint}}\, U_{\text{Joint}}^T \quad \text{and} \quad L_j = U_j\, \Sigma_j\, U_j^T,$$

for $j \in \{1, \ldots, M\}$. Here, U_{Joint} and U_j's contain eigenvectors while Σ_{Joint} and Σ_j's contain corresponding eigenvalues in non-increasing order. The initial iterates for the proposed algorithm are given by

$$U_{\text{Joint}}^{(0)} = U_{\text{Joint}}^r \quad \text{and} \quad U_j^{(0)} = U_j^r,$$

where U_{Joint}^r and U_j^r contain the r largest eigenvectors in U_{Joint} and U_j, respectively.

6.3.3.2 Convergence Criterion

Let $\boldsymbol{f}^{(t)}$ denote the value of the objective function $\boldsymbol{f}$ evaluated using $U_{\text{Joint}}^{(t)}$ and $U_j^{(t)}$'s, obtained at iteration t. For the proposed algorithm, the step lengths for optimization on both the manifolds are chosen to be identical, that is, $\eta_{\mathsf{K}} = \eta_{\mathsf{S}} = \eta$. The direction of movement at each iteration is always along the negative gradient (as in (6.13) and (6.19)), which should lead to a reduction in the objective function. To ensure convergence, the proposed algorithm moves to the next iterate only when there is a sufficient reduction in the value of the objective function (according to the Armijo criterion [9], see Section S2 of supplementary). Otherwise, both the step sizes are reduced by a factor $\delta \in (0, 1)$.

The proposed algorithm converges when, even with very small step sizes η_{K} and η_{S}, the difference in the objective function $\boldsymbol{f}$ in two consecutive iterations falls below the threshold $\epsilon > 0$, that is,

$$\boldsymbol{f}^{(t)} - \boldsymbol{f}^{(t+1)} < \epsilon. \tag{6.23}$$

The proposed algorithm is described in Algorithm 6.3.

6.3.3.3 Computational Complexity

Let $X_1, \ldots, X_m, \ldots, X_M$, where $X_m \in \Re^{n \times d_m}$, be M different views of a multi-view data set, all measured on the same set of n samples. The number of clusters in the data set is assumed to be known and is denoted by k, and let r be the rank of joint and individual subspaces, U_{Joint} and U_js, which is given as input to the proposed Algorithm 6.3. Given the similarity matrix W_m for view X_m, its graph Laplacian L_m is computed in step 2 in $\mathcal{O}(n^2)$ time. Then, the eigen-decomposition of L_m is computed in step 3 which takes $\mathcal{O}(n^3)$ time for the $(n \times n)$ matrix. The computation of relevance $\boldsymbol{\chi}_m$ in step 6 involves computation of Silhouette index which has pair-wise distance computation and takes $\mathcal{O}(n^2)$ time. For M views, the total complexity of steps 1–6 is bounded by $\mathcal{O}(Mn^3)$. The computation of joint Laplacian and its eigen-decomposition in steps 7 and 8, respectively, takes atmost $\mathcal{O}(n^3)$ time. Steps 9–10 are initializations, which take constant time. For a fixed j, optimization of U_j over Stiefel manifold takes $\mathcal{O}(n^2 r)$ time. The loop for j in step 12 runs once for each of the M views, which contributes to a total complexity of $\mathcal{O}(Mn^2 r)$ for steps 12–14. The optimization of U_{Joint} over k-means manifold in step

15 has $\mathcal{O}(n^3)$ time complexity due to the matrix exponential based retraction operation. The computation of the joint objective in step 16 takes $\mathcal{O}(Mn^2r)$ time. The evaluation of convergence criteria and variable updation in steps 17−21 takes $\mathcal{O}(1)$ time. Assuming that the algorithm takes t iterations to converge, the overall complexity of steps 11−22 is bounded by $\mathcal{O}(t\max\{n^3, Mn^2r\})$. The clustering on the final solution $U^{\star}_{\text{Joint}}$ in step 24 takes $\mathcal{O}(t_{km}nk^2)$ time, where t_{km} is the maximum number of iterations k-means clustering executes.

Hence, the overall computational complexity of the proposed MiMIC algorithm, to extract the subspace $U^{\star}_{\text{Joint}}$ and perform clustering, is $(\mathcal{O}(Mn^3+t\max\{n^3, Mn^2r\}+t_{km}nk^2) =)\mathcal{O}(tn^3)$, assuming $M, r, k << n$.

Algorithm 6.3 Proposed Algorithm: **MiMIC**

```
Input: Similarity matrices $W_1, \ldots, W_M$, number of clusters $k$, rank $r \geqslant k$, step lengths
    $\eta_{\mathsf{K}}$ and $\eta_{\mathsf{S}}$, $\mathsf{Km}$ parameter $\xi > 0$, convergence parameter $\epsilon > 0$ and $\delta \in (0, 1)$.
Output: Subspace $U^{\star}_{\text{Joint}}$ and clusters $A_1, \ldots, A_k$.
 1: for $m \leftarrow 1$ to $M$ do
 2:     Construct degree matrix $D_m$ and Laplacian $L_m$ as in (6.6).
 3:     Compute eigen-decomposition of $L_m$.
 4:     Store $r$ largest eigenvectors of $L_m$ in columns of $U^r_m$.
 5:     Compute weight $\alpha_m$ of $X_m$ (using (34) of the supplementary document).
 6: end for
 7: Compute joint Laplacian $\mathbf{L}^r_{\text{Joint}}$ using (7.4).
 8: Compute eigen-decomposition of $\mathbf{L}^r_{\text{Joint}}$.
 9: Initialize: $U^{(0)}_{\text{Joint}} \leftarrow U^r_{\text{Joint}}$, $U^{(0)}_j \leftarrow U^r_j$, $j = 1, .., M$.
10: $t \leftarrow 0$; $\boldsymbol{f}^{(0)} \leftarrow \boldsymbol{f}\left(U^{(0)}_{\text{Joint}}, U^{(0)}_1, \ldots, U^{(0)}_M\right)$.
11: do
12:     for each $j \in \{1, .., M\}$ do
13:         $U^{(t+1)}_j \leftarrow$ Optimize_Stiefel$\left( L^r_j, U^{(t)}_{\text{Joint}}, U^{(t)}_j, \eta_{\mathsf{S}}\right)$
14:     end for
15:     $U^{(t+1)}_{\text{Joint}} \leftarrow$ Optimize_$k$-means $\left(\mathbf{L}^r_{\text{Joint}}, U^{(t)}_{\text{Joint}}, U^{(t)}_1, \ldots \ldots, U^{(t)}_M, \eta_{\mathsf{K}}, \xi\right)$.
16:     Compute $\boldsymbol{f}^{(t+1)} \leftarrow \boldsymbol{f}\left(U^{(t+1)}_{\text{Joint}}, U^{(t+1)}_1, \ldots, U^{(t+1)}_M\right)$.
17:     if $\left(\boldsymbol{f}^{(t)} - \boldsymbol{f}^{(t+1)}\right) > \epsilon$ then
18:         Update to next iteration: $t = t + 1$.
19:     else
20:         Reduce step length: $\eta_{\mathsf{S}} = \delta\eta_{\mathsf{S}}$, $\eta_{\mathsf{K}} = \delta\eta_{\mathsf{K}}$.
21:     end if
22: while $(\eta_{\mathsf{K}} > 1e - 06 \;\&\; \eta_{\mathsf{S}} > 1e - 06)$
23: Optimal solution: $U^{\star}_{\text{Joint}} \leftarrow U^{(t+1)}_{\text{Joint}}$.
24: Perform $k$-means clustering on first $k$ columns of $U^{\star}_{\text{Joint}}$.
25: Return $U^{\star}_{\text{Joint}}$ and clusters $A_1, \ldots, A_k$ from $k$-means.
```

6.4 Asymptotic Convergence Analysis

This section presents the convergence analysis of the proposed MiMIC algorithm for the set of given U_j's under Armijo constraints [9] on the choice of step length. The asymptotic behavior of the algorithm is also studied to derive a bound on the difference between the objective function $\boldsymbol{f}$ evaluated at some iteration t and at the optimal solution, for sufficiently large values of t. The bound can be used to make inference about the compactness and separability of the clusters in the data set.

The proposed MiMIC algorithm for multi-view data clustering is provided in Algorithm 6.3. Before discussing the convergence result and analyzing its asymptotic behavior, the notation for the retraction operation on a manifold is re-stated next. Given a manifold $\mathcal{M}$, a point $y \in \mathcal{M}$, let $T_y\mathcal{M}$ denote the tangent space of the manifold rooted at point y. Given a tangent $\xi \in T_y\mathcal{M}$, the retraction operation $\mathsf{R}_y(\xi)$ denotes the combination of two steps. First, movement along ξ to get the point $y + \xi$ in the tangent space. Second, projection of the point $y + \xi$ back to the manifold $\mathcal{M}$. For minimization of a function $f(y)$ over $\mathcal{M}$, given the current iterate $y^{(t)}$ at iterarion t, the update equation for line-search [3] on $\mathcal{M}$ is given by

$$y^{(t+1)} = \mathsf{R}_{y^{(t)}}(\eta \mathscr{d}^{(t)}),$$

where $\mathscr{d}^{(t)}$ is a descent direction and η is the step length. For the proposed MiMIC algorithm, while optimizing the joint objective $\boldsymbol{f}$ with respect to U_{Joint} over the k-means manifold Km, the set of update equations is given by (Section 6.3.2.1)

$$\begin{aligned} \boldsymbol{Q}^{(t)}_{\text{Joint}} &= -\nabla_{U^{(t)}_{\text{Joint}}} \boldsymbol{f} \\ \boldsymbol{Z}^{(t)}_{\text{Joint}} &= \boldsymbol{\Pi}_{T_{U^{(t)}_{\text{Joint}}}\mathsf{Km}} \left(\boldsymbol{Q}^{(t)}_{\text{Joint}}\right) \\ \boldsymbol{Z}^{(t+1)}_{\text{Joint}} &= U^{(t)}_{\text{Joint}} + \eta \boldsymbol{Z}^{(t)}_{\text{Joint}} \\ U^{r(t+1)}_{\text{Joint}} &= \mathsf{PKm}_{U^{(t)}_{\text{Joint}}} \left(\boldsymbol{Z}^{(t+1)}_{\text{Joint}}\right), \end{aligned} \tag{6.24}$$

where $U^{(t)}_{\text{Joint}}$ denotes the value of U_{Joint} at iteration t. The set of equations in (6.24) can be coupled using the retraction operation R and written as

$$U^{(t+1)}_{\text{Joint}} = \mathsf{RKm}_{U^{(t)}_{\text{Joint}}} \left(-\eta \nabla_{U^{(t)}_{\text{Joint}}} \boldsymbol{f}\right), \tag{6.25}$$

where RKm denotes retraction on the k-means manifold. To prove the convergence of the proposed algorithm, certain restrictions are imposed on the descent direction and choice of step size during optimization. These are discussed in Appendix D.

6.4.1 Convergence

The following convergence result in Theorem 6.3 for line-search optimization over manifolds is motivated from their classical counterparts in $\Re^n$ [3].

Theorem 6.3 (**Convergence**). *Every limit point of the sequence* $\{U^{(t)}_{\text{Joint}}\}_{t=0,1,2,\ldots}$, *gener-*

ated by the proposed algorithm for a set of given U_j's for $j \in \{1,..,M\}$, is a critical point of the cost function $\boldsymbol{f}$.

Proof. (By contradiction) Let there be a subsequence of iterations $\{U^{(t)}_{\text{Joint}}\}_{t\in\mathcal{T}}$ that converges to some $U^{\star}_{\text{Joint}}$ which is not a critical point of $\boldsymbol{f}$, that is $\nabla_{U^{\star}_{\text{Joint}}}\boldsymbol{f} \neq 0$. The direction of movement at each iteration is the negative gradient along which the reduction of cost $\boldsymbol{f}$ is maximum. It follows that the whole sequence $\{\boldsymbol{f}(U^{(t)}_{\text{Joint}})\}$ is non-increasing and converges to $\boldsymbol{f}(U^{\star}_{\text{Joint}})$. So, the difference $\boldsymbol{f}(U^{(t)}_{\text{Joint}}) - \boldsymbol{f}(U^{(t+1)}_{\text{Joint}})$ goes gradually to zero. The Armijo criterion $C_{\mathcal{A}}$, given by (D.2), is evaluated at each iteration of the proposed MiMIC algorithm. The algorithm proceeds to the next iteration only if $C_{\mathcal{A}} \geqslant 0$. The k-means manifold, over which U_{Joint} is optimized, is a Riemannian manifold with the inner product given by $\langle Z_1, Z_2\rangle = tr(Z_1^T Z_2)$ [28]. This relation can used to replace the trace term in (D.2). Furthermore, for a set of given U_j's for $j \in \{1,..,M\}$, $\boldsymbol{f}$ becomes a function of U_{Joint} only. In that case, the negative gradient becomes $-\nabla_{U^{(t)}_{\text{Joint}}}\boldsymbol{f} = \boldsymbol{Q}^{(t)}_{\text{Joint}}$ (see (6.13)). Using (6.24), the Armijo criterion $C_{\mathcal{A}}$ can be written as

$$C_{\mathcal{A}} = \boldsymbol{f}\big(U^{(t)}_{\text{Joint}}\big) - \boldsymbol{f}\big(\mathsf{RKm}_{U^{(t)}_{\text{Joint}}}(\eta\boldsymbol{Q}^{(t)}_{\text{Joint}})\big) + \sigma\eta^{(t)}\big\langle\nabla_{U^{(t)}_{\text{Joint}}}\boldsymbol{f}, \boldsymbol{Q}^{(t)}_{\text{Joint}}\big\rangle. \tag{6.26}$$

The proposed MiMIC algorithm proceeds to the next iteration only if $C_{\mathcal{A}} \geqslant 0$, else it reduces the step size and checks again. Now, $C_{\mathcal{A}} \geqslant 0$ implies that at each iteration the proposed algorithm satisfies

$$\boldsymbol{f}\big(U^{(t)}_{\text{Joint}}\big) - \boldsymbol{f}\big(\mathsf{RKm}_{U^{(t)}_{\text{Joint}}}(\eta\boldsymbol{Q}^{(t)}_{\text{Joint}})\big) \geqslant -\sigma\eta^{(t)}\,\big\langle\nabla_{U^{(t)}_{\text{Joint}}}\boldsymbol{f},\ \boldsymbol{Q}^{(t)}_{\text{Joint}}\big\rangle.$$

The direction of movement at each iteration is

$$\boldsymbol{Q}^{(t)}_{\text{Joint}} = -\nabla_{U^{(t)}_{\text{Joint}}}\boldsymbol{f}$$

which implies that

$$\big\langle\nabla_{U^{(t)}_{\text{Joint}}}\boldsymbol{f}, \boldsymbol{Q}^{(t)}_{\text{Joint}}\big\rangle = -\|\nabla_{U^{(t)}_{\text{Joint}}}\boldsymbol{f}\|_F^2 < 0, \tag{6.27}$$

where $\|\ .\ \|_F$ deontes the Frobenius norm of a matrix. Thus, the sequence movement directions $\{\boldsymbol{Q}^{(t)}_{\text{Joint}}\}$ is gradient related. Moreover, as $\{\boldsymbol{f}(U^{(t)}_{\text{Joint}})\}$ is a convergent sequence, this implies that the step lengths $\{\eta^{(t)}\}_{t\in\mathcal{T}} \to 0$. As the step lengths $\eta^{(t)}$'s are determined from the Armijo rule, it follows that for all t greater than some $\bar{t}$, $\eta^{(t)} = \beta^{\omega_t}\eta$, where ω_t is an integer greater than zero. Therefore, the update $\frac{\eta^{(t)}}{\beta} = \beta^{(\omega_t-1)}\eta$ does not satisfy the

Armijo condition. So,

$$\boldsymbol{f}\big(U_{\text{Joint}}^{(t)}\big) - \boldsymbol{f}\left(\mathsf{RKm}_{U_{\text{Joint}}^{(t)}}\left(\frac{\eta^{(t)}}{\beta}\boldsymbol{Q}_{\text{Joint}}^{(t)}\right)\right) < -\sigma\frac{\eta^{(t)}}{\beta}\left\langle\nabla_{U_{\text{Joint}}^{(t)}}\boldsymbol{f}, \boldsymbol{Q}_{\text{Joint}}^{(t)}\right\rangle, \quad \forall t \in \tau, t \geqslant \bar{t}. \tag{6.28}$$

Let

$$\widehat{\boldsymbol{Q}}_{\text{Joint}}^{(t)} = \frac{\boldsymbol{Q}_{\text{Joint}}^{(t)}}{\|\boldsymbol{Q}_{\text{Joint}}^{(t)}\|} \quad \text{and} \quad \widehat{\eta}^{(t)} = \frac{\eta^{(t)}\|\boldsymbol{Q}_{\text{Joint}}^{(t)}\|}{\beta}.$$

For the function $\boldsymbol{f}$ over the manifold Km equipped with the retraction RKm, let $\widehat{\boldsymbol{f}} = \boldsymbol{f} \circ \mathsf{RKm}$ denote the pullback of $\boldsymbol{f}$ through RKm. For $U \in \mathsf{Km}$,

$$\widehat{\boldsymbol{f}}_U = \boldsymbol{f} \circ \mathsf{RKm}_U$$

denote the restriction of $\boldsymbol{f}$ to the tangent space $T_U\mathsf{Km}$. Denoting the zero element of tangent space $T_U\mathsf{Km}$ by $\mathbf{0}_U$, the inequality in (6.28) could be written as

$$\frac{\widehat{\boldsymbol{f}}_{U_{\text{Joint}}^{(t)}}\left(\mathbf{0}_{U_{\text{Joint}}^{(t)}}\right) - \widehat{\boldsymbol{f}}_{U_{\text{Joint}}^{(t)}}\left(\widehat{\eta}^{(t)}\widehat{\boldsymbol{Q}}_{\text{Joint}}^{(t)}\right)}{\widehat{\eta}^{(t)}} < -\sigma\left\langle\nabla_{U_{\text{Joint}}^{(t)}}\boldsymbol{f},\ \widehat{\boldsymbol{Q}}_{\text{Joint}}^{(t)}\right\rangle, \tag{6.29}$$

$\forall t \in \tau$, where $t \geqslant \bar{t}$. The mean-value theorem is used to replace the left-hand side of (6.29) by the directional derivative of $\widehat{\boldsymbol{f}}$ at point $U_{\text{Joint}}^{(t)}$ in the direction of $\widehat{\boldsymbol{Q}}_{\text{Joint}}^{(t)}$ (see Chapter 3, [3]). So, for some $c \in [0, \widehat{\eta}^{(t)}]$, (6.29) can be written as

$$-\mathsf{D}\widehat{\boldsymbol{f}}_{U_{\text{Joint}}^{(t)}}\left(c\,\widehat{\boldsymbol{Q}}_{\text{Joint}}^{(t)}\right)\left[\widehat{\boldsymbol{Q}}_{\text{Joint}}^{(t)}\right] < -\sigma\left\langle\nabla_{U_{\text{Joint}}^{(t)}}\boldsymbol{f},\ \widehat{\boldsymbol{Q}}_{\text{Joint}}^{(t)}\right\rangle, \tag{6.30}$$

$\forall t \in \tau$, where $t \geqslant \bar{t}$. Since $\{\eta^{(t)}\}_{t\in\tau} \to 0$ and $\boldsymbol{Q}_{\text{Joint}}^{(t)}$ is gradient-related, hence bounded, it follows that $\{\widehat{\eta}^{(t)}\}_{t\in\tau}$ also tends to 0. Moreover, as $\widehat{\boldsymbol{Q}}_{\text{Joint}}^{(t)}$ has unit norm, the set of unit norm vectors $\{\widehat{\boldsymbol{Q}}_{\text{Joint}}^{(t)}\}$ belongs to a compact set. Every sequence in a compact set converges to an element contained within the set. So, there must exist a index set $\widehat{\tau} \subset \tau$ such that $\{\widehat{\boldsymbol{Q}}_{\text{Joint}}^{(t)}\}_{t\in\widehat{\tau}} \to \widehat{\boldsymbol{Q}}_{\text{Joint}}^{\star}$ for some $\widehat{\boldsymbol{Q}}_{\text{Joint}}^{\star}$ having $\|\widehat{\boldsymbol{Q}}_{\text{Joint}}^{\star}\| = 1$. Taking limits in (6.30) over $\widehat{\tau}$, $\widehat{\eta}^{(t)} \to 0$, which implies that $c \to 0$ and $\widehat{\boldsymbol{Q}}_{\text{Joint}}^{(t)} \to \widehat{\boldsymbol{Q}}_{\text{Joint}}^{\star}$. Also, $\boldsymbol{f}$ is a continuous and differentiable scalar field over the Riemannian manifold Km. Therefore, from the definition of directional derivative D (see (3.31) in Chapter 3, [3]), it satisfies that

$$\mathsf{D}\widehat{\boldsymbol{f}}_{U_{\text{Joint}}^{(t)}}(\mathbf{0})\left[\widehat{\boldsymbol{Q}}_{\text{Joint}}^{(t)}\right] = \left\langle\nabla_{U_{\text{Joint}}^{(t)}}\boldsymbol{f}, \widehat{\boldsymbol{Q}}_{\text{Joint}}^{(t)}\right\rangle.$$

Taking limits, (6.30) becomes

$$-\left\langle\nabla_{U_{\text{Joint}}^{\star}}\boldsymbol{f}, \widehat{\boldsymbol{Q}}_{\text{Joint}}^{\star}\right\rangle < -\sigma\left\langle\nabla_{U_{\text{Joint}}^{\star}}\boldsymbol{f}, \widehat{\boldsymbol{Q}}_{\text{Joint}}^{\star}\right\rangle. \tag{6.31}$$

Since $0 < \sigma < 1$, it follows from (6.31) that

$$\left\langle \nabla_{U^\star_{\text{Joint}}} \boldsymbol{f}, \widehat{\boldsymbol{Q}}^\star_{\text{Joint}} \right\rangle > 0.$$

However, as $\{\boldsymbol{Q}^{(t)}_{\text{Joint}}\}$ is gradient related, therefore $\left\langle \nabla_{U^\star_{\text{Joint}}} \boldsymbol{f}, \widehat{\boldsymbol{Q}}^\star_{\text{Joint}} \right\rangle < 0$ (from (6.27)), which is a contradiction. Therefore, the subsequence of iterates $\{U^{(t)}_{\text{Joint}}\}_{t \in \mathcal{T}}$ converges to some critical point of the objective function $\boldsymbol{f}$. ■

Theorem 6.3 states that only critical points of the cost function $\boldsymbol{f}$ can be accumulation points of sequences $\{U^{(t)}_{\text{Joint}}\}$ generated by the MiMIC algorithm. It does not specify whether the accumulation points are local minimizers, local maximizers, or saddle points. Nevertheless, at each iteration, since the movement is always in the direction of negative gradient, unless the initial point $U^{(0)}_{\text{Joint}}$ is carefully crafted, Algorithm 6.3 produces sequences whose accumulation points are local minima of the cost function.

6.4.2 Asymptotic Analysis

The asymptotic convergence describes how fast the sequence of iterates generated by a search algorithm could arrive to an optimal solution, if exists. For a sufficiently large value of iteration number t, the properties of cost function $\boldsymbol{f}$ and the line-search nature of Algorithm 6.3 are used to upper bound the difference between the cost function at $U^{(t+1)}_{\text{Joint}}$ and at the optimal solution $U^\star_{\text{Joint}}$ in terms of the difference when evaluated at $U^{(t)}_{\text{Joint}}$ and $U^\star_{\text{Joint}}$. The result invokes the smallest and largest eigenvalues of the Hessian of $\boldsymbol{f}$ at the critical point.

Let $\{U^{(t)}_{\text{Joint}}\}_{t=0,1,2,\ldots}$ be an infinite sequence of iterates generated by the proposed Algorithm 6.3, for a set of given $\{U_j\}_{j=1}^M$. With the direction of movement being $\mathbf{Q}^{(t)}_{\text{Joint}} := -\nabla \boldsymbol{f}_{\text{Joint}^{(t)}}$, let the sequence $\{U^{(t)}_{\text{Joint}}\}_{t=0,1,\ldots}$ converge to a point $U^\star_{\text{Joint}}$, which is a critical point of $\boldsymbol{f}$ according to Theorem 6.3. Let the Hessian of the cost function at the converged solution be denoted by $\mathbf{H}_{U^\star_{\text{Joint}}} \boldsymbol{f}$, and $\lambda_{\mathbf{H},\min}$ and $\lambda_{\mathbf{H},\max}$ be the smallest and largest eigenvalues of the Hessian of $\mathbf{H}_{U^\star_{\text{Joint}}} \boldsymbol{f}$. Assume that $\lambda_{\mathbf{H},\min} > 0$ (hence $U^\star_{\text{Joint}}$ is a local minimizer of $\boldsymbol{f}$). The asymptotic bound is given by the following theorem.

Theorem 6.4. *There exists an integer $t' \geqslant 0$ such that*

$$\boldsymbol{f}\left(U^{(t+1)}_{\text{Joint}}\right) - \boldsymbol{f}\left(U^\star_{\text{Joint}}\right) \leqslant c\left(\boldsymbol{f}\left(U^{(t)}_{\text{Joint}}\right) - \boldsymbol{f}\left(U^\star_{\text{Joint}}\right)\right),$$

for all $t \geqslant t'$, where

$$c = 1 - 2\sigma\lambda_{\mathbf{H},\min} \min\left(\eta, \frac{2\beta(1-\sigma)}{\lambda_{\mathbf{H},\max}}\right), \tag{6.32}$$

where η is the step length, and σ and β are Armijo criterion parameters.

Proof. Let $(\mathcal{U}, \varphi)$ be a chart of the manifold $\mathcal{M} := \mathsf{Km}(n,k)$, with $U^\star_{\text{Joint}} \in \mathcal{U}$. Let the negative gradient of $\boldsymbol{f}$ at any point $U \in \mathcal{M}$ be given by $\zeta_U := -\nabla \boldsymbol{f}(U)$, where ζ_U belongs to the tangent space $T_U\mathcal{M}$. Let coordinate expressions for different elements in the corresponding Euclidean space $\Re^{n\times k}$ be denoted with a hat. The following notations are used for Euclidean space representations.

$\hat{U} := \varphi(U)$ $\rhd$ indicates that coordinate map $\hat{U}$ in $\Re^{n\times k}$ is equal to φ of U in $\mathcal{M}$,

$\hat{\mathcal{U}} := \varphi(\mathcal{U})$ $\rhd$ similar to above notation, but for the whole set $\mathcal{U}$,

$\hat{\boldsymbol{f}}(\hat{U}) := \boldsymbol{f}(U)$ $\rhd$ indicates that the value of $\hat{f}$ at $\hat{U} \in \Re^{n\times k}$ is equal to the value of f at $U \in \mathcal{M}$,

$\hat{\zeta}_{\hat{U}} := \mathsf{D}\varphi(U)[\zeta_U]$ $\rhd$ $\hat{\zeta}_{\hat{U}}$ is the coordinate expression corresponding to the directional derivative in manifold $\mathcal{M}$,

$\hat{\mathsf{R}}_{\hat{U}}(\hat{\zeta}) := \varphi(\mathsf{R}_U(\zeta))$ $\rhd$ the coordinate expression for the retracted point in $\Re^{n\times k}$ is given by the mapping φ of the retracted point $\mathsf{R}_U(\zeta)$ in $\mathcal{M}$.

Let $y_{\hat{U}}$ denote the Euclidean gradient of $\hat{\boldsymbol{f}}$ at $\hat{U}$, given by

$$y_{\hat{U}} = \begin{bmatrix} \partial_{11}\hat{\boldsymbol{f}}(\hat{U}) & \ldots & \partial_{1k}\hat{\boldsymbol{f}}(\hat{U}) \\ \partial_{21}\hat{\boldsymbol{f}}(\hat{U}) & \ldots & \partial_{2k}\hat{\boldsymbol{f}}(\hat{U}) \\ & \ldots & \\ \partial_{n1}\hat{\boldsymbol{f}}(\hat{U}) & \ldots & \partial_{nk}\hat{\boldsymbol{f}}(\hat{U}) \end{bmatrix}_{n\times k} \tag{6.33}$$

Let $G_{\hat{U}}$ denote the matrix representation of the Riemannian metric g of $\mathcal{M}$, in the coordinate space. Without loss of generality, we assume that the coordinate map of the critical point is $\hat{U}^{\star}_{\text{Joint}} = \mathbf{0}$ (the zero vector) and $G_{\hat{U}^{\star}_{\text{Joint}}} = \mathbf{I}_n$.

The main aim is to obtain, at a current iterate U, a suitable upper bound on $\boldsymbol{f}(\mathsf{R}_U(t^A\zeta_U))$, where t^A is the Armijo step and $t^A\zeta_U$ is the Armijo point in tangent space $\mathcal{T}_U\mathcal{M}$. The

Armijo condition implies that

$$\begin{aligned}\boldsymbol{f}(U) - \boldsymbol{f}(\mathsf{R}_U(t^A\zeta_U)) &\geqslant -\sigma\left\langle \nabla \boldsymbol{f}(U), t^A\zeta_U\right\rangle,\\ \Rightarrow \boldsymbol{f}(\mathsf{R}_U(t^A\zeta_U)) &\leqslant \boldsymbol{f}(U) - \sigma\left\langle \zeta_U, t^A\zeta_U\right\rangle\\ &\leqslant \boldsymbol{f}(U) - \sigma t^A\left\langle \zeta_U, \zeta_U\right\rangle. \end{aligned} \tag{6.34}$$

First a lower bound is obtained on $\langle\zeta_U, \zeta_U\rangle$ in terms of $\boldsymbol{f}(U)$. Given a smooth scalar field $\boldsymbol{f}$ on Riemannian manifold $\mathcal{M}$, ζ_U denotes the negative gradient of $\boldsymbol{f}$ at U, given by $\zeta_U := -\nabla\boldsymbol{f}(U)$. The coordinate expression for ζ_U in $\Re^{n\times k}$ is given in terms of the Euclidean gradient $y_{\hat{U}}$ and the matrix representation of Riemannian metric G as follows (Section 3.6 in [3]):

$$\hat{\zeta}_{\hat{U}} = G_{\hat{U}}^{-1}(-y_{\hat{U}}).$$

Also, from (3.29) in [3],

$$\langle\zeta_U, \zeta_U\rangle \ = \hat{\zeta}_{\hat{U}} G_{\hat{U}} \hat{\zeta}_{\hat{U}} = y_{\hat{U}} G_{\hat{U}}^{-1} y_{\hat{U}} = \parallel y_{\hat{U}} \parallel^2 \left(1 + \mathcal{O}(\hat{U})\right), \tag{6.35}$$

as $G_{\hat{U}}$ is assumed to be the identity matrix at the critical point $\hat{U}^{\star}_{\text{Joint}}$. From Taylor expansion of the Euclidean gradient $y_{\hat{U}}$, we have

$$\begin{aligned}\nabla\hat{\boldsymbol{f}}(\hat{U}^{\star}_{\text{Joint}} + \hat{U}) &= \nabla\hat{\boldsymbol{f}}(\hat{U}^{\star}_{\text{Joint}}) + \mathbf{H}_{\hat{U}^{\star}_{\text{Joint}}}\ \hat{U} + \mathcal{O}(\parallel \hat{U} \parallel^2),\\ \Rightarrow y_{\hat{U}} = \nabla\hat{\boldsymbol{f}}(\hat{U}) &= \mathbf{H_0}\ \hat{U} + \mathcal{O}(\parallel \hat{U} \parallel^2)\end{aligned} \tag{6.36}$$

(as $\hat{U}^{\star}_{\text{Joint}} = \mathbf{0}$, so $\nabla\hat{\boldsymbol{f}}(\hat{U}^{\star}_{\text{Joint}}) = 0$, and from (6.33))

On the other hand, from the Taylor expansion of $\hat{\boldsymbol{f}}$, we have

$$\begin{aligned}\hat{\boldsymbol{f}}(\hat{U}^{\star}_{\text{Joint}} + \hat{U}) &= \hat{\boldsymbol{f}}(\hat{U}^{\star}_{\text{Joint}}) + \left(\nabla\hat{\boldsymbol{f}}(\hat{U}^{\star}_{\text{Joint}})\right)^T \hat{U} + \frac{1}{2}\hat{U}^T\ \mathbf{H}_{\hat{U}^{\star}_{\text{Joint}}}\ \hat{U} + \mathcal{O}(\parallel \hat{U} \parallel^3),\\ \Rightarrow \hat{\boldsymbol{f}}(\hat{U}) &= \hat{\boldsymbol{f}}(\mathbf{0}) + \frac{1}{2}\hat{U}^T\ \mathbf{H_0}\ \hat{U} + \mathcal{O}(\parallel \hat{U} \parallel^3).\end{aligned} \tag{6.37}$$

(applying $\hat{U}^{\star}_{\text{Joint}} = \mathbf{0}$ and $\nabla\hat{\boldsymbol{f}}(\hat{U}^{\star}_{\text{Joint}}) = 0$)

It follows from (6.36) and (6.37) that

$$\begin{aligned}\hat{\boldsymbol{f}}(\hat{U}) - \hat{\boldsymbol{f}}(\mathbf{0}) &= \frac{1}{2} y_{\hat{U}}^T\ \mathbf{H_0}^{-1}\ y_{\hat{U}} + \mathcal{O}(\parallel \hat{U} \parallel^3)\\ &\leqslant \frac{1}{2}\frac{1}{\lambda_{\mathbf{H},\min}} \parallel y_{\hat{U}} \parallel,\end{aligned} \tag{6.38}$$

holds for all $\hat{U}$ sufficiently close to $\hat{U}^{\star}_{\text{Joint}}$. This is because, in (6.38) above, $\lambda_{\mathbf{H},\min}$ denotes the minimum eigenvalue of the Hessian of $\hat{\boldsymbol{f}}$ at $\hat{U}^{\star}_{\text{Joint}}$, that is $\mathbf{H_0}$, and from the properties of eigenvalues, we have that, for any vector v, $v^T\mathbf{H_0}^{-1}v \leqslant (\lambda_{\mathbf{H},\min})^{-1}$. Therefore, from (6.35) and (6.38), it can be concluded that

$$\begin{aligned}\boldsymbol{f}(U) - \boldsymbol{f}(U^{\star}_{\text{Joint}}) &\leqslant \frac{1}{2}\frac{1}{\lambda_{H,\min}}\langle\zeta_U, \zeta_U\rangle,\\ \Rightarrow 2\lambda_{\mathbf{H},\min}\left(\boldsymbol{f}(U) - \boldsymbol{f}(U^{\star}_{\text{Joint}})\right) &\leqslant \langle\zeta_U, \zeta_U\rangle.\end{aligned} \tag{6.39}$$

Thus, (6.39) gives the desired lower bound on $\langle\zeta_U, \zeta_U\rangle$. Using the bound (6.39) in the Armijo condition (6.34) gives us that

$$\begin{aligned}&\boldsymbol{f}(\mathsf{R}_U(t^A\zeta_U)) \leqslant \boldsymbol{f}(U) - \sigma t^A 2\lambda_{\mathbf{H},\min}\left(\boldsymbol{f}(U) - \boldsymbol{f}(U^{\star}_{\text{Joint}})\right),\\ &\Rightarrow f(\mathsf{R}_x(t^A\zeta_U)) - \boldsymbol{f}(U^{\star}_{\text{Joint}}) \leqslant d\left(1 - 2\lambda_{\mathbf{H},\min}\sigma t^A\right)\left(\boldsymbol{f}(U) - \boldsymbol{f}(U^{\star}_{\text{Joint}})\right).\end{aligned} \tag{6.40}$$

Next a lower bound is obtained on the Armijo step size t^A to substitute in (6.40). Using the retraction operator R and the of negative gradient ζ_U, we can define a smooth curve on the manifold, from $\Re$ to $\mathcal{M}$, given by $t \to \mathsf{R}_U(t\zeta_U)$. This mapping can be further used to define a smooth function h on $\mathcal{M}$ from $\Re$ to $\Re$ with a well-defined classical derivative, given by

$$h_U(t) = \boldsymbol{f}\left(\mathsf{R}_U(t\zeta_U)\right). \tag{6.41}$$

The derivative of h_U is given by (Sections 3.5.1, 3.5.2, and 3.6 of [3])

$$\begin{aligned}\dot{h}_U(t=0) &= \frac{\mathsf{d}}{\mathsf{d}t}\boldsymbol{f}\left(\mathsf{R}_U(t\zeta_U)\right)\Big|_{t=0} = \mathsf{D}\boldsymbol{f}(U)[-\zeta_U]\\ &= \langle\nabla\boldsymbol{f}(U), \zeta_U\rangle = -\langle\zeta_U, \zeta_U\rangle.\end{aligned} \tag{6.42}$$

Using (6.41) and (6.42) the Armijo condition (6.34) reads

$$h_U(t^A) \leqslant h_U(0) + \sigma t^A \dot{h}_U(0). \tag{6.43}$$

The Taylor expansion of h_U gives us that

$$h_U(t) = h_U(0) + t\dot{h}_U(0) + t^2\frac{\ddot{h}_U(0)}{2}.$$

The t at which the left- and right-hand sides of (6.43) are equal is given by

$$\begin{aligned}h_U(0) + t\dot{h}_U(0) + t^2\frac{\ddot{h}_U(0)}{2} &= h_U(0) + \sigma t\dot{h}_U(0),\\ \Rightarrow t\frac{\ddot{h}_U(0)}{2} = -\dot{h}_U(0) + \sigma\dot{h}_U(0), &\qquad \Rightarrow t = \frac{-2(1-\sigma)\dot{h}_U(0)}{\ddot{h}_U(0)}.\end{aligned} \tag{6.44}$$

Using t in (6.44) and the definition of Armijo point (Definition D.3 and Section 4.2 of [3]), the step size t^A that satisfies (6.40) has the following lower bound

$$t^A \geqslant \min\left(\eta, \frac{-2\beta(1-\sigma)\dot{h}_U(0)}{\ddot{h}_U(0)}\right), \tag{6.45}$$

where $\bar{\eta}$ and β are Armijo step size parameters. The second derivative $\ddot{h}_u$ is given by

$$\begin{aligned} \ddot{h}_U(t=0) &= \frac{\mathsf{d}^2}{\mathsf{d}t^2} f\left(\mathsf{R}_U(t\zeta_U)\right)\Big|_{t=0} = \mathsf{D}^2 f(x)[-\zeta_U] \\ &= (-\zeta_U)^T \mathbf{H}_0(-\zeta_U) = \mathbf{H_0} \parallel \zeta_U \parallel^2 . \end{aligned} \tag{6.46}$$

From properties of eigenvalues, we have that for any vector v, $v^T\mathbf{H_0}v \leqslant \lambda_{\mathbf{H},\max}$. Therefore, using (6.42) and (6.46) in (6.45) gives that

$$t^A \geqslant \min\left(\eta, \frac{2\beta(1-\sigma)}{\lambda_{\mathbf{H},\max}}\right), \tag{6.47}$$

for all U sufficiently close to $U^\star_{\text{Joint}}$. Using the lower bound (6.47) in (6.40) gives

$$\boldsymbol{f}(\mathsf{R}_U(t^A\zeta_U)) - \boldsymbol{f}(U^\star_{\text{Joint}}) \leqslant c\left(\boldsymbol{f}(U) - f(U^\star_{\text{Joint}})\right) \tag{6.48}$$

where

$$c = 1 - 2\sigma\lambda_{\mathbf{H},\min}\min\left(\eta, \frac{2\beta(1-\sigma)}{\lambda_{\mathbf{H},\max}}\right). \tag{6.49}$$

In (6.48), t^A is the Armio step size corresponding to the Armijo point. When the next iterate $U^{(t+1)}_{\text{Joint}}$ is the Armio point, then the decrease in the value of the objective function from $U^{(t)}_{\text{Joint}}$ to $U^{(t+1)}_{\text{Joint}}$ is σ times the directional derivative at $U^{(t)}_{\text{Joint}}$. In Algorithm 6.3, the next iterate is

$$U^{(t+1)}_{\text{Joint}} = \mathsf{R}_{U^{(t)}_{\text{Joint}}}\left(t\zeta_{U^{(t+1)}_{\text{Joint}}}\right), \tag{6.50}$$

where t satisfies the Armijo condition, that is, with step length t, the decrease in the value of the objective function is greater than or equal to σ times the directional derivative at $U^{(t)}_{\text{Joint}}$. Hence using (6.50) in (6.48), we get

$$\boldsymbol{f}(U^{(t+1)}_{\text{Joint}}) - \boldsymbol{f}(U^\star_{\text{Joint}}) \leqslant c\left(\boldsymbol{f}(U^{(t)}_{\text{Joint}}) - \boldsymbol{f}(U^\star_{\text{Joint}})\right),$$

where c is given by (6.49). ■

Let the rank r of subspaces U_{Joint} and U_js be set to k, the number of clusters in the data set. Given a set of $\{U_j\}_{j=1}^M$, ignoring the constant factors, the objective function $\boldsymbol{f}$ in terms of U_{Joint} is given by

$$\boldsymbol{f}(U_{\text{Joint}}) = -tr\left(U_{\text{Joint}}^T\left(\mathbf{L}_{\text{Joint}}^k + \frac{1}{M}\sum_{j=1}^{M} U_j U_j^T\right)U_{\text{Joint}}\right) + \xi \parallel U_{\text{Joint}-} \parallel_F^2$$

The cost function is equivalent to the Rayleigh quotient function, given by

$$\begin{aligned} \boldsymbol{f}(U_{\text{Joint}}) &= tr\left(U_{\text{Joint}}^T \boldsymbol{\Xi} U_{\text{Joint}}\right) \\ \text{where } \boldsymbol{\Xi} &= \left(\xi \mathbf{I}_n - \mathbf{L}_{\text{Joint}}^k - \frac{1}{M}\sum_{j=1}^{M} U_j U_j^T\right). \end{aligned} \tag{6.51}$$

Let $\lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_n$ be the eigenvalues of $\boldsymbol{\Xi}$. The extreme eigenvalues of the Hessian $\mathbf{H}_{U_{\text{Joint}}^\star}\boldsymbol{f}$ is given by (Section 4.9 of [3])

$$\lambda_{\mathbf{H},\min} = \lambda_{k+1} - \lambda_k \text{ and } \quad \lambda_{\mathbf{H},\max} = \lambda_n - \lambda_1. \tag{6.52}$$

Using (7.31) in (6.32), the convergence bound states that for all t greater than some t'

$$\boldsymbol{f}\left(U_{\text{Joint}}^{(t+1)}\right) - \boldsymbol{f}\left(U_{\text{Joint}}^\star\right) \leqslant c\left(\boldsymbol{f}\left(U_{\text{Joint}}^{(t)}\right) - \boldsymbol{f}\left(U_{\text{Joint}}^\star\right)\right),$$

where c is convergence factor given by

$$c = 1 - 2\sigma(\lambda_{k+1} - \lambda_k) \min\left\{\eta, \frac{2\beta(1-\sigma)}{(\lambda_n - \lambda_1)}\right\}. \tag{6.53}$$

The convergence factor c determines how fast the proposed algorithm converges to an optimal solution of a given data set. A smaller value of c indicates greater decrease in value of the cost function from iteration t to $(t+1)$ while c close to 1 indicates minimal decrease. Also, matrix $\boldsymbol{\Xi}$ has form equivalent to that of the normalized graph Laplacian. For a data set with k well-separated clusters, the matrix $\boldsymbol{\Xi}$ tends to have close to block diagonal structure and each of the k smallest eigenvalues of $\boldsymbol{\Xi}$ is indicative of one of the k clusters. In this case, it is expected to have a greater gap between the eigenvalues λ_k and λ_{k+1}, which leads to a smaller value of c, indicating faster convergence. For poorly separated clusters, the difference $(\lambda_{k+1} - \lambda_k)$ tends to be very small and c is close to 1, indicating longer time to reach the optimal solution. Hence, c can be used to infer about the cluster structure of the data set.

6.5 Experimental Results and Discussion

In this work, experiments are conducted to study and compare the performance of the proposed MiMIC algorithm on several real-world and synthetic data sets. The clustering performance of the MiMIC algorithm is compared with that of eight multi-view clustering algorithms on five benchmark data sets, and nine cancer subtype identification algorithms on eight multi-omics data sets. The performance of different algorithms is evaluated using six external cluster evaluation indices, namely, accuracy, adjusted rand index (ARI), normalized mutual information (NMI), F-measure, Rand index, and purity, which compare the identified clusters with the clinically established cancer subtypes and the ground truth class information of the benchmark data sets.

In order to randomize the experiments, each algorithm is executed 10 times, and the means and standard deviations of each measure are reported. In all the tables, the numbers within parentheses are the standard deviations, $\rightarrow 0$ means that the value is close to zero (approximately $1e-17$), while 0.00 denotes exactly zero. For the proposed MiMIC algorithm, the step lengths η_{K} and η_{S} are set to 0.05. The value of convergence parameter ϵ is empirically set to 0.001 for benchmark data sets and 0.005 for omics data sets. The step reduction factor δ is set to 0.5, and ξ of (6.9) is set to 0.01. The R implementation of the proposed algorithm is available at https://github.com/Aparajita-K/MiMIC.

6.5.1 Description of Data Sets

In this work, experiments are performed on three different types of data sets as follows:

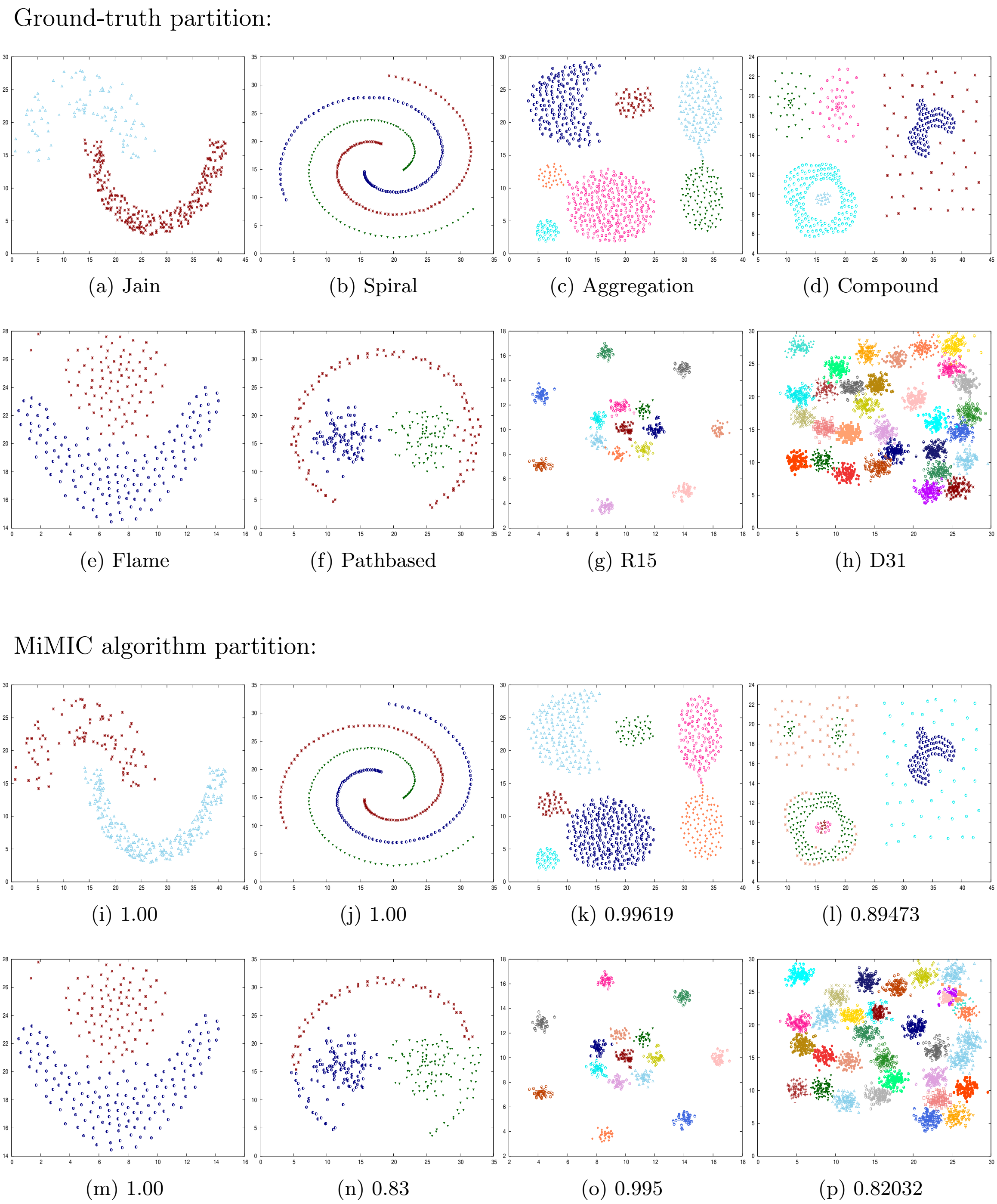
6.5.1.1 Synthetic Data Sets

Experiments are conducted on five two-dimensional shape data sets (http://cs.joensuu.fi/sipu/datasets/) to give a visual illustration of the capability of the proposed MiMIC algorithm. The data sets are Jain, Sipral, Aggregation, Compound, D31, Pathbased, Flame, and R15 consisting of 373, 312, 788, 399, 3100, 300, 240, and 600 samples, respectively, and number of clusters varying between 2 and 31. These are single view data sets for which two views are generated from two graphs constructed using k-nearest neighbors and Gaussian kernel.

