Graph-based Learning using Multiple Views

Supratim Manna

Graph-based Learning using Multiple Views

Thesis submitted to
Indian Institute of Technology Kharagpur
for the award of the degree

of

Master of Science (by Research)

by

Supratim Manna
(17EE72P02)

under the guidance of

Dr. Anirban Mukherjee

Department of Electrical Engineering
Indian Institute of Technology Kharagpur
India, 721302
March 2021

Arise, awake, stop not till the goal is reached

- Swami Vivekananda

~ Dedicated to my family and friends ~

Declaration

I certify that

a. The work contained in the thesis is original and has been done by myself under the general supervision of my supervisors.

b. The work has not been submitted to any other Institute for any degree or diploma.

c. I have followed the guidelines provided by the Institute in writing the thesis.

d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

Date: / / Supratim Manna

Place: Kharagpur (17EE72P02)

CERTIFICATE OF APPROVAL

Date: / /

Certified that the thesis entitled **Graph-based Learning using Multiple Views**, submitted by **Supratim Manna** to the Indian Institute of Technology Kharagpur, for the award of the degree of **Master of Science (by Research)** has been accepted by the external examiner and that the student has successfully defended the thesis in the viva-voce examination held today.

Dr. Anirban Mukherjee
(Supervisor)

Dr. Pranab Kumar Dutta
(Member of the DAC)

Dr. Nirmalya Ghosh
(Member of the DAC)

Dr. Debdoot Sheet
(Member of the DAC)

(Chairman)

(External Examiner)

Certificate

This is to certify that the thesis entitled **"Graph-based Learning using Multiple Views"**, submitted by **Supratim Manna**, to the Indian Institute of Technology Kharagpur, India, is a record of the bonafide research work carried out by him under my supervision and I consider it worthy of consideration for the award of the degree of Master of Science (By Research) of the institute.

Dr. Anirban Mukherjee
Department of Electrical Engineering
Indian Institute of Technology Kharagpur
Kharagpur, WB, India - 721302

I.I.T. Kharagpur
March, 2021

Acknowledgment

I would like to grab this golden opportunity to convey my heartfelt indebtedness to many individuals who have given me a lot of supports and strength during my tenure here in IIT Kharagpur. First and foremost, I express my deepest sense of gratitude to my supervisor Dr. Anirban Mukherjee whose expert guidance made my research work prolific here in IIT Kharagpur.

Apart from my supervisor, I have grasped many worthwhile ideas and technical aspects from Ms. Jessy Rimaya Khonglah whose altruistic help and encouragement have made my workplace environment as motivating and exciting as possible.

I would also like to thank Ministry of Human Resource Development, Government of India and Indian Council of Medical Research for sponsoring my research project.

This wonderful and memorable journey would not have been possible without the love and blessing of my parents. Their continuous support , and their faith in me has provided me the strength to reach the verge of completion of my degree.

I would also like to enunciate my wholehearted thanks to *Sudipto Trivedy, Archisman Datta, Rumia Masbura, Rasmita Nayak, Aritra Basu, Subhrasankha Ghosh, Debdeep Samajdar, Moumita Banik, Arnab Acharya, Gargi Das and Srijeeta Ghosh* for their continuous support and encouragement which have filled my journey with happiness and made it memorable.

Supratim Manna

Abstract

The representation of the data by means of a graph provides various advantages over feature vectors, and the similarity matrix (adjacency matrix) of the graph represents the relationship shared among the data samples. Therefore, various graph-based learning algorithms have been proposed in the literature where the optimal similarity matrix is determined to best represent the data samples, thus improving the learning performances.

In the graph-based learning methods, one of the principal aims is to find the optimal similarity matrix. This thesis presents a few methods to find the optimal similarity matrix from the graph representation of the data sets. A graph-based learning method named Self-weighted Multi-view Multiple Kernel Learning (SMVMKL) using multiple kernels on multiple views has been proposed to learn the optimal similarity matrix of the data sets. Owing to its limitation before the outliers present in the data set, an improved variant named Robust Self-weighted Multi-view Multiple Kernel Learning (RSMVMKL) has been proposed. To reduce the effect of noise, another method has been proposed which uses only the prominent features by solving a low-rank minimization problem. This proposed method is named as Low-rank Multi-view Multi-kernel Graph-based Clustering (LRMVMKC).

For all the proposed methods, multiple views and multiple kernels have been used to improve the learning performances. The efficacy of all the proposed algorithms has been supported by demonstrating their performance on the real-world benchmark data sets.

Keywords: Similarity matrix, multiple kernels, kernel matrix, multiple views.

Contents

Chapter 1

Introduction

A graph is a flow structure that represents the relationship among different objects. A graph consists of two basic components: nodes and edges. Where each node represents an object and an edge among any two nodes represents the relationship between these two objects. When a dataset is represented in terms of nodes and edges then it is called the graph representation of the dataset. One of the most important terminologies that is used to describe the graph representation of a dataset is *similarity matrix*.

Due to the advancement of sensing and storage technologies, large volume and large dimensional data are available nowadays and most of these data are unstructured thus making it difficult to learn the data. But the graph representation of the data makes them structured which is easy to analyze and learn. It is also known from the similarity matrix of the graph that how one sample in a dataset is related to other samples and how strong the relationship is.

Whenever a graph representation of a dataset is available, some valuable information can be extracted (learned) from the structure of the graph. This is known as graph-based learning. There are two types of graph-based learning:

- **Graph-based Clustering**: It is an unsupervised learning. Similar nodes of

the graph are grouped in a same set known as cluster.

- **Graph-based Semi-supervised Classification**: It is a semi supervised learning. Given some labelled samples, it classifies the nodes of the graph into different classes.

While studying graph-based learning, the concept of "view" is of paramount importance. To learn from a given dataset, a set of features is extracted from the dataset. This extracted set of features is known as view. This graph-based learning can be performed based on only a single view or multiple views. So, the graph-based learning can be sub-divided into two parts:

- **Single view learning**: Only a single set of features is extracted from a given dataset and the learning task is performed based on this set.
- **Multi-view learning**: Multiple views are extracted from the given dataset and the learning task is performed based on all the multiple set of features. It is worth noting that a single view may contain multiple features.

1.1 Background

1.1.1 Multiple Views of Data

A single set, consisting of multiple features, of a given dataset is known as view of the dataset. So, multiple distinct sets, each of having multiple features, are considered as multiple views of the dataset. In content based web-image retrieval, an image can be described from the visual feature from the image itself and also from the text that surrounds the image. Here the visual feature sets, extracting from the image, may be a single view and the features, derived from the surrounding text, may compose another view. Another example, may be a video clip, which is a combination of audio and visual frames. So, a video clip can be classified by its visual frames and also by its audio signal. Here the features, derived from visual frame, compose a view and the

audio signal-based features may contribute to form another, may be, complementary view. Various examples of multiple views of different dataset are shown in Fig. 1.1.

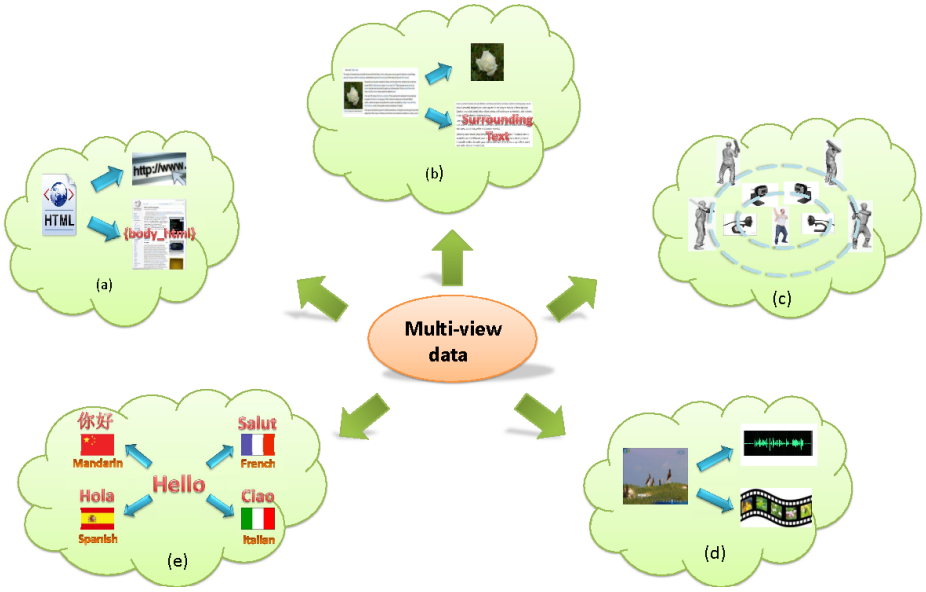


Figure 1.1: Multi-view data: a) a web document can be represented by its URL and words on the page, b) a web image can be depicted by an image alongwith the surrounding text, c) images of a 3D object taken from different viewpoints, d) video clips are combinations of audio signals and visual frames, e) multilingual documents have one view in each language.