6.5.1.2 Benchmark Data Sets

In this work, several publicly available data sets from a variety of application domains are considered. Among them, 3Sources (http://mlg.ucd.ie/datasets/3sources.html) and BBC (http://mlg.ucd.ie/datasets/segment.html) are multi-source news article clustering data sets, consisting of 169 and 685 news articles, annotated with 6 and 5 topics, respectively. Five benchmark image data sets are also considered, namely, Digits, 100Leaves, ALOI, ORL, and Caltech7. The Digits data set (https://archive.ics.uci.edu/ml/datasets/Multiple+Features) consists of 2000 images of handwritten numerals ('0'–'9'). The ALOI (http://elki.dbs.ifi.lmu.de/wiki/DataSets/MultiView) and 100Leaves[1] data sets are both 100 cluster data sets, where ALOI consists of 11,025 images of 100 small objects, while 100Leaves consists of 1,600 samples from 100 plant species. The ORL data set (https://cam-orl.co.uk/facedatabase.html) consists of 400 facial images of 40 subjects taken under varying lighting conditions and facial expressions. The Caltech7 data set (https://github.com/yeqinglee/mvdata) is a seven class object recognition data set. Apart from these data sets, six multi-view social network data sets, namely, Football, Olympics, Politics-UK, Politics-IE, Rugby, and CORA are also considered in this study.

[1] https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set



Ground-truth partition:

(a) Jain

(b) Spiral

(c) Aggregation

(d) Compound

(e) Flame

(f) Pathbased

(g) R15

(h) D31

MiMIC algorithm partition:

(i) 1.00

(j) 1.00

(k) 0.99619

(l) 0.89473

(m) 1.00

(n) 0.83

(o) 0.995

(p) 0.82032

Figure 6.2: Two-dimensional scatter plots of three synthetic shape data sets: ground truth clustering (top two rows: (a)-(h)) and MiMIC clustering (bottom two rows: (i)-(p)). The numbers in (i)-(p) denote the clustering accuracy obtained using the MiMIC algorithm.

Table 6.1: Performance Analysis of Proposed Algorithms on Synthetic Clustering Data Sets

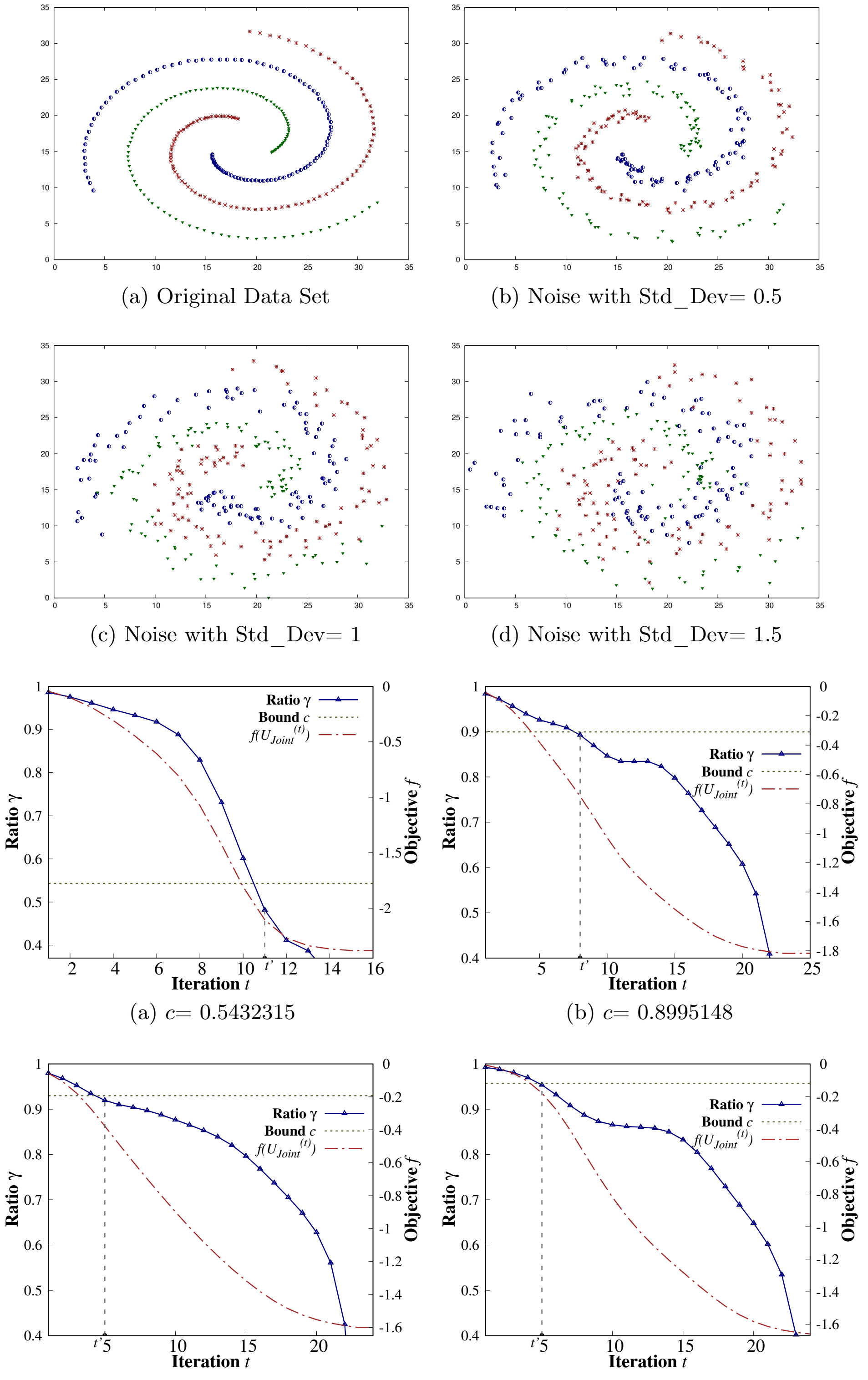
Data Sets→	**Aggregation**	**Compound**	**Pathbased**	**Spiral**	**Jain**	**Flame**	**R15**	**D31**
No. of Samples	788	399	300	312	373	240	600	3100
No. of Clusters	7	6	3	3	2	2	15	31
Accuracy	0.99619	0.89473	0.83000	1.00	1.00	0.98750	0.99500	0.82032
NMI	0.98839	0.89671	0.63926	1.00	1.00	0.89905	0.99135	0.91780
ARI	0.99198	0.92926	0.58486	1.00	1.00	0.95014	0.98921	0.68092
F-measure	0.99622	0.91264	0.81500	1.00	1.00	0.98748	0.99497	0.85341
Rand	0.99728	0.97343	0.81190	1.00	1.00	0.97520	0.99868	0.97419
Purity	0.99619	0.89724	0.83000	1.00	1.00	0.98750	0.99500	0.85096

6.5.1.3 Multi-Omics Cancer Data Sets

The subtype analysis is studied on eight types of cancers, namely, lower grade glioma (LGG), stomach adenocarcinoma (STAD), breast adenocarcinoma (BRCA), lung carcinoma (LUNG), ovarian carcinoma (OV), cervical carcinoma (CESC), colorectal carcinoma (CRC), and kidney carcinoma (KIDNEY), and corresponding data sets are obtained from The Cancer Genome Atlas (TCGA) (`https://cancergenome.nih.gov/`). The colorectal, lung and kidney cancer data sets have two, two, and three histological subtypes, respectively, identified by World Health Organization. For other five cancers, TCGA research network has identified four clinically relevant subtypes for both BRCA, STAD, and OV while three subtypes for LGG and CESC. For all the cancer data sets, four different omic data types, namely, DNA methylation (mDNA), gene expression (RNA), microRNA expression (miRNA), and reverse phase protein array expression (RPPA), are considered. All the real-world data sets are summarized in Table A.1 and are briefly described in Appendix A. For the data sets with feature vector based representation, the pairwise similarity matrices W_m's are computed using the Gaussian kernel.

6.5.2 Performance on Synthetic Data Sets

Figure 6.2 shows the scatter plots for five two dimensional shape data sets. The objects in Figure 6.2 are colored according to their ground truth partition information (top two rows: Figures6.2(a)-(h)) the partition obtained by the proposed MiMIC algorithm in (bottom two rows: Figure 6.2(i)-(p)).For these data sets, two views, generated using the Gaussian kernel and k-nearest neighbors, have related cluster structure, but they differ in their graph connectivity. The quantitative results on the synthetic data sets, in terms of the external indices, are reported in Table 6.1. The qualitative results in Figure 6.2 show that the MiMIC algorithm obtains almost perfect clustering for the Jain, Spiral, Aggregation, Flame, and R15 data sets. For the Compound, D31, and Pathbased data sets, the clustering performance is also very good, having accuracy 0.89473, 0.82032, 0.83, respectively. The scatter plots in Figure 6.2 show that the Spiral, Compound, Jain, and Pathbased data sets have non-linearly separable clusters, while the D31 data set has 3,100 samples and 31 clusters. All the results reported in Figure 6.2 show that the proposed algorithm can efficiently identify both non-linearly separable and large number of clusters.



(a) Original Data Set

(b) Noise with Std_Dev= 0.5

(c) Noise with Std_Dev= 1

(d) Noise with Std_Dev= 1.5

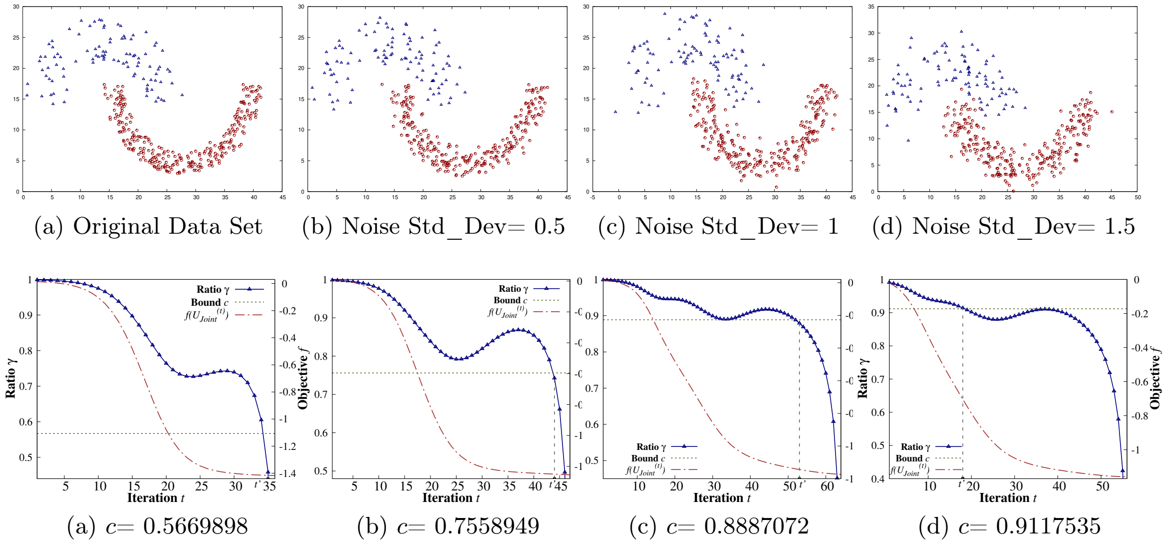
(a) c= 0.5432315

(b) c= 0.8995148

(c) c= 0.9298075

(d) c= 0.9569399

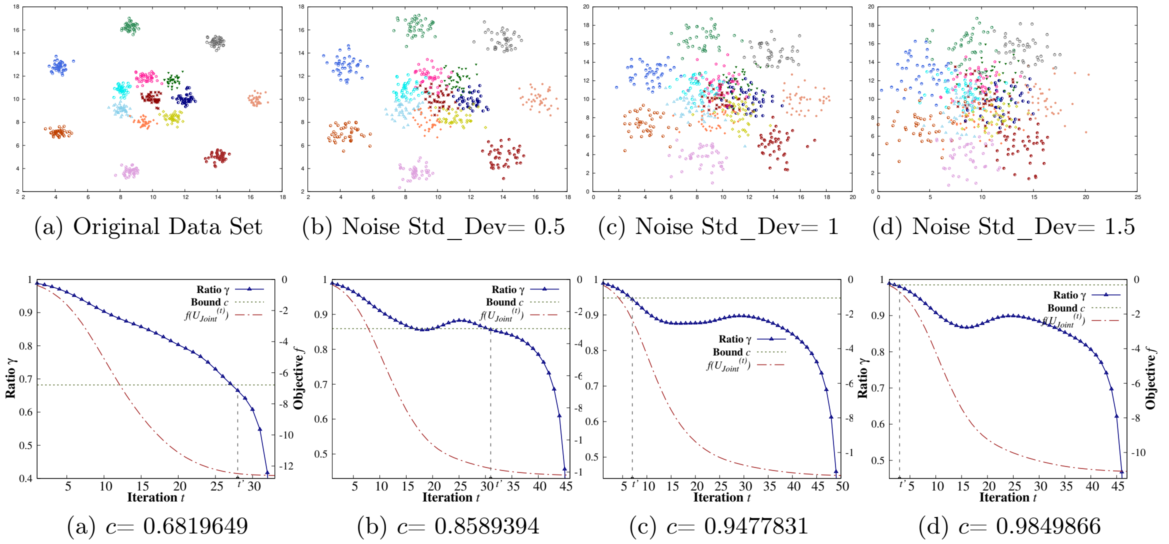
Figure 6.3: Asymptotic convergence analysis for Spiral data set: scatter plot of data with varying Gaussian noise (top row) and variation of convergence ratio and objective function with increase in iteration number t (bottom row).



(a) Original Data Set (b) Noise Std_Dev= 0.5 (c) Noise Std_Dev= 1 (d) Noise Std_Dev= 1.5

(a) c= 0.5669898 (b) c= 0.7558949 (c) c= 0.8887072 (d) c= 0.9117535

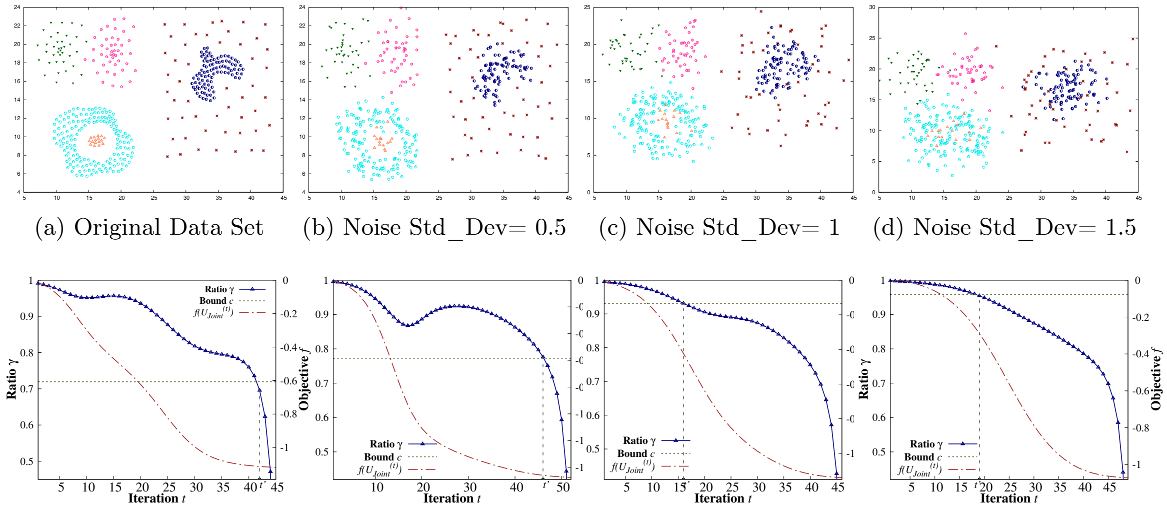
Figure 6.4: Asymptotic convergence analysis for Jain data set: scatter plot of data with varying Gaussian noise (top row) and variation of convergence ratio and objective function with increase in iteration number t (bottom row).



(a) Original Data Set (b) Noise Std_Dev= 0.5 (c) Noise Std_Dev= 1 (d) Noise Std_Dev= 1.5

(a) c= 0.6819649 (b) c= 0.8589394 (c) c= 0.9477831 (d) c= 0.9849866

Figure 6.5: Asymptotic convergence analysis for R15 data set: scatter plot of data with varying Gaussian noise (top row) and variation of convergence ratio and objective function with increase in iteration number t (bottom row).



(a) Original Data Set (b) Noise Std_Dev= 0.5 (c) Noise Std_Dev= 1 (d) Noise Std_Dev= 1.5

(a) c= 0.7194825 (b) c= 0.7725400 (c) c= 0.9311562 (d) c= 0.9583102

Figure 6.6: Asymptotic convergence analysis for Compound data set: scatter plot of data with varying Gaussian noise (top row) and variation of convergence ratio and objective function with increase in iteration number t (bottom row).

6.5.3 Significance of Asymptotic Convergence Bound

The asymptotic convergence bound obtained in Theorem 6.4 indicates how fast the sequence of iterates generated by the proposed algorithm converges to an optimal solution of a given data set. For a sufficiently large value of iteration number t, Theorem 6.4 bounds the difference between the cost function $\boldsymbol{f}$ evaluated at $U_{\text{Joint}}^{(t+1)}$ and at the optimal solution $U_{\text{Joint}}^{\star}$ in terms of the difference between that evaluated at $U_{\text{Joint}}^{(t)}$ and $U_{\text{Joint}}^{\star}$. Let γ_t be given by the ratio

$$\gamma_t = \frac{\boldsymbol{f}\left(U_{\text{Joint}}^{(t+1)}\right) - \boldsymbol{f}\left(U_{\text{Joint}}^{\star}\right)}{\boldsymbol{f}\left(U_{\text{Joint}}^{(t)}\right) - \boldsymbol{f}\left(U_{\text{Joint}}^{\star}\right)}. \tag{6.54}$$

Theorem 6.4 states that for all t greater or equal to some t', $\gamma_t \leqslant c$, where c is given by (6.32). The convergence factor c can be used to make inference about the underlying cluster structure of the data set. As discussed in Section 6.4.2, a value of c close to 1 indicates poor separation between the clusters present in the data set, while a value much lower than 1 indicates well-separated clusters. To experimentally establish this, multiple noisy data sets are generated from the synthetic shape data sets used in this work, by adding Gaussian noise of mean 0 and standard deviations 0.5, 1, and 1.5. Experiments are performed on noise-free and noisy variations of four shape data sets, namely, Spiral, Jain, R15, and Compound. The scatter plots for the noise-free and noisy variants of Spiral, Jain, R15, and Compound data sets are provided in the top rows of Figures 6.3, 6.4, 6.5, and 6.6, respectively. As stated in Section 6.5.1.1, for each variant of each data set, two views are generated using k-nearest neighbors and Gaussian kernel. Starting from a random initial iterate, the variation of γ_t and the cost function $\boldsymbol{f}\left(U_{\text{Joint}}^{(t)}\right)$ is observed for different values of $t = 1, 2, 3, \ldots$, until convergence. The variation of γ_t and $\boldsymbol{f}\left(U_{\text{Joint}}^{(t)}\right)$ along with the corresponding value of convergence factor c is provided in the bottom rows of Figures 6.3, 6.4, 6.5, and 6.6 for Spiral, Jain, R15, and Compound data sets, respectively. The value of the bound c is marked by a horizontal dashed green line in these figures.

For all the data sets, the top rows of Figures 6.3, 6.4, 6.5, and 6.6 show that the cluster structure and their separability degrades with the increase in noise, as expected. The bottom rows of these figures in turn show that with increase in noise in the data sets, the value of the convergence factor c increases and goes close to 1. For instance, for the Spiral data set, the value of c for the noise-free original data set in Figure 6.3(a) is 0.5432315, while that for the three increasingly noisy variants in Figures 6.4(b), 6.4(c), and 6.4(d) are 0.8995148, 0.9298075, and 0.9569399, respectively. Similar observations can be made for Jain, R15, and Compound data sets as well from the bottom rows of Figures 6.4, 6.5, and 6.6, respectively. Although the results are sensitive to the added noise and the choice of the random initial iterate, in general, it can be observed that lower values of c imply faster convergence. For instance, the bottom rows of Figures 6.3, 6.4, 6.5, and 6.6 show that for all four data sets, the proposed algorithm converges in lesser number of iterations in the noise-free case compared to the noisy ones. The value of the iteration threshold t', above which the asymptotic bound is satisfied by all the iterations until convergence, is marked by a dashed vertical line in the figures. In general, it can be observed from Figures 6.3, 6.4, 6.5, and 6.6 that for all data sets, as noise increases, the value of t' decreases implying a longer path until convergence. In brief, the results show that the convergence bound c can be used to make inference about the quality of the clusters and the speed of convergence of the proposed algorithm, for a given data set.

6.5.4 Choice of Rank

The proposed algorithm identifies the set of k clusters by performing clustering on the first k columns of $U_{\text{Joint}}^{\star}$. Although clustering is performed in a k-dimensional subspace, the proposed algorithm works with rank r subspaces, where r is generally greater or equal to k, in order to incorporate better information from the individual views. The optimal value of rank r is obtained using the same procedure as described in Section 5.5.2 of Chapter 5. The value of r is varied from k to $\max\{50, 2k\}$ and for each value of r, the Silhouette index $\mathcal{S}(r)$ is evaluated for clustering using the first k columns of $U_{\text{Joint}}^{\star}$. The optimal rank, $r^{\star}$, is the one that maximizes $\mathcal{S}(r)$ over different values of r.

In order to validate the choice of rank, based on the Silhouette index, the variation of both $\mathcal{S}(r)$ and F-measure is observed for different values of rank r. Figure 6.7 shows the variation of these two indices for Digits and LGG data sets, as examples. Similar to Figures 5.1 and 5.2 of Chapter 5, Figure 6.7 shows that $\mathcal{S}(r)$ and F-measure values tend to vary in a similar fashion for the data sets. The optimal values of rank for three image data sets, namely, Digits, 100Leaves, and ALOI are 12, 180, and 150, respectively. For both news article data sets, namely, 3Sources and BBC, the optimal rank is 21, while for eight omics data sets, namely, LGG, STAD, BRCA, LUNG, CRC, CESC, OV, and KIDNEY, the ranks are 43, 16, 4, 3, 8, 3, 5, and 5, respectively. For BRCA, LGG, STAD, LUNG, 100Leaves, and ALOI data sets, it is also observed that the F-measure corresponding to $r^{\star}$ coincides with the best F-measure obtained over different values of rank. In order to establish the importance of considering the optimal rank $r^{\star}$, Table 6.2 compares the performance of

Table 6.2: Performance Analysis of Proposed Algorithm at Rank k and Optimal Rank $r^\star$

Measure		Rank k	Rank $r^\star$		Rank k	Rank $r^\star$
Rank	**Digits**	10	12	**3Sources**	6	21
Accuracy		0.7905(0.0)	**0.9207**(4.21e-4)		0.6153(6.23e-3)	**0.7360**(5.92e-2)
NMI		0.7556(→0)	**0.8597**(4.88e-4)		0.5721(1.38e-2)	**0.6433**(3.59e-2)
ARI		0.6754(0.0)	**0.8352**(8.18e-4)		0.4635(1.88e-2)	**0.5957**(6.69e-2)
F-measure		0.8070(0.0)	**0.9209**(4.15e-4)		0.6786(5.28e-3)	**0.7581**(5.04e-2)
Rand		0.9409(0.0)	**0.9703**(1.49e-4)		0.8162(5.25e-3)	**0.8514**(2.61e-2)
Purity		0.7980(0.0)	**0.9207**(4.21e-4)		0.7455(6.23e-3)	**0.7946**(2.28e-2)
Rank	**BBC**	5	21	**100Leaves**	100	180
Accuracy		0.7275(8.35e-2)	**0.8715**(0.0)		0.6976(1.80e-2)	**0.8185**(1.55e-2)
NMI		0.6123(7.39e-2)	**0.7182**(→0)		0.8976(6.07e-3)	**0.9302**(4.12e-3)
ARI		0.5844(1.43e-1)	**0.7273**(0.0)		0.6148(2.12e-2)	**0.7431**(2.53e-2)
F-measure		0.7539(7.55e-2)	**0.8613**(0.0)		0.7524(1.55e-2)	**0.8492**(1.13e-2)
Rand		0.8253(7.10e-2)	**0.8959**(0.0)		0.9907(7.31e-4)	**0.9913**(1.17e-3)
Purity		0.7284(8.28e-2)	**0.8715**(0.0)		0.7380(1.62e-2)	**0.7772**(1.53e-2)
Rank	**BRCA**	4	4	**LGG**	3	43
Accuracy		**0.7964**(0.0)	**0.7964**(0.0)		0.6292(0.0)	**0.9625**(0.0)
NMI		**0.5553**(→0)	**0.5553**(→0)		0.4106(→0)	**0.8543**(→0)
ARI		**0.5474**(0.0)	**0.5474**(0.0)		0.2765(0.0)	**0.8790**(0.0)
F-measure		**0.7997**(0.0)	**0.7997**(0.0)		0.6316(0.0)	**0.9623**(0.0)
Rand		**0.8152**(0.0)	**0.8152**(0.0)		0.6590(0.0)	**0.9424**(0.0)
Purity		**0.7964**(0.0)	**0.7964**(0.0)		0.6741(0.0)	**0.9625**(0.0)
Rank	**STAD**	4	16	**LUNG**	2	3
Accuracy		0.5123(0.0)	**0.7727**(0.0)		0.9388(0.0)	**0.9463**(0.0)
NMI		0.2905(→0)	**0.5220**(→0)		0.6822(0.0)	**0.7173**(0.0)
ARI		0.1520(0.0)	**0.4650**(0.0)		0.7701(0.0)	**0.7965**(0.0)
F-measure		0.5239(0.0)	**0.7830**(0.0)		0.9386(0.0)	**0.9461**(0.0)
Rand		0.6362(0.0)	**0.7698**(0.0)		0.8850(0.0)	**0.8983**(0.0)
Purity		0.5909(0.0)	**0.7727**(0.0)		0.9388(0.0)	**0.9463**(0.0)

the proposed MiMIC algorithm when the rank is k with that of optimal rank $r^\star$ for four benchmark and four omics data sets. Table 6.2 shows that for LGG, STAD, LUNG, and all four benchmark data sets, there is a significant improvement in performance when considering rank $r^\star$ instead of k. For BRCA data, the performance is exactly same for both the cases.

6.5.5 Choice of Damping Factor in Joint Laplacian

The joint Laplacian $\mathbf{L}^r_{\text{Joint}}$, defined in (6.7), is a convex combination of the individual approximate graph Laplacians. The convex combination is set according to Section 5.3.5 of Chapter 5. In the convex combination, the Laplacians are weighted according to the relevance of the cluster information provided by the corresponding views. The relevance measure $\boldsymbol{\chi}$ in (5.31) gives a linear ordering of the views based on the quality of their underlying cluster structure. Based on this ordering, the relevance values are damped by powers of Δ and then used in the convex combination. This damping strategy upweights

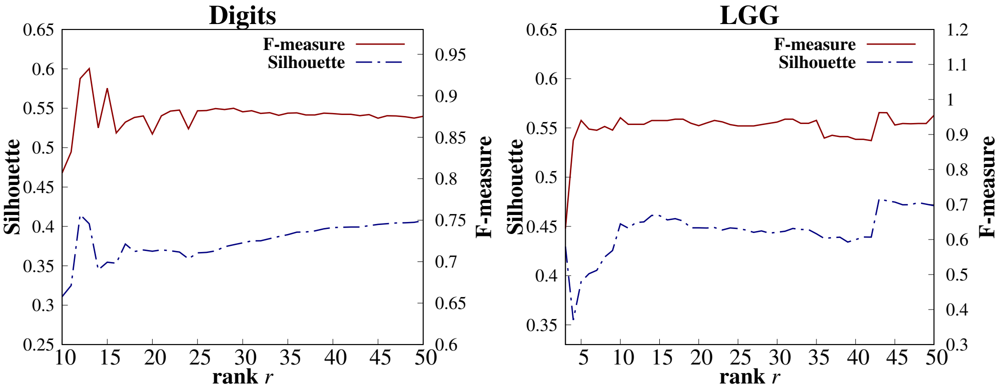
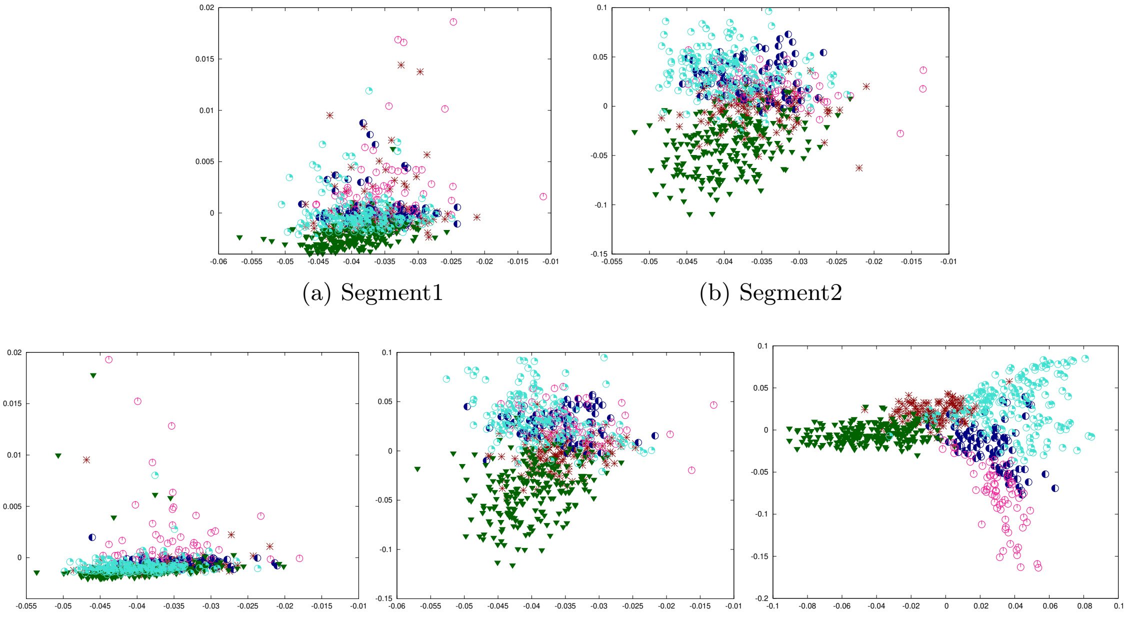


Figure 6.7: Variation of Silhouette index and F-measure for different values of rank r on Digits and LGG data sets.

the contribution of views with better cluster structure, while damping the effect of those having poorer structure. With damping factor $\Delta = 1$, the individual contributions are relatively close to each other depending upon their Fiedler values and Fiedler vectors. On the other hand, with $\Delta = 2$, the contributions of the views in decreasing order of relevance are $\frac{\mathcal{X}_{(1)}}{2}$, $\frac{\mathcal{X}_{(2)}}{4}$, $\frac{\mathcal{X}_{(3)}}{8}$, and so on. This indicates heavier damping resulting in higher difference between the individual contributions. The effect of the two damping factors is studied in Table 6.3 for different data sets.



(a) Segment1

(b) Segment2

(c) Segment3

(d) Segment4

(e) MiMIC

Figure 6.8: Two-dimensional scatter plots of individual views and proposed algorithm for BBC data set.

Table 6.3 shows that for four benchmark data sets, namely, Digits, 3Sources, BBC, and

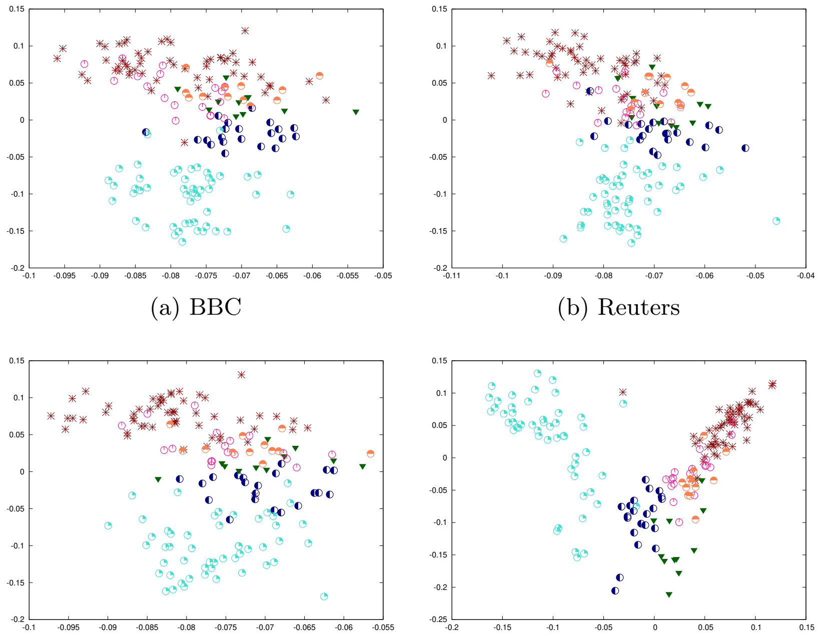
Table 6.3: Performance of the MiMIC Algorithm for Different Values of Damping Factor Δ on Benchmark and Multi-Omics Data Sets

	Measure		$\Delta = 1$	$\Delta = 2$		$\Delta = 1$	$\Delta = 2$
Benchmark	Rank	**Digits**	12	42	**3Sources**	21	26
	Accuracy		**0.9207**(4.21e-4)	0.7860(0.0)		**0.7360**(5.92e-2)	0.6520(3.74e-3)
	NMI		**0.8597**(4.88e-4)	0.8275(→0)		**0.6433**(3.59e-2)	0.6224(8.33e-3)
	ARI		**0.8352**(8.18e-4)	0.7367(0.0)		**0.5957**(6.69e-2)	0.5225(1.34e-2)
	F-measure		**0.9209**(4.15e-4)	0.8428(0.0)		**0.7581**(5.04e-2)	0.6941(3.91e-3)
	Rand		**0.9703**(1.49e-4)	0.9500(0.0)		**0.8514**(2.61e-2)	0.8191(7.10e-3)
	Purity		**0.9207**(4.21e-4)	0.8225(0.0)		**0.7946**(2.28e-2)	0.7763(3.74e-3)
Benchmark	Rank	**BBC**	21	5	**100Leaves**	180	50
	Accuracy		**0.8715**(0.0)	0.7976(3.04e-2)		**0.8185**(1.55e-2)	0.6765(1.80e-2)
	NMI		**0.7182**(→0)	0.6658(4.01e-2)		**0.9302**(4.12e-3)	0.8499(6.61e-3)
	ARI		**0.7273**(0.0)	0.7027(6.04e-2)		**0.7431**(2.53e-2)	0.5715(2.10e-2)
	F-measure		**0.8613**(0.0)	0.8127(3.42e-2)		**0.8492**(1.13e-2)	0.7067(1.46e-2)
	Rand		**0.8959**(0.0)	0.8874(2.89e-2)		**0.9913**(1.17e-3)	0.9910(5.93e-4)
	Purity		**0.8715**(0.0)	0.7991(3.04e-2)		**0.7772**(1.53e-2)	0.7120(1.32e-2)
Multi-Omics	Rank	**BRCA**	40	4	**LGG**	45	43
	Accuracy		0.6683(0.0)	**0.7964**(0.0)		0.9700(0.0)	**0.9625**(0.0)
	NMI		0.4503(→0)	**0.5553**(→0)		0.8646(→0)	**0.8543**(→0)
	ARI		0.3894(0.0)	**0.5474**(0.0)		0.9097(0.0)	**0.8790**(0.0)
	F-measure		0.6800(0.0)	**0.7997**(0.0)		0.9700(0.0)	**0.9623**(0.0)
	Rand		0.7499(0.0)	**0.8152**(0.0)		0.9574(0.0)	**0.9424**(0.0)
	Purity		0.6733(0.0)	**0.7964**(0.0)		0.9700(0.0)	**0.9625**(0.0)
Multi-Omics	Rank	**STAD**	25	16	**LUNG**	4	3
	Accuracy		0.7727(0.0)	**0.7727**(0.0)		0.9388(0.0)	**0.9463**(0.0)
	NMI		0.5183(→0)	**0.5220**(→0)		0.6920(0.0)	**0.7173**(0.0)
	ARI		0.4658(0.0)	**0.4650**(0.0)		0.7701(0.0)	**0.7965**(0.0)
	F-measure		0.7791(0.0)	**0.7830**(0.0)		0.9385(0.0)	**0.9461**(0.0)
	Rand		0.4591(0.0)	**0.7698**(0.0)		0.8850(0.0)	**0.8983**(0.0)
	Purity		0.7727(0.0)	**0.7727**(0.0)		0.9388(0.0)	**0.9463**(0.0)

100Leaves, lower damping ($\Delta = 1$) gives better performance compared to higher damping ($\Delta = 2$). The individual views of the benchmark data sets are relatively similar to each other, for instance, different segments of the same news article for the BBC data set, and RGB and HSV colour histograms of same image for ALOI data set. As a result, lower damping works better for the benchmark data sets. For the multi-oimcs data sets, however, Table 6.3 shows that heavier damping with $\Delta = 2$ gives better performance. Table 6.6 shows that there is a significant difference between the clustering performance of the most and the second most relevant views of LGG, BRCA, and LUNG data sets. Hence, significantly upweighting the most relevant view with $\Delta = 2$ gives better performance for the multi-oimcs data sets. Therefore, in this work, the damping factor Δ is chosen to be 2 for the multi-omics data sets, and 1 for the benchmark data sets.

Table 6.4: Performance Analysis of Spectral Clustering on Individual Views and Proposed MiMIC Algorithm for BBC and ALOI Data Sets

Views→		Segment1	Segment2	Segment3	Segment4	MiMIC
Accuracy	**BBC**	0.6202(2.1e-3)	0.6202(3.6e-2)	0.6102(3.6e-2)	0.5550(3.0e-3)	**0.8715**(0.0)
NMI		0.4312(1.7e-3)	0.4459(5.7e-2)	0.4097(1.3e-3)	0.4033(8.2e-3)	**0.7182**(→0)
ARI		0.3405(6.6e-2)	0.3895(8.9e-2)	0.3429(7.0e-3)	0.2518(1.1e-2)	**0.7273**(0.0)
F-measure		0.6514(1.2e-2)	0.6363(3.9e-2)	0.6435(1.2e-2)	0.6205(3.0e-3)	**0.8613**(0.0)
Rand		0.7256(1.7e-2)	0.7174(6.3e-2)	0.7425(7.4e-3)	0.6671(9.1e-3)	**0.8959**(0.0)
Purity		0.6212(3.5e-3)	0.6218(3.7e-2)	0.6120(3.6e-2)	0.5565(3.0e-3)	**0.8715**(0.0)
Views→		RGB	HSV	Haralick	ColorSimilarity	MiMIC
Accuracy	**ALOI**	0.4215(1.1e-2)	0.4433(7.0e-3)	0.1001(2.3e-3)	0.5191(1.1e-2)	**0.5742**(7.4e-3)
NMI		0.7179(3.9e-3)	0.7093(5.1e-3)	0.3659(4.1e-3)	0.7683(4.9e-3)	**0.7805**(2.3e-3)
ARI		0.2915(1.4e-2)	0.2979(1.9e-2)	0.0550(6.8e-4)	0.3745(2.2e-2)	**0.4233**(6.6e-3)
F-measure		0.4789(1.0e-2)	0.5136(7.9e-3)	0.1209(1.5e-3)	0.5843(1.1e-2)	**0.6221**(4.9e-3)
Rand		0.9745(1.8e-3)	0.9759(2.2e-3)	0.8938(7.0e-3)	0.9797(1.9e-3)	**0.9840**(3.7e-4)
Purity		0.4717(9.9e-3)	0.4876(7.3e-3)	0.1094(2.4e-3)	0.5547(8.6e-3)	**0.6119**(5.7e-3)



(a) BBC

(b) Reuters

(c) The Guardian

(d) MiMIC

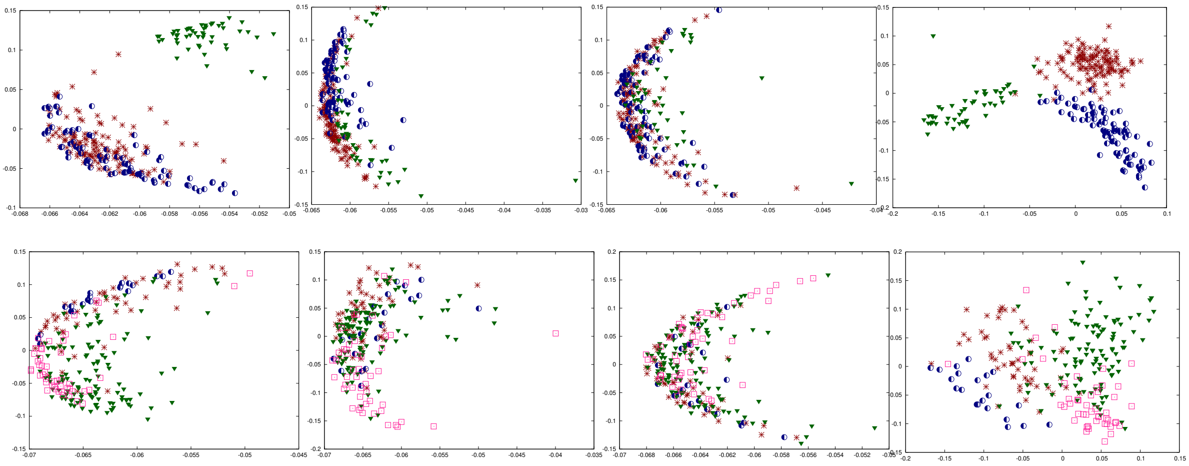
Figure 6.9: Two-dimensional scatter plots of individual views and proposed MiMIC algorithm for 3Sources data set.

6.5.6 Importance of Data Integration

The proposed algorithm integrates information by optimizing a joint clustering objective while reducing the disagreement between the joint and individual subspaces. To study the importance of integration, the performance of the proposed algorithm is compared with

Table 6.5: Performance Analysis of Spectral Clustering on Individual Views and Proposed MiMIC Algorithm for 100Leaves and 3Sources Data Sets

Views→		Shape	Texture	Margin	MiMIC
Accuracy	**100Leaves**	0.3095(9.1e-3)	0.4777(1.4e-2)	0.5786(1.1e-2)	**0.8185**(1.5e-2)
NMI		0.6479(6.7e-3)	0.7327(5.6e-3)	0.7940(4.4e-3)	**0.9302**(4.1e-3)
ARI		0.1820(5.8e-3)	0.3265(1.4e-2)	0.4478(9.8e-3)	**0.7431**(2.5e-2)
F-measure		0.3525(7.4e-3)	0.5139(1.1e-2)	0.6113(9.7e-3)	**0.8492**(1.1e-2)
Rand		0.9699(1.5e-3)	0.9839(7.2e-4)	0.9880(2.9e-4)	**0.9913**(1.17e-3)
Purity		0.3696(7.5e-3)	0.5216(1.2e-2)	0.6203(7.6e-3)	**0.7772**(1.53e-2)
Views→		BBC	Guardian	Reuters	MiMIC
Accuracy	**3Sources**	0.7159(0.0)	0.6508(0.0)	0.5562(0.0)	**0.7360**(5.9e-2)
NMI		0.6390(0.0)	0.5270(0.0)	0.5347(0.0)	**0.6433**(3.5e-2)
ARI		**0.6082**(0.0)	0.4119(0.0)	0.41434(0.0)	0.5957(6.6e-2)
F-measure		**0.7656**(0.0)	0.7036(0.0)	0.6482(0.0)	0.7581(5.0e-2)
Rand		**0.8624**(0.0)	0.7983(0.0)	0.7982(0.0)	0.8514(2.6e-2)
Purity		0.7869(0.0)	0.6982(0.0)	0.6982(0.0)	**0.7946**(2.2e-2)



(a) mDNA (b) RNA (c) miRNA (d) MiMIC

Figure 6.10: Two-dimensional scatter plots of three individual views and proposed MiMIC algorithm for multi-omics cancer data sets: LGG (top row) and STAD (bottom row).

that of spectral clustering on the individual views. The comparative results are reported in Tables 6.4 and 6.5 for the benchmark data sets, and in Table 6.6 for the multi-omics cancer data sets. The results in Tables 6.4 and 6.5 clearly show that for four benchmark data sets, namely, Digits, BBC, ALOI, and 100Leaves, there is significant improvement in performance of the proposed MiMIC algorithm considering multiple views over any single view clustering. For the 3Sources data set, there is lesser improvement in terms of NMI and accuracy, and the single view BBC news source gives the best performance in terms of ARI and F-measure. In case of the multi-omics data sets, Table 6.6 shows that for all four data sets, the proposed algorithm achieves the best clustering performance across all four evaluation indices. The performance gain is most evident for LGG and STAD data

Table 6.6: Performance Analysis of Spectral Clustering on Individual Views and Proposed MiMIC Algorithm for Multi-Omics Data Sets

Views→		mDNA	RNA	miRNA	RPPA	MiMIC
Accuracy	**LGG**	0.8352060	0.5917603	0.4307116	0.3970037	**0.9625468**
NMI		0.5734568	0.2176187	0.0498676	0.0254500	**0.8543905**
ARI		0.5567870	0.1801875	0.0510240	0.0238319	**0.8790253**
F-measure		0.8269248	0.5875701	0.4717221	0.4326018	**0.9623406**
Rand		0.7861508	0.6149925	0.5593760	0.5476050	**0.9424967**
Purity		0.8352060	0.5917603	0.5318352	0.5280899	**0.9625468**
Accuracy	**STAD**	0.5413223	0.4793388	0.3719008	0.4173554	**0.7727273**
NMI		0.2282198	0.1779419	0.0771419	0.0831100	**0.5220123**
ARI		0.1927570	0.1047749	0.0514998	0.0460928	**0.4650334**
F-measure		0.5469686	0.4781377	0.3998266	0.4469459	**0.7830757**
Rand		0.6509722	0.6239155	0.5989164	0.5883543	**0.7698296**
Purity		0.5867769	0.5495868	0.4917355	0.4917355	**0.7727273**
Accuracy	**BRCA**	0.5804020	0.7688442	0.4623116	0.4798995	**0.7964824**
NMI		0.3408150	0.5277072	0.1947561	0.3140984	**0.5553836**
ARI		0.3047769	0.5130244	0.1663564	0.2359641	**0.5474472**
F-measure		0.5982526	0.7690661	0.5105008	0.5630781	**0.7997020**
Rand		0.7193018	0.7995519	0.6455071	0.6689493	**0.8152728**
Purity		0.6532663	0.7688442	0.5703518	0.5879397	**0.7964824**
Accuracy	**LUNG**	0.8107303	0.9359165	0.8241431	0.5037258	**0.9463487**
NMI		0.2980508	0.6631276	0.3575188	0.0001449	**0.7173075**
ARI		0.3852741	0.7597207	0.4193820	-0.001743	**0.7965891**
F-measure		0.8104506	0.9357307	0.8237679	0.5630053	**0.9461134**
Rand		0.6926485	0.8798674	0.7097048	0.4992815	**0.8983028**
Purity		0.8107303	0.9359165	0.8241431	0.5365127	**0.9463487**

sets. Gene or RNA expression is the most relevant view for BRCA and LUNG data sets, while for LGG and STAD data sets it is DNA-methylation. For BRCA and LUNG data sets, the clustering performance of RNA expression is very close to that of the proposed multi-view algorithm. Evidently, most of the initial works of cancer subtype identification were based on gene expression study [75, 198].

The scatter plots of the first two dimensions of the subspaces extracted by the individual views and the proposed algorithm are given in Figures 6.9 and 6.8 for two benchmark data sets: 3Sources and BCC, and in Figure 6.10 for two multi-omics data sets: LGG and STAD, as examples. The objects in these figures are colored according to the ground truth or previously established TCGA cancer subtypes. The scatter plots for the individual views in Figures 6.9, 6.8 and 6.10 demonstrate the diversity of cluster structures exhibited by the views. The scatter plots for the proposed algorithm in Figures 6.9(d), 6.8(e), and 6.10(d) (top rpw) demonstrate significantly higher cluster separability compared to any of their individual views for 3Sources, BBC, and LGG data sets, respectively. The distinct omic views may exhibit disparate cluster structures, but Tables 6.4-6.6, and Figures 6.9-6.10 indicate that proper integration gives much better idea about the overall cluster structure of the data set.