1.1.2 Kernel Method

In order to linearly separate the non-linearly separable dataset, kernel methods are used. kernel methods are a class of algorithms for pattern analysis that use various kernel functions, which enable them to operate in a high-dimensional, implicit feature space. This bypasses the computation of the coordinates of the data, but rather by simply computing the inner products between the all pairs of data in the sample space. It is called the kernel trick.

Suppose there is a given dataset $X = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{\mathrm{d}\times\mathrm{n}}$ and there exists a mapping $\phi : X \mapsto \mathcal{V}$ which maps the data sample from input space, $\mathbb{R}^d$, to a reproducing

kernel Hilbert space, $\mathcal{V}\in\mathbb{R}^{n}$. Then the kernel trick is:

$$< \phi(x_i), \phi(x_j) > = \phi(\mathrm{x_i})^{\mathrm{T}}\phi(\mathrm{x_j}) = \mathrm{K}(\mathrm{x_i}, \mathrm{x_j})$$

where, $K=\mathrm{K}^{\mathrm{T}}\in\mathbb{R}_{+}^{n\times n}$ is the positive semi-definite kernel matrix. In Fig. 1.2, it is shown how the kernel trick makes a non-linearly separable data in input space to a linearly separable data in high dimensional feature space.

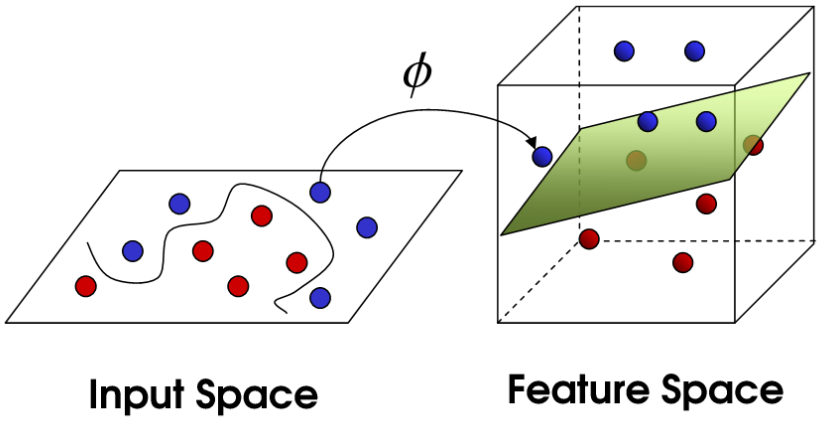


Figure 1.2: Kernel Trick.

But in kernel method, the choice of the kernel function is important. To overcome this, kernel learning algorithms are proposed where multiple kernel functions are used instead of a single kernel function. Then the optimal kernel is learned by an optimization algorithm.

1.1.3 Spectral Clustering

Given a set of data points $x_1, x_2, \cdots, x_n$ and some notion of similarity s_{ij} between all pair of data points x_i and x_j, the intuitive goal of the clustering is to partition the data points into several groups such that the points in the same group are similar and points in different group are dissimilar to each other. So, a nice way to represent