Table 6.7: Performance Analysis of Individual Manifolds and Proposed Algorithm

Manifold→		k-Means	Stiefel	MiMIC		k-Means	Stiefel	MiMIC
Accuracy	**Digits**	0.8205(0.0)	0.6480(2.2e-2)	**0.9207**(4.2e-4)	**BBC**	0.7345(8.5e-2)	0.7732(1.3e-3)	**0.8715**(0.0)
NMI		0.8350(→0)	0.6535(8.8e-3)	**0.8597**(4.8e-4)		0.6167(6.1e-2)	0.5983(2.4e-3)	**0.7182**(→0)
ARI		0.7687(0.0)	0.5155(1.5e-2)	**0.8352**(8.1e-4)		0.5910(1.2e-1)	0.6382(3.2e-3)	**0.7273**(0.0)
F-measure		0.8756(0.0)	0.6922(1.8e-2)	**0.9209**(4.1e-4)		0.7614(6.1e-1)	0.7806(1.3e-2)	**0.8613**(0.0)
Rand		0.9564(0.0)	0.9005(5.7e-03)	**0.9703**(1.4e-4)		0.8315(6.1e-2)	0.8659(1.1e-3)	**0.8959**(0.0)
Purity		0.8315(0.0)	0.6665(1.8e-02)	**0.9207**(4.2e-4)		0.7356(8.5e-2)	0.7747(1.3e-3)	**0.8715**(0.0)
Accuracy	**3Sources**	0.6497(2.4e-3)	0.6798(7.2e-2)	**0.7360**(5.9e-2)	**LGG**	0.9288(0.0)	0.6292(0.0)	**0.9625**(0.0)
NMI		0.6221(4.6e-3)	0.6020(7.2e-2)	**0.6433**(3.5e-2)		0.7949(→0)	0.4305(→0)	**0.8543**(→0)
ARI		0.5173(2.1e-3)	0.5226(1.2e-1)	**0.5957**(6.6e-2)		0.7790(0.0)	0.2842(0.0)	**0.8790**(0.0)
F-measure		0.6927(2.7e-4)	0.7330(6.3e-2)	**0.7581**(5.0e-2)		0.9269(0.0)	0.6313(0.0)	**0.9623**(0.0)
Rand		0.8170(1.7e-4)	0.8310(4.1e-2)	**0.8514**(2.61e-2)		0.8940(0.0)	0.6632(0.0)	**0.9424**(0.0)
Purity		0.7739(2.4e-3)	0.7455(5.9e-2)	**0.7946**(2.28e-2)		0.9288(0.0)	0.6779(0.0)	**0.9625**(0.0)
Accuracy	**100Leaves**	0.7139(2.2e-2)	0.6457(1.8e-2)	**0.8185**(1.5e-2)	**STAD**	0.6867(3.3e-2)	0.5165(0.0)	**0.7727**(0.0)
NMI		0.8887(5.4e-3)	0.8278(5.9e-3)	**0.9302**(4.1e-3)		0.4412(4.2e-2)	0.2985(→0)	**0.5220**(→0)
ARI		0.6374(2.1e-2)	0.5323(1.5e-2)	**0.7431**(2.5e-2)		0.3615(5.1e-2)	0.1616(0.0)	**0.4650**(0.0)
F-measure		0.7543(1.7e-2)	0.6741(1.5e-2)	**0.8492**(1.1e-2)		0.6930(3.1e-2)	0.5241(0.0)	**0.7830**(0.0)
Rand		0.9920(6.4e-4)	0.9903(3.8e-4)	**0.9913**(1.1e-3)		0.6938(1.1e-2)	0.6433(0.0)	**0.7698**(0.0)
Purity		0.7502(1.8e-2)	0.6764(1.5e-2)	**0.7772**(1.5e-2)		0.6884(2.9e-2)	0.5909(0.0)	**0.7727**(0.0)
Accuracy	**ALOI**	0.5044(1.4e-2)	0.5068(1.8e-2)	**0.5742**(7.4e-3)	**BRCA**	0.7085(0.0)	0.7889(0.0)	**0.7964**(0.0)
NMI		0.7461(4.5e-3)	0.7462(5.2e-3)	**0.7805**(2.3e-3)		0.4964(→0)	0.5373(→0)	**0.5553**(→0)
ARI		0.3874(1.6e-2)	0.3850(1.7e-2)	**0.4233**(6.6e-3)		0.4291(0.0)	0.5331(0.0)	**0.5474**(0.0)
F-measure		0.5739(1.0e-2)	0.5748(1.4e-2)	**0.6221**(4.9e-3)		0.7072(0.0)	0.7905(0.0)	**0.7997**(0.0)
Rand		0.9828(1.1e-3)	0.9826(1.2e-3)	**0.9840**(3.7e-4)		0.7670(0.0)	0.8075(0.0)	**0.8152**(0.0)
Purity		0.5461(1.1e-2)	0.5483(1.5e-2)	**0.6119**(5.7e-3)		0.7085(0.0)	0.7889(0.0)	**0.7964**(0.0)

6.5.7 Importance of k-Means and Stiefel Manifolds

The proposed objective function $\boldsymbol{f}$ in (6.9) is optimized over two different manifolds, namely, k-Means and Stiefel manifolds. The k-means manifold optimizes the joint clustering component, while Stiefel manifold minimizes the disagreement component. To establish the importance of the k-means manifold, only the disagreement minimization component corresponding to the U_j's is optimized over the Stiefel manifold. However, in this optimization problem, the joint subspace U_{Joint} does not get updated. So, to evaluate the performance of Stiefel manifold, the final clustering is performed on the subspace corresponding to the most relevant view (according to the relevance measure defined in Section 5.3.5 of Chapter 5). The comparative performance of Stiefel manifold optimization and the proposed MiMIC algorithm is presented in Table 6.7 for different benchmark and omics data sets. Table 6.7 shows that optimization over only the Stiefel manifold has led to significantly poor performance compared to the proposed algorithm for all data sets, except for BRCA and LUNG. For BRCA and LUNG data sets, the proposed algorithm outperforms the Stiefel manifold, but by a lower margin. This is attributed to the fact that for these two data sets, the most relevant view (that is, RNA) has performance close to the proposed algorithm (see Table 6.6), and the final clustering is also performed on the most relevant subspace. The significant difference in performance establishes the importance of k-means manifold in the proposed approach.

In order to study the importance of Stiefel manifold, the performance of the proposed algorithm is compared with that of the case where only the joint clustering component corresponding to U_{Joint} is optimized over k-means manifold. The comparative performance is reported in Table 6.7. For all the data sets, the proposed MiMIC algorithm optimized over two manifolds outperforms the joint clustering component optimized over only the k-means manifold. This establishes the essence of Stiefel manifold. Table 6.7 also indicates that apart from the 3Sources and BRCA data sets, k-means manifold optimization gives better performance compared to that of Stiefel manifold. For ALOI and LUNG data sets, the average performance of the two individual manifolds are competitive. However, best results are obtained when both manifolds are considered. This establishes the importance of considering two different manifolds.

6.5.8 Comparative Performance Analysis

Finally, the performance of the proposed MiMIC algorithm is extensively compared with that of several existing multi-view clustering algorithms on benchmark and multi-omics cancer data sets. Corresponding results are reported in Tables 6.8, 6.9, and 6.10, where best performance is highlighted in bold, while italicized values indicate second best performance.

6.5.8.1 Results on Benchmark Data Sets

For five benchmark data sets, the performance of MiMIC is compared with that of eight state-of-the-art methods, namely, multi-view k-means clustering (MKC) [26], co-regularized spectral clustering (CoregSC) [120], multi-view spectral clustering (MSC) [246], adaptive structure-based multi-view clustering (ASMV) [272], multiple graph learning (MGL) [164], multi-view clustering with graph learning (MCGL) [273], graph-based multi-view clustering (GMC) [236], and convex combination of approximate graph Laplacians (CoALa) [113]. Among these algorithms, MKC and CoregSC are subspace clustering and co-training based approaches, respectively, while others are graph-based approaches. The proposed MiMIC algorithm is compared with these eight existing approaches based on four external indices, namely, accuracy, NMI, ARI, and F-measure, similar to [236].

The comparative performance is provided in Table 6.8 for benchmark data sets. The results in Table 6.8 show that the proposed MiMIC algorithm gives the best performance on all five benchmark data sets across all measures, except for three cases, that is, ARI and NMI on Digits data set, accuracy on 100Leaves, and ARI on ALOI data set. For these three cases, MiMIC achieves the second best performance. The graph based algorithms like ASMV, MCGL, and GMC have standard deviations zero or close to zero as they do not require an additional k-means clustering step to determine the partition. The MiMIC algorithm performs robustly on Digits and BBC dats sets, with standard deviations close to zero. Small standard deviations are observed for 3Sources, 100Leaves, and ALOI data sets, among which the later two have as high as hundred clusters, while 3Sources exhibits poor separability in its joint subspace (as seen in Figure 6.9(d)). In general, algorithms like MKC, CoregSC, and MSC perform poorly compared to recently proposed graph algorithms like MCGL, GMC, and CoALa, all of which are again outperformed by the proposed MiMIC algorithm in 16 out of 20 cases. Similar to the proposed MiMIC algorithm, the CoALa algorithm is also based on Laplacian approximation, but CoALa obtains a closed form solution over the Euclidean space. The better performance of the proposed algorithm across

Table 6.8: Comparative Performance Analysis of Proposed and Existing Integrative Clustering Algorithms on Benchmark Data Sets

	Algorithm →	MKC	CoregSC	MSC	ASMV	MGL	MCGL	GMC	CoALa	**MiMIC**
Digits	Accuracy	0.4924(2.77e-1)	0.7556(5.96e-2)	0.7918(8.21e-2)	0.5745(→0)	0.7440(8.19e-2)	0.8530(0.0)	0.8820(→0)	*0.8835*(0.0)	**0.9207**(4.21e-4)
	NMI	0.5325(3.68e-1)	0.7421(3.27e-2)	0.7560(3.24e-2)	0.6709(→0)	0.8264(4.73e-2)	**0.9055**(0.0)	*0.9050*(→0)	0.7981(→0)	0.8597(4.88e-4)
	ARI	0.4280(2.99e-1)	0.6885(5.73e-2)	0.6803(6.28e-2)	0.4047(→0)	0.6888(1.07e-1)	0.8313(→0)	**0.8502**(→0)	0.7645(0.0)	*0.8352*(8.18e-4)
	F-measure	0.5130(2.33e-2)	0.6934(5.11e-2)	0.7129(5.58e-2)	0.4852(→0)	0.7238(9.37e-2)	0.8493(→0)	0.8658(→0)	*0.8839*(0.0)	**0.9209**(4.15e-4)
3Sources	Accuracy	0.4663(1.06e-1)	0.5479(2.99e-2)	0.4751(2.97e-2)	0.3373(→0)	0.6751(6.67e-2)	0.3077(→0)	*0.6923*(→0)	0.6508(0.0)	**0.7360**(5.92e-2)
	NMI	0.3665(1.00e-1)	0.5238(1.98e-2)	0.3850(2.27e-2)	0.0896(→0)	0.5768(8.61e-2)	0.1034(→0)	*0.6216*(0.0)	0.6198(→0)	**0.6433**(3.59e-2)
	ARI	0.2461(1.40e-1)	0.3339(2.85e-2)	0.2618(3.81e-2)	-0.021(→0)	0.4431(1.17e-1)	-0.033(→0)	0.4431(0.0)	*0.5183*(0.0)	**0.5957**(6.69e-2)
	F-measure	0.4114(1.08e-1)	0.4775(1.91e-2)	0.4087(3.05e-2)	0.3528(→0)	0.5966(7.12e-2)	0.3417(0.0)	0.6047(0.0)	*0.6929*(0.0)	**0.7581**(5.92e-2)
BBC	Accuracy	0.6034(1.10e-1)	0.4701(0.0)	0.6732(4.94e-2)	0.3372(0.0)	0.5396(1.10e-1)	0.3533(→0)	0.6934(→0)	*0.8108*(4.36e-3)	**0.8715**(0.0)
	NMI	0.4786(8.51e-2)	0.2863(0.0)	0.5531(1.44e-2)	0.0348(0.0)	0.3697(1.89e-1)	0.0741(→0)	0.5628(0.0)	*0.6536*(1.96e-2)	**0.7182**(→0)
	ARI	0.3450(1.21e-1)	0.2727(0.0)	0.4658(2.20e-2)	0.0018(→0)	0.3153(1.66e-1)	0.0053(→0)	0.4789(→0)	*0.7102*(2.78e-2)	**0.7273**(0.0)
	F-measure	0.5018(9.03e-2)	0.4879(0.0)	0.5877(1.83e-2)	0.3781(0.0)	0.5402(8.53e-2)	0.3762(0.0)	0.6333(→0)	*0.8138*(9.93e-4)	**0.8613**(0.0)
100Leaves	Accuracy	0.0100(0.0)	0.7706(2.58e-2)	0.7379(2.21e-2)	0.7906(→0)	0.6904(2.42e-2)	0.8106(→0)	**0.8238**(→0)	0.7384(1.34e-2)	*0.8185*(1.56e-2)
	NMI	0.0000(0.0)	0.9165(5.90e-3)	0.9014(7.60e-3)	0.9009(→0)	0.8753(7.60e-3)	0.9130(0.0)	*0.9292*(0.0)	0.8893(4.06e-3)	**0.9302**(4.12e-3)
	ARI	0.0000(0.0)	0.7229(1.92e-2)	*0.6788*(2.26e-2)	0.6104(→0)	0.3858(5.65e-2)	0.5155(→0)	0.4974(→0)	0.6550(1.41e-2)	**0.7431**(2.53e-2)
	F-measure	0.0186(0.0)	0.7257(1.90e-2)	0.6821(2.23e-2)	0.6148(→0)	0.3944(5.53e-2)	0.5217(0.0)	0.5042(→0)	*0.7672*(1.19e-2)	**0.8492**(1.13e-2)
ALOI	Accuracy	0.0101(→0)	0.5217(2.13e-2)	0.4738(7.65e-2)	0.4555(→0)	0.4807(1.51e-2)	0.4625(→0)	*0.5705*(→0)	0.5594(1.44e-2)	**0.5742**(7.44e-3)
	NMI	0.0000(0.0)	0.6993(1.32e-2)	0.6358(5.44e-2)	0.6767(→0)	0.7052(7.00e-3)	0.6657(→0)	0.7350(0.0)	*0.7654*(3.72e-3)	**0.7805**(2.39e-3)
	ARI	0.0000(→0)	0.4097(4.52e-2)	0.3305(4.81e-2)	0.0533(0.0)	0.1987(4.37e-2)	0.0441(0.0)	0.4305(→0)	**0.4352**(1.18e-2)	*0.4233*(6.66e-3)
	F-measure	0.0196(→0)	0.4051(2.38e-2)	0.3366(3.68e-2)	0.0712(→0)	0.2112(4.22e-2)	0.0621(→0)	0.4366(→0)	*0.6213*(1.15e-2)	**0.6221**(4.90e-3)

all four benchmark data sets in Table 6.8 establishes the importance of iterative line-search based manifold optimization in the proposed formulation compared to Euclidean space optimization in CoALa.

6.5.8.2 Results on Multi-Omics Cancer Data Sets

For the cancer data sets, the performance of MiMIC is compared with nine integrative cancer subtype identification algorithms, namely, cluster of cluster analysis (COCA) [93], multivariate normality based joint subspace clustering (NormS) [111], LRAcluster [243], iCluster [192], principal component analysis on naively concatenated data (PCA-con), selective update of relevant eigenspaces (SURE) [112], joint and individual variance explained (JIVE) [141], similarity network fusion (SNF) [234], and CoALa [113]. The COCA is a two-stage consensus clustering based approach, LRAcluster, NormS, and iCluster are probabilistic model based approaches, while JIVE and SURE are low-rank subspace based approaches. The SNF and CoALa algorithms are graph based approaches. The experimental setup followed for the existing multi-omics cancer subtyping algorithms is same as that followed in Chapter 3.

The comparative performance analysis is reported in Table 6.9. The results in Table 6.9 show that for BRCA and STAD data sets, the proposed MiMIC algortihm has the closest resemblance with the previously established TCGA and WHO subtypes of these cancers, in terms of all external indices. For LGG, LUNG, CRC, and KIDNEY data sets, although the best performance is obtained by either CoALa or SNF, the performance of MiMIC is very competitive. Among the five existing probabilistic model based approaches, NormS has superior performance in majority of the cases. The iCluster algorithm has poor performance on CESC, KIDNEY, STAD, and LUNG data sets, and LRAcluster has comparatively poor performance on STAD and LGG data sets. This poor performance is attributed to the poor fitting of their probabilistic model on the real-life data sets. The PCA-con, JIVE, and SURE algorithms are SVD based low-rank approaches, among them PCA-con and SURE have comparable performance. The results reported in both Tables 6.8, 6.9, and 6.10 show that the proposed MiMIC algorithm performs significantly better than the existing ones on majority of benchmark data sets and some omics data sets.

6.5.8.3 Results on Social Network and General Image Data Sets

Apart from the results on various data sets reported in Section 6.5.8.1 and Section 6.5.8.2, experiments are also carried out on five Twitter data sets: Football, Olympics, Politics-IE, Politics-UK, and Rugby; two general image data sets: Caltech7 and ORL; and one citation network data set: CORA. These data sets mostly have graph/network based views [78]. The performance of the proposed MiMIC algorithm on these eight data sets is compared with that of the two individual manifolds, namely, k-means and Stiefel manifolds, and with that of two graph based approaches, namely, SNF and CoALa (proposed in Chapter 5). The comparative results are provided in Tables 6.11 and 6.12.

The results in Tables 6.11 and 6.12 show that for most of the external indices, the proposed algorithm has better performance compared to the both individual manifolds: k-means and Stiefel manifolds, for all social network and image data sets, except Politics-UK. For the Politics-UK data set, better clustering performance is achieved when considering

Table 6.9: Comparative Performance Analysis of Proposed and Existing Integrative Clustering Algorithms on Multi-Omics Data Sets: BRCA, LGG, STAD, LUNG

	Algorithm→	Consensus	Statistical Model Based			Subspace Based			Graph Based		Manifold
		COCA	NormS	LRAcluster	iCluster	PCA-con	SURE	JIVE	SNF	CoALa	**MiMIC**
BRCA	Accuracy	0.7434(7.94e-4)	*0.7688(0.0)*	0.7110(0.0)	0.7638(0.0)	0.7587(0.0)	0.7663(0.0)	0.6859(0.0)	0.6783(0.0)	0.7613(0.0)	**0.7964**(0.0)
	NMI	0.5002(3.48e-4)	0.4287(→0)	0.5437(→0)	0.5176(→0)	0.5506(→0)	0.4558(0.0)	0.4368(0.0)	*0.5528*(→0)	0.5281(→0)	**0.5553**(→0)
	ARI	0.4864(4.50e-4)	0.5090(0.0)	0.4035(0.0)	0.4745(0.0)	0.5038(0.0)	*0.5104*(0.0)	0.3772(0.0)	0.4111(0.0)	0.4874(0.0)	**0.5474**(0.0)
	F-measure	0.7457(8.13e-4)	*0.7699*(0.0)	0.7101(0.0)	0.7658(0.0)	0.7601(0.0)	0.7683(0.0)	0.6889(0.0)	0.6865(0.0)	0.7660(0.0)	**0.7997**(0.0)
	Rand	0.7905(1.92e-4)	*0.7999*(0.0)	0.7521(0.0)	0.7842(0.0)	0.7984(0.0)	0.8010(0.0)	0.7464(0.0)	0.7602(0.0)	0.7922(0.0)	**0.8152**(0.0)
	Purity	0.7434(7.95e-4)	*0.7688*(0.0)	0.7110(0.0)	0.7638(0.0)	0.7587(0.0)	0.7663(0.0)	0.6859(0.0)	0.6959(0.0)	0.7613(0.0)	**0.7964**(0.0)
LGG	Accuracy	0.6591(0.0)	0.7940(0.0)	0.4719(0.0)	0.4382(0.0)	0.6666(0.0)	0.7940(0.0)	0.5617(0.0)	0.8689(0.0)	**0.9737**(0.0)	*0.9625*(0.0)
	NMI	0.2772(0.0)	0.5325(0.0)	0.1240(→0)	0.1379(→0)	0.3438(0.0)	0.5335(0.0)	0.2299(→0)	0.6253(0.0)	**0.8689**(→0)	*0.8543*(→0)
	ARI	0.2533(0.0)	0.4649(0.0)	0.1030(0.0)	0.0996(0.0)	0.3031(0.0)	0.4668(0.0)	0.1606(0.0)	0.6331(0.0)	**0.9199**(0.0)	*0.8790*(0.0)
	F-measure	0.6608(0.0)	0.7916(0.0)	0.5137(0.0)	0.5187(0.0)	0.6574(0.0)	0.7904(0.0)	0.5757(0.0)	0.8720(0.0)	**0.9737**(0.0)	*0.9623*(0.0)
	Rand	0.6454(0.0)	0.7465(0.0)	0.5831(0.0)	0.5821(0.0)	0.6616(0.0)	0.7465(0.0)	0.6056(0.0)	0.8268(0.0)	**0.9622**(0.0)	*0.9424*(0.0)
	Purity	0.6591(0.0)	0.7940(0.0)	0.5280(0.0)	0.5355(0.0)	0.6666(0.0)	0.7940(0.0)	0.5730(0.0)	0.8689(0.0)	**0.9737**(0.0)	*0.9625*(0.0)
STAD	Accuracy	0.4450(3.34e-2)	0.5702(0.0)	0.4256(0.0)	0.3512(0.0)	0.6900(0.0)	0.6983(0.0)	0.4049(0.0)	0.5661(0.0)	*0.7685*(0.0)	**0.7727**(0.0)
	NMI	0.1309(4.77e-3)	0.1805(→0)	0.1259(→0)	0.0650(→0)	0.3654(0.0)	0.3511(→0)	0.1288(→0)	0.3216(0.0)	*0.5107*(0.0)	**0.5220**(→0)
	ARI	0.0740(1.02e-2)	0.1625(0.0)	0.0912(0.0)	0.0288(0.0)	0.3204(0.0)	0.3445(0.0)	0.0657(0.0)	0.2694(0.0)	*0.4559*(0.0)	**0.4650**(0.0)
	F-measure	0.4558(2.50e-2)	0.5770(0.0)	0.4746(0.0)	0.3832(0.0)	0.6959(0.0)	0.7056(0.0)	0.4487(0.0)	0.6333(0.0)	*0.7778*(0.0)	**0.7830**(0.0)
	Rand	0.5981(1.32e-2)	0.6435(0.0)	0.6122(0.0)	0.5855(0.0)	0.7110(0.0)	0.7216(0.0)	0.5981(0.0)	0.6945(0.0)	*0.7661*(0.0)	**0.7698**(0.0)
	Purity	0.5173(9.50e-3)	0.5950(0.0)	0.5619(0.0)	0.4917(0.0)	0.6900(0.0)	0.6983(0.0)	0.5165(0.0)	0.6363(0.0)	*0.7685*(0.0)	**0.7727**(0.0)
LUNG	Accuracy	0.9284(0.0)	0.9359(0.0)	0.9344(0.0)	0.6333(0.0)	0.9388(0.0)	0.9418(0.0)	0.9269(0.0)	**0.9493**(0.0)	0.9403(0.0)	*0.9463*(0.0)
	NMI	0.6287(0.0)	0.6650(0.0)	0.6535(0.0)	0.0627(0.0)	0.6773(→0)	0.6878(0.0)	0.6333(0.0)	*0.7152*(0.0)	0.6970(0.0)	**0.7173**(0.0)
	ARI	0.7339(0.0)	0.7597(0.0)	0.7545(0.0)	0.0696(0.0)	0.7701(0.0)	0.7806(0.0)	0.7288(0.0)	**0.8072**(0.0)	0.7754(0.0)	*0.7965*(0.0)
	F-measure	0.9283(0.0)	0.9357(0.0)	0.9342(0.0)	0.6299(0.0)	0.9386(0.0)	0.9417(0.0)	0.9266(0.0)	**0.9492**(0.0)	0.9400(0.0)	*0.9461*(0.0)
	Rand	0.8669(0.0)	0.8798(0.0)	0.8772(0.0)	0.5348(0.0)	0.8850(0.0)	0.8903(0.0)	0.8644(0.0)	**0.9036**(0.0)	0.8877(0.0)	*0.8983*(0.0)
	Purity	0.9284(0.0)	0.9359(0.0)	0.9344(0.0)	0.6333(0.0)	0.9388(0.0)	0.9418(0.0)	0.9269(0.0)	**0.9493**(0.0)	0.9403(0.0)	*0.9463*(0.0)

Table 6.10: Comparative Performance Analysis of Proposed and Existing Integrative Clustering Algorithms on Multi-Omics Data Sets: CRC, CESC, KIDNEY, OV

	Algorithm→	Consensus	Statistical Model Based			Subspace Based			Graph Based		Manifold
		COCA	NormS	LRAcluster	iCluster	PCA-con	SURE	JIVE	SNF	CoALa	**MiMIC**
CRC	Accuracy	0.5323(5.56e-3)	0.6206(0.0)	0.5129(0.0)	0.6163(0.0)	0.5366(0.0)	0.5107(0.0)	0.6034(0.0)	0.5991(0.0)	**0.6400**(0.0)	*0.6228*(0.0)
	NMI	*0.0120*(1.27e-3)	0.0093(0.0)	0.0030(0.0)	0.0070(0.0)	0.0057(0.0)	0.0028(0.0)	0.0071(0.0)	0.0069(0.0)	**0.0185**(0.0)	0.0069(0.0)
	ARI	0.0007(1.86e-3)	0.0347(0.0)	-0.001(0.0)	0.0293(0.0)	0.0037(0.0)	-0.002(0.0)	0.0256(0.0)	0.0240(0.0)	**0.0548**(0.0)	*0.0310*(0.0)
	F-measure	0.5586(5.56e-3)	0.6345(0.0)	0.5410(0.0)	0.6298(0.0)	0.5642(0.0)	0.5416(0.0)	0.6210(0.0)	0.6178(0.0)	**0.6529**(0.0)	*0.6338*(0.0)
	Rand	0.5010(6.97e-4)	0.5281(0.0)	0.4992(0.0)	0.5260(0.0)	0.5016(0.0)	0.4991(0.0)	0.5203(0.0)	0.5186(0.0)	**0.5382**(0.0)	*0.5291*(0.0)
	Purity	**0.7370**(0.0)	**0.7370**(0.0)	**0.7370**(0.0)	**0.7370**(0.0)	**0.7370**(0.0)	**0.7370**(0.0)	**0.7370**(0.0)	**0.7370**(0.0)	**0.7370**(0.0)	**0.7370**(0.0)
CESC	Accuracy	0.6693(0.0)	**0.8870**(0.0)	0.8145(0.0)	0.5483(0.0)	0.8548(0.0)	*0.8629*(0.0)	0.7177(0.0)	0.6693(0.0)	0.8225(0.0)	0.8548(0.0)
	NMI	0.4172(4.77e-3)	**0.6854**(→0)	0.5176(→0)	0.1737(→0)	*0.6750*(→0)	0.6461(0.0)	0.4425(0.0)	0.4927(0.0)	0.5479(0.0)	0.6451(→0)
	ARI	0.3677(8.95e-4)	**0.7004**(0.0)	0.5384(0.0)	0.1017(0.0)	0.6333(0.0)	*0.6507*(0.0)	0.3860(0.0)	0.4239(0.0)	0.5637(0.0)	0.6236(0.0)
	F-measure	0.6865(2.49e-3)	**0.8801**(0.0)	0.8123(0.0)	0.5568(0.0)	0.8390(0.0)	*0.8512*(0.0)	0.7097(0.0)	0.7073(0.0)	0.8139(0.0)	0.8418(0.0)
	Rand	0.6971(6.33e-5)	**0.8587**(0.0)	0.7867(0.0)	0.5731(0.0)	0.8237(0.0)	*0.8339*(0.0)	0.7164(0.0)	0.7043(0.0)	0.7951(0.0)	0.8193(0.0)
	Purity	0.6774(0.0)	**0.8870**(0.0)	0.8145(0.0)	0.5645(0.0)	0.8548(0.0)	*0.8629*(0.0)	0.7177(0.0)	0.6935(0.0)	0.8225(0.0)	0.8548(0.0)
KIDNEY	Accuracy	0.9470(0.0)	0.9525(0.0)	0.9538(0.0)	0.6065(0.0)	0.9511(0.0)	0.9525(0.0)	0.9308(0.0)	**0.9579**(0.0)	0.9294(0.0)	*0.9552*(0.0)
	NMI	0.7493(0.0)	0.7726(→0)	*0.7862*(0.0)	0.2547(→0)	0.7670(→0)	0.7726(→0)	0.6955(→0)	**0.7946**(0.0)	0.6987(→0)	0.7767(0.0)
	ARI	0.8393(0.0)	0.8534(0.0)	*0.8579*(0.0)	0.1717(0.0)	0.8489(0.0)	0.8534(0.0)	0.7786(0.0)	**0.8796**(0.0)	0.7786(0.0)	0.8534(0.0)
	F-measure	0.9477(0.0)	0.9530(0.0)	0.9545(0.0)	0.6514(0.0)	0.9516(0.0)	0.9530(0.0)	0.9300(0.0)	**0.9590**(0.0)	0.9285(0.0)	*0.9551*(0.0)
	Rand	0.9199(0.0)	0.9269(0.0)	*0.9292*(0.0)	0.5842(0.0)	0.9246(0.0)	0.9269(0.0)	0.8893(0.0)	**0.9400**(0.0)	0.8893(0.0)	0.9268(0.0)
	Purity	0.9470(0.0)	0.9525(0.0)	0.9538(0.0)	0.6811(0.0)	0.9511(0.0)	0.9525(0.0)	0.9308(0.0)	**0.9579**(0.0)	0.9294(0.0)	*0.9552*(0.0)
OV	Accuracy	0.5943(7.09e-3)	*0.6976*(0.0)	0.6287(0.0)	0.5089(0.0)	0.6946(0.0)	**0.7215**(0.0)	0.5718(7.73e-3)	0.5269(0.0)	0.6736(0.0)	0.6595(2.84e-3)
	NMI	0.3131(1.22e-2)	*0.4504*(→0)	0.3745(→0)	0.2249(→0)	0.4424(0.0)	**0.4680**(→0)	0.2629(8.42e-4)	0.2753(0.0)	0.3381(→0)	0.3271(3.97e-4)
	ARI	0.2810(6.83e-3)	*0.4142*(0.0)	0.2999(0.0)	0.2005(0.0)	0.4068(0.0)	**0.4372**(0.0)	0.2027(4.21e-3)	0.2058(0.0)	0.3199(0.0)	0.3112(4.28e-3)
	F-measure	0.6068(4.28e-3)	*0.6910*(0.0)	0.6384(0.0)	0.4808(0.0)	0.6868(0.0)	**0.7148**(0.0)	0.5653(7.84e-3)	0.5642(0.0)	0.6700(0.0)	0.6611(2.54e-3)
	Rand	0.7039(2.64e-3)	*0.7766*(0.0)	0.7322(0.0)	0.6916(0.0)	0.7734(0.0)	**0.7857**(0.0)	0.6885(2.80e-3)	0.6557(0.0)	0.7379(0.0)	0.7383(1.92e-3)
	Purity	0.5943(7.09e-3)	*0.6976*(0.0)	0.6287(0.0)	0.5119(0.0)	0.6946(0.0)	**0.7215**(0.0)	0.5718(7.73e-3)	0.5389(0.0)	0.6736(0.0)	0.6595(2.84e-3)

only the *k*-means manifold. Compared to the graph based approaches SNF and CoALa, the proposed algorithm outperforms both of them for Football, Politics-IE, Caltech7, and CORA data sets on majority of external indices. For the Olympics, Politics-UK, and ORL data sets, the performance of CoALa and the proposed MiMIC algorithm is comparable. For the Rugby data set, the CoALa algorithm of Chapter 5 has the best performance.

Table 6.11: Comparative Performance Analysis of Proposed and Existing Algorithms on Twitter Data Sets

	Algorithm→	*k*-Means Manifold	Stiefel Manifold	Graph Based		**MiMIC**
				SNF	CoALa	
Football	Accuracy	*0.8673*(1.14e-2)	0.7366(1.09e-2)	0.8145(0.0)	0.8500(2.58e-2)	**0.8846**(2.27e-2)
	NMI	0.8804(9.42e-3)	0.7742(6.64e-3)	*0.8829*(0.0)	0.8625(1.90e-2)	**0.8958**(1.24e-2)
	ARI	*0.7566*(2.51e-2)	0.5584(1.63e-2)	0.7458(0.0)	0.7278(4.15e-2)	**0.7841**(4.61e-2)
	F-measure	*0.8792*(9.90e-2)	0.7610(9.99e-3)	0.8431(0.0)	0.8683(1.81e-2)	**0.8941**(1.74e-2)
	Rand	*0.9756*(3.02e-3)	0.9538(2.93e-3)	0.9735(0.0)	0.9739(4.74e-3)	**0.9781**(5.73e-3)
	Purity	*0.8713*(1.15e-2)	0.7576(1.03e-2)	0.8266(0.0)	0.8548(2.42e-2)	**0.8879**(2.13e-2)
Olympics	Accuracy	0.8228(2.88e-2)	0.7390(2.04e-2)	**0.9051**(0.0)	0.8443(1.49e-2)	*0.8844*(2.60e-2)
	NMI	0.9141(1.08e-2)	0.8075(1.05e-2)	*0.9381*(0.0)	0.9197(5.78e-3)	**0.9394**(9.10e-3)
	ARI	0.7890(6.58e-2)	0.5474(2.64e-2)	**0.9090**(0.0)	0.80712.40e-2)	*0.8699*(3.52e-2)
	F-measure	0.8520(2.87e-2)	0.7699(1.82e-2)	*0.9121*(0.0)	0.8682(1.46e-2)	**0.9006**(2.35e-2)
	Rand	0.9787(8.20e-3)	0.9391(5.51e-3)	**0.9911**(0.0)	0.9812(2.83e-3)	*0.9871*(3.69e-3)
	Purity	0.8782(1.66e-2)	0.7605(1.97e-2)	**0.9137**(0.0)	0.8991(1.03e-2)	*0.9112*(1.70e-2)
Politics-IE	Accuracy	0.8764(0.0)	0.8048(3.30e-2)	*0.9252*(0.0)	0.8735(0.0)	**0.9436**(1.45e-2)
	NMI	0.8246(→0)	0.6884(2.63e-2)	*0.8938*(0.0)	0.8170(0.0)	**0.8573**(1.88e-2)
	ARI	0.7408(0.0)	0.7096(3.17e-2)	**0.9409**(0.0)	0.8284(0.0)	*0.8693*(2.81e-2)
	F-measure	0.8662(0.0)	0.7988(3.18e-2)	*0.9258*(0.0)	0.8583(0.0)	**0.9447**(1.21e-2)
	Rand	0.8910(0.0)	0.8780(1.55e-2)	**0.9772**(0.0)	0.9305(0.0)	*0.9499*(1.01e-2)
	Purity	0.8850(0.0)	0.8275(2.02e-2)	*0.9310*(0.0)	0.8793(0.0)	**0.9436**(1.45e-2)
Politics-UK	Accuracy	**0.9785**(0.0)	0.9245(7.65e-3)	*0.9737*(0.0)	0.9665(0.0)	0.9727(2.01e-3)
	NMI	*0.93331*(→0)	0.7365(1.69e-2)	0.9194(0.0)	**0.9434**(0.0)	0.9225(5.69e-3)
	ARI	**0.9640**(0.0)	0.8175(1.58e-2)	0.9608(0.0)	*0.9633*(0.0)	0.9522(4.83e-3)
	F-measure	*0.9735*(0.0)	0.9182(8.98e-3)	0.9701(0.0)	**0.9736**(0.0)	0.9692(3.97e-3)
	Rand	**0.9829**(0.0)	0.9133(7.73e-3)	0.9814(0.0)	*0.9826*(0.0)	0.9774(2.32e-3)
	Purity	**0.9785**(0.0)	0.9274(5.30e-3)	*0.9761*(0.0)	**0.9785**(0.0)	0.9727(2.01e-3)
Rugby	Accuracy	0.6822(2.11e-2)	0.6001(3.63e-2)	*0.7611*(0.0)	**0.8305**(2.41e-3)	0.6841(2.40e-2)
	NMI	0.6513(1.11e-2)	0.6283(1.56e-2)	*0.6768*(0.0)	**0.7093**(3.13e-3)	0.6552(9.61e-3)
	ARI	0.4345(3.03e-2)	0.4057(1.91e-2)	*0.5485*(0.0)	**0.6627**(1.88e-3)	0.4344(2.72e-2)
	F-measure	0.7320(2.21e-2)	0.6629(3.27e-2)	*0.7778*(0.0)	**0.8349**(1.03e-3)	0.7331(2.50e-2)
	Rand	0.8622(7.46e-3)	0.8591(7.48e-3)	*0.8818*(0.0)	**0.9067**(4.61e-4)	0.8631(4.87e-3)
	Purity	0.8512(1.39e-2)	0.8418(1.91e-2)	0.8454(0.0)	**0.8606**(2.26e-3)	*0.8600*(8.94e-3)

6.6 Conclusion

This chapter presents a novel manifold optimization based algorithm for integrative clustering of high dimensional multi-view data sets. A joint objective is proposed, consisting

Table 6.12: Comparative Performance Analysis of Proposed and Existing Algorithms on ORL, Caltech7, and CORA Data Sets

Algorithm→		k-Means Manifold	Stiefel Manifold	Graph Based		**MiMIC**
				SNF	CoALa	
ORL	Accuracy	0.6602(4.30e-2)	0.7127(2.97e-2)	0.6907(2.57e-2)	**0.7715**(2.18e-2)	*0.7307*(2.36e-2)
	NMI	0.8396(1.87e-2)	0.8756(1.07e-2)	0.8616(1.00e-2)	**0.8980**(1.15e-2)	0.8814(1.35e-2)
	ARI	0.5141(5.14e-2)	0.6027(2.97e-2)	0.6054(3.04e-2)	**0.6932**(2.82e-2)	*0.6208*(3.83e-2)
	F-measure	0.7047(3.40e-2)	0.7620(2.22e-2)	0.7257(2.44e-2)	**0.7962**(1.78e-2)	*0.7677*(2.29e-2)
	Rand	0.9728(4.29e-3)	0.9792(2.19e-3)	*0.9804*(2.04e-3)	**0.9850**(1.63e-3)	0.9802(2.65e-3)
	Purity	0.7197(3.40e-2)	0.7635(1.91e-2)	0.7450(2.26e-2)	**0.8090**(1.75e-2)	*0.7737*(1.78e-2)
Caltech7	Accuracy	0.5655(5.72e-4)	0.4181(3.27e-4)	0.5440(3.42e-2)	*0.5685*(0.0)	**0.5773**(0.0)
	NMI	*0.5730*(1.17e-3)	0.3141(7.64e-5)	0.5676(2.41e-2)	0.5650(→0)	**0.5880**(→0)
	ARI	*0.4463*(1.12e-3)	0.2831(2.27e-4)	0.4126(2.86e-2)	0.4397(0.0)	**0.4608**(0.0)
	F-measure	0.6471(4.11e-4)	0.5290(3.63e-4)	0.6363(4.12e-2)	**0.6689**(0.0)	*0.6600*(0.0)
	Rand	*0.7613*(4.63e-4)	0.6958(8.45e-5)	0.7482(1.13e-2)	0.7583(0.0)	**0.7674**(0.0)
	Purity	*0.8654*(5.72e-4)	0.7648(6.55e-4)	0.8516(1.09e-2)	0.8548(0.0)	**0.8751**(0.0)
CORA	Accuracy	0.4823(4.46e-3)	0.3980(3.45e-2)	0.5450(2.79e-2)	*0.5896*(3.41e-3)	**0.6120**(2.46e-3)
	NMI	0.3284(4.92e-3)	0.2573(2.67e-2)	0.3829(1.14e-2)	*0.4364*(2.81e-3)	**0.4686**(6.17e-3)
	ARI	0.1275(3.02e-3)	0.0885(3.29e-3)	0.2941(1.86e-2)	*0.3256*(2.87e-3)	**0.3479**(3.73e-3)
	F-measure	0.4726(8.67e-3)	0.3932(2.36e-2)	*0.5957*(1.96e-2)	0.5844(4.98e-3)	**0.6373**(3.50e-3)
	Rand	0.5253(4.25e-2)	0.4862(4.36e-2)	**0.7936**(9.31e-3)	0.7460(2.14e-3)	*0.7709*(9.35e-4)
	Purity	0.5031(6.78e-3)	0.4468(1.60e-2)	0.6012(1.62e-2)	*0.6206*(3.41e-3)	**0.6423**(2.46e-3)

of two components, namely, a joint clustering component to identify compact and well-separated clusters, and a disagreement minimization component to look for consistent clusters across different views. The joint objective is optimized over two different manifolds, namely, k-means and Stiefel manifolds. The Stiefel manifold models the differential clusters in the individual views, while the k-means manifold tries to infer the best-fit global cluster structure in the data. The optimization is performed separately along the manifolds of each view, so that individual non-linearity within each view is not lost while looking for the shared cluster information. The convergence of the proposed algorithm is theoretically established over the manifold, while the analysis of its asymptotic behavior quantifies how fast it converges to an optimal solution. The derived asymptotic bound is used to make inference regarding the separability of the clusters present in the data set. The clustering performance of the proposed algorithm is studied and compared with several state-of-the-art integrative clustering approaches on several multi-omics cancer data sets and benchmark data sets. Comparative studies demonstrate that the proposed algorithm can efficiently leverage information from multiple views, and for majority of the data sets, it reveals clusters that have closest resemblance with the previously established cancer subtypes and the ground-truth class information.

The MiMIC algorithm proposed in this chapter optimizes only the joint and individual clustering subspaces to capture the underlying structure of the data set. However, simultaneous optimization of the individual graphs, their corresponding weightage in the joint view, as well as the joint and individual subspaces, is likely to give a more comprehensive idea of the clusters present in the data set. In this regard, Chapter 7 presents another manifold optimization algorithm that harnesses the geometry and structure preserving properties of symmetric positive definite manifold and Grassmannian manifold for efficient multi-view clustering.

Chapter 7

Geometry Aware Multi-View Clustering over Riemannian Manifolds

7.1 Introduction

Multi-view clustering, now a major hot spot in unsupervised machine learning, aims to gather similar subjects in the same group and dissimilar ones in different groups, utilizing the information of multiple views, instead of just one. The extensive literature on multi-view clustering can be classified into several categories [34,174], which are briefly described in Chapter 2. Among them, *graph based models* form the most common category, which fuse graphs from different views and extract a lower dimensional subspace or spectral embedding of the fused graph to perform clustering [128,164,234,236,246,272,273]. The weighted graph fusion has been proposed in several approaches [128,164,273]. Among them, the algorithms proposed in Chapters 5 and 6, and in [199] fix graph weights a priori, while those proposed in [128, 164, 236, 272, 273] use adaptive weight optimization techniques. A major issue with graph-driven approaches is that the real-world views inherently contain measurement errors, redundancy, and noise, which propagate during the graph fusion process distorting the learned cluster structure. In this regard, Chapter 5 introduces the fusion of de-noised approximations of view-specific graph Laplacians to obtain better cluster separation in the subsequent approximate subspace. However, the approach focuses on extracting only the consistent information of different views via a joint subspace of the fused graph. The complementary information of individual graphs is ignored during the fusion process.

In several real-world applications, data appears to be point-cloud, but, it's meaningful structure resides on a lower dimensional manifold embedded in the higher dimensional space [180, 186, 232]. The conventional manifold learning algorithms exploit the property that a manifold, although non-linear, has a locally linear geometry that resembles the Euclidean space. The locally linear property is used to identify neighboring points in a cluster [180,186]. In these algorithms, however, the form of the manifold is unknown. As a result, the metric and properties of the space are generally not defined. In a separate line of approach, the data is assumed to originate from a clearly known manifold, preferably a Riemannian one, as they are endowed with a smoothly varying inner product [74,232,233]. Some widely used Riemannian manifolds are Stiefel [57], Grassmannian [3], and symmetric positive definite (SPD) [74] manifolds. The Stiefel manifold is used in the optimization of cost functions with orthogonality constraints, where, in addition to the subspace structure, the specific choice of basis vectors is also important [143]. The Grassmannian manifold's geometric properties have been utilized in vision problems involving subspace constraints. Examples include affine invariant shape clustering [10], subspace tracking [195, 201], and face recognition from image sets and video clustering [224, 232, 233]. The covariance matrices of features, used as region descriptors, have been looked as points on the SPD manifold [209]. Nevertheless, the use of these manifolds is primarily restricted to image/video based applications. Their strength in general multi-view graph and spectral clustering applications is yet to be fully explored.

The MiMIC algorithm proposed in Chapter 6 uses Stiefel manifold to extract spectral embeddings of joint and individual views for clustering. However, the graphs constructed from inherently noisy real-life views may not be ideal to extract the best fit cluster structure of the data set. Although the algorithm proposed in Chapter 6 extracts both consistent information of fused graph and complementary information of individual graphs, it does not address the issue of graph refinement. Furthermore, the spectral embeddings extracted by the algorithm are sensitive to the choice of basis. So, it does not take into account the intrinsic geometry of the solution space. Simultaneous optimization of the individual graph structures, their fusion weights, and the joint and individual subspaces, is likely to give a more comprehensive idea of the clusters present in the data set.

In this regard, the current chapter presents a manifold based multi-view clustering algorithm, termed as GeARS (Geometry Aware Riemannian Spectral clustering). The proposed algorithm harnesses the geometry and structure preserving properties of Grassmannian and SPD manifolds for efficient multi-view clustering. It optimizes the spectral clustering objective separately on de-noised approximations of joint and individual views to extract the shared as well as view-specific complementary cluster structures. To impose consistency between the clustering in different views, it also minimizes the distance between the cluster solutions of joint and individual views, as well as that between pairwise individual views. The optimization is performed using a gradient based line-search that alternates between the SPD and Grassmannian manifolds. The SPD manifold is used to optimize the graph Laplacians corresponding to the individual views while preserving their symmetricity, positive definiteness, and related properties. The Grassmannian manifold, on the other hand, is used to optimize and reduce the disagreement between different clustering subspaces. Grassmannian modeling additionally enforces the clustering solutions to be basis invariant cluster indicator subspaces. The basis invariance property takes into account geometry of the space and maps multiple orthonormal cluster indicators spanning the same subspace into a single solution, as they are merely rotations of each other which does not essentially change the cluster structure conveyed by the subspace. The graph weights are also optimized at each iteration of the algorithm to obtain the optimal combination of the views. The asymptotic convergence behavior of the proposed algorithm is studied to obtain an upper bound that quantifies how fast the algorithm converges to a local optimal solution. The matrix perturbation theory is used to theoretically bound the disagreement or Grassmannian distance between the joint and individual subpaces at any given iteration of the proposed algorithm. The disagreement is empirically shown to minimize as the algorithm progresses and converges to a local minima. The multi-view clustering performance of the GeARS algorithm is extensively studied and compared with that of existing ones on diverse benchmark data sets. Its application in cancer subtype identification from multiple omics data types is also established.

The rest of the chapter is organized as follows: Section 7.2 presents the proposed model of multi-view data integration and clustering. Section 7.3 introduces the proposed line-search optimization technique over the Grassmannian and SPD manifolds and the proposed GeARS algorithm, and analyzes its convergence behavior. In Section 7.4, an upper bound on the Grassmannian distance between the joint and the individual subspaces is derived using matrix perturbation theory. Case studies on different multi-view benchmark data sets and multi-omics cancer data sets, along with a comparative performance analysis with existing approaches, are presented in Section 7.5. Concluding remarks are provided in Section 7.6.

7.2 GeARS: Proposed Method

A multi-view data set is a collection of $M(\geqslant 2)$ views for a common set of n samples, $\{x_i\}_{i=1}^n$. Each view is usually represented by a matrix $X_m \in \Re^{n \times d_m}$, for $m = 1, \ldots, M$, consisting of d_m-dimensional observations from the m-th data source for the common n samples. The view X_m can be encoded as a n node similarity graph G_m whose vertices represent the samples and edges represent the pairwise similarities between the samples. Let its affinity matrix be given by $W_m = [w_m(i,j)]_{n \times n}$. Its (i,j)-th element $w_m(i,j) \geqslant 0$ represents the affinity between samples x_i and x_j in view X_m. Given affinity W_m, the degree matrix D_m represents the total affinity at each vertex of the graph. It is given by $D_m = diag(\bar{d}_1^m, \ldots, \bar{d}_i^m, \ldots, \bar{d}_n^m)$, where $\bar{d}_i^m = \sum_{j=1}^{n} w_m(i,j)$. The shifted normalized Laplacian of graph G_m, as defined in Chapter 5, is given by

$$L_m = \mathbf{I}_n + D_m^{-1/2} W_m D_m^{-1/2}, \tag{7.1}$$

where $\mathbf{I}_n$ denotes the $(n \times n)$ identity matrix. The advantage of shifted Laplacian over the conventional definition [45] of $\mathbf{I}_n - D_m^{-1/2} W_m D_m^{-1/2}$ is that it merges the best rank k approximation of L_m as well as its cluster information into the same eigenspace. The spectral clustering problem in terms of L_m is a negative trace minimization problem given by

$$\underset{U_m \in \Re^{n \times k}}{\text{minimize}} - tr(U_m^T L_m U_m) \text{ such that } U_m^T U_m = \mathbf{I}_k, \tag{7.2}$$

where $tr(.)$ denotes the matrix trace function. The Laplacian and its spectrum provides insight into the edge-connectivity of the graph. This connectivity differs in each network, thus conveying varying cluster information. A truly integrative approach should (i) capture the joint or consistent clustering across different views while preserving the complementary cluster pattern of each view, (ii) refine the connectivity of the graphs based on joint and individual cluster structures, (iii) automatically estimate the weight or contribution of individual views during construction of the joint view (iv) be resilient to noise and heterogeneity of the high-dimensional views. A manifold based multi-view clustering approach that captures all these properties is described next. The term 'Laplacian' in the following sections would refer to its shifted normalized variant L as defined in (7.1), unless explicitly specified.

7.2.1 Geometry Aware Multi-View Integration

The solution U_m to the spectral clustering problem in (7.2) gives the best fit cluster indicator for view X_m given Laplacian L_m. Solving this for each of the M views gives M different cluster indicators that preserve the complementary cluster structure of different views. The shared or joint cluster structure, on the other hand, can be obtained by constructing a joint view and then solving its corresponding spectral clustering problem. The joint view can be constructed by integrating the individual Laplacians using a convex combination with weights proportional to the separability of the clusters in the views. However, the Laplacians constructed from the high-dimensional real-world views invariably contain noise, which gets reflected in the joint view during the combination process. To prevent this noise propagation, Chapter 5 proposes to combine de-noised approximations of the Laplacians. Specifically, let the best rank r approximation of Laplacian L_m be given by its eigenvalue decomposition as follows:

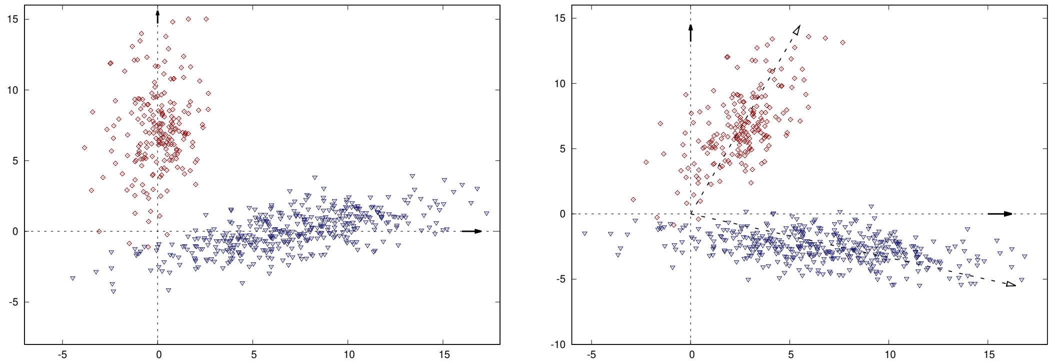
$$L_m^r = V_m^r \Sigma_m^r (V_m^r)^T, \tag{7.3}$$

where $V_m^r \in \Re^{n \times r}$ contains the r largest eigenvectors of L_m in its columns and Σ_m^r is a diagonal matrix consisting of the corresponding r largest non-zero eigenvalues. Conventionally, spectral clustering uses the k largest eigenvectors of L_m to obtain a clustering of view X_m. However, during Laplacian approximation, the rank r in (7.3) is considered to be greater or equal to k, the number of clusters, in order to capture more information from each view. To make the integration step resilient to noise, the "approximate" Laplacians L_m^r are integrated in the weighted combination, as opposed to "full-rank" Laplacians L_m, as presented in Chapter 5. The approximation tends to preserve the stronger pairwise similarities as opposed to the weaker ones. The approximate joint Laplacian corresponding to the fused network is given by

$$\mathbf{L}_{\text{Joint}}^r = \sum_{m=1}^{M} \alpha_m L_m^r, \text{ such that } \alpha_m \geqslant 0 \text{ and } \sum_{m=1}^{M} \alpha_m = 1. \tag{7.4}$$

The above approximation automatically filters the noise in the $(n-r)$ least significant eigenpairs of the individual L_m's from propagating into the joint network. Solving the spectral clustering problem on $\mathbf{L}_{\text{Joint}}^r$ gives a joint cluster indicator, say U_{Joint}, that captures the consistent clustering accross different views.

The spectral clustering solutions U_{Joint} and U_m are all $(n \times k)$ orthonormal matrices, which are treated as projection of the n points in some k-dimensional subspace. The k columns act as a set of k orthonormal basis vectors for the corresponding subspace. However, any k-dimensional subspace can be represented by an infinite number of orthogonal bases. A change in the orthonormal basis for the same subspace amounts to a linear transformation of the projected points which does not essentially change the cluster structure reflected in that subspace. Figure 7.1 shows an illustrative example in two dimensions.



(a) Axes aligned basis

(b) Rotated basis

Figure 7.1: Effect of basis rotation on the cluster structure of a data set.

The data points in Figure 7.1(b) are rotations of those in Figure 7.1(a). In Figure 7.1(a) the basis is aligned with the trivial $x - y$ axes, while that in Figure 7.1(b) is rotated by some angle θ. However, the geometry of the data points in Figures 7.1(a) and 7.1(b) show that this rotation does not change the cluster structure of the data set. This motivates the search for basis invariant solutions. The basis invariance property implies that the solution is a cluster indicator subspace instead of a representative cluster indicator matrix. Indicator subspaces are obtained by optimizing the spectral clustering objective in (7.2) over $\mathsf{span}(U_m)$, as opposed to a particular U_m, where, $\mathsf{span}(A)$ denotes the linear subspace spanned by the columns of matrix A. Optimization over the column space, $\mathsf{span}(U_m)$, restricts the search space to be a Riemannian quotient space, known as the Grassmannian manifold [3], defined by

$$\mathsf{Gr}(n,k) := \{\mathsf{span}(U) \in \Re^{n\times k} \mid U^T U = \mathbf{I}_k\}. \tag{7.5}$$

The Grassmannian manifold, $\mathsf{Gr}(n,k)$, with the integers $n \geqslant k > 0$ is the space formed by all k-dimensional linear subspaces embedded in the n-dimensional Euclidean space. A point on $\mathsf{Gr}(n,k)$ is represented by any orthonormal basis for a subspace. Clearly, the choice of representative basis is not unique. Hence, a Grassmannian point is an equivalence class $[U]$ of the set of all orthogonal matrices whose columns span the same subspace as those of U. For any matrix $U \in \Re^{n\times k}$, its column span is rotation invariant, that is, $\mathsf{span}(U) = \mathsf{span}(UR)$ for any $R \in O(k)$, where $O(k)$ is the set of $k \times k$ orthogonal rotation matrices. Hence, the equivalence classes of the Grassmannian manifold are obtained by the action of $(k \times k)$ orthogonal rotation matrices over the set of $(n \times k)$ orthonormal matrices, denoted by

$$\mathsf{Gr}(n,k) := \{U \in \Re^{n\times k} \mid U^T U = \mathbf{I}_k\}/O(k). \tag{7.6}$$

The curved surface in Figure 7.2 shows the Grassmannian manifold, while the points on the manifold, marked in black, are linear subspaces represented by rectangular planes. For instance, $\mathsf{span}(U_1)$ in Figure 7.2 is a Grassmann point and the equivalance class $[U_1]$ consists

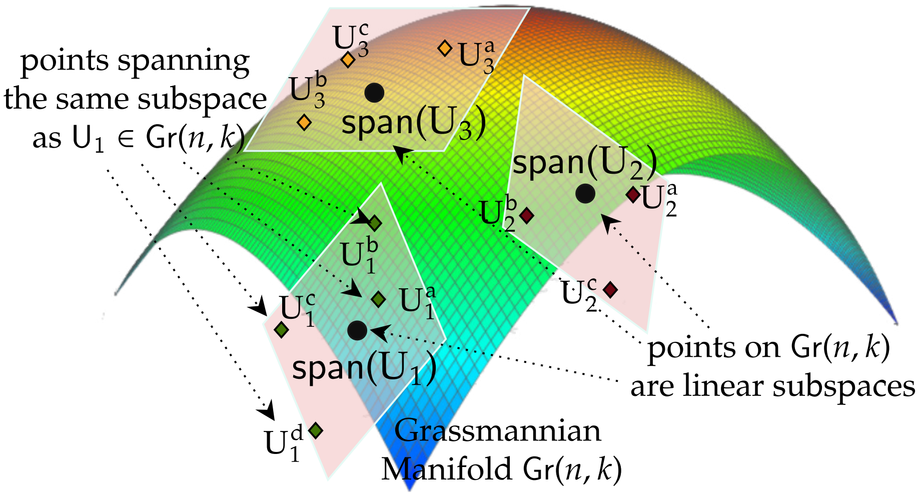


Figure 7.2: The Grassmannian manifold.

of all orthonormal matrices whose columns span the same subspace as those of U_1 (denoted by points U_1^a, U_1^b, U_1^c, and U_1^d in Figure 7.2. The quotient geometry of Grassmannian manifold in (7.6) enables the search for subspaces as opposed to representative matrices.

In the proposed formulation, the spectral clustering problem of (7.2) is solved for the approximate joint view as well as approximate individual ones, in order to obtain the global clustering while preserving the complementary cluster patterns of individual views. Furthermore, to incorporate geometry awareness, the solutions are made basis invariant by considering the search space to be the Grassmannian manifold $\mathsf{Gr}(n,k)$ of k-dimensional subspaces, as opposed to representative matrices in the Euclidean space, $\Re^{n\times k}$. This reduces the problem to an unconstrained optimization over the Grassmannian manifold as the orthonormality constraints on indicator subspaces are inherently incorporated into the manifold structure. The problem is given by

$$\underset{\left.\begin{array}{c}\mathsf{span}(U_{\text{Joint}})\\ \mathsf{span}(U_m)\end{array}\right\}\in\mathsf{Gr}(n,k)}{\text{minimize}} \quad -\,tr\left(U_{\text{Joint}}^T \mathbf{L}_{\text{Joint}}^r U_{\text{Joint}}\right) - \frac{1}{M}\sum_{m=1}^{M} tr(U_m^T L_m^r U_m). \tag{7.7}$$

In this optimization framework, k-dimensional linear subspaces, $\mathsf{span}(U_{\text{Joint}})$ and $\mathsf{span}(U_m)$, simply reduce to Grassmannian points. To impose consistency between the global clustering and clustering reflected in different views, the Grassmannian distance between the joint subspace and each of the individual subspaces, as well as that between pairwise individual subspaces are minimized. The distance between two Grassmann points can be computed in terms of the k principal angles $\{\theta_i\}_{i=1}^k$ between the subspaces [280]. The projection distance between points $\mathsf{span}(U_{\text{Joint}})$ and $\mathsf{span}(U_m)$ is given by

$$d_\theta(U_{\text{Joint}}, U_m) = \sum_{i=1}^{k} \sin^2\theta_i^m = \| U_{\text{Joint}}U_{\text{Joint}}^T - U_m U_m^T \|_F^2, \tag{7.8}$$

where θ_i^m denotes the i-th largest principal angle between corresponding subspaces. This distance can be reduced to

$$d_\theta(U_{\text{Joint}}, U_m) = k - tr\left(U_{\text{Joint}} U_{\text{Joint}}^T U_m U_m^T\right). \tag{7.9}$$

Incorporating the individual and pairwise distance minimization terms, the optimization problem becomes

$$\begin{aligned} \underset{\substack{\mathsf{span}(U_{\text{Joint}}) \\ \mathsf{span}(U_m)}\Big\}\in \mathsf{Gr}(n,k)}{\text{minimize}} \quad & \boldsymbol{f}(U_{\text{Joint}}, U_1, ..., U_M) = -\frac{1}{2} tr\left(U_{\text{Joint}}^T \mathbf{L}_{\text{Joint}}^r U_{\text{Joint}}\right) \qquad (7.10) \\ & + \frac{1}{2M(M-1)} \sum_{\substack{i,j=1 \\ i\neq j}}^{M} d_\theta(U_i, U_j) + \frac{1}{2M} \sum_{m=1}^{M} \Big[- tr(U_m^T L_m^r U_m) + d_\theta(U_{\text{Joint}}, U_m)\Big]. \end{aligned}$$

In the above problem, only the cluster indicator subspaces are optimized. However, the connectivity of the individual graphs also plays a crucial role in determining the global and view-specific clustering of the data set. Therefore, for better understanding of the true nature of the data set, it is essential to use the cluster information of indicator subspaces to modify the graph connectivity, as well as use the modified graph connections to update the cluster assignments.

7.2.2 Updation of Graph Connectivity

The graph Laplacian L_m in (7.1) contains the pairwise connectivity information of the similarity matrix W_m, while its spectrum contains the graph partition information. These information are pivotal in identifying the clusters in the data set. The Laplacian L_m is symmetric and positive semi-definite with n eigenvalues in $[0, 2]$. Its approximation, L_m^r, is constructed using the r largest non-zero eigenvalues and corresponding eigenvectors. Hence, L_m^r is symmetric and positive definite (SPD). Due to this property, the approximate Laplacians become elements of the Riemannian manifold of symmetric and positive definite matrices, known as the SPD manifold [74], which is defined as follows

$$\mathcal{S}_{++}^n = \{A \in \Re^{n\times n} \mid A = A^T \text{and } v^T A v > 0 \text{ for } v \in \Re^n, v \neq 0\}.$$

In order to update the similarity graphs based on information present in the cluster indicator subspaces, the approximate Laplacians L_m^rs are optimized over the SPD manifold, along with the subspaces $\mathsf{span}(U_{\text{Joint}})$ and $\mathsf{span}(U_m)$'s, which are optimized over the Grassmannian manifold. Modification of the Laplacians changes their edge connectivity, which in turn changes their inherent cluster structure. Accordingly, the contribution of the individual graphs in the fused network should also change. This motivates the optimization of graph weights α_m's of (7.4), as well. Incorporating these variables, the final optimization problem over two different manifolds is given by

$$\begin{aligned}
&\underset{\substack{\mathsf{span}(U_{\text{Joint}}),\, \mathsf{span}(U_m)\in\, \mathsf{Gr}(n,k)\\ L_m^r\in\mathcal{S}_{++}^n,\, \alpha_m\in\Re}}{\text{minimize}} \; -\frac{1}{2} tr\left(U_{\text{Joint}}^T\Big(\sum_{m=1}^{M}\alpha_m^{\kappa}L_m^r\Big)U_{\text{Joint}}\right)\\
&-\frac{1}{2M}\sum_{m=1}^{M}\left[tr(U_m^T L_m^r U_m)+tr\left(U_{\text{Joint}}U_{\text{Joint}}^T U_m U_m^T\right)\right]-\frac{1}{2M(M-1)}\sum_{\substack{i,j=1\\ i\neq j}}^{M} tr(U_iU_i^TU_jU_j^T),\\
&\text{such that } \alpha_m \geqslant 0,\ \sum_{m=1}^{M}\alpha_m = 1. \qquad (7.11)
\end{aligned}$$

In the above problem, while combining the Laplacians L_m^r's, their weights α_m's are raised to the power κ with $\kappa > 1$ as $\kappa = 1$ favors the trivial solution, where the v-th view with minimum loss has $\alpha_v = 1$, and 0 otherwise. The problem in (7.11) is solved iteratively by alternating optimization over Grassmannian manifold, SPD manifold, and the real-valued space. Note that, U_{Joint} and U_m are optimized over the Grassmannian manifold which is a quotient space formed by the action of $(k \times k)$ orthogonal rotation matrices. Hence, for a set of given $L_1^r, ..., L_M^r$ and $\alpha_1..., \alpha_M$, the objective in (7.10) should be rotation invariant in terms of U_{Joint} and U_ms.

Theorem 7.1. *$\boldsymbol{f}(U_{\text{Joint}}, U_1, ..., U_M)$ is rotation invariant.*

Proof. Let $R_{\text{Joint}}, R_1, \ldots, R_M \in O(k)$, the set of $(k \times k)$ orthogonal rotation matrices. So, it satisfies that

$$R_{\text{Joint}}R_{\text{Joint}}^T = \mathbf{I}_r \text{ and } R_mR_m^T = \mathbf{I}_r, \text{ for } m = 1, \ldots, M.$$

$$tr\left((U_mR_m)^T L_m^r (U_mR_m)\right) = tr\left(U_m^T L_m^r U_m R_m R_m^T\right) = tr\left(U_m^T L_m^r U_m\right). \qquad (7.12)$$

$$\text{Similarly, } tr\left((U_{\text{Joint}}R_{\text{Joint}})^T \mathbf{L}_{\text{Joint}}^r (U_{\text{Joint}}R_{\text{Joint}})\right) = tr\left(U_{\text{Joint}}^T L_{\text{Joint}}^r U_{\text{Joint}}\right). \qquad (7.13)$$

$$\text{Also, } U_mR_m(U_mR_m)^T = U_mU_m^T \quad \text{and } U_{\text{Joint}}R_{\text{Joint}}(U_{\text{Joint}}R_{\text{Joint}})^T = U_{\text{Joint}}U_{\text{Joint}}^T. \qquad (7.14)$$

Substituting (7.12), (7.13), and (7.14) in the function $\boldsymbol{f}$ of (7.10) gives

$$\boldsymbol{f}(U_{\text{Joint}}R_{\text{Joint}}, U_1R_1, ..., U_MR_M) = \boldsymbol{f}(U_{\text{Joint}}, U_1, ..., U_M).$$

■

7.3 Optimization Strategy

The manifold based formulation in (7.11) has two major advantages. First, manifolds are expected to better capture the underlying non-linear geometry of complex real-world data sets. The other advantage is that optimization over non-linear manifolds like Grassmannian and SPD manifolds does not require vector space assumptions on the search space. The gradient descent algorithm, which adds a multiple of descent direction to the previous iterate, obviously requires the structure of a vector space and is not possible on general manifolds. Optimization over manifolds is performed using line search [3] which substitutes the standard linear step in gradient descent by more general paths based on retractions [3]. It proceeds by projecting the negative gradient onto the tangent space of the manifold. The tangent space is essentially a vector space, which allows linear movement. Hence, a linear step is taken in the tangent space from the current iterate towards the projected gradient. Finally, retraction maps the updated point from tangent space back to the manifold. Let $\boldsymbol{f}$ denote the proposed joint objective function in (7.11), and $U^{(0)}_{\text{Joint}}$, $U^{(0)}_{m}$, $L^{r(0)}_{m}$, and $\alpha^{(0)}_{m}$, for $m \in \{1, \ldots, M\}$, denote the initial iterates for the respective variables. The optimization of (7.11) based on line-search is performed as follows.

7.3.1 Optimization over Grassmannian Manifold

Given a set of fixed U_m, L^r_m, α_m, for $m = 1, \ldots, M$, and the t-th iterate of U_{Joint}, denoted by $U^{(t)}_{\text{Joint}}$, let $\mathbb{U} = \sum\limits_{m=1}^{M} U_m U_m^T$. The negative gradient of $\boldsymbol{f}$ in (7.11) at $U^{(t)}_{\text{Joint}}$ is given by

$$\begin{aligned} -\nabla_{U^{(t)}_{\text{Joint}}} \boldsymbol{f} &= -\nabla_{U^{(t)}_{\text{Joint}}} - \frac{1}{2}\Bigg[- tr\Big(U^T_{\text{Joint}}\big(\sum_{m=1}^{M} \alpha^{\kappa}_{m} L^{r}_{m}\big) U_{\text{Joint}}\Big) - tr\big(U^T_{\text{Joint}}\big(\sum_{m=1}^{M} U_m U_m^T\big)\ U_{\text{Joint}}\big)\Bigg] \\ &= \big(\sum_{m=1}^{M} \alpha^{\kappa}_{m} L^{r}_{m} + \mathbb{U}\big) U^{(t)}_{\text{Joint}} = \boldsymbol{Q}^{(t)}_{\text{Joint}} \text{ (say).} \end{aligned} \tag{7.15}$$

In the notation of Grassmannian manifold, $\mathsf{Gr}(n, k)$, the parameters n and k are always fixed and are dropped for notational simplicity. Let $\mathcal{T}_{U^{(t)}_{\text{Joint}}}\mathsf{Gr}$ denote the tangent space of the Grassmannian manifold. The subscript $U^{(t)}_{\text{Joint}}$ implies that the tangent space has its origin at the Grassmannian point $\mathsf{span}(U^{(t)}_{\text{Joint}})$. Let $\Pi_X(Y)$ denote the projection of Y onto the tangent space of a manifold rooted at point X. The negative gradient $\boldsymbol{Q}^{(t)}_{\text{Joint}}$ is orthogonally projected on the tangent space $\mathcal{T}_{U^{(t)}_{\text{Joint}}}\mathsf{Gr}$ as follows [3]:

$$\Pi_{U^{(t)}_{\text{Joint}}}\left(\boldsymbol{Q}^{(t)}_{\text{Joint}}\right) = \left(\mathbf{I}_n - U^{(t)}_{\text{Joint}}\big(U^{(t)}_{\text{Joint}}\big)^T\right)\boldsymbol{Q}^{(t)}_{\text{Joint}} = \underset{\text{(say)}}{\boldsymbol{Z}^{(t)}_{\text{Joint}}}. \tag{7.16}$$

Given tangent $\boldsymbol{Z}^{(t)}_{\text{Joint}} \in \mathcal{T}_{U^{(t)}_{\text{Joint}}}\mathsf{Gr}$ and step size $\eta_{\mathsf{G}} > 0$, a linear step is taken within the tangent space from $U^{(t)}_{\text{Joint}}$ in the direction of $\boldsymbol{Z}^{(t)}_{\text{Joint}}$ as follows:

$$\boldsymbol{Z}^{(t+1)}_{\text{Joint}} = U^{(t)}_{\text{Joint}} + \eta_{\mathsf{G}} \boldsymbol{Z}^{(t)}_{\text{Joint}}. \tag{7.17}$$

Then the obtained point (7.17) is mapped from the tangent space back to the manifold. This is done by projective retraction, denoted by PGr, of the point $\boldsymbol{Z}_{\text{Joint}}^{(t+1)} \in \mathcal{T}_{U_{\text{Joint}}^{(t)}}\,\mathsf{Gr}$ back to manifold Gr. For the Grassmannian manifold, retraction is performed using singular value decomposition (SVD) of the representative matrix $\boldsymbol{Z}_{\text{Joint}}^{(t+1)}$ [3]. Let the SVD of $\boldsymbol{Z}_{\text{Joint}}^{(t+1)}$ be given by

$$\boldsymbol{Z}_{\text{Joint}}^{(t+1)} = E_{\text{Joint}}^{(t+1)}\ \Xi_{\text{Joint}}^{(t+1)}\ \left(V_{\text{Joint}}^{(t+1)}\right)^T,$$

where $E_{\text{Joint}}^{(t+1)}$ and $V_{\text{Joint}}^{(t+1)}$ are orthonormal matrices of left and right singular vectors of $\boldsymbol{Z}_{\text{Joint}}^{(t+1)}$, respectively, while $\Xi_{\text{Joint}}^{(t+1)}$ contains the singular values in its diagonal. The retractive projection is given by

$$\mathsf{PGr}_{U_{\text{Joint}}^{(t)}}\left(\boldsymbol{Z}_{\text{Joint}}^{(t+1)}\right) = \mathsf{span}\left(E_{\text{Joint}}^{(t+1)}\ \left(V_{\text{Joint}}^{(t+1)}\right)^T\right). \tag{7.18}$$

The point obtained after retraction in (7.18) becomes the next iterate of U_{Joint}, that is

$$\mathsf{span}\left(U_{\text{Joint}}^{(t+1)}\right) = \mathsf{PGr}_{U_{\text{Joint}}^{(t)}}\left(\boldsymbol{Z}_{\text{Joint}}^{(t+1)}\right). \tag{7.19}$$

Figure 7.3 shows the diagrammatic representation of a single step of line-search optimization over the Grassmanian manifold. In Figure 7.3, the curved surface denotes the manifold. The transparent stripped plane denotes the tangent space of the manifold, rooted at the current iterate $\mathsf{span}\left(U_{\text{Joint}}^{(t)}\right)$, denoted by the point lying at the intersection of the plane

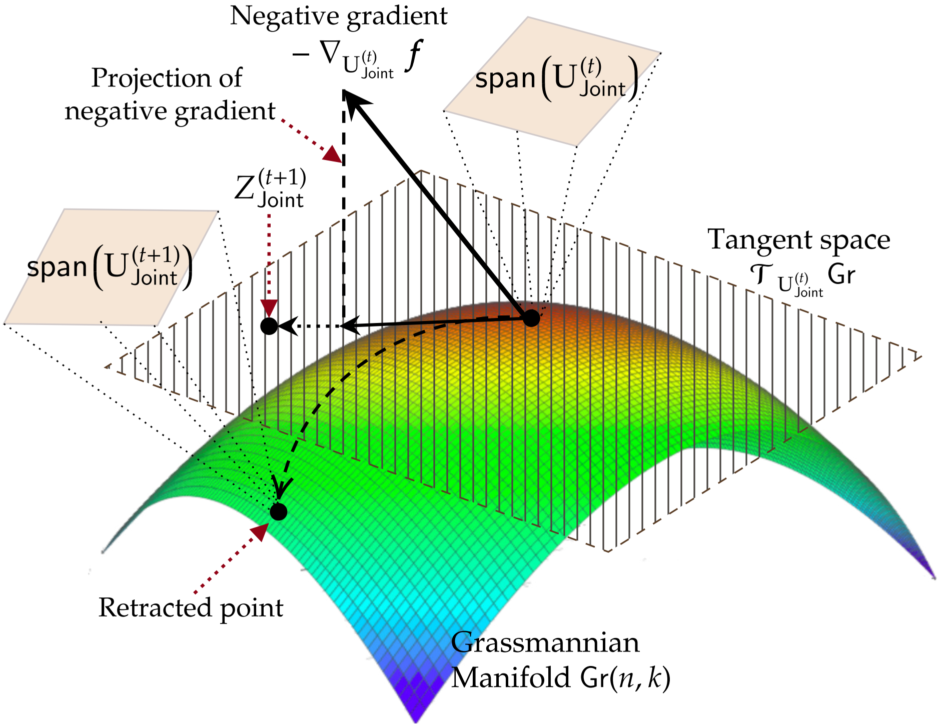

Negative gradient
$-\nabla_{\mathrm{U}_{\text{Joint}}^{(t)}} f$
Projection of
negative gradient
$\mathsf{span}\left(\mathrm{U}_{\text{Joint}}^{(t)}\right)$
$Z_{\text{Joint}}^{(t+1)}$
$\mathsf{span}\left(\mathrm{U}_{\text{Joint}}^{(t+1)}\right)$
Tangent space
$\mathcal{T}_{\mathrm{U}_{\text{Joint}}^{(t)}}\mathsf{Gr}$
Retracted point
Grassmannian
Manifold $\mathsf{Gr}(n,k)$

Figure 7.3: Optimization of U_{Joint} over the Grassmannian manifold.

and the manifold. That point is actually a k-dimensional linear subspace denoted by the shaded plane connected to the point using dotted lines. The vector pointing outwards from the tangent plane is the negative gradient direction, and its perpendicular projection $\boldsymbol{Z}_{\text{Joint}}^{(t)}$ lies on the tangent plane. A small step is taken in the tangent plane in the direction of the projected gradient, marked by the horizontal dotted line in Figure 7.3. The obtained point is then retracted back to the manifold. The retracted point $\mathsf{span}\left(U_{\text{Joint}}^{(t+1)}\right)$ lies on the curved surface. It is also a linear subspace denoted by the second shaded plane. The following theorem proves that the next iterate obtained by retraction in (7.19) belongs to the Grassmannian manifold.

Theorem 7.2. $\mathsf{span}\left(U_{\text{Joint}}^{(t+1)}\right)$ *belongs to the Grassmannian manifold.*

Proof. According to (7.5), for $\mathsf{span}\left(U_{\text{Joint}}^{(t+1)}\right)$ to belong to the Grassmanian manifold, $U_{\text{Joint}}^{(t+1)}$ should be orthonormal. From (7.18) and (7.19) we get

$$\left(U_{\text{Joint}}^{(t+1)}\right)^T U_{\text{Joint}}^{(t+1)} = V_{\text{Joint}}^{(t+1)}\left(E_{\text{Joint}}^{(t+1)}\right)^T E_{\text{Joint}}^{(t+1)}\left(V_{\text{Joint}}^{(t+1)}\right)^T = \mathbf{I}_k,$$

as $E_{\text{Joint}}^{(t+1)}$ and $V_{\text{Joint}}^{(t+1)}$ contain left and right singular vectors of $\boldsymbol{Z}_{\text{Joint}}^{(t+1)}$, respectively, and are therefore orthonormal. Hence, $U_{\text{Joint}}^{(t+1)}$ is also orthonormal. ∎

The algorithm for computing a single iteration of U_{Joint} over the Grassmannian manifold is given in Algorithm 7.1.

Algorithm 7.1 `Optimize_U`$_{\text{Joint}}$

```
▷ Optimization of $U_{\text{Joint}}$ over Grassmannian manifold $\mathsf{Gr}(n,k)$
Input: Cluster indicator subspaces $U_m$ and weights $\alpha_m$, for $m = 1,...,M$, joint Laplacian $\mathbf{L}_{\text{Joint}}^r$, joint subspace $U_{\text{Joint}}^{(t)}$ of iteration $t$, step size $\eta_{\mathsf{G}} > 0$.
Output: $U_{\text{Joint}}^{(t+1)}$.
1: Compute negative gradient $\boldsymbol{Q}_{\text{Joint}}^{(t)} \leftarrow \left[-\nabla_{U_{\text{Joint}}^{(t)}} \boldsymbol{f}\right]$ using (7.15).
2: Project negative gradient onto tangent space:
       $\boldsymbol{Z}_{\text{Joint}}^{(t)} \leftarrow \Pi_{U_{\text{Joint}}^{(t)}}\left(\boldsymbol{Q}_{\text{Joint}}^{(t)}\right)$ using (7.16).
3: $\boldsymbol{Z}_{\text{Joint}}^{(t+1)} \leftarrow U_{\text{Joint}}^{(t)} + \eta_{\mathsf{G}}\boldsymbol{Z}_{\text{Joint}}^{(t)}$.
4: Find retractive projection $\mathsf{PGr}_{U_{\text{Joint}}^{(t)}}\left(\boldsymbol{Z}_{\text{Joint}}^{(t+1)}\right)$ using (7.18).
5: Next iterate: $\mathsf{span}\left(U_{\text{Joint}}^{(t+1)}\right) \leftarrow \mathsf{PGr}_{U_{\text{Joint}}^{(t)}}\left(\boldsymbol{Z}_{\text{Joint}}^{(t+1)}\right)$.
6: Return $U_{\text{Joint}}^{(t+1)}$.
```

Similar to the joint subspace, $\mathsf{span}\left(U_{\text{Joint}}\right)$, the individual subspaces $\mathsf{span}\left(U_m\right)$s are also elements of the Grassmannian manifold. For a specific $m \in \{1,\ldots,M\}$, let U_{Joint}, weights α_j, Laplacians $L_j^r, \forall j \in \{1,\ldots,M\}$, and all other U_i's to be fixed for $i \in \{1,\ldots,M\}$, such that $i \neq m$. For that view X_m, let $U_m^{(t)}$ denote the representative matrix corresponding to cluster indicator subspace obtained at iteration t. The negative gradient of $\boldsymbol{f}$ at point $U_m^{(t)}$

is given by

$$-\nabla_{U_m^{(t)}} \boldsymbol{f} = -\nabla_{U_m^{(t)}} \frac{1}{2}\left[-tr\left(U_m^T \left(L_m^r + U_{\text{Joint}} U_{\text{Joint}}^T + \sum_{\substack{j=1 \\ j \neq m}}^{M} U_j U_j^T \right) U_m \right) \right]$$

$$= \left(L_m^r + U_{\text{Joint}} U_{\text{Joint}}^T + \sum_{\substack{j=1 \\ j \neq m}}^{M} U_j U_j^T \right) U_m^{(t)} = \boldsymbol{Q}_m^{(t)} \text{ (say)}. \tag{7.20}$$

Similar to (7.16), projection of negative gradient onto tangent space of $\mathsf{span}\left(U_m^{(t)}\right)$ is given by

$$\Pi_{U_m^{(t)}} \left(\boldsymbol{Q}_m^{(t)} \right) = \left(\mathbf{I}_n - U_m^{(t)} \left(U_m^{(t)} \right)^T \right) \boldsymbol{Q}_m^{(t)} = \boldsymbol{Z}_m^{(t)} \text{ (say)}. \tag{7.21}$$

Then, a linear step in the tangent space in the direction of projected negative gradient is taken as follows:

$$\boldsymbol{Z}_m^{(t+1)} = U_m^{(t)} + \eta_{\mathsf{G}} \boldsymbol{Z}_m^{(t)}.$$

After the linear step, retractive projection maps the obtained point $\boldsymbol{Z}_m^{(t+1)}$ back to the manifold, as follows:

$$\mathsf{PGr}_{U_m^{(t)}} \left(\boldsymbol{Z}_m^{(t+1)} \right) = \mathsf{span} \left(E_m^{(t+1)} \left(V_m^{(t+1)} \right)^T \right), \tag{7.22}$$

where $E_m^{(t+1)}$ and $V_m^{(t+1)}$ are orthonormal matrices containing left and right singular vectors of $\boldsymbol{Z}_m^{(t+1)}$, respectively. As before, the retracted point in (7.22) becomes the next iterate of U_m, that is,

$$\mathsf{span} \left(U_m^{(t+1)} \right) = \mathsf{PGr}_{U_m^{(t)}} \left(\boldsymbol{Z}_m^{(t+1)} \right).$$

It follows from Theorem 7.2 that $\mathsf{span}\left(U_m^{(t+1)}\right)$ belongs to the Grassmannian manifold. The pseudocode for a single iteration of U_m over the Grassmannian manifold is given in Algorithm 7.2. The same follows for each of the M views.

7.3.2 Optimization over SPD Manifold

Apart from the indicator subspaces, there are M other variables L_m^rs, each corresponding to the similarity graph of one of the views. These L_m^rs are optimized over the SPD manifold $\mathcal{S}_{++}^n$. For a specific m, assume that the indicator subspaces $\mathsf{span}(U_{\text{Joint}})$, $\mathsf{span}(U_j)$, weights α_j, for $j = 1, ..., M$, and the shifted Laplacians L_i^r are fixed for $i = 1, ..., M$, and $i \neq m$. Let $L_m^{r(t)}$ denote the t-th iterate of L_m^r. The negative gradient of $\boldsymbol{f}$ with respect to $L_m^{r(t)}$ is

Algorithm 7.2 `Optimize_U`$_{\text{m}}$

```
▷ Optimization of $U_m$ over Grassmannian manifold $\mathsf{Gr}(n,k)$
Input: Joint subspace $U_{\text{Joint}}$, other individual subspaces $U_j$ for $j = 1, ..., M, j \neq m$, Laplacian $\mathbf{L}^r_m$, subspace $U^{(t)}_m$ of iteration $t$, step size $\eta_{\mathsf{G}}$.
Output: $U^{(t+1)}_m$.
1: Compute negative gradient $\boldsymbol{Q}^{(t)}_m \leftarrow \left[-\nabla_{U^{(t)}_m} \boldsymbol{f}\right]$ by (7.20).
2: Project negative gradient onto tangent space
       $\boldsymbol{Z}^{(t)}_m \leftarrow \Pi_{U^{(t)}_m}\left(\boldsymbol{Q}^{(t)}_m\right)$ using (7.21).
3: $\boldsymbol{Z}^{(t+1)}_{\text{Joint}} \leftarrow U^{(t)}_{\text{Joint}} + \eta_{\mathsf{G}} \boldsymbol{Z}^{(t)}_{\text{Joint}}$.
4: Find retractive projection $\mathsf{PGr}_{U^{(t)}_m}\left(\boldsymbol{Z}^{(t+1)}_m\right)$ using (7.22).
5: Next iterate: $\mathsf{span}\left(U^{(t+1)}_m\right) \leftarrow \mathsf{PGr}_{U^{(t)}_m}\left(\boldsymbol{Z}^{(t+1)}_m\right)$.
6: Return $U^{(t+1)}_m$.
```

given by

$$-\nabla_{L^{r(t)}_m} \boldsymbol{f} = -\nabla_{L^{r(t)}_m} \frac{1}{2}\Big[-\alpha^\kappa_m tr\left(U^T_{\text{Joint}} L^r_m U_{\text{Joint}}\right) - tr\left(U^T_m L^r_m U_m\right)\Big]$$
$$= \left(\alpha^\kappa_m U_{\text{Joint}} U^T_{\text{Joint}} + U_m U^T_m\right) = \boldsymbol{Q}^{(t)}_{Lm} \text{ (say)}. \tag{7.23}$$

The tangent space of the manifold of SPD matrices is the set of symmetric matrices. Therefore, the projection of $\boldsymbol{Q}^{(t)}_{Lm}$ onto the tangent space of SPD manifold is given by [74]

$$\Pi_{L^{r(t)}_m}\left(\boldsymbol{Q}^{(t)}_{Lm}\right) = L^{r(t)}_m \mathtt{symm}\left(\boldsymbol{Q}^{(t)}_{Lm}\right) L^{r(t)}_m = \boldsymbol{Z}^{(t)}_{Lm}, \text{ (say)} \tag{7.24}$$

where $\mathtt{symm}(A) = \frac{(A+A^T)}{2}$. The symmetrization of $\boldsymbol{Q}^{(t)}_{Lm}$ is mathematically unnecessary as its structure in (7.23) implies that it would always be symmetric. Next, a linear step in taken in the tangent space of SPD manifold from $L^{r(t)}_m$ towards the projected gradient $\boldsymbol{Z}^{(t)}_{Lm}$ with step size $\eta_{\mathcal{S}} > 0$ as follows:

$$\boldsymbol{Z}^{(t+1)}_{Lm} = L^{r(t)}_m + \eta_{\mathcal{S}} \boldsymbol{Z}^{(t)}_{Lm}. \tag{7.25}$$

It can be shown from the following theorem that the obtained point $\boldsymbol{Z}^{(t+1)}_{Lm}$ in the tangent space itself belongs to the SPD manifold. This property is attributed to the form of the negative gradient in (7.23).

Theorem 7.3. *$\boldsymbol{Z}^{(t+1)}_{Lm}$ belongs to the SPD manifold.*

Proof. To prove the belongingness of $\boldsymbol{Z}^{(t+1)}_{Lm}$ to the SPD manifold, we first show that the tangent space point $\boldsymbol{Z}^{(t)}_{Lm}$ is symmetric and positive semi-definite. The U_{Joint} and U_m are points on the Grassmannian manifold and are both orthonormal matrices of order $(n \times k)$.

So, we can write

$$U_m U_m^T = U_m \mathbf{I}_k U_m^T \text{ and } U_{\text{Joint}} U_{\text{Joint}}^T = U_{\text{Joint}} \mathbf{I}_k U_{\text{Joint}}^T.$$

It follows from above that both $U_m U_m^T$ and $U_{\text{Joint}} U_{\text{Joint}}^T$ have k eigenvalues, each equals to 1. Hence, they are symmetric positive definite matrices. For any $z \neq 0$ in $\Re^n$, from (7.23), we have

$$\begin{gathered} z^T \boldsymbol{Q}_{Lm}^{(t)} z = z^T U_m U_m^T z + \alpha_m z^T U_{\text{Joint}} U_{\text{Joint}}^T z, \\ \text{where } z^T U_m U_m^T z > 0, z^T U_{\text{Joint}} U_{\text{Joint}}^T z > 0, \text{ and } \alpha_m \geqslant 0. \\ \text{Hence, } z^T \boldsymbol{Q}_{Lm}^{(t)} z > 0. \end{gathered} \tag{7.26}$$

So, the negative gradient $\boldsymbol{Q}_{Lm}^{(t)}$ is symmetric positive definite. Again, for any $z \neq 0$ in $\Re^n$, from (7.24), we have that

$$\begin{gathered} z^T \boldsymbol{Z}_{Lm}^{(t)} z = z^T L_m^{r(t)} \boldsymbol{Q}_{Lm}^{(t)} L_m^{r(t)} z = y^T \boldsymbol{Q}_{Lm}^{(t)} y > 0, \\ \text{where } y = L_m^{r(t)} z \neq 0 \Leftrightarrow z \neq 0. \end{gathered}$$

So, the projected gradient is also positive definite. While taking the linear step in the tangent space, the obtained point $\boldsymbol{Z}_{Lm}^{(t+1)}$ in (7.25) is the sum of two symmetric positive definite matrices, which similar to (7.26) is symmetric positive definite. Therefore, $\boldsymbol{Z}_{Lm}^{(t+1)}$ belongs to the SPD manifold. ■

As $\boldsymbol{Z}_{Lm}^{(t+1)} \in \mathcal{S}_{++}^n$, therefore, the retraction of $\boldsymbol{Z}_{Lm}^{(t+1)}$ from the tangent space of $\mathcal{S}_{++}^n$ to the manifold $\mathcal{S}_{++}^n$ is not required. This prevents the computationally expensive matrix exponential based retraction [74] step in case of SPD manifold optimization. Hence, $\boldsymbol{Z}_{Lm}^{(t+1)}$ itself is the next iterate of L_m^r, that is

$$L_m^{r(t+1)} = \boldsymbol{Z}_{Lm}^{(t+1)}.$$

The algorithm for a single update of L_m^r over the SPD manifold is given in Algorithm 7.3.

7.3.3 Optimization of Graph Weights

Considering all other variables fixed, the optimization problem of (7.11) in terms of the network weights α_m's is given by

$$\begin{gathered} \underset{\alpha_m \in \Re}{\text{minimize}} \sum_{m=1}^{M} -\alpha_m^{\kappa} \mathfrak{g}_m \text{ such that } \alpha_m \geqslant 0, \sum_{m=1}^{M} \alpha_m = 1, \\ \text{where } \mathfrak{g}_m = \frac{1}{2} tr\Big(U_{\text{Joint}}^T L_m^r U_{\text{Joint}} \Big). \end{gathered} \tag{7.27}$$

Algorithm 7.3 `Optimize_Lm`

```
▷ Optimization of $L^r_m$ over SPD manifold
Input: Indicators $U_{\text{Joint}}$ and $U_m$, Laplacian $L^{r(t)}_m$ of iteration $t$, step size $\eta_{\mathcal{S}} > 0$, weight factor $\alpha_m$.
Output: $L^{r(t+1)}_m$.
1: Compute negative gradient $\boldsymbol{Q}^{(t)}_{L_m} \leftarrow -\nabla_{L^{r(t)}_m} \boldsymbol{f}$ using (7.23).
2: Project negative gradient onto tangent space:
       $\boldsymbol{Z}^{(t)}_{L_m} \leftarrow L^{r(t)}_m \texttt{symm}\left(\boldsymbol{Q}^{(t)}_{L_m}\right) L^{r(t)}_m$ using (7.24).
3: Take linear step $\boldsymbol{Z}^{(t+1)}_{L_m} \leftarrow L^{r(t)}_m + \eta_{\mathcal{S}} \boldsymbol{Z}^{(t)}_{L_m}$.
4: Next iterate: $L^{r(t+1)}_m \leftarrow \boldsymbol{Z}^{(t+1)}_{L_m}$.
5: Return $L^{r(t+1)}_m$.
```

Ignoring the non-negativity constraints, the Lagrangian of the above problem in given by

$$\sum_{m=1}^{M} -\alpha_m^{\kappa} \mathfrak{g}_m + \xi\Big(\sum_{m=1}^{M} \alpha_m - 1\Big),$$

where ξ is the Lagrange multiplier. Taking the derivative of the Lagrangian on α_m and setting it to 0 gives

$$\begin{aligned} &-\kappa\alpha_m^{\kappa-1}\mathfrak{g}_m + \xi = 0, \\ \Rightarrow &\alpha_m = \left(\frac{\xi}{\kappa\mathfrak{g}_m}\right)^{\frac{1}{\kappa-1}} = \xi^{\frac{1}{\kappa-1}} \left(\kappa\mathfrak{g}_m\right)^{\frac{1}{1-\kappa}}. \end{aligned} \tag{7.28}$$

Since $\sum\limits_{m=1}^{M} \alpha_m = 1$, therefore, we get

$$\sum_{m=1}^{M} \xi^{\frac{1}{\kappa-1}} \left(\kappa\mathfrak{g}_m\right)^{\frac{1}{1-\kappa}} = 1, \qquad \Rightarrow \xi^{\frac{1}{\kappa-1}} = \frac{1}{\sum\limits_{m=1}^{M} \left(\kappa\mathfrak{g}_m\right)^{\frac{1}{1-\kappa}}}.$$

Substituting the value of ξ in (7.28) gives

$$\alpha_m = \frac{\left(\mathfrak{g}_m\right)^{\frac{1}{1-\kappa}}}{\sum\limits_{m=1}^{M} \left(\mathfrak{g}_m\right)^{\frac{1}{1-\kappa}}}. \tag{7.29}$$

In the above deduction, the non-negativity constraints on α_m are neglected. Nevertheless, the positive semi-definite property of L^r_m implies that $\mathfrak{g}_m \geqslant 0$. So, the derived expression of α_m in (7.29) automatically satisfies the non-negativity constraint.

7.3.4 Proposed Algorithm

Given M graphs with affinity matrices $W_1, \ldots, W_M$, corresponding to M views $X_1, \ldots, X_M$ and rank parameter $r \geqslant k$, the proposed GeARS algorithm extracts a low-rank subspace $\mathsf{span}(U_{\text{Joint}})$ that reflects the multi-view consensus clusters of the data set. Then, k-means on the rows of U_{Joint} identifies the final clusters. Similar to Chapter 6, the L_m^r's are initialized to the rank r approximations of the individual graph Laplacians, while the individual cluster indicator subspaces U_m's are initialized to the orthonormal matrices containing k largest eigenvectors of corresponding Laplacians. The initial view weights, $\alpha_{(m)}^{(0)}$'s, are determined based on eigenvalues of the corresponding Laplacians, as done in previously in Chapters 5 and 6. The initial weights are given by $\alpha_{(m)}^{(0)} = \frac{tr\left(\Sigma_{(m)}^k\right)}{\Delta^m}$, with $\Delta \geqslant 1$, where $\Sigma_{(m)}^k$ denotes the m-th largest order statistic of the sequence $\Sigma_1^k, ..., \Sigma_M^k$, and $\alpha_{(m)}$ denotes the weight corresponding to the view having the m-th largest order statistic. Given initializations of Laplacians, $L_m^{r(0)}$, their weights, $\alpha_m^{(0)}$, and $\kappa > 1$, the initial iterate for joint cluster indicator, $U_{\text{Joint}}^{(0)}$, is set to the k largest eigenvectors of the matrix $\sum_{m=1}^M (\alpha_m^{(0)})^\kappa L_m^{r(0)}$. The proposed GeARS algorithm is described in Algorithm 7.4.

7.3.4.1 Convergence

Let the proposed objective function $\boldsymbol{f}$ in (7.11), evaluated at the t-th iterate, be denoted by $\boldsymbol{f}^{(t)}$. Since the movement on each manifold is directed towards the negative gradient, which is a descent direction, the line-search ensures that there would be a reduction in objective function at each iteration. However, similar to Chapter 6, the algorithm proceeds to the next set of iterates only when the reduction is sufficient determined based on the Armijo convergence criterion [9]. If not, then the step lengths η_{G} and $\eta_{\mathcal{S}}$ are iteratively decreased by a factor of $\delta \in (0, 1)$ until an iterate is obtained with sufficient reduction. The proposed algorithm converges to a local optima when the difference between the value of $\boldsymbol{f}$ in two consecutive iterations t and $(t+1)$ falls below a given threshold ϵ, that is, $\boldsymbol{f}^{(t)} - \boldsymbol{f}^{(t+1)} < \epsilon$.

The proposed GeARS algorithm performs line search optimization over Riemannian manifolds. The convergence of line search over Riemannian manifolds is theoretically established in Chapter 6 and in [3]. The convergence result reported in Theorem 6.3 of Chapter 6 analogically holds for the proposed GeARS algorithm as well. It states that only critical points of the cost function where the gradient of $\boldsymbol{f}$ vanishes can be accumulation points of the sequence of iterates generated by the proposed algorithm. However, it does not necessarily guarantee that the obtained optimal solution is a local minimizer, and not a saddle point. Nevertheless, as the line-search at each iteration is directed towards the negative gradient, unless the initial iterate $U_{\text{Joint}}^{(0)}$ is specifically designed, Algorithm 7.4 is unlikely to produce sequences whose accumulation points are not local minima of the cost function.

7.3.4.2 Computational Complexity

Similar to the MiMIC algorithm of Chapter 6, the GeARS algorithm also starts by extracting a low-rank cluster indicator subspace corresponding to each view. This involves computing the $(n \times n)$ graph Laplacian of each view and it's $\mathcal{O}(n^3)$ eigendecomposition.