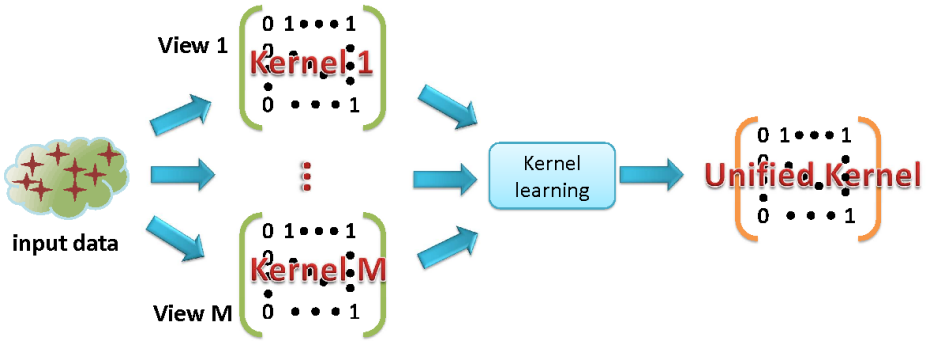


Figure 1.3: Multiple Kernel Learning

the data is in form of the similarity graph $G = (V, E)$, where $V = v_1, \cdots, v_n$ is the of the vertex and E is the set of edges of the graph, G. Each vertex v_i in this graph represents a data point x_i. Two vertices are connected with a edge whose weighted is denoted as s_{ij} where s_{ij} is symmetric and non-negative. Spectral clustering tries to divide the data samples into C clusters by finding n indicators $p_1, p_2, \cdots, p_n$ which satisfies:

$$\underset{f_1,f_2,\cdots,f_n}{minimize} \sum_{i,j} s_{ij} \|p_i - p_j\|^2$$

Let, S be the $n \times n$ matrix constituted of the similarities s_{ij}, D be the diagonal matrix with its i^{th} diagonal element being the sum of i^{th} row of S, i.e $D_{ii} = W_{i1} + W_{i2} + \cdots + W_{in}$. Matrix D is called as degree matrix. Spectral clustering solves the above equation by finding the smallest eigenvalues and their corresponding eigenvectors of the Laplacian matrix, $L = D - W$. Since the smallest eigenvalue λ_1 of L is always 0 which corresponds to the trivial solution of the constant-one eigenvector $\mathbf{1}$, the solution of spectral clustering is constructed by the eigenvectors corresponding to the next C smallest eigenvalues, $\lambda_2, \lambda_3, \cdots, \lambda_{C+1}$. After stacking these C eigenvectors into a matrix P, we get $P \in \mathbb{R}^{n \times C}$. The i^{th} row of P corresponds to the indicator p_i for x_i and the matrix P is called a indicator matrix.

1.1.4 Kernelized Graph-based Learning

Let, $X \in \mathbb{R}^{d \times n}$ be the data matrix with n data points each having d dimensional feature. Now, according to the self-expressive property [1], each data point can be represented as the linear combination of the other data points.

$$x_i = \sum_j x_j s_{ij}$$

where, s_{ij} denotes the similarity between the i^{th} and j^{th} data point and $s_{ij} \geq 0$. Now, $S \in \mathbb{R}^{n \times n}$ can be treated as a similarity matrix where the ij^{th} element of S is s_{ij} and it represents the global structure of the data. The similarity matrix S can be obtained by solving the following minimization problem:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \|X - XS\|_F^2 + \lambda \|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0. \end{aligned} \tag{1.1}$$

where, λ is a tuning parameter. It can be observed from Eq. 1.1 that it assumes linear relationship among the data points. Since, the real world data sets shows non-linear relationship between the data points, to recover the non-linear relationship, Eq. 1.1 can be extended to kernel space as following:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \|\phi(x) - \phi(x)S\|_F^2 + \lambda \|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0. \end{aligned} \tag{1.2}$$

where, ϕ is the kernel mapping function. Now using the kernel trick, Eq. 1.2, the can be rewritten as:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & Tr(K - 2KS + S^T K S) + \lambda \|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0. \end{aligned} \tag{1.3}$$

1.2 Related Work

To appreciate the proposed work, it is necessary to present the existing graph-based learning method briefly. The related existing work is categorized into two parts:

- Graph-based clustering and
- Graph-based semi-supervised classification.