Algorithm 7.4 Proposed Algorithm: **GeARS**

```
Input: Similarity matrices $W_1, \ldots, W_M$, clusters $k$, rank parameter $r \geqslant k$, $\kappa > 1$, step
    sizes $\eta_{\mathsf{G}}, \eta_{\mathcal{S}} > 0$, convergence parameters $\epsilon > 0$ and $\delta \in (0, 1)$.
Output: Multi-view clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$.
 1: for each view $m \leftarrow 1$ to $M$ do
 2:     Construct graph Laplacian $L_m$ as in (7.1).
 3:     Compute eigen-decomposition of $L_m$.
 4:     $L^r_m \leftarrow$ Rank $r$ approximation of $L_m$.
 5:     $V^k_m \leftarrow k$ largest eigenvectors of $L_m$.
 6:     Compute graph weight $\alpha_m$ based on $k$ largest eigenvalues of $L_m$.
 7: end for
 8: Compute $\mathbf{L}^{r\kappa}_{\text{Joint}} \leftarrow \sum_{m=1}^{M} \alpha^{\kappa}_m L^r_m$.
 9: Compute eigen-decomposition of $\mathbf{L}^{r\kappa}_{\text{Joint}}$ and store the $k$ largest eigenvectors in $U^k_{\text{Joint}}$.
10: Initialize variables: $U^{(0)}_{\text{Joint}} \leftarrow U^k_{\text{Joint}}$, $U^{(0)}_m \leftarrow V^k_m$, $L^{r(0)}_m \leftarrow L^r_m$, $\alpha^{(0)}_m \leftarrow \alpha_m$,
                        for each $m = 1, .., M$.
11: $t \leftarrow 0$; $\boldsymbol{f}^{(0)} \leftarrow \boldsymbol{f}\left(U^{(0)}_{\text{Joint}}, U^{(0)}_m, L^{r(0)}_m, \alpha^{(0)}_m, \forall m = 1, ..., M\right)$.
12: do
13:     $U^{(t+1)}_{\text{Joint}} \leftarrow \texttt{Optimize\_U}_{\text{Joint}}\left(U^{(t)}_{\text{Joint}}, L^{r(t)}_j, U^{(t)}_j, \alpha^{(t)}_j, \forall j = 1, ..., M, \eta_{\mathsf{G}}\right)$.
14:     $U^{(t+1)}_m \leftarrow \texttt{Optimize\_U}_{\text{m}}\left(L^{r(t)}_m, U^{(t+1)}_{\text{Joint}}, \alpha^{(t)}_m, U^{(t)}_m, U^{(t)}_j, j \neq m, \eta_{\mathsf{G}}\right)$
                 for each $m \in \{1, .., M\}$.
15:     $L^{r(t+1)}_m \leftarrow \texttt{Optimize\_Lm}\left(L^{r(t)}_m, \alpha^{(t)}_m, U^{(t+1)}_{\text{Joint}}, U^{(t+1)}_m, \eta_{\mathcal{S}}\right)$
                 for each $m \in \{1, .., M\}$.
16:     Update $\alpha^{(t+1)}_m$ according to (7.29), $\forall m \in \{1, .., M\}$.
17:     Compute $\boldsymbol{f}^{(t+1)} \leftarrow \boldsymbol{f}\left(U^{(t+1)}_{\text{Joint}}, U^{(t+1)}_m, L^{r(t+1)}_m, \alpha^{(t+1)}_m, \forall m = 1, ..., M\right)$.
18:     if $\left(\boldsymbol{f}^{(t)} - \boldsymbol{f}^{(t+1)}\right) > \epsilon$ then
19:         Move to next set of iterates: $t = t + 1$.
20:     else
21:         Decrease step size by $\delta$: $\eta_{\mathsf{G}} = \delta\eta_{\mathsf{G}}$,   $\eta_{\mathcal{S}} = \delta\eta_{\mathcal{S}}$.
22:     end if
23: while ($\eta_{\mathsf{G}} > 1e-03$ & $\eta_{\mathcal{S}} > 1e-03$)
24: Optimal joint subspace: $U^{\star}_{\text{Joint}} \leftarrow U^{(t+1)}_{\text{Joint}}$.
25: Perform $k$-means clustering on the rows of $U^{\star}_{\text{Joint}}$.
26: Return clustering $\mathcal{C}_1, \ldots, \mathcal{C}_k$ from $k$-means.
```

For M views, the individual subspace construction in steps 1–7 takes $\mathcal{O}(Mn^3)$ time with sequential operation. The computation of joint Laplacian and its eigendecomposition in steps 8 and 9, respectively, takes atmost $\mathcal{O}(n^3)$ time. The initialization steps in 10 and 11, take constant $\mathcal{O}(1)$ time.

In each iteration of the gradient based line-search in steps 12-23, the indicator subspaces U_{Joint} and U_j are optimized over the Grassmannian manifold. The Grassmannian manifold is a quotient manifold of the Stiefel manifold. Hence, computing a single iterate over the Grassmannian manifold has the same complexity as that for the Stiefel manifold, as in Chapter 6. The SVD based retraction over Grassmannian and Stiefel manifolds takes $\mathcal{O}(n^2k)$ time for steps 13 and 14. The L^r_m optimization in step 15 does not involve the matrix exponential based retraction operation, as established in Theorem 7.3. So, it has a time complexity of $\mathcal{O}(n^2)$ instead of $\mathcal{O}(n^3)$. The graph weight updation in step 16 takes $\mathcal{O}(1)$ time. The computation of the joint objective in step 17 takes $\mathcal{O}(Mn^2r)$ time. The evaluation of convergence criteria and variable updation in steps 18–22 takes $\mathcal{O}(1)$ time. Assuming that the algorithm takes t iterations to converge, the overall complexity of steps 12–23 is bounded by $\mathcal{O}\left(t\left(n^2k + Mn^2k + Mn^2 + 1\right)\right) = \mathcal{O}\left(tMn^2k\right)$. The clustering on the final solution $U^\star_{\text{Joint}}$ in step 25 takes $\mathcal{O}(t_{km}nk^2)$ time, where t_{km} is the maximum number of iterations k-means clustering executes.

Hence, the overall computational complexity of the proposed GeARS algorithm, to extract the subspace $U^\star_{\text{Joint}}$ and perform clustering, is $(\mathcal{O}(Mn^3 + tMn^2k + t_{km}nk^2) =)\mathcal{O}\left(\max\{Mn^3 + tMn^2k\}\right)$, assuming $M, r, k << n$.

7.3.4.3 Asymptotic Convergence Bound

The asymptotic behavior of the proposed algorithm is analyzed to obtain a convergence bound that indicates how fast the algorithm arrives at a local optima starting from a random initial iterate. For a sufficiently large value of iteration number t, the difference between the cost function $\boldsymbol{f}$ evaluated at $U^{(t+1)}_{\text{Joint}}$ and at the optimal solution $U^\star_{\text{Joint}}$ can be upper bounded in terms of the difference in $\boldsymbol{f}$ evaluated at $U^{(t)}_{\text{Joint}}$ and $U^\star_{\text{Joint}}$. The bound involves eigenvalues of the Hessian of $\boldsymbol{f}$ at the optimal solution. Given a set of fixed subspaces U_m's, Laplacians L^r_m's, and weights α_m's, for $m = 1, ..., M$, the proposed objective function $\boldsymbol{f}$ becomes a function of only U_{Joint}, given by

$$\boldsymbol{f}(U_{\text{Joint}}) = -tr\left(U^T_{\text{Joint}} \sum_{j=1}^{M} \left(\alpha^\kappa_m L^r_m + U_m U^T_m\right) U_{\text{Joint}}\right).$$

The above function has its form equivalent to that of the Rayleigh quotient function, given by $\boldsymbol{f}(U_{\text{Joint}}) = tr\left(U^T_{\text{Joint}}\ \boldsymbol{\Xi}\ U_{\text{Joint}}\right)$, where

$$\boldsymbol{\Xi} = -\sum_{j=1}^{M}\left(\alpha^\kappa_m L^r_m + U_m U^T_m\right) \text{ and } U_{\text{Joint}} \in \Re^{n\times k}. \tag{7.30}$$

Let $\lambda_1 \leqslant \ldots \leqslant \lambda_k \leqslant \lambda_{k+1} \leqslant \ldots \leqslant \lambda_n$ be the eigenvalues of $\boldsymbol{\Xi}$. Also, let the Hessian of $\boldsymbol{f}$ at the optimal solution be denoted by $\mathbf{H}_{U^\star_{\text{Joint}}}\boldsymbol{f}$, and $\lambda_{\mathbf{H},\max}$ and $\lambda_{\mathbf{H},\min}$, respectively, be the maximum and minimum eigenvalues of the Hessian matrix $\mathbf{H}_{U^\star_{\text{Joint}}}\boldsymbol{f}$. For the Rayleigh quotient form, these two eigenvalues are given by (Section 4.9 of [3])

$$\lambda_{\mathbf{H},\max} = \lambda_n - \lambda_1 \quad \text{and} \quad \lambda_{\mathbf{H},\min} = \lambda_{k+1} - \lambda_k. \tag{7.31}$$

Similar to the asymptotic analysis reported in Section 6.4.2 of Chapter 6, the asymptotic bound for the proposed model is given as follows. There exists an iteration number $t' \geqslant 0$ such that

$$\boldsymbol{f}\left(U^{(t+1)}_{\text{Joint}}\right) - \boldsymbol{f}\left(U^\star_{\text{Joint}}\right) \leqslant c\left(\boldsymbol{f}\left(U^{(t)}_{\text{Joint}}\right) - \boldsymbol{f}\left(U^\star_{\text{Joint}}\right)\right),$$

for all $t \geqslant t'$, where

$$c = 1 - 2\sigma(\lambda_{k+1} - \lambda_k) \min \left\{ \eta_{\mathsf{G}}, \frac{2\beta(1-\sigma)}{(\lambda_n - \lambda_1)} \right\}. \tag{7.32}$$

Here, σ and β are Armijo parameters, and η_{G} is the Grassmannian step size.

The bound c in (7.32) determines the relative decrease in the value of the cost function $\boldsymbol{f}$ from iteration number t to $(t+1)$, for large values of t. In case of negligible reduction in $\boldsymbol{f}$ the value of c is close to 1, while smaller values indicate higher reduction in cost function. The Ξ matrix in (7.30), whose eigenvalues determine the convergence factor c, has its form similar to Laplacian matrix. Hence, similar to the graph Laplacian, in case of k well-separated clusters, the Ξ matrix is expected to have a greater gap between the eigenvalues λ_k and λ_{k+1}. This results in a smaller value of c and faster convergence to the local minima. In case of poor inter-cluster separation, the gap $(\lambda_{k+1} - \lambda_k)$ also reduces resulting in c being close to 1 and slower convergence. Hence, the convergence factor c can predict separation between the clusters in a data set.

7.4 Grassmannian Disagreement Bounds

The Grassmannian distance $d_\theta(U_{\text{Joint}}, U_m)$ in (7.8) quantifies the disagreement between the joint cluster indicator subspace $\mathsf{span}(U_{\text{Joint}})$ and the indicator subspace $\mathsf{span}(U_m)$ corresponding to the m-th view. The disagreement is given by the sum of the squared principal sines of k angles between the two subspaces. In order to impose consistency between the clusterings reflected in different views, the disagreement between the joint and each of the individual indicator subspaces is minimized in the proposed formulation. This subsection uses matrix perturbation theory [202] to derive an upper bound on the Grassmannian distance $d_\theta(U_{\text{Joint}}, U_m)$ between the joint and an individual subspace, at any given iteration t of the proposed GeARS algorithm.

Let $U_{\text{Joint}}^{(t-1)}$, $U_m^{(t-1)}$, and $L_m^{r(t-1)}$ denote the values of the corresponding variables at iteration $(t-1)$ of the proposed algorithm (Algorithm 7.4). Without loss of generality, it is assumed that the views are equally weighted, that is, $\alpha_m^{(t-1)} = \frac{1}{M}$ and $\kappa = 1$, $\forall m = 1, 2, \ldots, M$. The proposed objective function, in (7.11), is given by

$$\underset{U_{\text{Joint}},\, U_m\, L_m^r}{\text{minimize}} \; - tr\left(U_{\text{Joint}}^T \big(\sum_{m=1}^{M} \frac{1}{M} L_m^r \big) U_{\text{Joint}} \right) \tag{7.33}$$
$$- \frac{1}{M} \sum_{m=1}^{M} \left[tr(U_m^T L_m^r U_m) + tr\left(U_{\text{Joint}} U_{\text{Joint}}^T U_m U_m^T \right) \right] - \frac{1}{M(M-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{M} tr(U_i U_i^T U_j U_j^T),$$

with orthonormality constraints on U_{Joint} and U_m, and symmetric positive definite constraints on L_m^r. The constant factor 2 in the denominator of (7.11) is not taken into consideration as it does not affect the angles between the subspaces. Given the $(t-1)$-th iterates of U_m and L_m^r, the subproblem of (7.33) with respect to U_{Joint}, at iteration t, is given by

$$\underset{U_{\text{Joint}}}{\text{minimize}} - \frac{1}{M} tr \left(U_{\text{Joint}}^T \left[\sum_{j=1}^{M} L_j^{r(t-1)} + U_j^{(t-1)} U_j^{(t-1)^T} \right] U_{\text{Joint}} \right). \tag{7.34}$$

The solution to the negative trace minimization problem in (7.34) is the t-th iterate of U_{Joint} and is given by the k largest eigenvectors of

$$\mathbf{V}_{\text{Joint}}^{(t)} = \frac{1}{M} \left[\sum_{j=1}^{M} \left(L_j^{r(t-1)} + U_j^{(t-1)} U_j^{(t-1)^T} \right) \right]. \tag{7.35}$$

Similarly, for a fixed m, the subproblem with respect to the individual subspace U_m, keeping all other variables fixed to their $(t-1)$-th iterates, is given by

$$\underset{U_m}{\text{minimize}} - tr \left(U_m^T \left[\frac{1}{M} L_m^{r(t-1)} + \frac{1}{M} U_{\text{Joint}}^{(t-1)} U_{\text{Joint}}^{(t-1)^T} + \frac{1}{M(M-1)} \sum_{\substack{j=1 \\ j \neq m}}^{M} U_j^{(t-1)} U_j^{(t-1)^T} \right] U_m \right) \tag{7.36}$$

The solution to (7.36) is the t-th iterate of U_m and is given by the k largest eigenvectors of

$$\mathbf{W}_m^{(t)} = \frac{1}{M} L_m^{r(t-1)} + \frac{1}{M} U_{\text{Joint}}^{(t-1)} U_{\text{Joint}}^{(t-1)^T} + \frac{1}{M(M-1)} \sum_{\substack{j=1 \\ j \neq m}}^{M} U_j^{(t-1)} U_j^{(t-1)^T}. \tag{7.37}$$

The t-th iterates of U_{Joint} and U_m are given by the k largest eigenvectors of $\mathbf{V}_{\text{Joint}}^{(t)}$ and $\mathbf{W}_m^{(t)}$, respectively. The bound on the Grassmannian distance between subspaces $U_{\text{Joint}}^{(t)}$ and $U_m^{(t)}$ is obtained by computing the distance between the k dimensional eigenspaces of $\mathbf{V}_{\text{Joint}}^{(t)}$ and $\mathbf{W}_m^{(t)}$. This is done using matrix perturbation theory [202], which analyzes the difference between the eigenspaces of a matrix and its perturbation. From the expressions of $\mathbf{V}_{\text{Joint}}^{(t)}$ and $\mathbf{W}_m^{(t)}$ in (7.35) and (7.37), respectively, $\mathbf{V}_{\text{Joint}}^{(t)}$ can be written as a perturbation of $\mathbf{W}_m^{(t)}$ by $\mathbf{E}_m^{(t)}$, given by

$$\mathbf{V}_{\text{Joint}}^{(t)} = \mathbf{W}_m^{(t)} + \mathbf{E}_m^{(t)}, \tag{7.38}$$

where

$$\mathbf{E}_m^{(t)} = \sum_{\substack{j=1 \\ j \neq m}}^{M} \left(\frac{1}{M} L_j^{r(t-1)} + \frac{M-2}{M(M-1)} U_j^{(t-1)} U_j^{(t-1)^T} \right) + \frac{1}{M} \left(U_m^{(t-1)} U_m^{(t-1)^T} - U_{\text{Joint}}^{(t-1)} U_{\text{Joint}}^{(t-1)^T} \right). \tag{7.39}$$

Applying the Davis Kahan theorem [49] (see Appendix C) on the perturbation relation in (7.38), the squared principal sines between the k largest eigenvectors of $\mathbf{V}_{\text{Joint}}^{(t)}$ and $\mathbf{W}_m^{(t)}$ is bounded as follows.

Result 7.1. *The Grassmannian distance between t-th iterates of the joint subspace, U_{Joint},* *and the individual subspace corresponding to the m-th view, U_m, is given by*

$$d_\theta(U^{(t)}_{\text{Joint}}, U^{(t)}_m) \leqslant \frac{\left\| \mathbf{E}^{(t)}_m \; U^{(t)}_m \right\|^2_F}{\Psi_{\mathbf{V}_{\text{Joint}^{(t)}}}(k) - \Phi_{\mathbf{W}^{(t)}_m}(k+1)}, \tag{7.40}$$

where $\Psi_{\mathbf{V}_{\text{Joint}^{(t)}}}(k)$ and $\Phi_{\mathbf{W}^{(t)}_m}(k+1)$ denote the k-th and $(k+1)$-th largest eigenvalues of $\mathbf{V}^{(t)}_{\text{Joint}}$ and $\mathbf{W}^{(t)}_m$, respectively.

In multi-view clustering, the views are expected to agree upon a uniform global clustering of the data set. Hence, the distance $d_\theta(U^{(t)}_{\text{Joint}}, U^{(t)}_m)$ between the joint and individual clustering subspace, for each $m = 1, 2, ..., M$, is expected to minimize as the proposed algorithm progresses starting from a random initial iterate to a local minima.

7.5 Experimental Results and Discussion

The clustering performance of the proposed GeARS algorithm is studied and compared with that of the existing approaches on several real-world data sets. Among them, four are benchmark data sets on which GeARS is compared with nine multi-view clustering algorithms, while eight others are multi-omics cancer data sets on which GeARS is evaluated against ten cancer subtyping algorithms. The clustering results are evaluated by measuring the closeness of the identified clusters with ground truth class information for the benchmark data sets, and with the clinically established cancer subtypes for the multi-omics data sets. Six indices, namely, clustering accuracy, adjusted Rand index (ARI), normalized mutual information (NMI), F-measure, Rand index, and purity are used to evaluate the clustering performance.

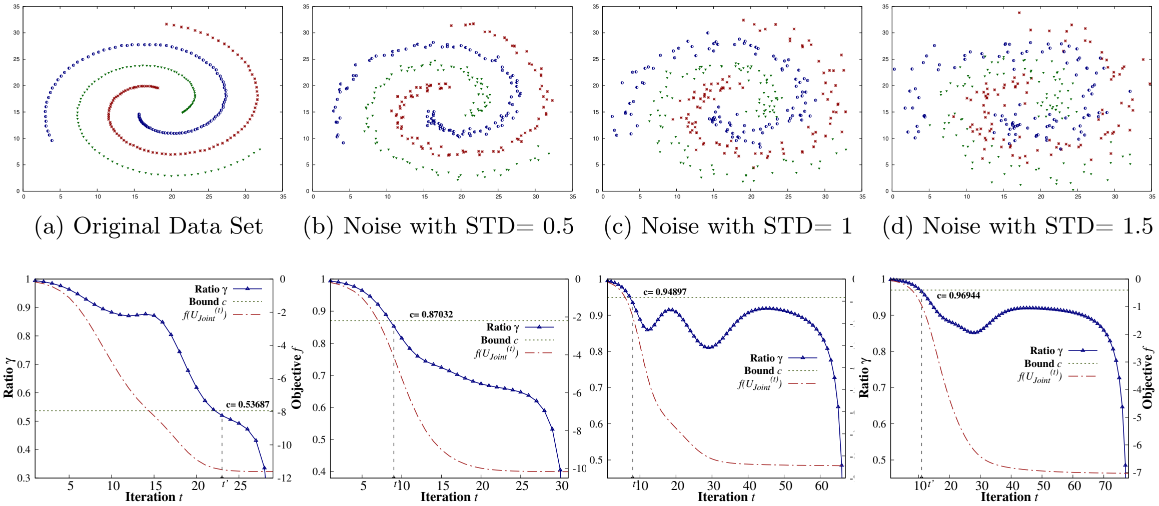
To add randomization in the clustering results, each algorithm is executed 10 times, and the evaluation indices are reported in the mean ± standard deviation form. Similar to Chapter 6, in Tables 7.1-7.6, the numbers within brackets denote the standard deviations, $\rightarrow 0$ denotes a value close to zero ($\sim 1e-16$), while '0.0' denotes the exact zero. For the proposed GeARS algorithm, the step sizes η_{G} for Grassmannian manifold is set to 0.01, while $\eta_{\mathcal{S}}$ for SPD manifold is set to 0.001 for all data sets. The values of ϵ and δ are empirically set to 0.01 and 0.5, respectively. The weight initialization parameter Δ is set to 1 for benchmark data sets and 2 for multi-omic cancer data sets, as in Chapter 6. The value of κ in the weighted combination α is set to 2 for all data sets. The source code of the proposed algorithm in R is available at `https://github.com/Aparajita-K/GeARS`.

7.5.1 Description of Data Sets

In this work, multi-view clustering is performed on the following benchmark and multi-omics cancer data.

Benchmark Data Sets: Seven publicly available benchmark data sets, namely, 3Sources, BBC, Digits, 100Leaves, ORL, Caltech7, and CORA from diverse application domains are considered in this study. The BBC and 3Sources are multi-source document clustering data sets, Digits, 100Leaves, ORL, and Caltech7 are image data sets, while CORA is a social network data set.

Multi-Omics Cancer Data Sets: Disease subtype identification is performed on eight different cancers, namely, ovarian carcinoma (OV), breast adenocarcinoma (BRCA), lower grade glioma (LGG), stomach adenocarcinoma (STAD), colorectal carcinoma (CRC), cervical carcinoma (CESC), lung carcinoma (LUNG),fig:rankGr and kidney carcinoma (KIDNEY). All the cancer data sets are obtained from The Cancer Genome Atlas [1] (TCGA). The number of cancer subtypes for CRC and LUNG is two, for LGG, CESC, and KIDNEY is three, while for OV, BRCA, and STAD is four. For each of these cancer data sets, four genomic views are considered, namely, microRNA expression (miRNA), gene expression (RNA), DNA methylation (mDNA), and reverse phase protein array expression (RPPA). The benchmark and multi-omics data sets, are described in Appendix A, while the cluster evaluation indices are described Appendix B.



(a) Original Data Set (b) Noise with STD= 0.5 (c) Noise with STD= 1 (d) Noise with STD= 1.5

(a) c= 0.53687 (b) c= 0.87032 (c) c= 0.94897 (d) c= 0.96944

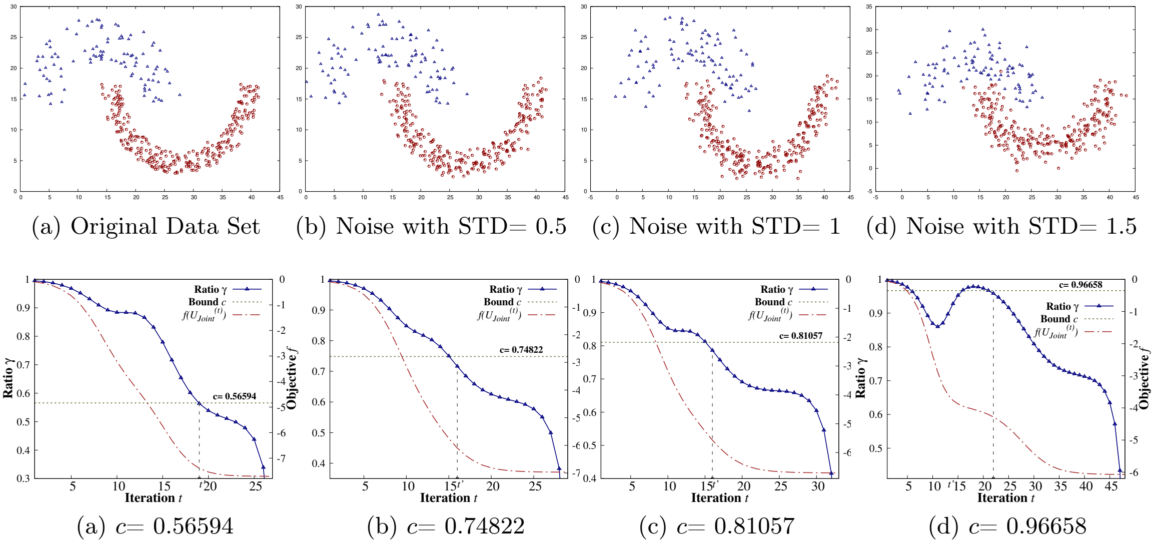
Figure 7.4: Asymptotic convergence analysis for Spiral data set: scatter plot of data with varying Gaussian noise (top row) and variation of convergence ratio and objective function with increase in iteration number t (bottom row).

7.5.2 Significance of Asymptotic Convergence Bound

The convergence factor c in (7.32) bounds the difference between the cost function $\boldsymbol{f}$ evaluated at point $U_{\text{Joint}}^{(t+1)}$ and at the optimal solution $U_{\text{Joint}}^{\star}$ in terms of the difference between that evaluated at $U_{\text{Joint}}^{(t)}$ and $U_{\text{Joint}}^{\star}$. Let this ratio be given by

$$\gamma_t = \frac{\boldsymbol{f}\left(U_{\text{Joint}}^{(t+1)}\right) - \boldsymbol{f}\left(U_{\text{Joint}}^{\star}\right)}{\boldsymbol{f}\left(U_{\text{Joint}}^{(t)}\right) - \boldsymbol{f}\left(U_{\text{Joint}}^{\star}\right)}. \tag{7.41}$$

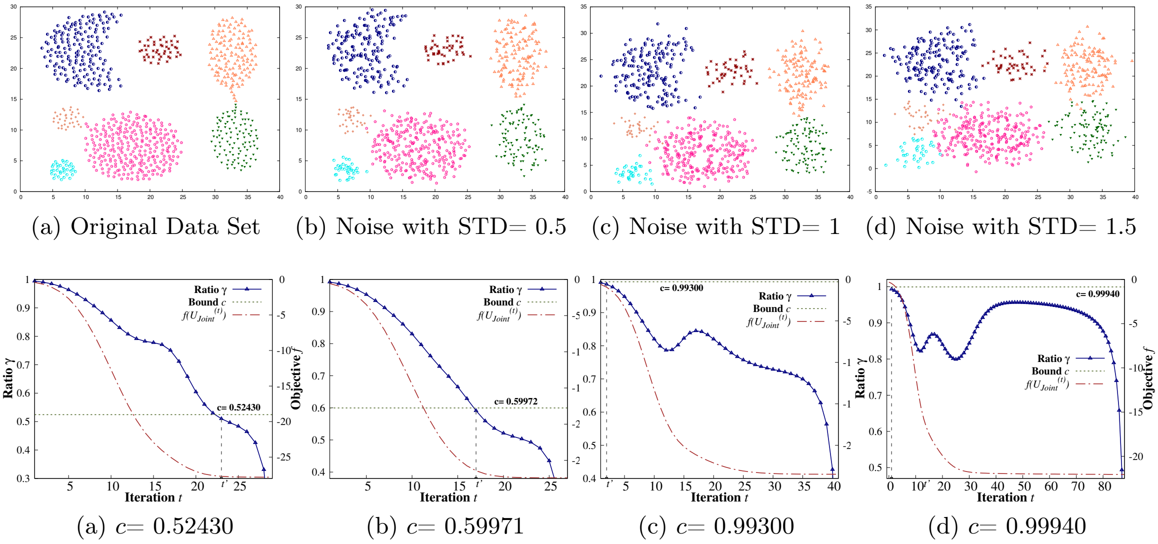
[1] https://cancergenome.nih.gov/



(a) Original Data Set (b) Noise with STD= 0.5 (c) Noise with STD= 1 (d) Noise with STD= 1.5

(a) c= 0.56594 (b) c= 0.74822 (c) c= 0.81057 (d) c= 0.96658

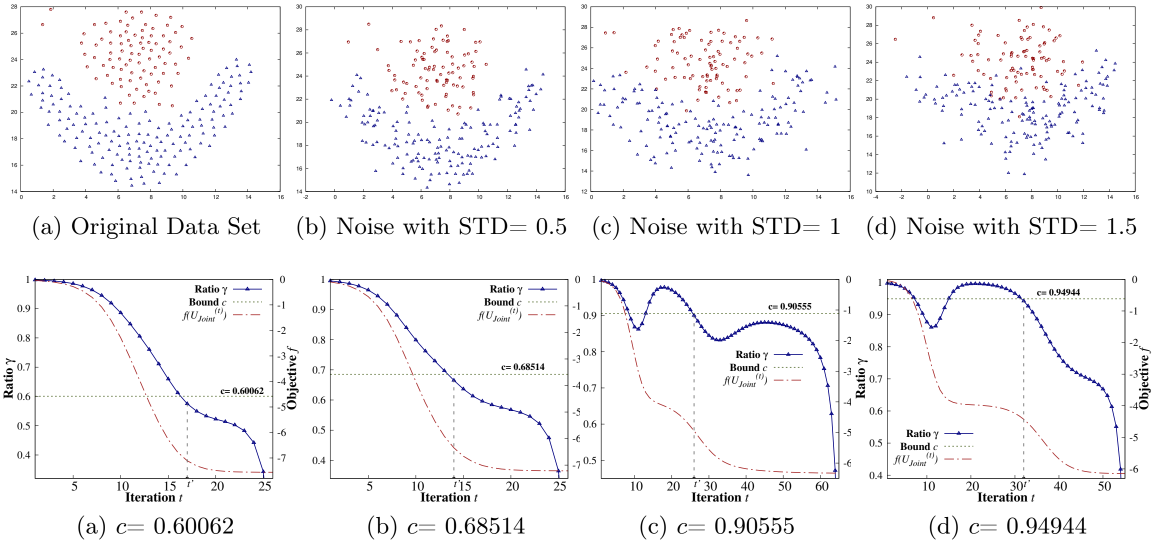
Figure 7.5: Asymptotic convergence analysis for Jain data set: scatter plot of data with varying Gaussian noise (top row) and variation of convergence ratio and objective function with increase in iteration number t (bottom row).



(a) Original Data Set (b) Noise with STD= 0.5 (c) Noise with STD= 1 (d) Noise with STD= 1.5

(a) c= 0.52430 (b) c= 0.59971 (c) c= 0.99300 (d) c= 0.99940

Figure 7.6: Asymptotic convergence analysis for Aggregation data set: scatter plot of data with varying Gaussian noise (top row) and variation of convergence ratio and objective function with increase in iteration number t (bottom row).



(a) Original Data Set (b) Noise with STD= 0.5 (c) Noise with STD= 1 (d) Noise with STD= 1.5

(a) c= 0.60062 (b) c= 0.68514 (c) c= 0.90555 (d) c= 0.94944

Figure 7.7: Asymptotic convergence analysis for Flame data set: scatter plot of data with varying Gaussian noise (top row) and variation of convergence ratio and objective function with increase in iteration number t (bottom row).

Similar to the results and analysis presented in Section 6.5.3 of Chapter 6, the scatter plots for the noise-free and noisy variants of four synthetic shape data sets, namely, Spiral, Jain, Aggregation, and Flame, are provided in the top rows of Figures 7.4, 7.5, 7.6, and 7.7, respectively. The variation of γ_t and the cost function $\boldsymbol{f}\left(U_{\text{Joint}}^{(t)}\right)$, for different values of iteration number $t = 1, 2, 3, \ldots$, along with the corresponding value of convergence factor c for the data sets, is provided in the bottom rows of Figures 7.4, 7.5, 7.6, and 7.7. In these figures, the value of the bound c is marked by a horizontal dashed green line, while the vertical dashed line denotes the iteration threshold t' above which the asymptotic bound is satisfied by all the iterations until convergence.

For all four data sets, the top rows of Figures 7.4, 7.5, 7.6, and 7.7 show that the cluster structure and their separability degrades with the increase in noise, as expected. The bottom rows of these figures in turn show that with increase in noise in the data sets, the value of the convergence factor c increases and goes close to 1. For instance, in case of the Spiral data set, the value of c for the noise-free original data set in Figure 7.4(a) is 0.53687, while that for the three increasingly noisy variants in Figure 7.4(b), 7.4(c), and 7.4(d) are 0.87032, 0.94897, and 0.96944, respectively. Similar pattern in the values of c can be observed for Jain, Aggregation, and Flame data sets from the bottom rows of Figures 7.5, 7.6, and 7.7, respectively. Although the results are sensitive to the added noise and the choice of the random initial iterate, in general, it can be observed that lower values of c implies faster convergence. For instance, the bottom row of Figure 7.4 shows that the proposed algorithm converges in much lesser number of iterations ($\leqslant 30$) in the noise-free case (Figure 7.4(a)) compared to the noisy ones (iterations $\geqslant 65$ in Figures 7.4(c) and 7.4(d)). It can also be observed that the value of the iteration threshold t', above which the asymptotic bound is satisfied by all the iterations until convergence, decreases as the amount of noise increases (from Figure 7.4(a) to Figure 7.4(d)), implying a longer path until convergence due to noise. The value of the minimization based objective function $\boldsymbol{f}$ at the optimal solution also increases from -11.6 in the noise-free case (Figure 7.4(a)) to -7.02 in heavily noised case (Figure 7.4(d)), implying degradation of cluster structure. Similar observations can be made from Figures 7.5, 7.6, and 7.7 for Jain, Aggregation, and Flame data sets, respectively. These results show that, similar to Chapter 6, the convergence factor c in (7.32) can be used to make inference about the quality of the clusters and the speed of convergence of the proposed algorithm, for a given data set.

7.5.3 Empirical Study on Subspace Disagreement Bound

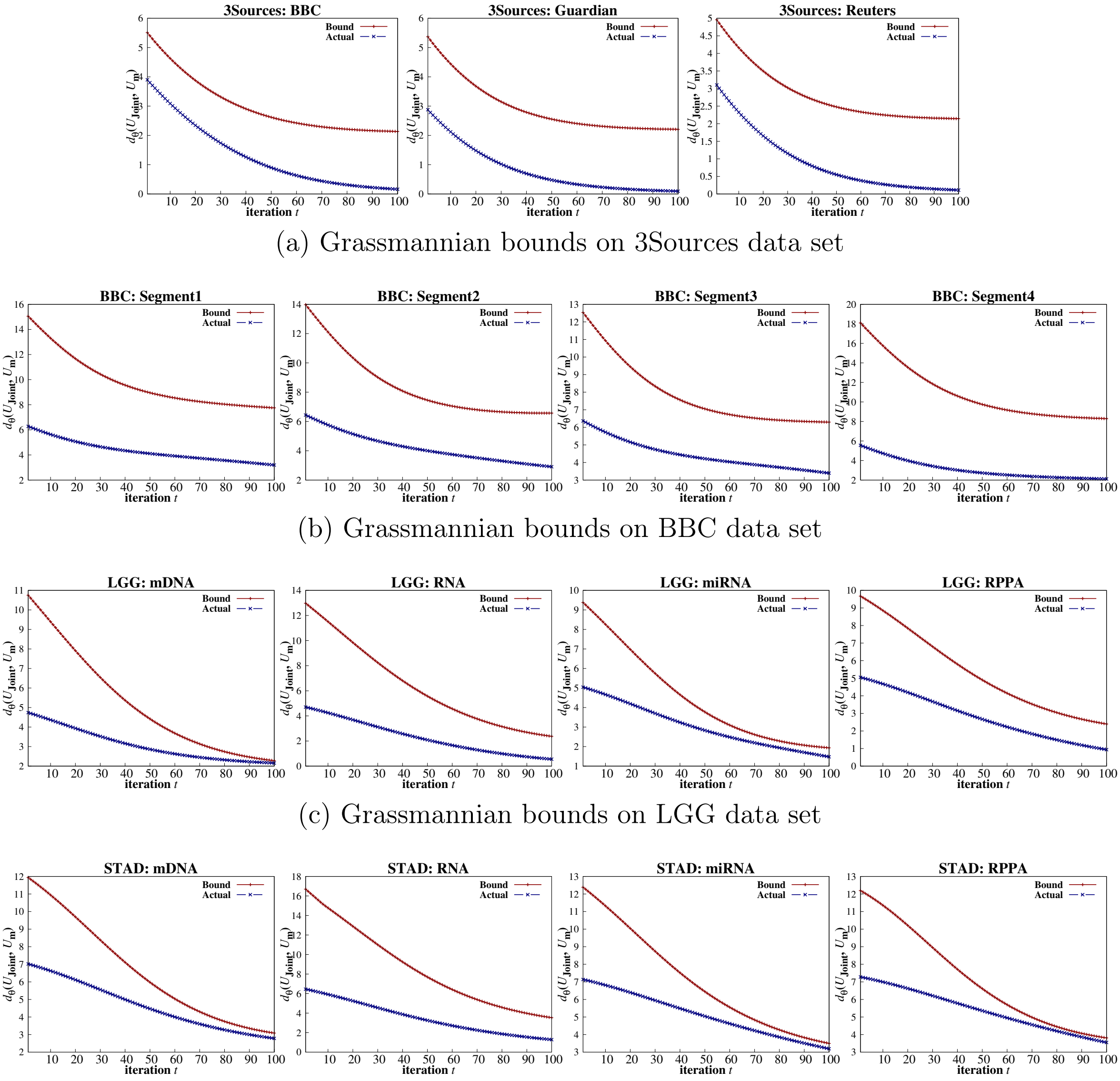
The relation established in (7.40) gives an upper bound on the Grassmannian distance $d_\theta\big(U_{\text{Joint}}^{(t)}, U_m)^{(t)}\big)$ between the joint subspace $\mathsf{span}(U_{\text{Joint}})$ and the individual one $\mathsf{span}(U_m)$ corresponding to the m-th view, at iteration t of the proposed algorithm. Let the bound for view m at iteration t be denoted by

$$\mathbf{\Gamma}_m^{(t)} = \frac{\left\|\mathbf{E}_m^{(t)}\ U_m^{(t)}\right\|_F^2}{\Psi_{\mathbf{V}_{\text{Joint}^{(t)}}}(k) - \Phi_{\mathbf{W}_m^{(t)}}(k+1)}. \tag{7.42}$$

The variation of the upper bound $\mathbf{\Gamma}_m^{(t)}$ and the actual distance between those subspaces is empirically studied for two omics data sets, namely, LGG and STAD, and two benchmark data sets, namely, 3Sources and BBC, with different values of iteration number t. The variations are reported in Figure 7.8, for each view of LGG, STAD, 3Sources, and BBC data sets. Figure 7.8 shows that for all views of each data set the theoretical bound (marked in red) is satisfied by the actual distance (marked in blue) between the joint and the individual subspaces. At each iteration of the proposed GeARS algorithm, the next set of iterates is computed by taking a small step in the direction of the negative gradient of objective function $\boldsymbol{f}$. Since the Grassmannian bound in Section 7.4 is computed from the closed form solutions of U_{Joint} and U_m, the upper bound $\mathbf{\Gamma}_m^{(t)}$, as seen in Figure 7.8, is satisfied by the actual distance observed between those subspaces at each iteration t of the proposed algorithm. The bottom two rows of Figure 7.8 for LGG and STAD data sets show that the theoretical bound is closer to the observed distance for the omics data sets compared to that for the 3Sources and BBC benchmark data sets, shown in top two rows of the same figure.

7.5.4 Choice of Rank

The optimal approximation rank, $r^\star$, of the individual Laplacians, is determined based on the Silhouette index, as described in Section 6.5.4 of Chapter 6. The variation of the Silhouette index and F-measure is shown in Figure 7.9 for LGG, OV, and Digits data sets, as examples. Figure 7.9 shows that both F-measure and Silhouette index tend to vary similarly for the data sets. Based on this criteria, the optimal rank for the benchmark data sets, namely, 3Sources, BBC, Digits, and 100Leaves are 45, 46, 13, and 160, respectively. For the eight cancer data sets, namely, OV, LGG, BRCA, STAD, CRC, CESC, LUNG, and KIDNEY, the ranks are 5, 31, 4, 19, 10, 3, 33, and 5, respectively. For OV, BRCA, LGG, Digits, and 3Sources data sets, the F-measure corresponding to selected rank $r^\star$



(a) Grassmannian bounds on 3Sources data set

(b) Grassmannian bounds on BBC data set

(c) Grassmannian bounds on LGG data set

(d) Grassmannian bounds on STAD data set

Figure 7.8: Variation of the theoretical upper bound $\mathbf{\Gamma}_m$ and the observed Grassmannian distance between U_{Joint} and U_m with increase in iteration number t for 3Sources, BBC, LGG, and STAD data sets. Sub-figures in each row shows the variation for different views of the corresponding data set.

coincides with the best F-measure obtained over different values of r. The importance of considering the approximation rank $r^\star$ to be greater than k, the number of clusters, is established by comparing the clustering performance of the proposed algorithm at rank $r^\star$ with that at k in Table 7.1. Table 7.1 shows that for all benchmark data sets, and three cancer data sets, namely, OV, LGG, and STAD, the clustering performance significantly improves when considering the optimal rank $r^\star$ instead of k. For BRCA data, $r^\star = k$ yields same performance in both the cases.

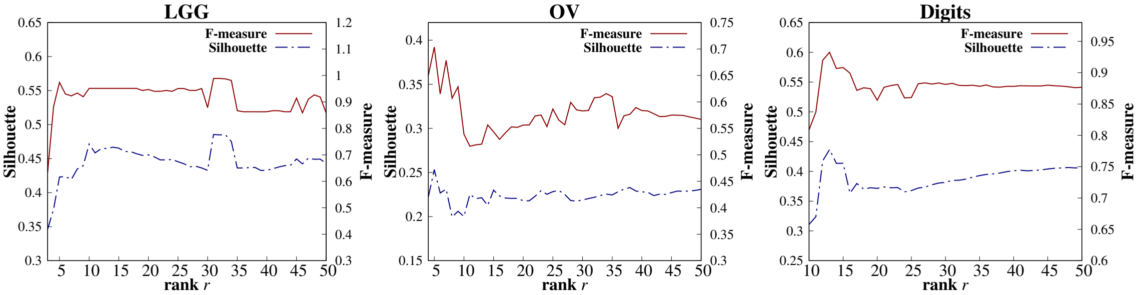


Figure 7.9: Variation of Silhouette index and F-measure for different values of rank r on LGG, OV, and Digits data sets.

7.5.5 Effectiveness of Proposed Algorithm

This subsection illustrates the significance of different components of the proposed formulation, such as, optimization of joint and individual cluster indicator subspaces over Grassmannian manifold, optimization of the graph Laplacians over SPD manifold, Laplacian eigenvalue based initialization of graph weights, and so on. The results are studied on four benchmark data sets: 3Sources, BBC, Digits, and 100Leaves, and four omics data sets: BRCA, STAD, LGG, and OV, as examples, in Tables 7.2 and 7.3.

7.5.5.1 Importance of Joint Subspace Optimization

The proposed algorithm extracts a global cluster indicator subspace, U_{Joint}, that optimizes the spectral clustering objective over the joint view, while reducing the discrepancy between the joint and the individual clustering subspaces. To establish the importance of optimizing U_{Joint} in the proposed formulation, the clustering performance of the proposed algorithm is compared with that of the case where the joint clustering subspace is not optimized, instead, all other components like the individual cluster indicator subspaces, individual Laplacians, and their corresponding weights are optimized. Since U_{Joint} is not optimized, the clusters are identified by performing k-means clustering on the indicator subspace corresponding to the highest weighted view according to the eigenvalue based measure. The clustering performance for this case is reported under the 'U_{Joint} optimize' component in Tables 7.2 and 7.3. These two tables show that for all data sets, except BRCA, clustering on the joint subspace having multi-view information gives a significant improvement in performance as opposed to even the most relevant single view. For the BRCA data set, RNA expression is the most relevant view according to α, and its spectral clustering result is considerably high (as shown in the result corresponding to the 'Best view' in Table 7.3), hence, the improvement in performance considering U_{Joint} is comparatively lower. Nevertheless, the increased performance of the proposed GeARS algorithm across all data sets establishes the importance of U_{Joint} optimization.

7.5.5.2 Importance of Individual Subspace Optimization

In order to establish the importance of optimizing the individual cluster indicator subspaces, the performance of the proposed algorithm is compared with that of the case where

Table 7.1: Performance Analysis of Proposed Algorithm at Rank k and Optimal Rank $r^\star$

Measure		Rank k	Rank $r^\star$		Rank k	Rank $r^\star$
Rank	**Digits**	10	13	**3Sources**	6	45
Accuracy		0.7925(0.0)	**0.9321**(3.16e-4)		0.6153(7.88e-3)	**0.7786**(7.48e-3)
NMI		0.7610(→ 0)	**0.8683**(5.59e-4)		0.5723(1.62e-2)	**0.6823**(1.06e-2)
ARI		0.6792(0.0)	**0.8557**(6.27e-4)		0.4537(1.28e-2)	**0.6591**(2.09e-2)
F-measure		0.8089(0.0)	**0.9325**(3.21e-4)		0.6786(5.43e-3)	**0.8063**(1.07e-2)
Rand		0.9416(0.0)	**0.9740**(1.12e-4)		0.8127(4.48e-3)	**0.8770**(6.61e-3)
Purity		0.8000(0.0)	**0.9321**(3.16e-4)		0.7455(7.89e-3)	**0.8142**(7.48e-3)
Rank	**BBC**	5	46	**100Leaves**	100	160
Accuracy		0.7132(6.52e-2)	**0.8804**(1.74e-3)		0.7893(1.69e-2)	**0.8372**(1.79e-2)
NMI		0.5856(5.70e-2)	**0.7335**(5.84e-3)		0.9145(6.09e-3)	**0.9346**(3.71e-3)
ARI		0.5371(1.00e-1)	**0.7566**(8.23e-3)		0.7042(1.94e-2)	**0.7643**(1.97e-2)
F-measure		0.7293(5.31e-2)	**0.8710**(1.98e-3)		0.8206(1.48e-2)	**0.8618**(1.11e-2)
Rand		0.8032(5.52e-2)	**0.9078**(3.35e-3)		0.9936(5.17e-4)	**0.9950**(5.35e-4)
Purity		0.7143(6.47e-2)	**0.8804**(1.75e-3)		0.8137(1.41e-2)	**0.8545**(1.41e-2)
Rank	**OV**	4	5	**LGG**	3	31
Accuracy		0.6497(0.0)	**0.7023**(2.89e-3)		0.6329(0.0)	**0.9887**(0.0)
NMI		0.3287(0.0)	**0.3687**(1.71e-3)		0.4312(→ 0)	**0.9397**(→ 0)
ARI		0.3070(0.0)	**0.3735**(4.88e-3)		0.2861(0.0)	**0.9655**(0.0)
F-measure		0.6503(0.0)	**0.7035**(3.02e-3)		0.6347(0.0)	**0.9887**(0.0)
Rand		0.7392(0.0)	**0.7621**(2.29e-3)		0.6639(0.0)	**0.9837**(0.0)
Purity		0.6497(0.0)	**0.7023**(2.89e-3)		0.6779(0.0)	**0.9887**(0.0)
Rank	**BRCA**	4	4	**STAD**	4	19
Accuracy		**0.7914**(0.0)	**0.7914**(0.0)		0.4214(0.0)	**0.7933**(0.0)
NMI		**0.5444**(→ 0)	**0.5444**(→ 0)		0.0982(→ 0)	**0.4970**(→ 0)
ARI		**0.5376**(0.0)	**0.5376**(0.0)		0.0560(0.0)	**0.5059**(0.0)
F-measure		**0.7936**(0.0)	**0.7936**(0.0)		0.4530(0.0)	**0.7942**(0.0)
Rand		**0.8104**(0.0)	**0.8104**(0.0)		0.5953(0.0)	**0.7847**(0.0)
Purity		**0.7914**(0.0)	**0.7914**(0.0)		0.5000(0.0)	**0.7933**(0.0)

the individual subspaces U_m's, for $m = 1, ..., M$, are set to their initial iterates and not optimized, but all other variables are optimized. This result is provided corresponding to 'U_m optimize' component in Tables 7.2 and 7.3. The proposed algorithm outperforms this restricted case across all data sets as shown in Tables 7.2 and 7.3. The difference is more significant in case of 3Sources, LGG, BRCA, and STAD data sets. When the individual subspaces U_m's are not optimized, the information in the joint subspace and the individual Laplacians do not update the individual subspaces. In absence of this information propagation, the joint subspace U_{Joint} is unable to reach a better consensus about the global cluster structure, which results in poorer performance as shown in Tables 7.2 and 7.3. This establishes the importance of optimizing the variables U_m's, corresponding to each individual view.

7.5.5.3 Importance of Pairwise Distance Reduction

Apart from reducing the Grassmannian distance between the joint and individual subspaces, the proposed model also reduces that between every pair of individual subspaces. Tables 7.2 and 7.3 report the clustering performance of the model where the pairwise

Table 7.2: Importance of Different Components of the Proposed Algorithm on Benchmark Data Sets

Module		Accuracy	NMI	ARI	F-measure	Rand	Purity
Best view	**Digits**	0.7096(7.7e-4)	0.6443(3.9e-4)	0.5416(9.2e-4)	0.7206(6.9e-4)	0.9173(2.1e-4)	0.7096(7.7e-4)
U_{Joint}		0.7735(0.0)	0.7256($\rightarrow$ 0)	0.6401(0.0)	0.7843(0.0)	0.9347(0.0)	0.7775(0.0)
U_m		0.9055(0.0)	0.8392($\rightarrow$ 0)	0.8064(0.0)	0.9067(0.0)	0.9650(0.0)	0.9055(0.0)
$d_\theta(U_i, U_j)$		0.9315(0.0)	0.8668($\rightarrow$ 0)	0.8543(0.0)	0.9318(0.0)	0.9738(0.0)	0.9315(0.0)
L_m		0.9060(0.0)	0.8395($\rightarrow$ 0)	0.8072(0.0)	0.9072(0.0)	0.9652(0.0)	0.9060 (0.0)
α_Equal		0.8965(0.0)	0.8363($\rightarrow$ 0)	0.7973(0.0)	0.8971(0.0)	0.9634(0.0)	0.8965(0.0)
$\alpha^{(0)}$_Eigen		0.9055(0.0)	0.8385($\rightarrow$ 0)	0.8060(0.0)	0.9067(0.0)	0.9650(0.0)	0.9055(0.0)
GeARS		**0.9321**(3.1e-4)	**0.8683**(5.5e-4)	**0.8555**(6.2e-4)	**0.9325**(3.2e-4)	**0.9740**(1.1e-4)	**0.9321**(3.1e-4)
Best view	**3Sources**	0.7159(0.0)	0.6390(0.0)	0.6082(0.0)	0.7656(0.0)	0.8624(0.0)	0.7869(0.0)
U_{Joint}		0.6343(1.3e-2)	0.5535(7.3e-3)	0.4432(1.1e-2)	0.6844(1.0e-2)	0.8136(2.8e-3)	0.7236(9.2e-3)
U_m		0.7165(4.3e-2)	0.6292(2.2e-2)	0.5635(4.9e-2)	0.7560(3.4e-2)	0.8455(1.9e-2)	0.7621(1.9e-2)
$d_\theta(U_i, U_j)$		0.7526(3.3e-2)	0.6615(3.7e-2)	0.6176(6.7e-2)	0.7855(3.8e-2)	0.8627(2.5e-2)	0.7881(3.3e-2)
L_m		0.7526(3.3e-2)	0.6590(3.4e-2)	0.6149(6.4e-2)	0.7838(3.7e-2)	0.5464(2.4e-2)	0.7881(3.3e-2)
α_Equal		0.7289(4.93e-2)	0.6434(3.44e-2)	0.5832(5.57e-2)	0.7629(3.78e-2)	0.8509(2.11e-2)	0.7745(2.53e-2)
$\alpha^{(0)}$_Eigen		0.7526(3.3e-2)	0.6590(3.4e-2)	0.6149(6.4e-2)	0.7838(3.7e-2)	0.8617(2.4e-2)	0.7881(3.3e-2)
GeARS		**0.7786**(7.4e-3)	**0.6823**(1.0e-2)	**0.6591**(2.0e-2)	**0.8063**(1.0e-2)	**0.8770**(6.6e-3)	**0.8142**(7.4e-3)
Best view	**BBC**	0.6202(2.1e-3)	0.4312(1.7e-3)	0.3405(6.6e-2)	0.6514(1.2e-2)	0.7256(1.7e-2)	0.6212(3.5e-3)
U_{Joint}		0.7868(0.0)	0.6402($\rightarrow$ 0)	0.6846(0.0)	0.8067(0.0)	0.8824(0.0)	0.7883(0.0)
U_m		0.8759(0.0)	0.7282($\rightarrow$ 0)	0.7499(0.0)	0.8663(0.0)	0.9053(0.0)	0.8759(0.0)
$d_\theta(U_i, U_j)$		0.8797(1.4e-3)	0.7308(5.2e-3)	0.7529(7.8e-3)	0.8701(1.7e-3)	0.9062(3.2e-3)	0.8797(1.4e-3)
L_m		0.8786(1.7e-3)	0.7293(4.6e-3)	0.7521(6.2e-3)	0.8692(1.7e-3)	0.9061(2.5e-3)	0.8786(1.7e-3)
α_Equal		0.8658(3.4e-2)	0.7120(3.2e-2)	0.7162(9.0e-2)	0.8575(3.0e-2)	0.8914(3.7e-2)	0.8658(3.4e-2)
$\alpha^{(0)}$_Eigen		0.8786(1.2e-3)	0.7291(3.5e-3)	0.7520(5.3e-3)	0.8692(1.2e-3)	0.9060(2.2e-3)	0.8786(1.2e-3)
GeARS		**0.8804**(1.7e-3)	**0.7335**(5.8e-3)	**0.7566**(8.2e-3)	**0.8710**(1.9e-3)	**0.9078**(3.3e-3)	**0.8804**(1.7e-3)
Best view	**100Leaves**	0.5786(1.1e-2)	0.7940(4.4e-3)	0.4478(9.8e-3)	0.6113(9.7e-3)	0.9880(2.9e-4)	0.6203(7.6e-3)
U_{Joint}		0.6338(1.3e-2)	0.8146(5.4e-3)	0.5059(1.3e-2)	0.6565(1.1e-2)	0.9898(3.1e-4)	0.6614(1.2e-2)
U_m		0.8228(1.7e-2)	0.9340(3.4e-3)	0.7553(1.4e-2)	0.8546(1.1e-2)	0.9948(4.0e-4)	0.8456(1.4e-2)
$d_\theta(U_i, U_j)$		0.8313(1.6e-2)	**0.9357**(3.8e-3)	0.7530(1.7e-2)	0.8585(1.0e-2)	0.9947(6.4e-4)	0.8512(1.0e-2)
L_m		0.8232(1.2e-2)	0.9336(4.8e-3)	0.7542(2.4e-2)	0.8546(1.0e-2)	0.9948(3.8e-4)	0.8457(1.4e-2)
α_Equal		0.8186(1.5e-2)	0.9307(3.9e-3)	0.7414(1.8e-2)	0.8502(9.8e-3)	0.9944(5.1e-4)	0.8421(1.1e-2)
$\alpha^{(0)}$_Eigen		0.8223(1.9e-2)	0.9328(4.4e-3)	0.7510(1.9e-2)	0.8530(1.3e-2)	0.9947(5.1e-4)	0.8449(1.5e-2)
GeARS		**0.8372**(1.7e-2)	0.9346(3.7e-3)	**0.7643**(1.9e-2)	**0.8618**(1.1e-2)	**0.9950**(5.3e-4)	**0.8545**(1.4e-2)

distance term $\sum d_\theta(U_i, U_j), \forall i \neq j$ in (7.11), is not considered into the optimization framework. The results are provided under '$d_\theta(U_i, U_j)$' component in Tables 7.2 and 7.3. They demonstrate that for OV and 3Sources data sets, there is a substantial decrease in clustering performance when not considering the pairwise distance minimization term. For the other data sets, the performance is reduced by a smaller margin. The small margin is due to the fact that the final k-means is performed on U_{Joint} and the pairwise distance term has an indirect effect on the cluster structure reflected in U_{Joint}.

7.5.5.4 Importance of Laplacian Optimization

Optimization of Laplacians L_m's plays an important role in updating the connectivity of the graphs based on the information reflected in the joint and individual cluster indicator

Table 7.3: Importance of Different Components of the Proposed Algorithm on Omics Data Sets

Module		Accuracy	NMI	ARI	F-measure	Rand	Purity
Best view	**OV**	0.6497(0.0)	**0.3748**(→ 0)	0.3548(0.0)	0.6444(0.0)	0.7536(0.0)	0.6497(0.0)
U_{Joint}		0.6556(0.0)	0.3297(→ 0)	0.3109(0.0)	0.6566(0.0)	0.7409(0.0)	0.6556(0.0)
U_m		0.6766(0.0)	0.3360(0.0)	0.3369(0.0)	0.6775(0.0)	0.7484(0.0)	0.6766(0.0)
$d_\theta(U_i, U_j)$		0.5889(8.0e-2)	0.3075(4.5e-2)	0.2684(7.6e-2)	0.5963(7.6e-2)	0.7208(3.0e-2)	0.6035(7.0e-2)
L_m		0.6485(6.5e-2)	0.3257(3.2e-2)	0.3152(5.7e-2)	0.6510(6.1e-2)	0.7397(2.3e-2)	0.6526(5.6e-2)
α_Equal		0.5598(0.0)	0.2588(→ 0)	0.2155(0.0)	0.5691(0.0)	0.7030(0.0)	0.5628(0.0)
$\alpha^{(0)}$_Eigen		0.5658(0.0)	0.2673(→ 0)	0.2443(0.0)	0.5699(0.0)	0.7162(0.0)	0.5658(0.0)
GeARS		**0.7023**(2.8e-3)	0.3687(1.7e-3)	**0.3735**(4.8e-3)	**0.7035**(3.0e-3)	**0.7621**(2.2e-3)	**0.7023**(2.8e-3)
Best view	**LGG**	0.8352(0.0)	0.5734(→ 0)	0.5567(0.0)	0.8269(0.0)	0.7861(0.0)	0.8352(0.0)
U_{Joint}		0.6292(0.0)	0.4106(→ 0)	0.2765(0.0)	0.6316(0.0)	0.6590(0.0)	0.6741(0.0)
U_m		0.8764(0.0)	0.6502(→ 0)	0.6328(0.0)	0.8742(0.0)	0.8194(0.0)	0.8764(0.0)
$d_\theta(U_i, U_j)$		0.9812(0.0)	0.9001(→ 0)	0.9449(0.0)	0.9812(0.0)	0.9740(0.0)	0.9812(0.0)
L_m		0.9565(3.9e-2)	0.8406(9.5e-2)	0.8689(1.2e-1)	0.9561(4.0e-2)	0.9367(6.0e-2)	0.9565(3.9e-2)
α_Equal		0.9101(0.0)	0.7167(→ 0)	0.7541(0.0)	0.9107(0.0)	0.8834(0.0)	0.9101(0.0)
$\alpha^{(0)}$_Eigen		0.9850(0.0)	0.9189(→ 0)	0.9527(0.0)	0.9850(0.0)	0.9776(0.0)	0.9850(0.0)
GeARS		**0.9887**(0.0)	**0.9397**(→ 0)	**0.9655**(0.0)	**0.9887**(0.0)	**0.9837**(0.0)	**0.9887**(0.0)
Best view	**BRCA**	0.7688(0.0)	0.5277(→ 0)	0.5130(0.0)	0.7690(0.0)	0.7995(0.0)	0.7688(0.0)
U_{Joint}		0.7788(0.0)	0.5331(→ 0)	0.5169(0.0)	0.7812(0.0)	0.8023(0.0)	0.7788(0.0)
U_m		0.7060(0.0)	0.4698(→ 0)	0.4094(0.0)	0.7134(0.0)	0.7611(0.0)	0.7060(0.0)
$d_\theta(U_i, U_j)$		0.7814(0.0)	0.5361(→ 0)	0.5210(0.0)	0.7840(0.0)	0.8041(0.0)	0.7814(0.0)
L_m		0.7085(0.0)	0.4795(→ 0)	0.4144(0.0)	0.7158(0.0)	0.7628(0.0)	0.7085(0.0)
α_Equal		0.6783(0.0)	0.4535(→ 0)	0.3793(0.0)	0.6858(0.0)	0.7495(0.0)	0.6783(0.0)
$\alpha^{(0)}$_Eigen		0.7788(0.0)	0.5341(→ 0)	0.5162(0.0)	0.7817(0.0)	0.8023(0.0)	0.7788(0.0)
GeARS		**0.7914**(0.0)	**0.5444**(→ 0)	**0.5376**(0.0)	**0.7936**(0.0)	**0.8104**(0.0)	**0.7914**(0.0)
Best view	**STAD**	0.5413(0.0)	0.2282(0.0)	0.1927(0.0)	0.5469(0.0)	0.6509(0.0)	0.5867(0.0)
U_{Joint}		0.4297(0.0)	0.1178(0.0)	0.0636(0.0)	0.4597(0.0)	0.5971(0.0)	0.5000(0.0)
U_m		0.7148(0.0)	0.4548(→ 0)	0.3591(0.0)	0.7164(0.0)	0.7264(0.0)	0.7148(0.0)
$d_\theta(U_i, U_j)$		0.7768(0.0)	0.4648(→ 0)	0.4710(0.0)	0.7766(0.0)	0.7682(0.0)	0.7768(0.0)
L_m		0.7685(0.0)	0.4537(→ 0)	0.4508(0.0)	0.7687(0.0)	0.7615(0.0)	0.7685(0.0)
α_Equal		**0.7933**(0.0)	**0.5038**(→ 0)	0.5041(0.0)	**0.7956**(0.0)	**0.7864**(0.0)	**0.7933**(0.0)
$\alpha^{(0)}$_Eigen		0.7636(1.7e-3)	0.4452(6.4e-4)	0.4566(3.6e-3)	0.7650(4.8e-3)	0.7635(7.5e-4)	0.7636(1.7e-3)
GeARS		**0.7933**(0.0)	0.4970(→ 0)	**0.5059**(0.0)	0.7942(0.0)	0.7847(0.0)	**0.7933**(0.0)

subspaces. To study the significance of this, the performance of the proposed algorithm is compared with that of the case where the Laplacians are fixed to the original ones obtained from the input graphs, and only the indicator subspaces are optimized. The 'L_m optimize' component reports the results of this case in Tables 7.2 and 7.3. Similar to the other cases, the proposed GeARS algorithm also outperforms this case in all data sets. The difference in performance is significant for BRCA and OV data sets, and marginal for BBC and 100Leaves data sets. In this case also the effect of L_m optimization on global clustering performance is indirect as change in L_ms induces change in the individual indicator subspaces U_ms which in turn can influence the change in global cluster structure reflected in U_{Joint}. Even so, there is a decrease in performance when not considering the L_ms into the optimization model.

7.5.5.5 Importance of Weight Updation

The proposed algorithm initilizes the weight of each view based on its Laplacian eigenvalues which capture a notion of cluster separability. To study the importance of this weight initialization and its subsequent optimization, the performance of the proposed algorithm is compared with two cases, one where all the views are equally weighted, and other where the weights are fixed to their eigenvalue based initial weights. In both of these cases, the weights are kept fixed throughout the optimization procedure, in order to study the impact of weight updation. The former case is denoted by 'Equal Weight' in Tables 7.2 and 7.3, while the later is denoted by 'Eigen Weight'. Tables 7.2 and 7.3 shows that for all data sets, except STAD, the eigen weighted combination of views gives better clustering performance compared to the equally weighted combination. However, the proposed GeARS algorithm that iteratively optimizes the eigenvalue based weight initialization outperforms both these cases on each data set, except STAD. For the STAD data set, the equally weighted combination marginally outperforms GeARS on two indices, ARI and F-measure.

7.5.6 Comparision with Exisitng Approaches

The multi-view clustering performance of the proposed GeARS algorithm is compared with that of several existing approaches on benchmark and cancer data sets. The comparative results are provided in Tables 7.4, 7.5, and 7.6.

7.5.6.1 Performance Analysis on Benchmark Data Sets

For the benchmark data sets, the performance of GeARS is compared with that of nine state-of-the-art algorithms, namely, co-regularized spectral clustering (CoregSC) [120], multi-view k-means clustering (MKC) [26], adaptive structure-based multi-view clustering (ASMV) [272], multiple graph learning (MGL) [164], multi-view clustering with graph learning (MCGL) [273], multi-view spectral clustering (MSC) [246], convex combination of approximate graph Laplacians (CoALa) (proposed in Chapter 5) [113], graph-based multi-view clustering (GMC) [236], and multi-manifold integrative clustering (MiMIC) (proposed in Chapter 6) [114]. Among these approaches, CoregSC is a co-training based approach, MKC is multi-subspace based, MiMIC is based on manifold clustering, while all others are graph based approaches.

The comparative results are provided in Table 7.4 for the benchmark data sets. It shows that the proposed GeARS algorithm gives the best performance on all benchmark data sets across all measures, except for the NMI measure on Digits data set. The proposed algorithm performs third best in NMI after the graph based MCGL and GMC approaches. Among the existing approaches, two recently proposed graph based approaches, namely, GMC and CoALa outperform the co-training, multi-subspace clustering, and other graph based approaches. Nevertheless, the MiMIC algorithm majorarily outperforms both GMC and CoALa on the data sets. The GMC is a graph fusion based approach that automatically computes graph weights and produces clusters without any additional clustering step. The CoALa algorithm on the other hand fuses together approximate Laplacians, but all the graph weights are fixed a priori during fusion. Furthermore, both these algorithms perform Euclidean space optimization focusing only on constructing a fused graph, not aiming to preserve the complementary cluster structure of individual graphs. Although the manifold

Table 7.4: Comparative Performance Analysis of Proposed and Existing Multi-View Clustering Algorithms on Benchmark Data Sets

Algorithm		Accuracy	NMI	ARI	F-measure
MKC	**Digits**	0.4924(2.77e-1)	0.5325(3.68e-1)	0.4280(2.99e-1)	0.5130(2.33e-2)
CoregSC		0.7556(5.96e-2)	0.7421(3.27e-2)	0.6885(5.73e-2)	0.6934(5.11e-2)
MSC		0.7918(8.21e-2)	0.7560(3.24e-2)	0.6803(6.28e-2)	0.7129(5.58e-2)
ASMV		0.5745(→ 0)	0.6709(→ 0)	0.4047(→ 0)	0.4852(→ 0)
MGL		0.7440(8.19e-2)	0.8264(4.73e-2)	0.6888(1.07e-1)	0.7238(9.37e-2)
MCGL		0.8530(0.0)	**0.9055**(0.0)	0.8313(→ 0)	0.8493(→ 0)
CoALa		0.8835(0.0)	0.7981(→ 0)	0.7645(0.0)	0.8839(0.0)
GMC		0.8820(→ 0)	0.9050(→ 0)	0.8502(→ 0)	0.8658(→ 0)
MiMIC		0.9207(4.21e-4)	0.8597(4.88e-4)	0.8352(8.18e-4)	0.9209(4.15e-4)
GeARS		**0.9321**(3.16e-4)	0.8683(5.59e-4)	**0.8557**(6.27e-4)	**0.9325**(3.21e-4)
MKC	**3Sources**	0.4663(1.06e-1)	0.3665(1.00e-1)	0.2461(1.40e-1)	0.4114(1.08e-1)
CoregSC		0.5479(2.99e-2)	0.5238(1.98e-2)	0.3339(2.85e-2)	0.4775(1.91e-2)
MSC		0.4751(2.97e-2)	0.3850(2.27e-2)	0.2618(3.81e-2)	0.4087(3.05e-2)
ASMV		0.3373(→ 0)	0.0896(→ 0)	-0.021(→ 0)	0.3528(→ 0)
MGL		0.6751(6.67e-2)	0.5768(8.61e-2)	0.4431(1.17e-1)	0.5966(7.12e-2)
MCGL		0.3077(→ 0)	0.1034(→ 0)	-0.033(→ 0)	0.3417(0.0)
CoALa		0.6508(0.0)	0.6198(→ 0)	0.5183(0.0)	0.6929(0.0)
GMC		0.6923(→ 0)	0.6216(0.0)	0.4431(0.0)	0.6047(0.0)
MiMIC		0.7360(5.92e-2)	0.6433(3.59e-2)	0.5957(6.69e-2)	0.7581(5.92e-2)
GeARS		**0.7786**(7.48e-3)	**0.6823**(1.06e-2)	**0.6591**(2.09e-2)	**0.8063**(1.07e-2)
MKC	**BBC**	0.6034(1.10e-1)	0.4786(8.51e-2)	0.3450(1.21e-1)	0.5018(9.03e-2)
CoregSC		0.4701(0.0)	0.2863(0.0)	0.2727(0.0)	0.4879(0.0)
MSC		0.6732(4.94e-2)	0.5531(1.44e-2)	0.4658(2.20e-2)	0.5877(1.83e-2)
ASMV		0.3372(0.0)	0.0348(0.0)	0.0018(→ 0)	0.3781(0.0)
MGL		0.5396(1.10e-1)	0.3697(1.89e-1)	0.3153(1.66e-1)	0.5402(8.53e-2)
MCGL		0.3533(→ 0)	0.0741(→ 0)	0.0053(→ 0)	0.3762(0.0)
CoALa		0.8108(4.36e-3)	0.6536(1.96e-2)	0.7102(2.78e-2)	0.8138(9.93e-4)
GMC		0.6934(→ 0)	0.5628(0.0)	0.4789(→ 0)	0.6333(→ 0)
MiMIC		0.8715(0.0)	0.7182(→ 0)	0.7273(0.0)	0.8613(0.0)
GeARS		**0.8804**(1.74e-3)	**0.7335**(5.84e-3)	**0.7566**(8.23e-3)	**0.8710**(1.98e-3)
MKC	**100Leaves**	0.0100(0.0)	0.0000(0.0)	0.0000(0.0)	0.0186(0.0)
CoregSC		0.7706(2.58e-2)	0.9165(5.90e-3)	0.7229(1.92e-2)	0.7257(1.90e-2)
MSC		0.7379(2.21e-2)	0.9014(7.60e-3)	0.6788(2.26e-2)	0.6821(2.23e-2)
ASMV		0.7906(→ 0)	0.9009(→ 0)	0.6104(→ 0)	0.6148(→ 0)
MGL		0.6904(2.42e-2)	0.8753(7.60e-3)	0.3858(5.65e-2)	0.3944(5.53e-2)
MCGL		0.8106(→ 0)	0.9130(0.0)	0.5155(→ 0)	0.5217(0.0)
CoALa		0.7384(1.34e-2)	0.8893(4.06e-3)	0.6550(1.41e-2)	0.7672(1.19e-2)
GMC		0.8238(→ 0)	0.9292(0.0)	0.4974(→ 0)	0.5042(→ 0)
MiMIC		0.8185(1.56e-2)	0.9302(4.12e-3)	0.7431(2.53e-2)	0.8492(1.13e-2)
GeARS		**0.8372**(1.79e-2)	**0.9346**(3.71e-3)	**0.7643**(1.97e-2)	**0.8618**(1.11e-2)

based algorithm, MiMIC, optimizes over the non-linear Steifel manifold to better capture the lower-dimensional non-linear geometry of complex data sets, it does not optimize the Laplacians based on the indicator subspaces. The graph weights, like CoALa, are also fixed a priori. However, the increased performance of the proposed GeARS algorithm for all data sets in Table 7.4 establishes the importance of joint and individual subspace optimization, as well as graph and its corresponding weight updation in context of multi-view clustering.

Table 7.5: Comparative Performance Analysis of Proposed and Existing Subtype Identification Algorithms on Multi-Omics Cancer Data Sets: OV, LGG, BRCA, and STAD

Algorithm		Accuracy	NMI	ARI	F-measure	Rand	Purity
COCA	**OV**	0.5943(7.0e-3)	0.3131(1.2e-2)	0.2810(6.8e-3)	0.6068(4.2e-3)	0.7039(2.6e-3)	0.5943(7.0e-3)
NormS		0.6976(0.0)	0.4504(0.0)	0.4142(0.0)	0.6910(0.0)	*0.7766*(0.0)	*0.6976*(0.0)
LRAcluster		0.6287(0.0)	0.3745(→ 0)	0.2999(0.0)	0.6384(0.0)	0.7322(0.0)	0.6287(0.0)
iCluster		0.5089(0.0)	0.2249(→ 0)	0.2005(0.0)	0.4808(0.0)	0.6916(0.0)	0.5119(0.0)
PCA-con		0.6946(0.0)	0.4424(→ 0)	0.4068(0.0)	0.6868(0.0)	0.7734(0.0)	0.6946(0.0)
SURE		**0.7215**(0.0)	**0.4680**(→ 0)	**0.4372**(0.0)	**0.7148**(0.0)	**0.7857**(0.0)	**0.7215**(0.0)
JIVE		0.5718(7.7e-3)	0.2629(8.4e-3)	0.2027(4.2e-3)	0.5653(7.8e-3)	0.6885(2.8e-3)	0.5718(7.7e-3)
SNF		0.5269(0.0)	0.2753(0.0)	0.2058(0.0)	0.5642(0.0)	0.6557(0.0)	0.5389(0.0)
CoALa		0.6736(0.0)	0.3381(→ 0)	0.3199(0.0)	0.6700(0.0)	0.7379(0.0)	0.6736(0.0)
MiMIC		0.6595(2.8e-3)	0.3271(3.9e-4)	0.3112(4.2e-3)	0.6611(2.5e-3)	0.7383(1.9e-3)	0.6595(2.8e-3)
GeARS		0.7023(2.8e-3)	0.3687(1.7e-3)	0.3735(4.8e-3)	0.7035(3.0e-3)	0.7621(2.2e-3)	0.7023(2.8e-3)
COCA	**LGG**	0.6591(0.0)	0.2772(0.0)	0.2533(0.0)	0.6608(0.0)	0.6454(0.0)	0.6591(0.0)
NormS		0.7940(0.0)	0.5325(0.0)	0.4649(0.0)	0.7916(0.0)	0.7465(0.0)	0.7940(0.0)
LRAcluster		0.4719(0.0)	0.1240(→ 0)	0.1030(0.0)	0.5137(0.0)	0.5831(0.0)	0.5280(0.0)
iCluster		0.4382(0.0)	0.1379(→ 0)	0.0996(0.0)	0.5187(0.0)	0.5821(0.0)	0.5355(0.0)
PCA-con		0.6666(0.0)	0.3438(0.0)	0.3031(0.0)	0.6574(0.0)	0.6616(0.0)	0.6666(0.0)
SURE		0.7940(0.0)	0.5335(0.0)	0.4668(0.0)	0.7904(0.0)	0.7465(0.0)	0.7940(0.0)
JIVE		0.5617(0.0)	0.2299(→ 0)	0.1606(0.0)	0.5757(0.0)	0.6056(0.0)	0.5730(0.0)
SNF		0.8689(0.0)	0.6253(0.0)	0.6331(0.0)	0.8720(0.0)	0.8268(0.0)	0.8689(0.0)
CoALa		0.9737(0.0)	0.8689(→ 0)	0.9199(0.0)	0.9737(0.0)	0.9622(0.0)	0.9737(0.0)
MiMIC		0.9625(0.0)	0.8543(→ 0)	0.8790(0.0)	0.9623(0.0)	*0.9424*(0.0)	*0.9625*(0.0)
GeARS		**0.9887**(0.0)	**0.9397**(→ 0)	**0.9655**(0.0)	**0.9887**(0.0)	**0.9837**(0.0)	**0.9887**(0.0)
COCA	**BRCA**	0.7434(7.9e-4)	0.5002(3.4e-4)	0.4864(4.5e-4)	0.7457(8.1e-4)	0.7905(1.9e-4)	0.7434(7.9e-4)
NormS		0.7688(0.0)	0.4287(→ 0)	0.5090(0.0)	0.7699(0.0)	*0.7999*(0.0)	*0.7688*(0.0)
LRAcluster		0.7110(0.0)	0.5437(→ 0)	0.4035(0.0)	0.7101(0.0)	0.7521(0.0)	0.7110(0.0)
iCluster		0.7638(0.0)	0.5176(→ 0)	0.4745(0.0)	0.7658(0.0)	0.7842(0.0)	0.7638(0.0)
PCA-con		0.7587(0.0)	0.5506(→ 0)	0.5038(0.0)	0.7601(0.0)	0.7984(0.0)	0.7587(0.0)
SURE		0.7663(0.0)	0.4558(0.0)	0.5104(0.0)	0.7683(0.0)	0.8010(0.0)	0.7663(0.0)
JIVE		0.6859(0.0)	0.4368(0.0)	0.3772(0.0)	0.6889(0.0)	0.7464(0.0)	0.6859(0.0)
SNF		0.6783(0.0)	0.5528(→ 0)	0.4111(0.0)	0.6865(0.0)	0.7602(0.0)	0.6959(0.0)
CoALa		0.7613(0.0)	0.5281(→ 0)	0.4874(0.0)	0.7660(0.0)	0.7922(0.0)	0.7613(0.0)
MiMIC		**0.7964**(0.0)	**0.5553**(→ 0)	**0.5474**(0.0)	**0.7997**(0.0)	**0.8152**(0.0)	**0.7964**(0.0)
GeARS		0.7914(0.0)	0.5444(→ 0)	0.5376(0.0)	0.7936(0.0)	0.8104(0.0)	0.7914(0.0)
COCA	**STAD**	0.4450(3.3e-2)	0.1309(4.7e-3)	0.0740(1.0e-2)	0.4558(2.5e-2)	0.5981(1.3e-2)	0.5173(9.5e-3)
NormS		0.5702(0.0)	0.1805(→ 0)	0.1625(0.0)	0.5770(0.0)	0.6435(0.0)	0.5950(0.0)
LRAcluster		0.4256(0.0)	0.1259(→ 0)	0.0912(0.0)	0.4746(0.0)	0.6122(0.0)	0.5619(0.0)
iCluster		0.3512(0.0)	0.0650(→ 0)	0.0288(0.0)	0.3832(0.0)	0.5855(0.0)	0.4917(0.0)
PCA-con		0.6900(0.0)	0.3654(0.0)	0.3204(0.0)	0.6959(0.0)	0.7110(0.0)	0.6900(0.0)
SURE		0.6983(0.0)	0.3511(→ 0)	0.3445(0.0)	0.7056(0.0)	0.7216(0.0)	0.6983(0.0)
JIVE		0.4049(0.0)	0.1288(→ 0)	0.0657(0.0)	0.4487(0.0)	0.5981(0.0)	0.5165(0.0)
SNF		0.5661(0.0)	0.4558(0.0)	0.1522(0.0)	0.5521(0.0)	0.6945(0.0)	0.6363(0.0)
CoALa		0.7685(0.0)	0.5107(0.0)	0.4559(0.0)	0.7778(0.0)	*0.7661*(0.0)	*0.7685*(0.0)
MiMIC		0.7727(0.0)	**0.5220**(→ 0)	0.4650(0.0)	0.7830(0.0)	0.7698(0.0)	0.7727(0.0)
GeARS		**0.7933**(0.0)	0.4970(→ 0)	**0.5059**(0.0)	**0.7942**(0.0)	**0.7847**(0.0)	**0.7933**(0.0)

7.5.6.2 Performance Analysis on Cancer Data Sets

In case of the cancer data sets, the performance of proposed GeARS algorithm is compared with ten multi-omics cancer subtyping algorithms, namely, LRAcluster [243], iCluster [192], multivariate normality based joint subspace clustering (NormS) (proposed in Chapter 3) [111], cluster of cluster analysis (COCA) [93], joint and individual variance explained (JIVE) [141], selective update of relevant eigenspaces (SURE) (proposed in Chapter 4) [112], principal component analysis on naively concatenated data (PCA-con), similarity network fusion (SNF) [234], CoALa (proposed in Chapter 5) [113], and MiMIC (proposed in Chapter 6) [114]. The experimental setup followed for the existing multi-omics cancer subtyping algorithms is same as that followed in Chapter 3.

The comparative results are reported in Tables 7.5 and 7.6. All the results show that for LGG and STAD data sets, the proposed algorithm outperforms all the existing ones with respect to all the clustering indices, except for the NMI measure on STAD data set. For the OV data set, SVD based SURE algorithm of Chapter 4 has the highest performance, while the MiMIC algorithm, proposed in Chapter 6, has that for the BRCA data set, outperforming the proposed one by a small margin. Apart from COCA which is two-stage consensus clustering approach, all the approaches studied in Tables 7.5 and 7.6 perform clustering on a low-rank subspace. For NormS, LRAcluster, and iCluster algorithms, the subspace is based on a probabilistic model, for JIVE, SURE, and PCA-con, the subspace is SVD based variance maximization subspace, while for CoALa, SNF, MiMIC, and GeARS, it is the graph-cut minimization based spectral clustering subspace. The results in Tables 7.5 and 7.6 show that except for the CESC and OV data sets, in general, the spectral clustering subspace outperforms the probabilistic and the variance maximization based subspaces. A possible explanation for this is that the probabilistic model often fits poorly in real-life data sets, while the variance maximization property tends to reflect the variance due to the cluster pattern as well as noise in its principal subspace. The combined results of Tables 7.5, and 7.6 show that the proposed GeARS algorithm has the best performance for LGG and STAD data sets, and has the second or third best performance in the remaining six cancer data sets, thus achieving competitive results with respect to the state-of-the-art in all data sets.

7.5.6.3 Performance Analysis on Social Network and General Image Data Sets

The performance of the proposed GeARS algorithm is also studied on the CORA social network data set and two general image data sets, namely, Caltech7 and ORL. These data sets consist of mostly network or graph based views, hence, the proposed algorithm is compared with SNF, CoALa, and MiMIC algorithms, that can work graphical representation of views. The comparative results are reported in Table 7.7. The results in Table 7.7 show that the proposed algorithm has the best clustering performance for the ORL data set and the second best performance for Caltech7 and CORA data sets. For Caltech7 and CORA data sets, the MiMIC algorithm proposed in Chapter 6 has the best performance for majority of the external indices. The competitive performance of the proposed GeARS algorithm on these data sets indicates that the algorithm can correctly identify the community structure in large-scale social networks and recognize faces or objects from multi-feature image data sets.

Table 7.6: Comparative Performance Analysis of Proposed and Existing Subtype Identification Algorithms on Multi-Omics Cancer Data Sets: CRC, CESC, KIDNEY, and LUNG

Algorithm		Accuracy	NMI	ARI	F-measure	Rand	Purity
COCA	**CRC**	0.5323(5.56e-3)	*0.0120*(1.27e-3)	0.0007(1.86e-3)	0.5586(5.56e-3)	0.5010(6.97e-4)	**0.7370**(0.0)
NormS		0.6206(0.0)	0.0093(0.0)	0.0347(0.0)	0.6345(0.0)	0.5281(0.0)	**0.7370**(0.0)
LRAcluster		0.5129(0.0)	0.0030(0.0)	-0.001(0.0)	0.5410(0.0)	0.4992(0.0)	**0.7370**(0.0)
iCluster		0.6163(0.0)	0.0070(0.0)	0.0293(0.0)	0.6298(0.0)	0.5260(0.0)	**0.7370**(0.0)
PCA-con		0.5366(0.0)	0.0057(0.0)	0.0037(0.0)	0.5642(0.0)	0.5016(0.0)	**0.7370**(0.0)
SURE		0.5107(0.0)	0.0028(0.0)	-0.002(0.0)	0.5416(0.0)	0.4991(0.0)	**0.7370**(0.0)
JIVE		0.6034(0.0)	0.0071(0.0)	0.0256(0.0)	0.6210(0.0)	0.5203(0.0)	**0.7370**(0.0)
SNF		0.5991(0.0)	0.0069(0.0)	0.0240(0.0)	0.6178(0.0)	0.5186(0.0)	**0.7370**(0.0)
CoALa		**0.6400**(0.0)	**0.0185**(0.0)	**0.0548**(0.0)	**0.6529**(0.0)	**0.5382**(0.0)	**0.7370**(0.0)
MiMIC		*0.6228*(0.0)	0.0069(0.0)	*0.0310*(0.0)	*0.6338*(0.0)	*0.5291*(0.0)	**0.7370**(0.0)
GeARS		0.6206(0.0)	0.0065(0.0)	0.0295(0.0)	0.6321(0.0)	0.5281(0.0)	**0.7370**(0.0)
COCA	**CESC**	0.6693(0.0)	0.4172(4.77e-3)	0.3677(8.95e-4)	0.6865(2.49e-3)	0.6971(6.33e-5)	0.6774(0.0)
NormS		**0.8870**(0.0)	**0.6854**(→0)	**0.7004**(0.0)	**0.8801**(0.0)	**0.8587**(0.0)	**0.8870**(0.0)
LRAcluster		0.8145(0.0)	0.5176(→0)	0.5384(0.0)	0.8123(0.0)	0.7867(0.0)	0.8145(0.0)
iCluster		0.5483(0.0)	0.1737(→0)	0.1017(0.0)	0.5568(0.0)	0.5731(0.0)	0.5645(0.0)
PCA-con		0.8548(0.0)	*0.6750*(→0)	0.6333(0.0)	0.8390(0.0)	0.8237(0.0)	0.8548(0.0)
SURE		*0.8629*(0.0)	0.6461(0.0)	*0.6507*(0.0)	*0.8512*(0.0)	*0.8339*(0.0)	*0.8629*(0.0)
JIVE		0.7177(0.0)	0.4425(0.0)	0.3860(0.0)	0.7097(0.0)	0.7164(0.0)	0.7177(0.0)
SNF		0.6693(0.0)	0.4927(0.0)	0.4239(0.0)	0.7073(0.0)	0.7043(0.0)	0.6935(0.0)
CoALa		0.8225(0.0)	0.5479(0.0)	0.5637(0.0)	0.8139(0.0)	0.7951(0.0)	0.8225(0.0)
MiMIC		0.8548(0.0)	0.6451(→0)	0.6236(0.0)	0.8418(0.0)	0.8193(0.0)	0.8548(0.0)
GeARS		0.8548(0.0)	0.6451(→0)	0.6236(0.0)	0.8418(0.0)	0.8193(0.0)	0.8548(0.0)
COCA	**KIDNEY**	0.9470(0.0)	0.7493(0.0)	0.8393(0.0)	0.9477(0.0)	0.9199(0.0)	0.9470(0.0)
NormS		0.9525(0.0)	0.7726(→0)	0.8534(0.0)	0.9530(0.0)	0.9269(0.0)	0.9525(0.0)
LRAcluster		0.9538(0.0)	*0.7862*(0.0)	*0.8579*(0.0)	0.9545(0.0)	*0.9292*(0.0)	0.9538(0.0)
iCluster		0.6065(0.0)	0.2547(→0)	0.1717(0.0)	0.6514(0.0)	0.5842(0.0)	0.6811(0.0)
PCA-con		0.9511(0.0)	0.7670(→0)	0.8489(0.0)	0.9516(0.0)	0.9246(0.0)	0.9511(0.0)
SURE		0.9525(0.0)	0.7726(→0)	0.8534(0.0)	0.9530(0.0)	0.9269(0.0)	0.9525(0.0)
JIVE		0.9308(0.0)	0.6955(→0)	0.7786(0.0)	0.9300(0.0)	0.8893(0.0)	0.9308(0.0)
SNF		**0.9579**(0.0)	**0.7946**(0.0)	**0.8796**(0.0)	**0.9590**(0.0)	**0.9400**(0.0)	**0.9579**(0.0)
CoALa		0.9294(0.0)	0.6987(→0)	0.7786(0.0)	0.9285(0.0)	0.8893(0.0)	0.9294(0.0)
MiMIC		*0.9552*(0.0)	0.7767(0.0)	0.8534(0.0)	*0.9551*(0.0)	0.9268(0.0)	*0.9552*(0.0)
GeARS		0.9565(0.0)	0.7797(→0)	0.8580(0.0)	0.9566(0.0)	0.9291(0.0)	0.9565(0.0)
COCA	**LUNG**	0.9284(0.0)	0.6287(0.0)	0.7339(0.0)	0.9283(0.0)	0.8669(0.0)	0.9284(0.0)
NormS		0.9359(0.0)	0.6650(0.0)	0.7597(0.0)	0.9357(0.0)	0.8798(0.0)	0.9359(0.0)
LRAcluster		0.9344(0.0)	0.6535(0.0)	0.7545(0.0)	0.9342(0.0)	0.8772(0.0)	0.9344(0.0)
iCluster		0.6333(0.0)	0.0627(0.0)	0.0696(0.0)	0.6299(0.0)	0.5348(0.0)	0.6333(0.0)
PCA-con		0.9388(0.0)	0.6773(→0)	0.7701(0.0)	0.9386(0.0)	0.8850(0.0)	0.9388(0.0)
SURE		0.9418(0.0)	0.6878(0.0)	0.7806(0.0)	0.9417(0.0)	0.8903(0.0)	0.9418(0.0)
JIVE		0.9269(0.0)	0.6333(0.0)	0.7288(0.0)	0.9266(0.0)	0.8644(0.0)	0.9269(0.0)
SNF		**0.9493**(0.0)	*0.7152*(0.0)	**0.8072**(0.0)	**0.9492**(0.0)	**0.9036**(0.0)	**0.9493**(0.0)
CoALa		0.9403(0.0)	0.6970(0.0)	0.7754(0.0)	0.9400(0.0)	0.8877(0.0)	0.9403(0.0)
MiMIC		*0.9463*(0.0)	**0.7173**(0.0)	*0.7965*(0.0)	*0.9461*(0.0)	*0.8983*(0.0)	*0.9463*(0.0)
GeARS		0.9433(0.0)	0.7035(0.0)	0.7859(0.0)	0.9431(0.0)	0.8929(0.0)	0.9433(0.0)

Table 7.7: Comparative Performance Analysis of Proposed and Existing Algorithms on ORL, Caltech7, and CORA Data Sets

Algorithms→		Graph Based		Manifold Based	
		SNF	CoALa	MiMIC	**GeARS**
Accuracy	**ORL**	0.6907(2.57e-2)	*0.7715*(2.18e-2)	0.7307(2.36e-2)	**0.8052**(3.06e-2)
NMI		0.8616(1.00e-2)	*0.8980*(1.15e-2)	0.8814(1.35e-2)	**0.9118**(1.73e-2)
ARI		0.6054(3.04e-2)	*0.6932*(2.82e-2)	0.6208(3.83e-2)	**0.7201**(4.22e-2)
F-measure		0.7257(2.44e-2)	*0.7962*(1.78e-2)	0.7677(2.29e-2)	**0.8297**(2.61e-2)
Rand		0.9804(2.04e-3)	*0.9850*(1.63e-3)	0.9802(2.65e-3)	**0.9864**(2.28e-3)
Purity		0.7450(2.26e-2)	*0.8090*(1.75e-2)	0.7737(1.78e-2)	**0.8315**(2.66e-2)
Accuracy	**Caltech7**	0.5440(3.42e-2)	0.5685(0.0)	*0.5773*(0.0)	**0.5852**(2.31e-2)
NMI		0.5676(2.41e-2)	0.5650(→0)	**0.5880**(→0)	*0.5734*(2.69e-2)
ARI		0.4126(2.86e-2)	0.4397(0.0)	**0.4608**(0.0)	*0.4582*(3.65e-2)
F-measure		0.6363(4.12e-2)	*0.6689*(0.0)	0.6600(0.0)	**0.6761**(2.99e-2)
Rand		0.7482(1.13e-2)	0.7583(0.0)	**0.7674**(0.0)	*0.7666*(1.49e-2)
Purity		0.8516(1.09e-2)	0.8548(0.0)	**0.8751**(0.0)	*0.8603*(7.65e-3)
Accuracy	**CORA**	0.5450(2.79e-2)	*0.5896*(3.41e-3)	**0.6120**(2.46e-3)	0.5801(3.35e-2)
NMI		0.3829(1.14e-2)	0.4364(2.81e-3)	**0.4686**(6.17e-3)	*0.4416*(1.64e-2)
ARI		0.2941(1.86e-2)	*0.3256*(2.87e-3)	**0.3479**(3.73e-3)	0.3079(4.17e-2)
F-measure		0.5957(1.96e-2)	0.5844(4.98e-3)	**0.6373**(3.50e-3)	*0.6140*(1.96e-2)
Rand		**0.7936**(9.31e-3)	0.7460(2.14e-3)	0.7709(9.35e-4)	*0.7736*(9.66e-3)
Purity		0.6012(1.62e-2)	0.6206(3.41e-3)	**0.6423**(2.46e-3)	*0.6217*(2.38e-2)

7.6 Conclusion

This chapter presents a multi-view clustering algorithm based on line-search optimization of two different Riemannian manifolds, namely, Grassmannian and SPD manifolds. While the Grassmannian manifold is used to optimize the lower dimensional cluster indicator subspaces corresponding to different views, the SPD manifold optimizes the graph structure represented by the corresponding Laplacians. The SPD manifold automatically preserves the symmetricity and positive definiteness of the Laplacians during optimization. Additionally, the basis invariance property of the Grassmannian manifold finds cluster indicator subspaces as opposed to representative indicator matrices. The convergence and asymptotic properties of the proposed line-search algorithm are analyzed in order to predict noise and separability of the clusters in the data set. The matrix perturbation theory is used to derive a theoretical upper bound on the Grassmannian distance between the joint and individual clustering subspaces. The distance is also empirically shown to minimize as the algorithm converges to a local minima of the objective function. The clustering performance of the proposed algorithm is studied and compared with that of several state-of-the-art multi-view clustering approaches on four benchmark and eight multi-omics cancer data sets. Experimental results show that simultaneous optimization of the clustering subspaces, graph Laplacians, and their corresponding weights, in the proposed manifold algorithm, has superior performance in several data sets, compared to existing algorithms that optimize over the Euclidean space or only a subset of the variables.

Chapter 8

Conclusion and Future Directions

This chapter summarizes the major contributions of the research reported in different chapters of the thesis. It also provides future research directions, including possible extensions and applications of the proposed research work, in multi-view clustering.

8.1 Major Contributions

The thesis presents different approaches for multi-view data clustering. Primarily, there are four major challenges in multi-view clustering: (i) the high-dimensional low-sample size nature of views, (ii) selection of relevant and informative views over noisy and redundant ones during data integration, (iii) prevent the propagation of noise from the real-life individual views to the joint one during information fusion, and (iv) modelling the lower dimensional non-linear geometry of views. The algorithms proposed in this thesis address these issues using three different baseline strategies, namely, subspace based approach (Chapters 3 and 4), graph based approach (Chapter 5), and manifold based approach (Chapter 6 and 7). A brief summary, highlighting the key attributes of the proposed approaches, is discussed as follows.

Chapter 3 presents a new algorithm for the extraction of a low-rank joint subspace from high-dimensional multi-view data sets. The algorithm uses hypothesis testing to estimate efficiently the rank of each individual view by separating its signal or structural component from the noise component. In order to address the major challenge of appropriate view selection during data integration, two evaluation measures are proposed. One evaluates the relevance of a view in terms of the quality of cluster structure embedded within it, while the other measures the amount of shared information contained within the views. The views with highest relevance and maximum shared information are selected for integration. Next, in Chapter 4, in order to reduce the computational complexity of joint subspace construction, the problem of updating the SVD of a data matrix is formulated and solved for multi-view data sets. The theoretical formulation introduced in this chapter enables the proposed algorithm to extract the principal components in lesser time compared to performing PCA on the concatenated data. Some new quantitative indices are proposed to theoretically evaluate the gap between joint subspace extracted by the proposed algorithm and the principal subspace extracted by PCA. Similar to the previous chapter, the algorithm proposed in Chapter 4 also evaluates and then integrates only the relevant views for joint eigenspace construction. The effectiveness of the algorithms proposed in Chapters 3 and 4 is studied and compared with several existing integrative clustering approaches on real-life multi-omics cancer data sets.

Chapter 5 presents a novel algorithm, for the integration of multiple similarity graphs, that prevents the noise of the individual graphs from being propagated into the unified one. The algorithm first approximates each graph using the most informative eigenpairs of its Laplacian which contains its cluster information. Thus, the noise in the individual graphs is not reflected in their approximations. These denoised approximations are then integrated for the construction of a low-rank subspace that best preserves the overall cluster structure of multiple graphs. Using the matrix perturbation theory, theoretical bounds are derived as a function of the approximation rank, in order to precisely evaluate how far the approximate subspace deviates from the full-rank subspace. The clustering performance of the approximate subspace is compared with that of different existing integrative clustering approaches on several real-life cancer data sets as well as benchmark data sets from varying application domains.

Chapter 6 presents a novel manifold optimization-based algorithm for integrative clustering of high-dimensional multi-view data sets. A joint clustering objective is optimized over two different manifolds, namely, k-means and Stiefel manifolds. The Stiefel manifold models the differential clusters of the individual views, whereas the k-means manifold tries to infer the best-fit global cluster structure in the data. The optimization is performed separately along the manifolds of each view so that individual non-linearity within each view is not lost while looking for the shared cluster information. The convergence of the proposed algorithm is theoretically established over the manifold, while the analysis of its asymptotic behavior quantifies how fast it converges to an optimal solution. Chapter 7, on the other hand, demonstrates that simultaneous optimization of the individual graph structures, their weights, and the joint and individual subspaces, is likely to give a more comprehensive idea of the clusters present in the data set. It presents another manifold optimization algorithm that harnesses the geometry and structure preserving properties of symmetric positive definite manifold (SPD) and Grassmannian manifold for efficient multi-view clustering. The SPD manifold is used to optimize the graph Laplacians corresponding to the individual views while preserving their symmetricity, positive definiteness, and related properties. The Grassmannian manifold is used to optimize and reduce the disagreement between the joint and individual clustering subspaces. The clustering performance of the manifold optimization algorithms proposed in Chapters 6 and 7 is studied and compared with several state-of-the-art integrative clustering approaches on various multi-omics cancer and benchmark data sets.

The concept of approximate graph Laplacians proposed in this thesis is unique.

8.2 Future Directions

There are many important aspects of the research reported in this thesis that can be extended for the advancement of multi-view data analysis. Future directions are enlisted as follows:

1. **Eigenspace model when data does not follow a mixture of Guassians**: The

basic assumption of the signal-plus-noise model proposed in Section 3.2 of Chapter 3 is that the data in each view is drawn from a mixture of Gaussian distributions. Under this assumption, the signal component of each view and its corresponding rank is estimated by those principal components which show deviance from the multivariate normal distribution. However, real-life data may sometimes fail to satisfy the mixture of Gaussian assumption, for which generalized models of signal and noise component estimation may be developed.

2. **Parallel computation of joint eigenspace**: The joint eigenspace of the integrated data, proposed in Section 4.3.1 of Chapter 4, is constructed sequentially in M steps for M views, $X_1, \ldots, X_M$. A possible extension of this model is to reduce the computational complexity by constructing the joint eigenspace parallelly in a single step from the individual eigenspaces. Parallel computation would involve solving a single SVD problem of larger size compared to those solved in each of the M steps of sequential eigenspace construction.

3. **Non-linear combination of graph Laplacians**: In Chapter 5, the joint graph Laplacian is constructed by taking a convex combination of individual Laplacians. The convex combination weights are determined by a heuristic, based on the eigenvalues and eigenvectors of the individual Laplacians. This model can be extended by considering a non-linear combination of individual Laplacians. The non-linear combination coefficients may also be determined automatically by solving an optimization problem.

4. **Tensor spectral clustering**: The graph based multi-view clustering approaches proposed in Chapters 5, 6, and 7 work with similarity graphs represented by pairwise similarity between the samples. This model may be extended by considering higher-order relationships between the samples represented by p-th order tensors, where $p > 2$. The higher-order relationships may better capture the non-linear distribution or neighborhood of the samples, resulting in better clustering. Once the higher-order relationships are modelled using tensors, a joint spectral clustering objective can be optimized over the Euclidean space (as in Chapter 5) or over manifolds (as in Chapters 6 and 7).

5. **Second-order geometry in manifold optimization**: The line-search optimization proposed in Chapters 6 and 7 obtains a local optimal solution by always moving in the negative gradient direction starting from an initial iterate. The gradient only considers the first order geometry of the manifold while optimizing at a given iterate. Other optimization techniques, like Newton's method, trust-region method [3], can be developed that consider the second order geometry of the space during optimization and obtain solutions with global convergence properties.

6. **Data drawn from an union of overlapping manifolds**: The algorithms proposed in Chapters 6 and 7 extract a single lower dimensional manifold corresponding to each view of a multi-view data set. Generalizations of this model can be proposed where the data in each view is considered to be lying on a union of multiple, possibly overlapping, non-linear manifolds. This generic model is expected to better capture non-linear cluster patterns embedded in non-Euclidean spaces.

7. **Incomplete views**: All the multi-view clustering algorithms proposed in the thesis assume that all the samples are completely observed in all the views. However, due to measurement and pre-processing errors, the data sets often have incomplete views, where some of samples are not observed in one view (missing view), or only some of the variables are observed corresponding to a sample in some view (missing variables). Multi-view clustering algorithms can be developed that can work in presence of incomplete views. It would require utilizing the connection between the views to restore the samples in the incomplete views with the help of corresponding samples in the complete views.