1.2.1 Graph-based Clustering

Spectral clustering (SC) method [2] is considered as the baseline in the field of graph-based clustering. In this method, a given set of data points $x_1, x_2, \cdots, x_n$ can be expressed in the form of *similarity graph* $G = (V, E)$ where, $V = (\mathrm{v}_1, \mathrm{v}_2, \cdots, \mathrm{v}_\mathrm{n})$ is a set of vertices and each vertex denotes a data point. Different methods [3], such as: the ε-neighborhood graph, k-nearest neighbor graph, the fully connected graph, are used to create the similarity graph in spectral clustering. The Laplacian matrix L is formed from the similarity graph and then the clustering task is performed by using K-means algorithm on some of the largest eigenvectors of L. But the traditional SC method uses only a single affinity (similarity) matrix. But in many applications, multiple useful features are available therefore multiple affinity matrices. To obtain an optimal affinity matrix by using those multiple affinity matrices, affinity aggregation for spectral clustering (AASC) has been proposed in [4]. In forming a Laplacian matrix, the choice of the number of neighbours, the dealing with data outliers and noise are important. To solve those issues, simplex sparse representation (SSR) method [5] has been proposed which automatically chooses the number of neighbours and the sparse representation of the Laplacian matrix leads to the reduced computational cost and robustness to the data outliers. Most of the graph-based clustering methods create a graph from the data. If the graph is of low quality then the clustering performance gets affected. To solve this issue, a constrained Laplacian rank graph-based clustering algorithm based on l_1 and l_2 norm has been proposed in [6]. Real

data are often of compound quality and it affects the graph learning resulting in poor clustering performances. To make the graph-based clustering less sensitive to the noise, a robust graph learning has been demonstrated in [7] where the raw data are decomposed into two low-rank matrices, one is called "clean data" and other is called "noise/errors" and then the similarity matrix is learned from the "clean data".

In real-world datasets, samples are generally non-linearly separable. The methods, described earlier, don't consider that nonlinearity present in the dataset. In order to incorporate this nonlinearity into the linear graph-based framework, various kernelized graph-based framework have been proposed. In kernel k-means method (KKM) [8], representation of data points into a higher dimensional feature space is performed by using a nonlinear function called kernel function and then the data points are separated linearly in the high dimensional feature space. In [9], a weighted Kernel Principal Component Analysis approach has been stated for classical graph-based image segmentation problem where the clustering model can be trained and validated on sub-sampled parts of the image to be segmented thus reducing the computational time for image segmentation. But these kernelized methods rely only on a single kernel thus the learning performance depends on the choice of the kernel. To solve this issue, various multiple kernel learning methods have been proposed in [10], [11], [12]. In [13], multiple kernel k-means (MKKM) method has been described. MKKM method is same as KKM method but the issue of kernel choice in KKM is taken care of by using multiple kernel instead of a single kernel. But both the KKM and MKKM methods are sensitive to noise and data outlier. To take care of this problem, a robust multiple kernel k-means (RKKM) using $l_{2,1}$ norm has been proposed in [14]. A robust kernelized group sparse graph construction has been presented in [15] where an informative graph is created by using auto-grouped sparse regularization based on the l^1-graph [16]. A low-rank kernel learning method for graph-based clustering has been described in [17] where multiple kernels are used and low-rank optimization of the kernel matrices make the framework less sensitive to outliers. This low-rank kernel matrices exploit the similarity nature of the kernel matrices and seek an optimal

kernel matrix from the neighbourhood of candidate kernels. While using multiple kernels, an important goal is to assign proper weight to each of the kernels. A self-weighted multiple kernel learning has been stated in [18] where an optimal kernel is learned from the multiple candidate kernels and a proper weight is assigned to each of the kernels automatically. But all these methods use a single view for the clustering task.

The clustering performances can be improved by using multiple views instead of single view and the multi-view graph-based clustering methods have been proposed for better clustering performances. A robust multi-view k-means clustering has been described in [19] where multiple views of the dataset are used and $l_{2,1}$ norm is used to make the framework robust. A weight assignment parameter is needed for proper weight assignment to each view. A co-training approach for multi-view spectral clustering has been proposed in [20] where a graph is formed for each view and then spectral clustering is performed on each graph. Now the clustering of data points is done by using discriminative eigenvectors of one view and this clustering is used to improve the graph structure of the other views and vice-versa. When multiple views of a dataset are available, then one approach to do clustering task is to look for the cluster indicator matrix which is consistent across all the views, i.e., a certain data samples should have the same membership in all the views. This is achieved by the co-regularization [21] of the clustering hypothesis. Auto-weighted multiple graph learning has been proposed in [22] where a graph is created for each view and each of them has consistent cluster indicator matrix for all the views but each individual graph has partial information to learn the real manifold. Also proper weight is assigned to every graph. In [23], two multi-view clustering with multiple graph methods are stated. One is parameter weighted multi-view clustering with multiple graph where a proper weight is assigned to each view by introducing a hyper-parameter. Another one is self-weighted multi-view clustering with multiple graph where a proper weight is assigned to each view automatically according to their contribution to the clustering task. A multi-view clustering with soft capped norm algorithm has been

proposed in [24] where the use of soft capped norm reduces the different label noises present in the data and make the algorithm less sensitive to the noises.

1.2.2 Graph-based Semi-supervised Classification

It is often challenging to obtain labelled data whereas unlabelled data is easily available. That's why semi-supervised learning is important and useful in machine learning and many research works have been conducted on this and one of them is graph-based semi supervised learning method. Graph-based methods consider the labelled and unlabelled samples as the vertices of the graph and utilizes the weight of the edges to pass information from labelled samples to unlabelled samples.

One of the most important assumptions for semi supervised learning is the assumption of consistency. It says: (i) neighbourhood points are likely to have same labels and (ii) points on the same structure are likely to have same label. In [25], a method called learning with local and global consistency has been proposed using the assumption of consistency where a classifying function has been designed which is sufficiently smooth with respect to the intrinsic structure revealed by the labelled and unlabelled samples. Most of the semi supervised learning methods are having two stages: (i) creating an affinity or similarity matrix form the dataset and (ii) propagates information to the unlabelled data from labelled data through the affinity matrix. But in [26] a unified framework for semi supervised learning has been proposed where both the affinity matrix and unknown labels are learned simultaneously. Another semi supervised learning framework with adaptive neighbours has been proposed in [27] where the label of the unlabelled data and local structure of the graph is learned simultaneously. The graph-based semi-supervised learning methods mainly consider the single label problem. But in real life the problems are associated with multiple labels. To address this multiple label issue, a multi labels semi supervised learning framework has been described in [28]. In most of the cases, when a new unlabelled data point is presented then for labeling that data point the graph is again rebuilt and

the classification algorithm is run from the starting which is computationally heavy. So when unlabelled dataset is large then scalability becomes a concerning issue. To solve this scalability issue, a graph-based harmonic mixture model is presented in [29].

Later to consider the nonlinear relation among different data samples, kernel methods have been incorporated in linear graph-based semi supervised framework. Kernelized semi supervised methods have improved the learning performances drastically. A kernelized graph-based semi supervised framework has been stated in [30] where spectral kernel is designed for the given dataset. Another kernelized method called hyperparameter and kernel learning for graph-based semi supervised classification has been presented in [31] where the hyperparameter that defines the structure of the similarity graph is learned as well as the kernel matrix while adhering to a Bayesian framework. In [32], a kernel based semi supervised learning has been proposed where the vector-based and graph-based approach have been combined together. In [18] a self-weighted multiple kernel semi supervised classification framework has been presented where the kernel choice is resolved by using multiple kernels and also proper weight is assigned to each kernel according to their importance to the learning task.

All these graph-based semi-supervised learning methods use only a single view of the dataset. Later to improve the semi supervised learning performances, various methods and frameworks using multiple views have been proposed. A graph-based multi modality method has been stated in [33] where each kind of feature is considered as a modality and it is represented by one independent graph. An iterative fusion approach has been presented in [34] for graph-based semi supervised learning. In this framework, each feature is considered as a view and label propagation is performed by using multiple views. Also the proper weights are assigned to the views dynamically to reduce the adverse effects of irrelevant views on the learning task. A co-regularization framework has been presented in [35] for semi supervised classification task. In this method, the classifier is learned in each view through forms

of multi-view regularization. Multiple graph label propagation method has been described in [36]. This method improves the semi supervised learning performances by eliminating noisy graph while integrating multiple graphs. There exists a sparse weight co-efficient which helps to find the more important graph thus improving the performance.

1.3 Objectives of the Thesis

Based on the aforementioned discussion, it is observed that graph-based learning using multiple kernels on multiple views has received considerably less recognition. The objective of this thesis is to propose a graph-based learning method that uses multiple kernels as well as multiple views so that the unsupervised graph-based learning as well as semi-supervised graph-based learning can achieve improved performances than other existing graph-based methods and to make the proposed method robust to noise.