8. **Views observed in heterogeneous measurement spaces**: The approaches proposed in Chapters 3 and 4 assume that in case of feature space based representation, the views $X_1, \ldots, X_m, \ldots, X_M$ are all observed in a real-valued Euclidean space, that is, $X_m \in \Re^{n \times d_m}$. However, some of the views may not be observed in the real-valued space. For example, the single nucleotide polymorphism (SNP) data is binary, with a one if a nucleotide has undergone mutation in a sample, and zero otherwise. Similarly, the views can also consist of categorical, integer count, or textual data. The proposed clustering algorithms can be extended to work with heterogeneous multi-view data where different views are observed in different measurement spaces.

9. **Deep network based optimization**: All the algorithms, proposed in different chapters of this thesis, perform shallow optimization and obtain either eigenvalue-eigenvector or gradient based solutions. However, the multi-view clustering objective proposed especially in Chapters 6 and 7 can also be optimized using a network based on deep leaning framework. The eigenvector based solutions can be used to initialize or guide the deep optimization model.

10. **Weak supervision model**: The algorithms proposed in Chapters 3, 4, 5, 6, and 7 are designed for an unsupervised setting, which does not consider the label information during the learning process. In large-scale real-life data sets, although it may not be possible to annotate all the samples in a data set, it may be possible to obtain labels of only a subset of the samples. New approaches can be designed that can improve the learning performance by allowing the multi-view clustering algorithms to be supervised by a small number of labelled samples.

Appendix A

Description of Data Sets

The appendix presents a brief description of the multi-omics cancer and multi-view benchmark data sets used in the thesis for comparative analysis of the proposed and the existing multi-view clustering algorithms.

A.1 Multi-Omics Cancer Data Sets

Throughout Chapters 3-7 nine real-life multi-omics cancer data sets from TCGA are extensively studied in the thesis. The cancer data sets and their genomic views are described as follows.

1. **Cervical Carcinoma** (**CESC**): The cervical cancer data set consists of 124 samples. The recent integrative study by TCGA Research Network [218] has identified three molecular subtypes cervical cancer, namely, Keratin-low Squamous subgroup, Keratin-high Squamous subgroup, and Adenocarcinoma-rich subgroup. The data set consists of 37 samples of Keratin-low Squamous subgroup, 58 samples of Keratin-high Squamous subgroup, and 29 samples of Adenocarcinoma-rich subgroup.

2. **Colorectal Carcinoma** (**CRC**): It is the third most commonly diagnosed cancer in both men and women and account for nine percent of all cancer deaths [65]. The colon and rectum are parts of the digestive system and cancer forms in the colon and/or the rectum. There are 307 samples in the OV data set. Depending on the site of origin, the samples of OV are divided into two subtypes, namely, colon carcinoma and rectum carcinoma, having 236 and 71 samples, respectively.

3. **Lower Grade Glioma** (**LGG**): Diffuse low-grade and intermediate-grade gliomas which together make up the lower-grade gliomas have highly variable clinical behaviour that is not adequately predicted on the basis of histological class. Integrative analysis of data from RNA, DNA-copy-number, and DNA-methylation platforms has uncovered three prognostically significant subtypes of lower-grade glioma [217]. The LGG data set consists of 267 samples. The first subtype has 134 samples which exhibit IDH mutation and no 1p/19q codeletion. The second subtype exhibits both IDH mutation and 1p/19q codeletion and has 84 samples. The third one is called the wild-type IDH subtype and has 49 samples.

4. **Breast Invasive Carcinoma** (**BRCA**): Breast cancer is one of the most common cancers with greater than 1,300,000 cases and 450,000 deaths each year worldwide [214]. During the last 15 years, four intrinsic molecular subtypes of breast cancer, namely, Luminal A, Luminal B, HER2-enriched, and Basal-like, have been identified and intensively studied [99], [198], [214]. The BRCA data set consists of 398 samples comprising of 171, 98, 49, and 80 samples of LuminalA, LuminalB, HER2-enriched, and Basal-like subtype, respectively.

5. **Ovarian Carcinoma** (**OV**): Ovarian cancer is the eighth most commonly occurring cancer in women and there were nearly 300,000 new cases in 2018 [22]. Ovarian cancer encompasses a heterogeneous group of malignancies that vary in etiology, molecular biology, and numerous other characteristics. TCGA researchers have identified four robust expression subtypes of high-grade serous ovarian cancer [215]. The OV data set consists of 334 samples. The four subtypes are termed as immunoreactive, differentiated, proliferative, and mesenchymal, consisting of 74, 91, 90, and 79 samples, respectively.

6. **Stomach Adenocarcinome** (**STAD**): Stomach/Gastric cancer was the worldâĂŹs third leading cause of cancer mortality in 2012, responsible for 723,000 deaths [64]. TCGA research network has proposed a molecular classification dividing gastric cancer into four subtypes [216]. The STAD data set has 242 samples which consists of 54 samples from microsatellite unstable tumours, which show elevated mutation rates, 21 samples of tumours showing positivity for EpsteinBarr virus, 119 samples of tumours having chromosomal instability, and 48 samples of genomically stable tumors.

7. **Glioblastoma Multiforme** (**GBM**): GBM is the most common and malignant form of brain cancer and has four subtypes identified in the study by Veerhak *et al.* [228]. The subtypes are Proneural, Neural, Classical, and Mesenchymal. The data set consists of 168 samples from three genomic modalities, namely, Gene, miRNA, and CNV, as the DNA and the Protein modalities are available for a small number of samples. The data set contains 51, 24, 37, and 56 samples of Proneural, Neural, Classical, and Mesenchymal subtypes, respectively.

8. **Lung Carcinoma**(**LUNG**): Based on the primary site of origin, lung cancer set can be categorized in two subtypes, namely, adenocarcinoma and squamous cell carcinoma. These were also the two major subtypes of lung cancer in 2015 WHO classification [222]. The LUNG data set consists of 671 samples with 360 samples of lung adenocarcinoma and 311 samples of lung squamous cell carcinoma.

9. **Kidney Carcinoma**(**KIDNEY**): There are three subtypes of kidney cancer in TCGA based on their tissue type of the site of origin. These are, namely, kidney renal clear cell carcinoma, kidney renal papillary cell carcinoma and kidney chromophobe. The data set consists of 737 samples of kidney cancer with 460 samples of kidney renal clear cell carcinoma, 214 samples of kidney renal papillary cell carcinoma, and 63 samples of kidney chromophobe.

A.1.1 Pre-Processing of Multi-Omics Data Sets

For all the data sets except GBM, four different omic modalities are considered, namely, DNA methylation (mDNA), gene expression (RNA), microRNA expression (miRNA), and reverse phase protein array expression (RPPA). For the GBM data set three modalities namely RNA, miRNA, and copy number variation (CNV) are considered as mDNA and RPPA modalities are not available for a majority of the samples in the data set. In order to avoid considering features with too many missing values, for all the omic modalities, those features for which the corresponding omic expression value is not present for more than 5% of the total number of samples are excluded. For the remaining features, missing values are replaced using 0.

- **RNA and miRNA pre-processing**: For all data sets except GBM and OV, sequence based RNA and miRNA expression data from Il- lumina HiSeq and Illumina GA platforms are used. The RNA and miRNA modalities contain expression signals for 20, 502 annotated genes and 1046 miRNAs, respectively. However, fil- tering out miRNAs with more than 5% missing values reduced the number miRNAs for the these data sets to around 300. The under- lying assumption of the proposed work is that the data follows multivariate Gaussian distribution. However, the sequence based RNA and miRNA expression modalities of the data sets contain normalized RPKM (reads per kilobase of exon per million) counts for the genes. Count data are known to follow a skewed distribution and have the property that the variance depends on the mean value [300]. It is observed that genes having larger mean expression values also tend to have larger variances and are not normally distributed. Log transformation is generally performed on the sequence based expression data to make the data more or less normally distributed [300]. The degree of normality attained depends on the skewness of the data before transformation. Therefore, for modalities with sequence based count data, the 0 entries are replaced by 1, and then the data is log-transformed using base 10. On the other hand, for OV and GBM data sets, array based RNA and miRNA expression data from AgilentG4502A_07_3 and H-miRNA_8x15Kv2 platforms are used. As the RNA and miRNA expression data for the OV data set is observed on microarray based platforms which contain log-ratio based expression data, so the data is not log-transformed as in case of the other four data sets. The RNA modality of OV data set consists of expression for 17,814 genes amongst which 2,000 most variable genes are considered. The miRNA expression data is available for 799 microRNAs.

- **mDNA pre-processing**: For the DNA methylation modality, methylation β-values from Illumina HumanMethylation450 and HumanMethylation450 beadarray platforms are used. The HumanMethylation450 beadarray gives methylation β-values of 485,577 CpG sites, while HumanMethylation27 beadarray covers 27,578 CpG sites. These two platforms share a common set of 25,978 CpG locations. Over 94% of loci present on HumanMethylation27 array are included in the HumanMethylation450 array content. Moreover, the correlation between the β-value measurements across the two platforms were compared in [14] which showed strong correlation of $R^2 > 0.97$. Therefore, for all the data set, methylation data across those common 25,978 CpG locations are considered from both the platforms. Additionally, CpG locations with missing gene information were filtered out from the study. The top 2,000 most variable CpG sites are used for clustering.

- **RPPA pre-processing**: For protein modality, reverse phase protein array data from the MDA_RPPA_Core platform is used. The protein expression data is available in log-ratio form with values ranging between $[-10, 10]$. Taking intersections of the protein IDs available for different samples, expression levels of around 200 proteins are obtained for different data sets.

- **CNV pre-processing**: For the GBM data set, CNV data from affymetrix SNP array 6.0 platform is used. The raw copy number segmented data is processed using the CNregions function of iCluster+ [155] R-package to reduce the redundant copy number regions. The CNregions function has a *epsilon* parameter which denotes the maximum Euclidean distance between adjacent probes tolerated for defining a non-redundant region. The number of non-redundant copy number regions extracted for a data set depends on the value of the *epsilon* parameter and is proportional to the number of samples in the data set. It is recomended in [155] to choose a value of *epsilon* such that the reduced dimension is less than $10,000$. The default value of 0.005 is considered for the *epsilon* parameter of the CNregions function for all the data sets.

These five modalities, measured on different platforms represent a wide variety of biological information. The summary of the data sets in terms of their sample size, dimension of their individual modalities, and their number of clusters is provided in Table A.1.

A.2 Multi-View Benchmark Data Sets

Benchmark data sets from different application domains like social networking, information retrieval, handwritten digits identification, and object detection are considered in this work. The data sets are briefly described as follows.

A.2.1 Social Network Data Sets

Two types of social network data sets are studied in the thesis: Twitter network data sets and citation network data sets.

A.2.1.1 Twitter Data Sets

A brief description of five Twitter data sets used in this work are as follows.

1. **Football**: This data set is a collection of 248 English Premier League football players and clubs active on Twitter. The disjoint ground truth communities correspond to the 20 individual clubs in the league.

2. **Politics-UK**: This data set consists of 419 Members of Parliament (MPs) in the United Kingdom. The ground truth consists of five groups, corresponding to political parties.

3. **Rugby**: The Rugby data set is a collection of 854 international Rugby Union players, clubs, and organizations currently active on Twitter. The ground truth consists of over- lapping communities corresponding to 15 countries. In the case of players, these user accounts can potentially be assigned to both their home nation and the nation in which they play club rugby. As the full names or screen names of the Twitter users are not available, so the overlapping Rugby players are assigned either to their country or their club.

4. **Olympics**: A dataset of 464 users, covering athletes and organizations that were involved in the London 2012 Summer Olympics. The disjoint ground truth communities correspond to 28 different sports.

5. **Politics-IE**: A collection of Irish politicians and political organisations, assigned to seven disjoint ground truth groups, according to their affiliation.

Views of Twitter Data Sets

For each Twitter data set, a heterogeneous collection of nine network and content-based modalities are available. In all cases, cosine similarity is used to compute the pairwise similarities between the Twitter users. All the Twitter data sets are publicly available at http://mlg.ucd.ie/aggregation/. Description of the nine different modalities of each Twitter data set is given below:

1. **Tweets500**: User content profiles, constructed from the concatenation of the 500 most recently-posted tweets for each user.

2. **Lists500**: List content profiles, constructed from the concatenation of both the names and the descriptions of the 500 Twitter lists to which each user has most recently been assigned.

3. **Follows**: From the unweighted directed follower graph, construct binary user profile vectors based on the users whom they follow (i.e. out-going links).

4. **Followed-by**: From the unweighted directed follower graph, construct binary user profile vectors based on the users that follow them (that is, incoming links). A pair of users are deemed to be similar if they are frequently âĂIJco-followedâĂİ by the same users.

5. **Mentions**: From the weighted directed mention graph, construct user profile vectors based on the users whom they mention.

6. **Mentioned-by**: From the weighted directed mention graph, construct binary user profile vectors based on the users that mention them. A pair of users are deemed to be similar if they are frequently âĂIJco-mentionedâĂİ by the same users.

7. **Retweets**: From the weighted directed retweet graph, construct user profile vectors based on the users whom they retweet.

8. Retweeted-by: From the weighted directed retweet graph, construct user profile vectors based on the users that retweet them. Users are deemed to be similar if they are frequently âĂIJco-retweetedâĂİ by the same users.

9. **ListMerged500**: Based on Twitter user list memberships, construct an unweighted bipartite graph, such that an edge between a list and a user indicates that the list contains the specified user. A pair of users are deemed to be similar if they are frequently linked to the same lists. Again, we only consider the 500 lists to which each user has been assigned most recently.

A.2.1.2 Citation Network Data Set

The **CORA** citation network data set consists of 2708 machine learning papers. The data set has two views. The citation relation view consists of 5429 links indicating inbound and outbound citations among the papers. The other view is a content based view where each publication in the data set is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from a dictionary of 1433 unique machine learning keywords. The machine learning articles are classified into seven topics, namely, neural networks, rule learning, reinforcement learning, probabilistic methods, theory, genetic algorithms, and case based study. The pre-processed citation and content views are publicly available at https://github.com/KunyuLin/Multi-view-Datasets.

A.2.2 Image Data Sets

1. **Digits**: This data set consists of features of handwritten numerals (‘0’-‘9’) extracted from a collection of Dutch utility maps with 200 patterns per class (for a total of 2,000 patterns) have been digitized in binary images. The data set is publicly available at https://archive.ics.uci.edu/ml/datasets/Multiple+Features. The samples are represented in terms of the following six feature sets:

   (a) mfeat-fou: 76 Fourier coefficients of the character shapes.

   (b) mfeat-fac: 216 profile correlations.

   (c) mfeat-kar: 64 Karhunen-Love coefficients.

   (d) mfeat-pix: 240 pixel averages in 2 x 3 windows.

   (e) mfeat-zer: 47 Zernike moments.

   (f) mfeat-mor: 6 morphological features.

2. **100Leaves**: It is a one-hundred plant species leaves data set https://archive.ics.uci.edu/ml/datasets/One-hundred+ plant+species+leaves+data+set. The data set consists of 1,600 samples, with sixteen samples of each type of leaf for each of the one-hundred plant species. Each sample is represented by three sets of image features: shape descriptors, fine scale margin, and texture histogram.

3. **ALOI**: This is the Amsterdam Library of Object Image data set http://elki.dbs.ifi.lmu.de/wiki/DataSets/MultiView. The data set is from the work of [18]. The data set consists of 11,025 images of 100 small objects. Each image is represented

with four types of features, that is, RGB color histogram, HSV color histogram, color similiarity and Haralick features.

4. **ORL**:The ORL database of faces contains 400 face images. There are ten different images of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting conditions, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). The size of each image is (92×112) pixels, with 256 grey levels per pixel. Following [276], the images in the data set are resized to (48×48) and three types of image features are extracted: View1 intensity (4096 dimensions), View2 local binary pattern (LBP) (3304 dimensions), and View3 Gabor (6750 dimensions). The standard LBP feature is extracted from (72×80) loosely cropped images with a histogram size of 59 over 910 pixel patches. The Gabor feature is extracted with one scale $\lambda = 4$ at four orientations $\theta = 0°, 45°, 90°, 135°$ with a loose face crop at the resolution of (25×30) pixels. The ORL data set is available at `https://cam-orl.co.uk/facedatabase.html`.

5. **Caltech7**: The Caltech7 is a subset of the Caltech 101 data set `http://www.vision.caltech.edu/Image_Datasets/Caltech101/` for image based object recognition problem. The data set consists of 1474 images from seven widely used classes, namely, Face, Motorbikes, Dolla-Bill, Garfield, Snoopy, Stop-Sign, and Windsor-Chair. Six types of image features are extracted from all the images: 48 dimensional Gabor features, 40 dimensional wavelet moments features, 254 dimensional centrist features, 1984 dimensional histogram of oriented gradients (HOG) features, 512 dimensional GIST descriptors, and 928 dimensional LBP features. The processed image features for the Caltech7 data set are available at `https://github.com/yeqinglee/mvdata`.

A.2.3 Multi-Source News Article Data Sets

1. **3Sources**: This is a multi-view text data set available at `http://mlg.ucd.ie/datasets/3sources.html`. It consists of 169 news articles collected from three well-known online news sources: BBC, Reuters, and The Guardian, from the period February to April 2009. Each news article story was manually annotated with one or more of the six topical labels: business, entertainment, health, politics, sport, and technology. The labels roughly correspond to the primary section headings used across the three news sources. The data set has three views, one corresponding of each of the three news sources.

2. **BBC**: This is also a multi-view news article clustering data set constructed from the single-view BBC news corpora `http://mlg.ucd.ie/datasets/segment.html`. It consists of 685 news documents. Each raw document was split into four segments by separating the documents into paragraphs, and merging sequences of consecutive paragraphs. The segments for each document were then randomly assigned to views.

Table A.1: Summary of Data Sets with Feature Space based Representation

	Data Set	Sample	Cluster	View	d_1	d_2	d_3	d_4	d_5	d_6
Benchmark	**Digits**	2000	10	6	216	76	64	6	240	47
	3Sources	169	6	3	3560	3631	3068	-	-	-
	BBC	685	5	4	4659	4633	4665	4684	-	-
	100Leaves	1600	100	3	64	64	64	-	-	-
	ALOI	11025	100	4	64	64	13	77	-	-
	ORL	400	40	3	4096	3304	6750	-	-	-
	Caltech7	1474	7	6	48	40	254	1984	512	928
Multi-Omics	**BRCA**	398	4	4	2000	2000	278	212	-	-
	LGG	267	3	4	2000	2000	333	209	-	-
	STAD	242	4	4	2000	2000	291	218	-	-
	LUNG	671	2	4	2000	2000	296	180	-	-
	KIDNEY	737	3	4	2000	2000	261	174	-	-
	CESC	124	3	4	2000	2000	311	219	-	-
	CRC	464	2	4	2000	2000	291	178	-	-
	OV	737	4	4	2000	2000	334	192	-	-
	GBM	169	4	3	2000	2000	534	-	-	-

Each document is annotated with one of the five topical labels: business, entertainment, politics, sport, and technology. The data set has four views corresponding to the four segments.

Appendix B

Cluster Evaluation Indices

B.1 External Cluster Evaluation Measures

Four external cluster evaluation measures are used to compare the performance different approaches, namely, accuracy, adjusted rand index (ARI), normalized mutual information (NMI), and F-measure. Since there are different definitions of some of the measures, like accuracy and NMI, in clustering, the definitions used in this work is are described next. A higher value indicates a better performance for each metric. Let $\mathcal{T} = \{t_1, \ldots, t_j, \ldots, t_k\}$ be the true partition of n samples of a data set into k clusters. Let $\mathcal{C} = \{c_1, \ldots, c_i, \ldots, c_k\}$ be the k clusters returned by a clustering algorithm. Let the number of samples in the data set be denoted by n. The external evaluation indices measure how close is the clustering $\mathcal{C}$ with respect to true partition $\mathcal{T}$. Also, let ↑ denote that a higher value of that index means a "better" clustering, while ↓ means the exact opposite. The four external evaluation indices are as follows.

1. **Accuracy**↑ [275]: Given a sample x_p , let its cluster and class labels be denoted by c_p and t_p, respectively. The clustering accuracy is given by

$$\text{Accuracy} = \frac{1}{n}\sum_{p=1}^{n} \delta(t_p, map(c_p)),$$

where $\delta(a, b) = 1$ when $a = b$, otherwise $\delta(a, b) = 0$. The function $map(c_p)$ is the permutation map function, which maps the cluster labels into class labels. The best map can be obtained by the Kuhn-Munkres algorithm [118].

2. **NMI**↑ [68] measures the concordance of cluster assignments in $\mathcal{T}$ and $\mathcal{C}$. NMI is defined as follows:

$$\text{NMI} = \frac{2\,\mathbb{I}\,(\mathcal{T}, \mathcal{C})}{[\mathbb{H}(\mathcal{T}) + \mathbb{H}(\mathcal{C})]}; \tag{B.1}$$

where $\mathbb{H}(\mathcal{C})$ is the entropy of $\mathcal{C}$ and $\mathbb{I}\,(\mathcal{T}, \mathcal{C})$ is the mutual information between $\mathcal{T}$ and

$\mathcal{C}$, which are as follows:

$$\mathbb{H}(\mathcal{C}) = -\sum_{i=1}^{k} Pr(c_i)\log Pr(c_i);$$

$$\mathbb{I}(\mathcal{T},\mathcal{C}) = \sum_{i=1}^{k}\sum_{j=1}^{k} Pr(c_i \cap t_j)\log\left[\frac{Pr(c_i \cap t_j)}{Pr(c_i)Pr(t_j)}\right];$$

where $Pr(S)$ denotes the probability of the set S.

3. **ARI**↑ [8] is an adjustment of the rand index, given by,

$$\text{ARI} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k}\binom{|c_i \cap t_j|}{2} - n_3}{\frac{1}{2}(n_1 + n_2) - n_3}.$$

where $n_1 = \sum_{i=1}^{k}\binom{|c_i|}{2}$, $n_2 = \sum_{j=1}^{k}\binom{|t_j|}{2}$, $n_3 = \frac{2n_1n_2}{n(n-1)}$.

4. **F-measure**↑ [122] of a cluster c_i with respect to a class t_j evaluates how well cluster cluster c_i describes class t_j and is given by the harmonic mean of precision and recall.

$$\text{Precision } P_{ij} = \frac{|c_i \cap t_j|}{|c_i|}.$$

$$\text{Recall } R_{ij} = \frac{|c_i \cap t_j|}{|t_j|}.$$

$$\begin{aligned}\text{F-measure } \ \mathcal{F}(t_j, c_i) &= \frac{2P_{ij}R_{ij}}{P_{ij} + R_{ij}} \\ &= \frac{2|c_i \cap t_j|}{|c_i| + |t_j|}.\end{aligned}$$

The overall F-measure is given by the weighted average of the maximum F-measure over the clusters in $\mathcal{C}$.

$$\text{F-measure} = \frac{1}{n}\sum_{j=1}^{k} n_j \max_i\{\mathcal{F}(t_j, c_i)\},$$

where n_j denotes the number of points in class t_j.

5. **Purity**↑ [177]: It measures the extent to which each cluster contains samples primarily from one class. Each cluster is first assigned with the true class which is most frequent in the cluster and then the purity of the clustering solution is assessed by

the proportion of correctly assigned samples. Formally it is given by,

$$\text{Purity} = \frac{1}{n}\sum_{i=1}^{k} \underset{j}{max} \; \{|c_i \cap t_j|\}. \tag{B.2}$$

In general, higher the value of purity, better is the cluster solution. However, purity does not penalize large number of clusters.

6. **Rand↑** [173]: Rand index is a pair-counting based cluster evaluation index which measures the pairs of points on which the two clusterings agree or disagree. In a n sample data set, the $\binom{n}{2}$ pairs of points can be divided into four categories. Let a represent the number of pairs that are in the same cluster both in $\mathcal{C}$ and $\mathcal{T}$, b represent the number of pairs that are in the same cluster in $\mathcal{C}$ but in different clusters in $\mathcal{T}$, c represents the number of pairs that are in different clusters in $\mathcal{C}$ but in the same cluster in $\mathcal{T}$, and d represent the number of pairs that are in different clusters both in $\mathcal{C}$ and $\mathcal{T}$.The values a and d count the agreements while b and c the disagreements. The Rand index is defined as the ratio of the total number of agreements to the total number of pairs, given by

$$\text{Rand} = \frac{a+d}{a+b+c+d}. \tag{B.3}$$

All the external cluster validation indices lie in [0,1] and a higher value indicates better clustering.

B.2 Internal Cluster Evaluation Measures

Internal cluster validity indices evaluate the quality of clustering based on the information intrinsic to data like compactness and separation of the identified clusters. The information of the correct partition of the data is not used during internal cluster evaluation. In the proposed CoALa algorithm, k-means clustering is performed in an approximate subspace of rank k, where k is the number of clusters in the data set. To establish the effectiveness of the proposed algorithm, the quality of clustering in the k-dimensional approximate subspace is compared with that of the rank k subspaces of the individual modalities and the rank k true subspace using seven internal cluster evaluation indices. These indices are described as follows.

Let $X = \{x_1, ..., x_i, ..., x_n\}$ be the set of n samples, where $x_i \in \mathbb{R}^k$ represents the i-th sample in a k-dimensional subspace. Let the Euclidean distance between samples x_i and x_j be denoted as $d_e(x_i, x_j)$. The k clusters are represented as $\mathcal{C} = C_1, ..., C_k$, and the centroids of each of k clusters are $v_1, ..., v_k$. Let the centroid of the dataset be given by $\bar{x} = \frac{1}{n}\sum\limits_{i=1}^{n} x_i$.

1. **Silhouette ↑** [179]: It is a normalized summation-type index. The compactness within the clusters is measured based on the distance between all the samples in the same cluster and the separation between the clusters is based on the nearest neighbor

distance. It is defined as

$$Silhouette = \frac{1}{n} \sum_{C_j \in \mathcal{C}} \sum_{x_i \in C_j} \frac{b(x_i, C_j) - a(x_i, C_j)}{\max\{b(x_i, C_j), a(x_i, C_j)\}}, \tag{B.4}$$

where

$$a(x_i, C_j) = \frac{1}{|C_j| - 1} \sum_{\substack{x_m \in C_j, \\ x_i \neq x_m}} d_e(x_i, x_m); \tag{B.5}$$

$$b(x_i, C_j) = \min_{C_l \in \mathcal{C} \setminus C_j} \left\{ \frac{1}{|C_l|} \sum_{x_m \in C_l} d_e(x_i, x_m) \right\}. \tag{B.6}$$

2. **Dunn Index ↑** [55]: It is a ratio-type index where compactness is estimated by the nearest neighbor distance and the separation by the maximum cluster diameter. It is defines as

$$Dunn = \frac{\min_{C_j \in \mathcal{C}}\{ \min_{C_l \in \mathcal{C} \setminus C_j} \{\delta(C_j, C_l)\}\}}{\max_{C_j \in \mathcal{C}}\{\Delta(C_j)\}}, \tag{B.7}$$

$$\text{where } \delta(C_j, C_l) = \min_{x_i \in C_j} \min_{x_m \in C_l} \{d_e(x_i, x_m)\}, \tag{B.8}$$

$$\text{and } \Delta(C_j) = \max_{x_i, x_m \in C_j} \{d_e(x_i, x_m)\}. \tag{B.9}$$

3. **Davies-Bouldin (DB) Index ↓** [48]: This index estimates the compactness based on the distance from the samples in a cluster to its centroid and separation based on the distance between centroids. It is defined as

$$DB = \frac{1}{k} \sum_{C_j \in \mathcal{C}} \max_{C_l \in \mathcal{C} \setminus C_j} \left\{ \frac{S(C_j) + S(C_l)}{d_e(v_j, v_l)} \right\}, \tag{B.10}$$

$$\text{where } S(C_j) = \frac{1}{|C_j|} \sum_{x_i \in C_j} d_e(x_i, v_j). \tag{B.11}$$

4. **Xie-Beni Index ↓** [251]: It is an index of fuzzy clustering, but it is also applicable to crisp clustering. For crisp clustering it is defined as

$$Xie - Beni = \frac{1}{n} \frac{\sum_{C_j \in \mathcal{C}} \sum_{x_i \in C_j} d_e^2(x_i, v_j)}{\min_{C_j, C_l \in \mathcal{C}} \{d_e^2(v_j, v_l)\}}. \tag{B.12}$$

Appendix C

Basics of Matrix Perturbation Theory

This section provides a brief description of the theory of invariant subspaces and the related theorems that are used in the main paper. Let A be a matrix of order n. An invariant subspace of the matrix A and some of its properties are as follows [202].

Definition C.1. *A subspace $\mathcal{Z}$ is an invariant subspace of A if $A\mathcal{Z} \subset \mathcal{Z}$, that is, $\forall x \in \mathcal{Z}, Ax \in \mathcal{Z}$.*

Property C.1. *Let $\mathcal{Z}$ is an invariant subspace of A, and let the columns of matrix Z form a basis for subspace $\mathcal{Z}$. Then there exists a unique matrix B such that*

$$AZ = ZB. \tag{C.1}$$

The matrix B is considered to be the representation of A on subspace $\mathcal{Z}$ with respect to the basis Z.

Property C.2. *Let $\mathcal{Z}_1$ be an invariant subspace of A and the columns of Z_1 form an orthonormal basis for $\mathcal{Z}_1$. Let $\begin{bmatrix} Z_1 & W_2 \end{bmatrix}$ be unitary, where columns of W_2 spans the subspace orthogonal to $\mathcal{Z}_1$. Then we can write*

$$\begin{bmatrix} Z_1 & W_2 \end{bmatrix}^T A \begin{bmatrix} Z_1 & W_2 \end{bmatrix} = \begin{bmatrix} B_1 & W \\ 0 & B_2 \end{bmatrix}, \tag{C.2}$$

where $B_1 = Z_1^T A Z_1$, $B_2 = W_2^T A W_2$, and $W = Z_1^T A W_2$.

The equation in (C.2) is called the *reduced form* of A. The eigenvalues of B_1 are the eigenvalues of A associated with the basis $\mathcal{Z}_1$. The complementary set of eigenvalues are those of the matrix B_2. This leads to the notion of a simple invariant subspace, defined as follows.

Definition C.2. *Let $\mathcal{Z}_1$ be an invariant subspace of A, and let the reduced form of A with respect to the unitary matrix $\begin{bmatrix} Z_1 & W_2 \end{bmatrix}$ be given by (C.2). Then $\mathcal{Z}_1$ is called a simple invariant subspace of A if*

$$\Omega(B1) \cap \Omega E(B2) = \varnothing, \tag{C.3}$$

where $\Omega(B)$ denotes the set of all eigenvalues of matrix B.

A simple invariant subspace has a complementary subspace defined using the spectral resolution of A as follows.

Theorem C.1. *Let the simple invariant subspace $\mathcal{Z}_1$ of A with respect to the unitary matrix $\begin{bmatrix} Z_1 & W_2 \end{bmatrix}$ have the reduced form as given by (C.2). Then there exist matrices Z_2 and W_1 such that $\begin{bmatrix} Z_1 & Z_2 \end{bmatrix}^{-1} = \begin{bmatrix} W_1 & W_2 \end{bmatrix}^T$ and*

$$A = Z_1 B_1 W_1^T + Z_2 B_2 W_2^T, \tag{C.4}$$

where $B_j = W_j^T A Z_j$, for $j = 1, 2$. Also, $AZ_1 = Z_1 B_1$ and $AZ_2 = Z_2 B_2$ [202].

Since, $AZ_2 = Z_2 B_2$, this implies that $\mathcal{Z}_2 = \mathcal{C}(Z_2)$ is an invariant subspace of A. Thus if $\mathcal{Z}_1$ is an invariant subspace of A, then A also has a complementary invariant subspace $\mathcal{Z}_2$ [202]. The form of A in (C.4) is called the *spectral resolution* of A along $\mathcal{Z}_1$ and $\mathcal{Z}_2$.

In the matrix perturbation problem, let $\mathcal{Z}_1$ be a simple invariant subspace of a matrix A, and let $\widetilde{A} = A + E$ be a perturbation of A. If E is sufficiently small, then there is an invariant subspace $\widetilde{\mathcal{Z}}_1$ of $\widetilde{A}$, such that $\widetilde{\mathcal{Z}}_1$ approaches $\mathcal{Z}_1$ as E approaches zero. The Davis Kahan theorem [202] is used to bound the difference between $\widetilde{\mathcal{Z}}_1$ and $\mathcal{Z}_1$ in terms of the residual E.

Theorem C.2. ***Davis Kahan* sinΘ *theorem*** [202] *Let A be a matrix of order n. Let A have a spectral resolution given by*

$$A = Z_1 B_1 Z_1^T + Z_2 B_2 Z_2^T, \tag{C.5}$$

where $\begin{bmatrix} Z_1 & Z_2 \end{bmatrix}$ is unitary with $Z_1 \in \mathbb{C}^{n \times r}$. Let $\widetilde{Z} \in \mathbb{C}^{n \times r}$ have orthonormal columns, and for any Hermitian matrix B of order r, let residual

$$\mathcal{R} = A\widetilde{Z} - \widetilde{Z}B. \tag{C.6}$$

If $\Omega(B) \subset [a, b]$ and for some $\delta > 0$, $\Omega(B2) \subset \mathbb{R} \backslash [a - \delta, b + \delta]$, then for any unitarily invariant norm $\| \, . \, \|$,

$$\left\| \sin \Theta \left(\mathcal{C}(Z_1), \mathcal{C}(\widetilde{Z}) \right) \right\| \leqslant \frac{\| \, \mathcal{R} \, \|}{\delta}. \tag{C.7}$$

Weyl's theorem and Weilandt-Hoffman theorem which bound the eigenvalues of the sum of two Hermitian matrices are as follows.

Theorem C.3. ***Weyl*** [202] *Let A and E be $n \times n$ Hermitian matrices with eigenvalues $a_1 \geqslant \ldots \geqslant a_n$ and $b_1 \geqslant \ldots \geqslant b_n$, respectively. The Hermitian matrix $\widetilde{A} = A + B$ having eigenvalues $\widetilde{a}_1 \geqslant \ldots \geqslant \widetilde{a}_n$ satisfy*

$$a_i + b_n \leqslant \widetilde{a}_i \leqslant a_i + b_1. \tag{C.8}$$

Theorem C.4. ***Weilandt-Hoffman theorem*** [76] *Let A and B be $n \times n$ be real symmetric matrices with eigenvalues $a_1 \geqslant \ldots \geqslant a_n$ and $b_1 \geqslant \ldots \geqslant b_n$, respectively. Let the Hermitian matrix $\widetilde{A} = A + B$ have eigenvalues $\widetilde{a}_1 \geqslant \ldots \geqslant \widetilde{a}_n$. Then the following bound holds*

$$\sum_{i=1}^{n} (\widetilde{a}_i - a_i)^2 \leqslant \|B\|_F^2 = \sum_{i=1}^{n} b_i^2. \tag{C.9}$$

Appendix D

Background on Manifold Optimization

Definition D.1 (**Gradient-related sequence**). *Given a cost function f on a Riemannian manifold $\mathcal{M}$, a sequence $\left\{\xi^{(t)}\right\}$, where $\xi^{(t)} \in T_{y^{(t)}}\mathcal{M}$, is gradient-related if, for any subsequence $\left\{y^{(t)}\right\}_{t\in\mathcal{T}}$ of $\left\{y^{(t)}\right\}$ that converges to a non-critical point of f, the corresponding subsequence $\left\{\xi^{(t)}\right\}_{t\in\mathcal{T}}$ is bounded and satisfies*

$$\limsup_{t\to\infty,\ t\in\mathcal{T}} \left\langle \nabla f(y^{(t)}), \xi^{(t)} \right\rangle < 0.$$

Here, $\langle .,. \rangle$ denotes the inner product. For a function f, descent direction at a point y refers to a vector moving along which leads to a reduction of the function. A direction ξ is a descent direction if the directional derivative along ξ is negative, that is,

$$\langle \nabla f(y), \xi \rangle < 0.$$

Definition D.1 implies that a sequence of directions $\left\{\xi^{(t)}\right\}$ on the tangent space of $\mathcal{M}$ is gradient related if it contains a subsequence of descent directions of f. Thus, moving along a gradient-related sequence at each iteration would lead to a reduction of the function f.

To ensure the convergence of the proposed algorithm, the Armijo condition [9] is imposed on the choice of step size during the optimization. The condition is defined as follows:

Definition D.2 (**Armijo criterion**). *Given a cost function f on a Riemannian manifold $\mathcal{M}$ with retraction R, a point $y \in \mathcal{M}$, a tangent vector $\xi \in T_y\mathcal{M}$, and scalars $\bar{\eta} > 0$ and $\sigma \in (0,1)$, the step length $\bar{\eta}$ is said to satisfy the Armijo condition restricted to the direction ξ if the following inequality holds:*

$$f(y) - f\big(\mathsf{R}_y(\bar{\eta}\xi)\big) \geqslant -\sigma\bar{\eta}\big\langle \nabla f(y), \xi \big\rangle. \tag{D.1}$$

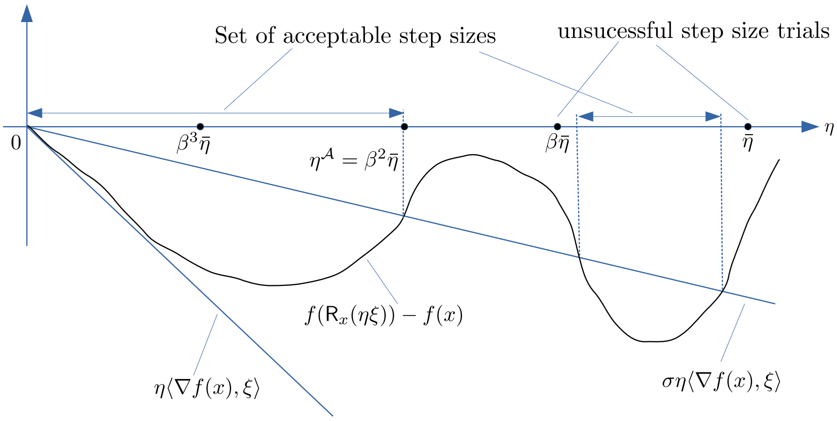

Set of acceptable step sizes
unsucessful step size trials
η
0
$\beta^3\bar{\eta}$
$\eta^{\mathcal{A}} = \beta^2\bar{\eta}$
$\beta\bar{\eta}$
$\bar{\eta}$
$f(\mathsf{R}_x(\eta\xi)) - f(x)$
$\eta\langle\nabla f(x), \xi\rangle$
$\sigma\eta\langle\nabla f(x), \xi\rangle$

Figure D.1: Armijo condition for the choice of step size.

The Armijo condition is a popular line-search condition that states that the reduction in f, given by $f(y) - f\big(\mathsf{R}_y(\bar{\eta}\xi)\big)$, should be proportional to both the step length $\bar{\eta}$ and the directional derivative $\big\langle \nabla f(y), \xi \big\rangle$, where $\sigma \in (0, 1)$ is the constant of proportionality.

Let $\boldsymbol{f}^{(t)}$ denote the value of the objective function $\boldsymbol{f}$ evaluated using $U_{\text{Joint}}^{(t)}$ and $U_j^{(t)}$'s, obtained at iteration t of the proposed algorithm. For the proposed algorithm, the step lengths for optimization on both the manifolds are chosen to be identical, that is, $\eta_{\mathsf{K}} = \eta_{\mathsf{S}} = \eta$. Also, the direction of movement on the tangent space is always the negative gradient $-\nabla \boldsymbol{f}$ (as in (6.13) and (6.19)), and the retracted point from the tangent space gives the next iterate. Between two consecutive iterations, the reduction in the objective function $\boldsymbol{f}$ is given by $\boldsymbol{f}^{(t)} - \boldsymbol{f}^{(t+1)}$. Inorder to satisfy the Armijo criterion, this reduction must be proportional to the directional derivative. This is evaluated using

$$C_{\mathcal{A}} = \boldsymbol{f}^{(t)} - \boldsymbol{f}^{(t+1)} + \sigma\eta \big\langle \nabla \boldsymbol{f}, -\nabla \boldsymbol{f} \big\rangle. \tag{D.2}$$

Here, $C_{\mathcal{A}} \geqslant 0$ implies that the Armijo condition is satisfied and there has been a sufficient reduction in the value of the objective function. The proposed algorithm moves to the next iterate only when the Armijo criterion is satisfied. The value of Armijo parameter σ is set to $1e - 05$ following [20].

Definition D.3 (**Armijo point**). *Given a cost function f on a Riemannian manifold $\mathcal{M}$ with retraction R, a point $y \in \mathcal{M}$, a tangent vector $\xi \in T_y\mathcal{M}$, and scalars $\bar{\eta} > 0$, $\beta, \sigma \in (0, 1)$, the Armijo point is $\xi^{\mathcal{A}} = \eta^{\mathcal{A}}\xi = \beta^{\omega}\bar{\eta}\xi$, where ω is the smallest non-negative integer such that*

$$f(x) - f\big(\mathsf{R}_y(\beta^{\omega}\bar{\eta}\xi)\big) \geqslant -\sigma\big\langle \nabla f(y), \beta^{\omega}\bar{\eta}\xi \big\rangle.$$

The real number $\eta^{\mathcal{A}}$ is called the Armijo step size [3].

The smallest step size that satisfies the Armijo condition is called the Armijo step size $\eta^{\mathcal{A}}$. It is given by $\eta^{\mathcal{A}} = \beta^{\omega}\bar{\eta}$, such that ω is the smallest non-negative integer to achieve this for a given $\bar{\eta} > 0$ and $\beta \in (0, 1)$. Figure D.1 shows an example of the Armijo condition for choosing the step size. To choose a step size that satisfies the Armijo condition, we start with a step length $\bar{\eta} > 0$ and then check for the choices $\beta\bar{\eta}$, $\beta^2\bar{\eta}$, $\ldots$, until $\beta^{\omega}\bar{\eta}$ falls under the set of acceptable step sizes that satisfy (D.1). This choice of step size would give a sufficient decrease in the value of the function f.

List of Publications

Published/Accepted:

J1. **Aparajita Khan** and Pradipta Maji. Multi-Manifold Optimization for Multi-View Subspace Clustering. ***IEEE Transactions on Neural Networks and Learning Systems***, pages 1-13, 2021. DOI: 10.1109/TNNLS.2021.3054789.

J2. **Aparajita Khan** and Pradipta Maji. Approximate Graph Laplacians for Multimodal Data Clustering. ***IEEE Transactions on Pattern Analysis and Machine Intelligence***, 43(3):798-813, 2021. DOI: 10.1109/TPAMI.2019.2945574.

J3. **Aparajita Khan** and Pradipta Maji. Selective Update of Relevant Eigenspaces for Integrative Clustering of Multimodal Data. ***IEEE Transactions on Cybernetics***, pages 1-13, 2020. DOI: 10.1109/TCYB.2020.2990112.

J4. **Aparajita Khan** and Pradipta Maji. Low-Rank Joint Subspace Construction for Cancer Subtype Discovery. ***IEEE/ACM Transactions on Computational Biology and Bioinformatics***, 17(4):1290-1302, 2020. DOI: 10.1109/TCBB.2019.2894635.

Submitted:

J5. **Aparajita Khan** and Pradipta Maji. Geometry Aware Multi-View Clustering over Riemannian Manifolds. ***IEEE Transactions on Pattern Analysis and Machine Intelligence***, pages 1-13, 2021 (Manuscript ID: TPAMI-2021-08-1458).

References

[1] M. Abavisani and V. M. Patel. Deep Multimodal Subspace Clustering Networks. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1601–1614, 2018.

[2] P. Absil and J. Malick. Projection-like Retractions on Matrix Manifolds. *SIAM Journal on Optimization*, 22(1):135–158, 2012.

[3] P. A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, New Jersey, 2008. ISBN:978-0-691-13298-3.

[4] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the Surprising Behavior of Distance Metrics in High Dimensional Spaces. In *Proceedings of the 8th International Conference on Database Theory*, ICDT '01, pages 420–434, Berlin, Heidelberg, 2001.

[5] G. Alexe, G. S. Dalgin, S. Ganesan, C. DeLisi, and G. Bhanot. Analysis of Breast Cancer Progression Using Principal Component Analysis and Clustering. *Journal of Biosciences*, 32:1027–1039, 2007.

[6] O. Alter, P. O. Brown, and D. Botstein. Singular Value Decomposition for Genome-Wide Expression Data Processing and Modeling. *Proceedings of the National Academy of Sciences USA*, 97(18):10101–10106, 2000.

[7] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep Canonical Correlation Analysis. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 1247–1255, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[8] P. Arabie and L. Hubert. Comparing Partitions. *Journal of Classification*, 2:193–218, 1985.

[9] L. Armijo. Minimization of Functions Having Lipschitz Continuous First Partial Derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, 1966.

[10] E. Begelfor and M. Werman. Affine Invariance Revisited. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2087–2094, 2006.

[11] A. Benton, R. Arora, and M. Dredze. Learning Multiview Embeddings of Twitter Users. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 14–19, Berlin, Germany, August 2016. Association for Computational Linguistics.

[12] A. Benton, H. Khayrallah, B. Gujral, D. A. Reisinger, S. Zhang, and R. Arora. Deep Generalized Canonical Correlation Analysis. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 1–6, Florence, Italy, August 2019. Association for Computational Linguistics.

[13] J. C. Bezdek, R. Ehrlich, and W. Full. FCM: The Fuzzy c-Means Clustering Algorithm. *Computers & Geosciences*, 10(2):191–203, 1984.

[14] M. Bibikova, B. Barnes, C. Tsan, V. Ho, B. Klotzle, J. M. Le, D. Delano, L. Zhang, G. P. Schroth, K. L. Gunderson, J.-B. Fan, and R. Shen. High Density DNA Methylation Array with Single CpG Site Resolution. *Genomics*, 98(4):288–295, 2011. New Genomic Technologies and Applications.

[15] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer-Verlag, New York, 2006. ISBN: 978-0-387-31073-2.

[16] A. Bjorck and G. H. Golub. Numerical Methods for Computing the Angles Between Linear Subspaces. *Mathematics of Computation*, 27:579–594, 1973.

[17] M. B. Blaschko and C. H. Lampert. Correlational Spectral Clustering. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pages 1–8, Los Alamitos, CA, USA, June 2008. Max-Planck-Gesellschaft, IEEE Computer Society.

[18] A. Blum and T. Mitchell. Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, COLT' 98, pages 92–100, New York, NY, USA, 1998. Association for Computing Machinery.

[19] A. Bojchevski, Y. Matkovic, and S. Günnemann. Robust Spectral Clustering for Noisy Data: Modeling Sparse Corruptions Improves Latent Embeddings. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 737âĂŞ746, New York, NY, USA, 2017. Association for Computing Machinery.

[20] Nicolas Boumal. *Optimization and Estimation on Manifolds.* PhD thesis, Université catholique de Louvain, 2014.

[21] M. Brand. Incremental Singular Value Decomposition of Uncertain Data with Missing Values. In *Proceedings of the European Conference on Computer Vision*, pages 707–720. Springer, 2002.

[22] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal. Global Cancer Statistics 2018: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. *CA: A Cancer Journal for Clinicians*, Sep 2018.

[23] E. Bruno and S. Marchand-Maillet. Multiview Clustering: A Late Fusion Approach Using Latent Models. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 736–737, 2009.

[24] D. Cai, X. He, J. Han, and T. S. Huang. Graph Regularized Nonnegative Matrix Factorization for Data Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, 2011.

[25] M. Cai and L. Li. Subtype Identification from Heterogeneous TCGA Datasets on a Genomic Scale by Multi-View Clustering with Enhanced Consensus. *BMC Medical Genomics*, 10(4):75, December 2017.

[26] X. Cai, F. Nie, and H. Huang. Multi-View K-Means Clustering on Big Data. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 2598–2604, Beijing, China, 2013.

[27] X. Cao, C. Zhang, H. Fu, Si Liu, and Hua Zhang. Diversity-Induced Multi-view Subspace Clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–594, Boston, Massachusetts, 2015.

[28] T. Carson, D. G. Mixon, and S. Villar. Manifold Optimization for k-means Clustering. In *2017 International Conference on Sampling Theory and Applications (SampTA)*, pages 73–77, July 2017.

[29] Y. Chahlaoui, K. Gallivan, and P. Van Dooren. Recursive Calculation of Dominant Singular Subspaces. *SIAM Journal on Matrix Analysis and Applications*, 25(2):445–463, 2003.

[30] M. A. Z. Chahooki and N. M. Charkari. Learning the Shape Manifold to Improve Object Recognition. *Machine Vision and Applications*, 1(24):33–46, 2013.

[31] P. Chalise, D. C. Koestler, M. Bimali, Q. Yu, and B. L. Fridley. Integrative Clustering Methods for High-Dimensional Molecular Data. *Translational Cancer Research*, 3(3):202, 2014.

[32] S. Chandrasekaran, B. S. Manjunath, Y. F. Wang, J. Winkeler, and H. Zhang. An Eigenspace Update Algorithm for Image Analysis. *Graphical Models and Image Processing*, 59(5):321–332, 1997.

[33] S. Chang, J. Hu, T. Li, H. Wang, and B. Peng. Multi-View Clustering via Deep Concept Factorization. *Knowledge-Based Systems*, 217:106807, 2021.

[34] G. Chao, S. Sun, and J. Bi. A Survey on Multi-View Clustering. *arXiv e-prints*, page arXiv:1712.06246, December 2017.

[35] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan. Multi-View Clustering via Canonical Correlation Analysis. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 129–136, New York, NY, USA, 2009. Association for Computing Machinery.