1.4 Contributions of the Thesis

The main contributions of this thesis are listed below:

- proposing a novel method to construct an optimal kernel and an optimal similarity matrix of the graph using multiple kernel on multiple views.
- proposing a robust framework for graph-based learning by using $l_{2,1}$ norm as well as by doing the low rank kernel optimization.
- proposing a novel kernelize graph-based learning method on high dimensional data to improve the learning performances by getting rid of redundant features.
- stating an algorithm to assign appropriate weight to each kernel of each view automatically without introducing any extra weight assignment parameter.
- integrating the graph construction, kernel and label learning, and dimension

reduction.

1.5 Dataset

Different graph-based learning algorithms are described in this thesis. To show the performances of these algorithms, these algorithms are tested on various real-world benchmark datasets. The details of the datasets are as follows :

Animal with Attributes

This dataset [37] consists of 50 classes and 6 features. Out of 50 classes, 10 classes (Antelope, Bat, Buffalo, Dolphin, Giraffe, Horse, Lion, Mouse, Seal, and Squirrel) are considered for the experiment with each of the 10 classes having 10 samples each. Three published features are considered: Color Histogram (CQ), Local Self-Similarity (LSS) [38] and Pyramid Histogram of Orientation Grdients (PHOG) [39].

MSRC-v1

This dataset [40] consists of 240 images divided into 8 classes. 7 classes [41] composing of Airplane, Bicycle, Building, Car, Tree, Cow, and Face each consisting of 30 images are selected for this experiment. From each image, three visual features are extracted: Color Moment (CM) with dimension 27, GIST with dimension 512 [42] and Local Binary Pattern (LBP) feature with dimension 59.

Jaffe

This dataset [43] consists of 213 images of various facial expressions of 10 different Japanese models. For the experiments, 200 images consists of 20 facial images for each model have been considered. Three different features: 512 dimension GIST, 420 dimension CLR and 512 dimension Scale-invariant Feature Transform (SIFT), have been extracted from each image.

Yale

This dataset [44] consists of 165 grayscale images of various facial expressions of 15 different individuals. Three different features: 512 dimension GIST, 22356 dimension HOG and 59 dimension LBP feature, have been extracted from each image.

Reuters Multilingual Data

This textual dataset [45] contains features of documents originally written in five languages (English, French, German, Spanish, Italian) and their translations over 6 categories. 180 documents are randomly sampled in a balanced manner with each of the 6 classes consisting of 30 documents each. Documents originally in English have been used as a first view and their French, German, Spanish and Italian translations have been used as second, third, fourth and fifth views.

Caltech101

It [46] contains 8677 images with 101 different categories. Five classes (Garfield, Motorbike, Snoopy, Stop-Sign and Windsor-Chair) are considered. From each image, six features: Gabor features with dimension 48, Wavelet Moments with dimension 40, CENTRIST features with dimension 254, HOG features with dimension 1984, GIST features with dimension 512, and LBP features with dimension 928, are extracted.

1.6 Evaluation Metric

To measure the performance of the learning tasks of all the stated algorithms in this thesis, three important evaluation metrics have been adopted : Accuracy (ACC), Normalized Mutual Information (NMI) and Purity (PUR).

$$ACC(y, s) = \frac{\sum_i^n \delta(s_i, map(s_i))}{n} \tag{1.4}$$

where y is the ground truth label with c classes and s is the resulting clustering label with $\hat{c}$ classes of any data sample x_i and $\delta(x, y) = 1$ if $x = y$, $\delta(x, y) = 0$ otherwise. $map(s_i)$ is the best mapping function. Kuhn-Munkres algorithm is used to permute the clustering labels to match the ground truth labels. A larger value of ACC denotes a better clustering performance.

The NMI is the shared information between a pair of clusters [47], defined as:

$$NMI(y, s) = \frac{I(y; s)}{H(y) + H(s)} \tag{1.5}$$

where, $I(y; s)$ is the mutual information between and y and s, $H(y)$ and $H(s)$ are the entropy of y and s.

PUR is another popular evaluation metric that is used to measure the clustering performance. If $y = (y_1, y_2, y_3, \cdots, y_n)$ is the ground truth label, and $s = (s_1, s_2, s_3, \cdots, s_n)$ is the clustering result label, then purity is computed by assigning each cluster to the class which is the most frequent in the cluster, followed by counting the number of correctly assigned objects and finally dividing by n.

$$PUR(y, s) = \frac{1}{n} \sum_{k} \max_{j} |s_k \cap y_j| \tag{1.6}$$

Like ACC and NMI, a larger value of PUR denotes a better clustering performance.