[36] F. Chen, G. Li, S. Wang, and Z. Pan. Multiview Clustering via Robust Neighboring Constraint Nonnegative Matrix Factorization. *Mathematical Problems in Engineering*, 2019:1–10, November 2019.

[37] J. Chen, G. Wang, and G. B. Giannakis. Graph Multiview Canonical Correlation Analysis. *IEEE Transactions on Signal Processing*, 67(11):2826–2838, 2019.

[38] J. Chen, G. Wang, and G. B. Giannakis. Multiview Canonical Correlation Analysis over Graphs. In *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2947–2951, 2019.

[39] Y. Chen, S. Wang, C. Peng, Z. Hua, and Y. Zhou. Generalized Nonconvex Low-Rank Tensor Approximation for Multi-View Subspace Clustering. *IEEE Transactions on Image Processing*, 30:4022–4035, 2021.

[40] Y. Chen, X. Xiao, and Y. Zhou. Multi-view Clustering via Simultaneously Learning Graph Regularized Low-Rank Tensor Representation and Affinity Matrix. In *Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1348–1353, 2019.

[41] Y. Chen, X. Xiao, and Y. Zhou. Jointly Learning Kernel Representation Tensor and Affinity Matrix for Multi-View Clustering. *IEEE Transactions on Multimedia*, 22(8):1985–1997, 2020.

[42] Y. Chen, X. Xiao, and Y. Zhou. Multi-view Subspace Clustering via Simultaneously Learning the Representation Tensor and Affinity Matrix. *Pattern Recognition*, 106:107441, 2020.

[43] C. M. Christoudias, R. Urtasun, and T. Darrell. Multi-View Learning in the Presence of View Disagreement. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, UAI'08, pages 88–96, Arlington, Virginia, USA, 2008.

[44] D. Chu, L. Liao, M. K. Ng, and X. Zhang. Sparse Canonical Correlation Analysis: New Formulation and Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):3050–3065, 2013.

[45] F. R. K. Chung. *Spectral Graph Theory*. Number 92. American Mathematical Society, Providence, Rhode Island, 1997. ISBN: 0-8218-0315-8.

[46] P. Coretto, A. Serra, and R. Tagliaferri. Robust Clustering of Noisy High-Dimensional Gene Expression Data for Patients Subtyping. *Bioinformatics*, 34(23):4064–4072, 2018.

[47] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273âĂŞ297, September 1995.

[48] D. L. Davies and D. W. Bouldin. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.

[49] C. Davis and W. Kahan. The Rotation of Eigenvectors by a Perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

[50] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[51] C. Dhanjal, R. Gaudel, and S. Clémençon. Efficient Eigen-Updating for Spectral Graph Clustering. *Neurocomputing*, 131:440–452, 2014.

[52] C. Ding and X. He. K-means Clustering via Principal Component Analysis. In *Proceedings of the 21st International Conference on Machine learning*, page 29. ACM, 2004.

[53] H. Ding, M. Sharpnack, C. Wang, K. Huang, and R. Machiraju. Integrative Cancer Patient Stratification via Subspace Merging. *Bioinformatics*, 35(10):1653–1659, May 2019.

[54] A. Djelouah, J. Franco, E. Boyer, F. Le Clerc, and P. PÃľrez. Sparse Multi-View Consistency for Object Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1890–1903, 2015.

[55] J. C. Dunn. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.

[56] C. Eckart and G. Young. The Approximation of One Matrix by Another of Lower Rank. *Psychometrika*, 1(3):211–218, Sep 1936.

[57] A. Edelman, T. A. Arias, and S. T. Smith. The Geometry of Algorithms with Orthogonality Constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, April 1999.

[58] Nour El Din Elmadany, Yifeng He, and Ling Guan. Multiview Learning via Deep Discriminative Canonical Correlation Analysis. In *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2409–2413, 2016.

[59] E. Elhamifar and R. Vidal. Sparse Subspace Clustering. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2797, 2009.

[60] E. Elhamifar and R. Vidal. Sparse Subspace Clustering: Algorithm, Theory, and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013.

[61] J. Fang, D. Lin, S. C. Schulz, Z. Xu, V. D. Calhoun, and Y. P. Wang. Joint Sparse Canonical Correlation Analysis for Detecting Differential Imaging Genetics Modules. *Bioinformatics*, 32(22):3480–3488, November 2016.

[62] J. Farquhar, D. Hardoon, H. Meng, J. Shawe-taylor, and S. Szedmák. Two View Learning: SVM-2K, Theory and Practice. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2006.

[63] Q. Feng, M. Jiang, J. Hannig, and J.S. Marron. Angle-Based Joint and Individual Variation Explained. *Journal of Multivariate Analysis*, 166:241–265, 2018.

[64] J. Ferlay, I. Soerjomataram, R. Dikshit, S. Eser, C. Mathers, M. Rebelo, D. M. Parkin, D. Forman, and F. Bray. Cancer Incidence and Mortality Worldwide: Sources, Methods and Major Patterns in GLOBOCAN 2012. *International Journal of Cancer*, 136(5):E359–386, Mar 2015.

[65] J. Ferlay, I. Soerjomataram, M. Ervik, R. Dikshit, S. Eser, C. Mathers, M. Rebelo, D. M. Parkin, D. Forman, and F. Bray. GLOBOCAN 2012 v1.0, Cancer Incidence and Mortality Worldwide: IARC CancerBase No. 11, 2013.

[66] M. Fiedler. A Property of Eigenvectors of Nonnegative Symmetric Matrices and its Application to Graph Theory. *Czechoslovak Mathematical Journal*, 25(4):619–633, 1975.

[67] P. Flach. *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press, New York, 2012. ISBN: 978-1-107-09639-4.

[68] A. L. N. Fred and A. K. Jain. Robust Data Clustering. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 3, pages 128–136, 2003.

[69] J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*, volume 1. Springer series in statistics New York, 2001. ISBN:978-0-387-84857-0.

[70] K. Fukui and A. Maki. Difference Subspace and Its Generalization for Subspace-Based Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2164–2177, Nov 2015.

[71] H. Gao, F. Nie, X. Li, and H. Huang. Multi-view subspace clustering. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4238–4246, 2015.

[72] Q. Gao, H. Lian, Q. Wang, and G. Sun. Cross-Modal Subspace Clustering via Deep Canonical Correlation Analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3938–3945, Apr 2020.

[73] Q. Gao, J. Ma, H. Zhang, X. Gao, and Y. Liu. Stable orthogonal local discriminant embedding for linear dimensionality reduction. *IEEE Transactions on Image Processing*, 22(7):2521–2531, 2013.

[74] Z. Gao, Y. Wu, Y. Jia, and M. Harandi. Learning to Optimize on SPD Manifolds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7697–7706, 2020.

[75] M. E. Garber, O. G. Troyanskaya, K. Schluens, S. Petersen, Z. Thaesler, M. Pacyna-Gengelbach, M. van de Rijn, G. D. Rosen, C. M. Perou, R. I. Whyte, R. B. Altman, P. O. Brown, D. Botstein, and I. Petersen. Diversity of Gene Expression in Adenocarcinoma of the Lung. *Proceedings of the National Academy of Sciences*, 98(24):13784–13789, 2001.

[76] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, MD, USA, 1996. ISBN:0-8018-5414-8.

[77] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning.* MIT Press, 2016. ISBN:978-0262035613.

[78] D. Greene and P. Cunningham. Producing a Unified Graph Representation from Multiple Social Network Views. In *Proceedings of the 5th Annual ACM Web Science Conference*, WebSci '13, pages 118–121, New York, NY, USA, 2013. ACM.

[79] Z. Gu, Z. Zhang, J. Sun, and B. Li. Robust Image Recognition by L1-norm Twin-Projection Support Vector Machine. *Neurocomputing*, 223:1–11, 2017.

[80] C. Guo and D. Wu. Canonical Correlation Analysis (CCA) Based Multi-View Learning: An Overview. *CoRR*, abs/1907.01693, 2019.

[81] D. Guo, J. Zhang, X. Liu, Y. Cui, and C. Zhao. Multiple Kernel Learning Based Multi-view Spectral Clustering. In *Proceedings of the 22nd International Conference on Pattern Recognition*, pages 3774–3779, 2014.

[82] J. Guo, Y. Sun, J. Gao, Y. Hu, and B. Yin. Low rank representation on product grassmann manifolds for multi-view subspace clustering. In *Proceedings of the 25th International Conference on Pattern Recognition, 2020. ICPR 2020.*, 08 2020.

[83] Y. Guo and M. Xiao. Cross Language Text Classification via Subspace Co-Regularized Multi-View Learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 915âĂŞ922, Madison, WI, USA, 2012. Omnipress.

[84] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On Clustering Validation Techniques. *Journal of intelligent information systems*, 17(2-3):107–145, 2001.

[85] P. Hall, D. Marshall, and R. Martin. Merging and Splitting Eigenspace Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):1042–1049, 2000.

[86] P. Hall, D. Marshall, and R. Martin. Adding and Subtracting Eigenspaces with Eigenvalue Decomposition and Singular Value Decomposition. *Image and Vision Computing*, (20):1009–1016, 2002.

[87] G. Hamerly and C. Elkan. Learning the k in k-means. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 281–288, 2004.

[88] Han, J. and Kamber, M. and Pei, J. *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011. ISBN: 978-0123814791.

[89] Y. Hasin, M. Seldin, and A. Lusis. Multi-Omics Approaches to Disease. *Genome Biology*, 18(1):83, May 2017.

[90] K. A. Heller and Z. Ghahramani. Bayesian Hierarchical Clustering. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 297–304, 2005.

[91] K. H. Hellton and M. Thoresen. Integrative Clustering of High-Dimensional Data with Joint and Individual Clusters. *Biostatistics*, 17(3):537–548, 02 2016.

[92] C. Hennig. Cluster-Wise Assessment of Cluster Stability. *Computational Statistics & Data Analysis*, 52(1):258–271, 2007.

[93] K. A Hoadley, C. Yau, et al. Multiplatform Analysis of 12 Cancer Types Reveals Molecular Classification Within and Across Tissues of Origin. *Cell*, 158(4):929–944, 2014.

[94] M. Horie and H. Kasai. Consistency-Aware and Inconsistency-Aware Graph-Based Multi-View Clustering. In *Proceedings of the 28th European Signal Processing Conference (EUSIPCO)*, pages 1472–1476, 2021.

[95] Paul Horst. Relations Among m Sets of Measures. *Psychometrika*, 26:129–149, 1961.

[96] D. W. Hosmer, S. Lemeshow, and S. May. *Applied Survival Analysis: Regression Modeling of Time to Event Data.* Wiley-Interscience, New York, NY, USA, 2nd edition, 2008. ISBN:9780471754992.

[97] Harold Hotelling. Relations Between Two Sets of Variates. *Biometrika*, 28(3/4):321–377, 1936.

[98] C. Hou, F. Nie, H. Tao, and D. Yi. Multi-View Unsupervised Feature Selection with Adaptive Similarity and View Weight. *IEEE Transactions on Knowledge and Data Engineering*, 29(9):1998–2011, 2017.

[99] Z. Hu et al. The Molecular Portraits of Breast Tumors are Conserved Across Microarray Platforms. *BMC Genomics*, 7:96, April 2006.

[100] C. Huang, F. Chung, and S. Wang. Multi-View L2-SVM and Its Multi-View Core Vector Machine. *Neural Networks*, 75(C):110–125, March 2016.

[101] J. Huang, F. Nie, H. Huang, and C. Ding. Robust Manifold Nonnegative Matrix Factorization. *ACM Transactions on Knowledge Discovery from Data*, 8(3), June 2014.

[102] S. Huang, K. Chaudhary, and L. X Garmire. More is Better: Recent Progress in Multi-omics Data Integration Methods. *Frontiers in Genetics*, 8:84, 2017.

[103] S. Huang, Z. Kang, and Z. Xu. Auto-weighted Multi-View Clustering via Deep Matrix Decomposition. *Pattern Recognition*, 97:107015, 2020.

[104] Y. Huang, W. Wang, L. Wang, and T. Tan. A General Nonlinear Embedding Framework Based on Deep Neural Network. In *Proceedings of the 22nd International Conference on Pattern Recognition*, pages 732–737, 2014.

[105] Y. Ji, Q. Wang, X. Li, and J. Liu. A Survey on Tensor Techniques and Applications in Machine Learning. *IEEE Access*, 7:162950–162990, 2019.

[106] S. Ji-guang. Perturbation of Angles Between Linear Subspaces. *Journal of Computational Mathematics*, 5(1):58–61, 1987.

[107] Y. Jia, H. Liu, J. Hou, S. Kwong, and Q. Zhang. Multi-View Spectral Clustering Tailored Tensor Low-Rank Representation. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2021.

[108] S. Jing-Tao and Z. Qiu-Yu. Completion of Multiview Missing Data based on Multi-manifold Regularised Non-negative Matrix Factorisation. *Artificial Intelligence Review*, 53(7):5411–5428, 2020.

[109] S. Jung and J. S. Marron. PCA Consistency in High Dimension, Low Sample Size Context. *The Annals of Statistics*, 37(6B):4104 – 4130, 2009.

[110] M. Kan, S. Shan, and X. Chen. Multi-View Deep Network for Cross-View Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4847–4855, 2016.

[111] A. Khan and P. Maji. Low-rank joint subspace construction for cancer subtype discovery. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 17(4):1290–1302, 2020. DOI: 10.1109/TCBB.2019.2894635.

[112] A. Khan and P. Maji. Selective Update of Relevant Eigenspaces for Integrative Clustering of Multimodal Data. *IEEE Transactions on Cybernetics*, pages 1–13, 2020. DOI: 10.1109/TCYB.2020.2990112.

[113] A. Khan and P. Maji. Approximate Graph Laplacians for Multimodal Data Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):798–813, 2021. DOI: 10.1109/TPAMI.2019.2945574.

[114] A. Khan and P. Maji. Multi-Manifold Optimization for Multi-View Subspace Clustering. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–13, 2021. DOI: 10.1109/TNNLS.2021.3054789.

[115] P. Kirk, J. E. Griffin, R. S. Savage, Z. Ghahramani, and D. L. Wild. Bayesian Correlated Clustering to Integrate Multiple Datasets. *Bioinformatics*, 28(24):3290–3297, Dec 2012.

[116] A. V. Knyazev and P. Zhu. Principal Angles Between Subspaces and their Tangents. Technical Report TR2012-058, Mitsubishi Electric Research Laboratories, September 2012.

[117] M. Kosinski. *RTCGA.clinical: Clinical Datasets from The Cancer Genome Atlas Project*, 2016. R package version 20151101.6.0.

[118] H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1âĂŘ2):83–97, 1955.

[119] A. Kumar and H. Daume III. A Co-Training Approach for Multi-View Spectral Clustering. In *Proceedings of the 28th International Conference on Machine Learning*, ICML'11, pages 393–400, Madison, WI, USA, 2011. Omnipress.

[120] A. Kumar, P. Rai, and H. Daumé. Co-Regularized Multi-View Spectral Clustering. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, pages 1413–1421, Red Hook, NY, USA, 2011. Curran Associates Inc.

[121] C. Lan, Y. Deng, X. Li, and J. Huan. Co-regularized Least Square Regression for Multi-view Multi-class Classification. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 342–347, 2016.

[122] B. Larsen and C. Aone. Fast and effective text mining using linear time document clustering. In *In Proc. Knowledge Discovery and Data mining*, pages 16–22, San Diego, USA, 1999.

[123] D. D. Lee and H. S. Seung. Algorithms for Non-Negative Matrix Factorization. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'00, page 535âĂŞ541, Cambridge, MA, USA, 2000. MIT Press.

[124] G. Li, S. C. H. Hoi, and K. Chang. Two-View Transductive Support Vector Machines. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA*, pages 235–244. SIAM, 2010.

[125] J. Li, N. Allinson, D. Tao, and X. Li. Multitraining Support Vector Machine for Image Retrieval. *IEEE Transactions on Image Processing*, 15(11):3597–3601, 2006.

[126] J. Li, L. Xie, Y. Xie, and F. Wang. Bregmannian Consensus Clustering for Cancer Subtypes Analysis. *Computer Methods and Programs in Biomedicine*, 189:105337, June 2020.

[127] J. Li, C. Xu, W. Yang, C. Sun, and D. Tao. Discriminative Multi-View Interactive Image Re-Ranking. *IEEE Transactions on Image Processing*, 26(7):3113–3127, July 2017.

[128] X. Li, H. Zhang, R. Wang, and F. Nie. Multi-view clustering: A scalable and parameter-free bipartite graph fusion method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.

[129] Y. Li, F. Nie, H. Huang, and J. Huang. Large-Scale Multi-View Spectral Clustering via Bipartite Graph. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, AAAI'15, page 2750âĂŞ2756. AAAI Press, 2015.

[130] Y. Li, M. Yang, and Z. Zhang. A Survey of Multi-View Representation Learning. *IEEE Transactions on Knowledge and Data Engineering*, 31(10):1863–1883, 2019.

[131] Z. Li, Q. Wang, Z. Tao, Q. Gao, and Z. Yang. Deep Adversarial Multi-View Clustering Network. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2952–2958, July 2019.

[132] Y. Liang, D. Huang, and C. Wang. Consistency Meets Inconsistency: A Unified Graph Learning Framework for Multi-View Clustering. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 1204–1209, 2019.

[133] D. Lin, J. Zhang, J. Li, V. D. Calhoun, H. W. Deng, and Y. P. Wang. Group Sparse Canonical Correlation Analysis for Genomic Data Integration. *BMC Bioinformatics*, 14:245, Aug 2013.

[134] Y. Lin, T. Liu, and C. Fuh. Multiple Kernel Learning for Dimensionality Reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(6):1147–1160, June 2011.

[135] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust Recovery of Subspace Structures by Low-Rank Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

[136] J. Liu, F. Cao, X.-Z. Gao, L. Yu, and J. Liang. A Cluster-Weighted Kernel K-Means Method for Multi-View Clustering. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 4860–4867. AAAI Press, 2020.

[137] J. Liu, C. Wang, J. Gao, and J. Han. Multi-View Clustering via Joint Nonnegative Matrix Factorization. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 252–260, 2013.

[138] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, and W. Gao. Late Fusion Incomplete Multi-View Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(10):2410–2423, 2019.

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

[140] E. F. Lock and D. B. Dunson. Bayesian consensus clustering. *Bioinformatics*, 29(20):2610–2616, 2013.

[141] E. F. Lock, K. A. Hoadley, J. S. Marron, and A. B Nobel. Joint and Individual Variation Explained (JIVE) for Integrated Analysis of Multiple Data Types. *The Annals of Applied Statistics*, 7(1):523–542, 2013.

[142] B. Long, P. S. Yu, and Z. Zhang. A General Model for Multiple View Unsupervised Learning. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 822–833. SIAM, 2008.

[143] Y. M. Lui, J. R. Beveridge, and M. Kirby. Canonical Stiefel Quotient and its Application to Generic Face Recognition in Illumination Spaces. In *Proceedings of the IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems*, pages 1–8, 2009.

[144] Y. Luo, D. Tao, K. Ramamohanarao, C. Xu, and Y. Wen. Tensor Canonical Correlation Analysis for Multi-View Dimension Reduction. *IEEE Transactions on Knowledge and Data Engineering*, 27(11):3111–3124, 2015.

[145] Y. Luo, D. Tao, C. Xu, C. Xu, H. Liu, and Y. Wen. Multiview Vector-Valued Manifold Regularization for Multilabel Image Classification. *IEEE Transactions on Neural Networks and Learning Systems*, 24(5):709–722, 2013.

[146] Y. Ma, X. Hu, T. He, and X. Jiang. Clustering and Integrating of Heterogeneous Microbiome Data by Joint Symmetric Nonnegative Matrix Factorization with Laplacian Regularization. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 17(3):788–795, 2020.

[147] M. Maila and J. Shi. A Random Walks View of Spectral Segmentation. In *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, volume R3 of *Proceedings of Machine Learning Research*, pages 203–208. PMLR, 04–07 Jan 2001. Reissued by PMLR on 31 March 2021.

[148] P. Maji and S. Paul. *Scalable Pattern Recognition Algorithms: Applications in Computational Biology and Bioinformatics.* Springer-Verlag, London, April 2014. ISBN: 978-3-319-05629-6.

[149] A. Mandal and P. Maji. FaRoC: Fast and Robust Supervised Canonical Correlation Analysis for Multimodal Omics Data. *IEEE Transactions on Cybernetics*, 48(4):1229–1241, 2018.

[150] C. J. Mecklin. *A Comparison of the Power of Classical and Newer Tests of Multivariate Normality.* PhD thesis, University of Northern Colorado, 2000.

[151] M. Meilă. The Uniqueness of a Good Optimum for k-means. In *Proceedings of the 23rd International Conference on Machine learning*, pages 625–632. ACM, 2006.

[152] M. Meila and J. Shi. Learning Segmentation by Random Walks. In *Proceedings of the Advances in Neural Information Processing Systems 13*, pages 873–879. MIT Press, 2001.

[153] M. Mendes and A. Pala. Type I Error Rate and Power of Three Normality Tests. *Pakistan Journal of Information and Technology*, 2(2):135–139, 2003.

[154] G. F. Miranda, C. E. Thomaz, and G. A. Giraldi. Geometric Data Analysis Based on Manifold Learning with Applications for Image Understanding. In *Proceedings of the 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, pages 42–62, Oct 2017.

[155] Q. Mo and R. Shen. *iClusterPlus: Integrative clustering of multi-type genomic data*, 2016. R package version 1.12.1.

[156] Q. Mo, S. Wang, V. E. Seshan, A. B. Olshen, N. Schultz, C. Sander, R S. Powers, M. Ladanyi, and R. Shen. Pattern Discovery and Cancer Gene Identification in Integrated Cancer Genomic Data. *Proceedings of the National Academy of Sciences*, 110(11):4245–4250, 2013.

[157] B. Moghaddam and A. Pentland. Probabilistic Visual Learning for Object Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, 1997.

[158] B. Mohar, Y. Alavi, G. Chartrand, and O. R. Oellermann. The Laplacian Spectrum of Graphs. *Graph Theory, Combinatorics, and Applications*, 2(871-898):12, 1991.

[159] C. Moler and C. Loan. Nineteen Dubious Ways to Compute the Exponential of a Matrix. *SIAM Review*, 20:801–836, 10 1978.

[160] Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus Clustering: a Resampling-based Method for Class Discovery and Visualization of Gene Expression Microarray Data. *Machine Learning*, 52:91–118, 2003.

[161] T. K. Moon. The Expectation-Maximization Algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996.

[162] A. Y. Ng, M. I. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an Algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pages 849–856, 2001.

[163] N. D. Nguyen, I. K. Blaby, and D. Wang. ManiNetCluster: a Novel Manifold Learning Approach to Reveal the Functional Links Between Gene Networks. *BMC Genomics*, 20(Suppl 12):1003, December 2019.

[164] F. Nie, J. Li, and X. Li. Parameter-Free Auto-Weighted Multiple Graph Learning: A Framework for Multiview Clustering and Semi-Supervised Classification. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 1881–1887. AAAI Press, 2016.

[165] F. Nie, J. Li, and X. Li. Self-weighted Multiview Clustering with Multiple Graphs. In *Proceedings of the26th International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2564–2570, 2017.

[166] Feiping Nie, Guohao Cai, and Xuelong Li. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 2408âĂŞ2414. AAAI Press, 2017.

[167] G. Niu, Y. Yang, and L. Sun. One-Step Multi-View Subspace Clustering with Incomplete Views. *Neurocomputing*, 438:290–301, 2021.

[168] L. Niu, W. Li, D. Xu, and J. Cai. An Exemplar-Based Multi-View Domain Generalization Framework for Visual Recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 29(2):259–272, 2018.

[169] W. Ou, S. Yu, G. Li, J. Lu, K. Zhang, and G. Xie. Multi-View Non-negative Matrix Factorization by Patch Alignment Framework with View Consistency. *Neurocomputing*, 204:116–124, 2016. Big Learning in Social Media Analytics.

[170] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, and W. Ziarko. Rough Sets. 38(11):88âĂŞ95, November 1995.

[171] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Multi-View and 3D Deformable Part Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2232–2245, November 2015.

[172] R. Peto and J. Peto. Asymptotically Efficient Rank Invariant Test Procedures. *Journal of the Royal Statistical Society. Series A (General)*, 135(2):185–207, 1972.

[173] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[174] N. Rappoport and R. Shamir. Multi-Omic and Multi-View Mlustering Algorithms: Review and Cancer Benchmark. *Nucleic Acids Research*, 46(20):10546–10562, November 2018.

[175] N. Rasiwasia, D. Mahajan, V. Mahadevan, and G. Aggarwal. Cluster Canonical Correlation Analysis. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 823–831, Reykjavik, Iceland, 22–25 Apr 2014. PMLR.

[176] N. M. Razali and Y. B. Wah. Power Comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling Tests. *Journal of statistical modeling and analytics*, 2(1):21–33, 2011.

[177] E. Rendón, I. M. Abundez, C. Gutierrez, S. Zagal, A. Arizmendi, E. M. Quiroz, and H. E. Arzate. A Comparison of Internal and External Cluster Validation Indexes. In *Proceedings of the 2011 American Conference on Applied Mathematics and the 5th WSEAS International Conference on Computer Engineering and Applications*, pages 158–163, 2011.

[178] W. Rong, E. Zhuo, H. Peng, J. Chen, H. Wang, C. Han, and H. Cai. Learning a Consensus Affinity Matrix for Multi-View Clustering via Subspaces Merging on Grassmann Manifold. *Information Sciences*, 547:68–87, 2021.

[179] P. J. Rousseeuw. Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[180] S. T. Roweis and L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, December 2000.

[181] J. P. Royston. An Extension of Shapiro and Wilk's W Test for Normality to Large Samples. *Applied Statistics*, pages 115–124, 1982.

[182] J. P. Royston. Some Techniques for Assessing Multivariate Normality Based on the Shapiro-Wilk W. *Applied Statistics*, pages 121–133, 1983.

[183] J. P. Royston. Approximating the Shapiro-Wilk W-test for Non-normality, journal=Statistics and Computing. 2(3):117–119, 1992.

[184] J. Ruiz-del-Solar and P. Navarrete. Recursive Estimation of Motion Parameters. *Computer Vision and Image Understanding*, 64(3):434–442, 1996.

[185] J. Ruiz-del-Solar and P. Navarrete. Eigenspace-Based Face Recognition: A Comparative Study of Different Approaches. *IEEE Transactions on Systems, Man, and Cybernetics*, 35(2):315–325, 2006.

[186] L. K. Saul and S. T. Roweis. Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds. *Journal of Machine Learning Research*, 4:119–155, December 2003.

[187] M. Seeland and P. MÃď der. Multi-View Classification with Convolutional Neural Networks. *PLOS ONE*, 16(1):1–17, 01 2021.

[188] H. S. Seung and D. D. Lee. Cognition. The Manifold Ways of Perception. *Science*, 290(5500):2268–2269, December 2000.

[189] S. S. Shapiro and M. B. Wilk. An Analysis of Variance Test for Normality (Complete Samples). *Biometrika*, 52(3/4):591–611, 1965.

[190] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, 2004. ISBN: 10.1017/CBO9780511809682.

[191] R. Shen, Q. Mo, N. Schultz, V. E. Seshan, A. B. Olshen, J. Huse, M. Ladanyi, and C. Sander. Integrative Subtype Discovery in Glioblastoma using iCluster. *PloS One*, 7(4):e35236, 2012.

[192] R. Shen, A. B. Olshen, and M. Ladanyi. Integrative Clustering of Multiple Genomic Data Types using a Joint Latent Variable Model with Application to Breast and Lung Cancer Subtype Analysis. *Bioinformatics*, 25(22):2906–2912, 2009.

[193] R. Shen, S. Wang, and Q. Mo. Sparse Integrative Clustering of Multiple Omics Data Sets. *The Annals of Applied Statistics*, 7(1):269–294, 2013.

[194] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug 2000.

[195] S. Shirazi, M. T. Harandi, B. C. Lovell, and C. Sanderson. Object Tracking via Non-Euclidean Geometry: A Grassmann Approach. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 901–908, 2014.

[196] V. Sindhwani and P. Niyogi. A Co-regularized Approach to Semi-supervised Learning with Multiple Views. In *Proceedings of the ICML Workshop on Learning with Multiple Views*, 2005.

[197] V. Sindhwani and D. S. Rosenberg. An RKHS for Multi-View Learning and Manifold Co-Regularization. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 976–983, New York, NY, USA, 2008. Association for Computing Machinery.

[198] T. Sorlie, C. M. Perou, R. Tibshirani, T. Aas, S. Geisler, H. Johnsen, T. Hastie, M. B. Eisen, M. van de Rijn, S. S. Jeffrey, T. Thorsen, H. Quist, J. C. Matese, P. O. Brown, D. Botstein, P. E. LÃÿnning, and A. L. Borresen-Dale. Gene Expression Patterns of Breast Carcinomas Distinguish Tumor Subclasses with Clinical Implications. *Proceedings of the National Academy of Sciences U.S.A.*, 98(19):10869–10874, September 2001.

[199] N. K. Speicher and N. Pfeifer. Integrating Different Data Types by Regularized Unsupervised Multiple Kernel Learning with Application to Cancer Subtype Discovery. *Bioinformatics*, 31(12):i268–i275, 06 2015.

[200] D. A. Spielman and S.-H. Teng. Spectral Partitioning Works: Planar Graphs and Finite Element Meshes. *Linear Algebra and its Applications*, 421(2):284 – 305, 2007. Special Issue in honor of Miroslav Fiedler.

[201] A. Srivastava and E. Klassen. Bayesian and Geometric Subspace Tracking. *Advances in Applied Probability*, 36(1):43–56, 2004.

[202] G. W. Stewart and J. Sun. *Matrix Perturbation Theory*. Academic press, New York, 1990. ISBN: 9780126702309.

[203] S. Sun, L. Mao, Z. Dong, and L. Wu. *Multiview Machine Learning*. Springer Singapore, Singapore, 2019. ISBN: 978-981-13-3029-2.

[204] S. Sun and J. Shawe-Taylor. Sparse Semi-supervised Learning Using Conjugate Functions. *Journal of Machine Learning Research*, 11(84):2423–2455, 2010.

[205] S. Sun, X. Xie, and C. Dong. Multiview Learning With Generalized Eigenvalue Proximal Support Vector Machines. *IEEE Transactions on Cybernetics*, 49(2):688–697, 2019.

[206] S. Sun, X. Xie, and M. Yang. Multiview Uncorrelated Discriminant Analysis. *IEEE Transactions on Cybernetics*, 46(12):3272–3284, 2016.

[207] S. Sun and D. Zong. LCBM: A Multi-View Probabilistic Model for Multi-label Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.

[208] T. Sun, S. Chen, J. Yang, and P. Shi. A Novel Method of Combined Feature Extraction for Recognition. In *Proceedings of the 8th IEEE International Conference on Data Mining*, pages 1043–1048, 2008.

[209] H. Tabia and H. Laga. Covariance-Based Descriptors for Efficient 3D Shape Matching, Retrieval, and Classification. *IEEE Transactions on Multimedia*, 17(9):1591–1603, 2015.

[210] J. Tang, Y. Tian, P. Zhang, and X. Liu. Multiview Privileged Support Vector Machines. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8):3463–3477, 2018.

[211] X. Tang, X. Tang, W. Wang, L. Fang, and X. Wei. Deep Multi-View Sparse Subspace Clustering. In *Proceedings of the 7th International Conference on Network, Communication and Computing*, ICNCC 2018, pages 115–119, New York, NY, USA, 2018. Association for Computing Machinery.

[212] H. Tao, C. Hou, Y. Qian, J. Zhu, and D. Yi. Latent Complete Row Space Recovery for Multi-View Subspace Clustering. *IEEE Transactions on Image Processing*, 29:8083–8096, 2020.

[213] H. Tao, C. Hou, J. Zhu, and D. Yi. Multi-View Clustering with Adaptively Learned Graph. In *Proceedings of the 9th Asian Conference on Machine Learning*, volume 77 of *Proceedings of Machine Learning Research*, pages 113–128. PMLR, November 2017.

[214] TCGA Network. Comprehensive Molecular Portraits of Human Breast Tumours. *Nature*, 490(7418):61–70, October 2012.

[215] TCGA Research Network. Integrated Genomic Analyses of Ovarian Carcinoma. *Nature*, 474(7353):609–615, Jun 2011.

[216] TCGA Research Network. Comprehensive Molecular Characterization of Gastric Adenocarcinoma. *Nature*, 513(7517):202–209, 2014.

[217] TCGA Research Network. Comprehensive, Integrative Genomic Analysis of Diffuse Lower-Grade Gliomas. *The New England Journal of Medicine*, 372(26):2481–2498, 2015.

[218] TCGA Research Network. Integrated Genomic and Molecular Characterization of Cervical Cancer. *Nature*, 543(7645):378–384, 2017.

[219] A. Tenenhaus and M. Tenenhaus. Regularized Generalized Canonical Correlation Analysis. *Psychometrika*, 76(2):257–284, April 2011.

[220] S. Theodoridis and K. Koutroumbas. *Pattern Recognition.* Academic Press, Inc., USA, 4th edition, 2008. ISBN: 9781597492720.

[221] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[222] W. D. Travis, E. Brambilla, A. P. Burke, A. Marx, and A. G. Nicholson. Introduction to The 2015 World Health Organization Classification of Tumors of the Lung, Pleura, Thymus, and Heart. *Journal of Thoracic Oncology*, 10(9):1240–1242, September 2015.

[223] A. Trivedi, P. Rai, H. Daume III, and S. L. DuVall. Multiview Clustering with Incomplete Views. In *Proceedings of the Neural Information Processing Systems Workshop*, volume 224, pages 1–8, 2010.

[224] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical Computations on Grassmann and Stiefel Manifolds for Image and Video-Based Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, 2011.

[225] G. Tzortzis and A. Likas. Kernel-Based Weighted Multi-View Clustering. In *Proceedings of the IEEE 12th International Conference on Data Mining*, pages 675–684, 2012.

[226] V. Vapnik and R. Izmailov. Learning Using Privileged Information: Similarity Control and Knowledge Transfer. *Journal of Machine Learning Research*, 16(61):2023–2049, 2015.

[227] V. Vapnik and A. Vashist. A New Learning Paradigm: Learning using Privileged Information. *Neural Networks*, 22(5):544–557, 2009. Advances in Neural Networks Research: IJCNN2009.

[228] R. G. Verhaak, K. A. Hoadley, E. Purdom, V. Wang, Y. Qi, M. D. Wilkerson, C. R. Miller, L. Ding, T. Golub, J. P. Mesirov, G. Alexe, M. Lawrence, M. O'Kelly, P. Tamayo, B. A. Weir, S. Gabriel, W. Winckler, S. Gupta, L. Jakkula, H. S. Feiler, J. G. Hodgson, C. D. James, J. N. Sarkaria, C. Brennan, A. Kahn, P. T. Spellman, R. K. Wilson, T. P. Speed, J. W. Gray, M. Meyerson, G. Getz, C. M. Perou, and D. N. Hayes. Integrated Genomic Analysis Identifies Clinically Relevant Subtypes of Glioblastoma Characterized by Abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer Cell*, (17):98–110, 2010.

[229] J. Via, I. Santamaria, and J. Perez. A Learning Algorithm for Adaptive Canonical Correlation Analysis of Several Data Sets. *Neural Networks*, 20(1):139–152, 2007.

[230] U. Von Luxburg. A Tutorial on Spectral Clustering. *Statistics and computing*, 17(4):395–416, 2007.

[231] D. Wagner and F. Wagner. Between Min Cut and Graph Bisection. In *Proceedings of the International Symposium on Mathematical Foundations of Computer Science*, pages 744–750. Springer, 1993.

[232] B. Wang, Y. Hu, J. Gao, Y. Sun, F. Ju, and B. Yin. Adaptive Fusion of Heterogeneous Manifolds for Subspace Clustering. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2020.

[233] B. Wang, Y. Hu, J. Gao, Y. Sun, F. Ju, and B. Yin. Learning Adaptive Neighborhood Graph on Grassmann Manifolds for Video/Image-Set Subspace Clustering. *IEEE Transactions on Multimedia*, 23:216–227, 2021.

[234] B. Wang, A. M. Mezlini, F. Demir, M. Fiume, Z. Tu, M. Brudno, B. Haibe-Kains, and A. Goldenberg. Similarity Network Fusion for Aggregating Data Types on a Genomic Scale. *Nature Methods*, 11:333–337, 2014.

[235] H. Wang, F. Nie, and H. Huang. Multi-View Clustering and Feature Learning via Structured Sparsity. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 352–360, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[236] H. Wang, Y. Yang, and B. Liu. GMC: Graph-Based Multi-View Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1116–1129, 2020.

[237] H. Wang, Y. Yang, B. Liu, and H. Fujita. A Study of Graph-Based System for Multi-View Clustering. *Knowledge-Based Systems*, 163:1009–1019, 2019.

[238] Q. Wang, J. Cheng, Q. Gao, G. Zhao, and L. Jiao. Deep Multi-View Subspace Clustering with Unified and Discriminative Learning. *IEEE Transactions on Multimedia*, pages 1–1, 2020.

[239] Q. Wang, Z. Ding, Z. Tao, Q. Gao, and Y. Fu. Partial multi-view clustering via consistent gan. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 1290–1295, 2018.

[240] X. Wang, X. Guo, Z. Lei, C. Zhang, and S. Z. Li. Exclusivity-Consistency Regularized Multi-View Subspace Clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, Los Alamitos, CA, USA, July 2017. IEEE Computer Society.

[241] Y. Wang, X. Lin, L. Wu, W. Zhang, Q. Zhang, and X. Huang. Robust Subspace Clustering for Multi-View Data by Exploiting Correlation Consensus. *IEEE Transactions on Image Processing*, 24(11):3939–3949, 2015.

[242] P. Wedin. Perturbation Bounds in Connection with Singular Value Decomposition. *BIT Numerical Mathematics*, 12(1):99–111, 1972.

[243] D. Wu, D. Wang, M. Q. Zhang, and J. Gu. Fast Dimension Reduction and Integrative Clustering of Multi-omics Data using Low-rank Approximation: Application to Cancer Molecular Classification. *BMC genomics*, 16(1):1022, 2015.

[244] J. Wu, Z. Lin, and H. Zha. Essential Tensor Learning for Multi-View Spectral Clustering. *IEEE Transactions on Image Processing*, 28(12):5910–5922, 2019.

[245] J. Wu, X. Xie, L. Nie, Z. Lin, and H. Zha. Unified Graph and Low-Rank Tensor Learning for Multi-View Clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 2020.

[246] R. Xia, Y. Pan, L. Du, and J. Yin. Robust Multi-View Spectral Clustering via Low-Rank and Sparse Decomposition. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 2149–2155, 2014.

[247] T. Xia, D. Tao, T. Mei, and Y. Zhang. Multiview Spectral Embedding. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(6):1438–1446, Dec 2010.

[248] D. Xie, Q. Gao, S. Deng, X. Yang, and X. Gao. Multiple Graphs Learning with a New Weighted Tensor Nuclear Norm. *Neural Networks*, 133:57–68, 2021.

[249] D. Xie, W. Xia, Q. Wang, Q. Gao, and S. Xiao. Multi-View Clustering by Joint Manifold Learning and Tensor Nuclear Norm. *Neurocomputing*, 380:105–114, 2020.

[250] M. Xie, Z. Ye, G. Pan, and X. Liu. Incomplete Multi-View Subspace Clustering with Adaptive Instance-sample Mapping and Deep Feature Fusion. *Applied Intelligence*, 01 2021.

[251] X. L. Xie and G. Beni. A Validity Measure for Fuzzy Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):841–847, 1991.

[252] Y. Xie, J. Liu, Y. Qu, D. Tao, W. Zhang, L. Dai, and L. Ma. Robust Kernelized Multiview Self-Representation for Subspace Clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):868–881, 2021.

[253] Y. Xie, D. Tao, W. Zhang, Y. Liu, L. Zhang, and Y. Qu. On Unifying Multi-View Self-Representations for Clustering by Tensor Multi-Rank Minimization. *International Journal of Computer Vision*, 126:1157–1179, 11 2018.

[254] Xie Xijiong and Shiliang Sun. Multi-view laplacian twin support vector machines. *Intelligent Data Analysis*, 19:701–712, 07 2015.

[255] C. Xu, H. Liu, Z. Guan, X. Wu, J. Tan, and B. Ling. Adversarial Incomplete Multiview Subspace Clustering Networks. *IEEE Transactions on Cybernetics*, pages 1–14, 2021.

[256] C. Xu, D. Tao, and C. Xu. Large-margin multi-viewinformation bottleneck. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1559–1572, August 2014.

[257] C. Xu, D. Tao, and C. Xu. Multi-view learning with incomplete views. *IEEE Transactions on Image Processing*, 24(12):5812–5825, 2015.

[258] H. Xu, X. Zhang, W. Xia, Q. Gao, and X. Gao. Low-rank Tensor Constrained Co-regularized Multi-view Spectral Clustering. *Neural Networks*, 132:245–252, 2020.

[259] J. Xu, X. Zhang, W. Li, X. Liu, and J. Han. Joint Multi-view 2D Convolutional Neural Networks for 3D Object Classification. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3202–3208, July 2020.

[260] Z. Xue, J. Du, D. Du, and S. Lyu. Deep Low-rank Subspace Ensemble for Multi-view Clustering. *Information Sciences*, 482:210–227, 2019.

[261] B. Yang, X. Zhang, F. Nie, F. Wang, W. Yu, and R. Wang. Fast Multi-View Clustering via Nonnegative and Orthogonal Factorization. *IEEE Transactions on Image Processing*, 30:2575–2586, 2021.

[262] D. Yang, Z. Ma, and A. Buja. A Sparse Singular Value Decomposition Method for High-Dimensional Data. *Journal of Computational and Graphical Statistics*, 23(4):923–942, 2014.

[263] Mo Yang and Shiliang Sun. Multi-view Uncorrelated Linear Discriminant Analysis with Applications to Handwritten Digit Recognition. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 4175–4181, 2014.

[264] Y. Yang and H. Wang. Multi-view Clustering: A Survey. *Big Data Mining and Analytics*, 01(02):83, 2018.

[265] Y. Yao, Y. Li, B. Jiang, and H. Chen. Multiple Kernel k-Means Clustering by Selecting Representative Kernels. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2020.

[266] Y. Ye, X. Liu, J. Yin, and E. Zhu. Co-regularized Kernel k-means for Multi-view Clustering. In *Proceedings of the 23rd International Conference on Pattern Recognition (ICPR)*, pages 1583–1588, 2016.

[267] C. Z. You, H. H. Fan, and Z. Q. Shu. Non-negative Sparse Laplacian regularized Latent Multi-view Subspace Clustering. In *Proceedings of the 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, pages 210–213, 2020.

[268] H. Yu, T. Zhang, and Y. Lian, Y.and Cai. Co-regularized Multi-view Subspace Clustering. In *Proceedings of the 10th Asian Conference on Machine Learning*, volume 95 of *Proceedings of Machine Learning Research*, pages 17–32. PMLR, 14–16 Nov 2018.

[269] Y. Yu, L. Zhang, and S. Zhang. Simultaneous Clustering of Multiview Biomedical Data using Manifold Optimization. *Bioinformatics*, 35(20):4029–4037, 03 2019.

[270] L. A. Zadeh. Fuzzy Sets. *Information and Control*, 8(3):338–353, 1965.

[271] H. Zha, X. He, C. Ding, H. Simon, and M. Gu. Spectral Relaxation for k-means Clustering. In *Proceedings of the Neural Information Processing Systems*, volume 14, pages 1057–1064, Vancouver, Canada, 2001.

[272] K. Zhan, X. Chang, J. Guan, L. Chen, Z. Ma, and Y. Yang. Adaptive Structure Discovery for Multimedia Analysis Using Multiple Features. *IEEE Transactions on Cybernetics*, 49(5):1826–1834, 2019.

[273] K. Zhan, C. Zhang, J. Guan, and J. Wang. Graph Learning for Multiview Clustering. *IEEE Transactions on Cybernetics*, 48(10):2887–2895, 2018.

[274] C. Zhang, E. Adeli, T. Zhou, X. Chen, and D. Shen. Multi-Layer Multi-View Classification for Alzheimer's Disease Diagnosis. 2018:4406–4413, February 2018.

[275] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, and D. Xu. Generalized Latent Multi-View Subspace Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(1):86–99, 2020.

[276] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao. Low-Rank Tensor Constrained Multiview Subspace Clustering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1582–1590, 2015.

[277] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao. Latent Multi-view Subspace Clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4333–4341, 2017.

[278] C. Zhang, J. Liu, Q. Shi, X. Yu, T. Zeng, and L. Chen. Integration of Multiple Heterogeneous Omics Data. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 564–569, December 2016.

[279] D. Zhang and S. Chen. Clustering Incomplete Data Using Kernel-Based Fuzzy C-means Algorithm. *Neural Processing Letters*, 18(3):155–162, 2003.

[280] J. Zhang, G. Zhu, R. W. Heath Jr., and K. Huang. Grassmannian Learning: Embedding Geometry Awareness in Shallow and Deep Learning. *Computing Research Repository (CoRR)*, abs/1808.02229, 2018.

[281] S. Zhang, C. C. Liu, W. Li, H. Shen, P. W. Laird, and X. J. Zhou. Discovery of Multidimensional Modules by Integrative Analysis of Cancer Genomic Data. *Nucleic Acids Research*, 40(19):9379–9391, October 2012.

[282] W. Zhang, Y. Liu, N. Sun, D. Wang, J. Boyd-Kirkup, X. Dou, and J. D. Han. Integrating Genomic, Epigenomic, and Transcriptomic Features Reveals Modular Signatures Underlying Poor Prognosis in Ovarian Cancer. *Cell Reports*, 4(3):542–553, 2013.

[283] X. Zhang, L. Zhao, L. Zong, X. Liu, and H. Yu. Multi-view Clustering via Multi-manifold Regularized Nonnegative Matrix Factorization. In *Proceedings of the IEEE International Conference on Data Mining*, pages 1103–1108, 2014.

[284] X. Zhang, L. Zong, X. Liu, and H. Yu. Constrained NMF-Based Multi-View Clustering on Unmapped Data. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, AAAI'15, page 3174âĂŞ3180. AAAI Press, 2015.

[285] Y. Zhang, W. Yang, B. Liu, G. Ke, Y. Pan, and J. Yin. Multi-view Spectral Clustering via Tensor-SVD Decomposition. In *Proceedings of the IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 493–497, 2017.

[286] Z. Zhang, Z. Zhai, and L. Li. Uniform Projection for Multi-View Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(8):1675–1689, August 2017.

[287] H. Zhao, Z. Ding, and Y. Fu. Multi-View Clustering via Deep Matrix Factorization. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, AAAI'17, page 2921âĂŞ2927. AAAI Press, 2017.

[288] J. Zhao, X. Xie, X. Xu, and S. Sun. Multi-View Learning Overview: Recent Progress and New Challenges. *Information Fusion*, 38(C):43–54, November 2017.

[289] L. Zhao, T. Yang, J. Zhang, Z. Chen, Y. Yang, and Z. J. Wang. Co-Learning Non-Negative Correlated and Uncorrelated Features for Multi-View Data. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–11, 2020.

[290] W. Zhao, S. Tan, Z. Guan, B. Zhang, M. Gong, Z. Cao, and Q. Wang. Learning to Map Social Network Users by Unified Manifold Alignment on Hypergraph. *IEEE Transactions on Neural Networks and Learning Systems*, 29(12):5834–5846, December 2018.

[291] X. Zhao, N. Evans, and J. Dugelay. A Subspace Co-training Framework for Multi-view Clustering. *Pattern Recognition Letters*, 41:73–82, 2014. Supervised and Unsupervised Classification Techniques and their Applications.

[292] Q. Zheng, J. Zhu, Z. Li, S. Pang, Jun Wang, and Lei Chen. Consistent and Complementary Graph Regularized Multi-view Subspace Clustering. *arXiv*, 2004.03106, 2020.

[293] Q. Zheng, J. Zhu, Z. Tian, Z. Li, S. Pang, and X. Jia. Constrained Bilinear Factorization Multi-view Subspace Clustering. *Knowledge-Based Systems*, 194:105514, 2020.

[294] D. Zhou and C. JC Burges. Spectral Clustering and Transductive Learning with Multiple Views. In *Proceedings of the 24th International Conference on Machine Learning*, pages 1159–1166. ACM, 2007.

[295] L. Zhou, G. Du, K. LÃij, and L. Wang. A Network-based Sparse and Multi-manifold Regularized Multiple Non-negative Matrix Factorization for Multi-view Clustering. *Expert Systems with Applications*, 174:114783, 2021.

[296] P. Zhou, Y. Shen, L. Du, and F. Ye. Incremental Multi-view Support Vector Machine. In *Proceedings of the 2019 SIAM International Conference on Data Mining (SDM)*, pages 1–9, 2019.

[297] F. Zhuang, G. Karypis, X. Ning, Q. He, and Z. Shi. Multi-view Learning via Probabilistic Latent Semantic Analysis. *Information Sciences*, 199:20–30, 2012.

[298] M. Zitnik and B. Zupan. Data Fusion by Matrix Factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1):41–53, 2015.

[299] L. Zong, X. Zhang, L. Zhao, H. Yu, and Q. Zhao. Multi-view Clustering via Multi-manifold Regularized Non-negative Matrix Factorization. *Neural Networks*, 88:74–89, 2017.

[300] I. Zwiener, B. Frisch, and H. Binder. Transforming RNA-Seq Data to Improve the Performance of Prognostic Gene Signatures. *PloS One*, 9(1):e85150, 2014.