1.7 Organization of the Thesis

This thesis is organized as follows.

A robust graph-based learning framework is discussed in Chapter 2. Instead of a single view, multiple views are considered for the learning task. Kernel method is incorporated in the framework. The advantages of using multiple kernels are discussed in this chapter. The overall learning performances of the proposed framework are verified by extensive experiments on different real-world datasets.

A robust graph-based clustering method is discussed in Chapter 3. Here a low-rank kernel optimization is performed by using nuclear norm instead of using $l_{2,1}$ norm as discussed in Chapter 2. The proposed method is less sensitive to the noise present in the dataset. Here also for better performance of the clustering task multiple kernel and multiple views are incorporated to the framework. The performance of the clustering task are verified by several experiments on different real-world benchmark datasets.

A kernelized graph-based learning method for high dimensional data is described in Chapter 4. Here Kernel Principal Component Analysis (KPCA) is used to incorporate nonlinear relationship between different data samples into the framework as well as to reduce the dimension of the dataset thus getting rid of redundant features present in the high dimensional dataset. Various experiments on different real-world datasets validates the excellent learning performances of the proposed method.

Chapter 2

Robust Graph-based Learning using Multiple Kernel on Multiple Views

2.1 Introduction

A number of existing graph-based learning models are linear [2], [48], [49] and they don't consider the nonlinear relationship that may be present in a dataset. But most real-world applications have data that are randomly distributed and therefore, not linearly separable. So, the use of linear graph-based learning framework on such data may degrade the learning performances. Therefore, to consider the existing nonlinear relationship between different data samples and to introduce nonlinearity into linear models, different kernel methods have been widely applied in many machine learning tasks [12], [15], [50]. But one of the disadvantages of kernel method is that it requires a predefined kernel to be selected and tuned. If the choice of kernel is poor then it may affect the learning performances. So, it is a challenging task to choose the most suitable kernel for a specific task. To address this, several multiple Kernel Learning

(MKL) algorithm is proposed [51], [52], [53]. Later those multiple kernel learning methods are used in different graph-based learning methods [13], [18].

Different kernelized graph-based learning methods perform the learning task only on a single view of the data. But methods, combining different views and containing different partial information, have been shown to improve the overall learning performances [23], [54], [55]. In many real life applications and scientific data analytic problems such as multi-camera surveillance system, social computing, web page classification, etc., data are collected from different views, where each view contains different information but belong to the same class. For example, in web page classification, a web page can be described by the document text and at the same time by the anchor text attached to hyperlinks pointing to a particular page where the document text is considered as the first view and the attached hyperlink is considered as the second view.

To use different views of a dataset with different partial information while incorporating the kernel method in the learning task, a novel self-weighted multi-view multiple kernel graph-based learning (SMVMKL) has been proposed. A key step in SMVMKL is to use multiple views while incorporating nonlinearity into the model by defining multiple kernels for each view and automatically assigning an optimal weight to each kernel of each view without the need of introducing an additional weight assignment parameter [11], [19]. But in real life, noises may present in the dataset and the performance of the proposed SMVMKL method may get effected due to the presence of the data outliers and noises. To nullify this effect, another novel framework named robust self-weighted multi-view multiple kernel graph-based learning (RSMVMKL) has been proposed by using the $l_{2,1}$-norm [7], [14].

The experiments for both the proposed SMVMKL and RSMVMKL are implemented on several benchmark datasets and the results validate that the better performances of both the proposed methods.

2.2 Methodology

Different features of a dataset are introduced as different views and multiple kernels are assigned to each of the views. Using multi-views and multiple kernels, the optimal similarity matrix is learned which can be directly partitioned into several clusters equal to the number of data class. The issue of weight assignment to each kernel of each view is also addressed.

2.2.1 Kernelized Graph-based Learning

There is a given the dataset $X \in \mathbb{R}^{d \times n}$ with only one view where $X = [\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \cdots, \mathbf{x_n}]$, $\mathbf{x_i} \in \mathbb{R}^{d \times 1}$, n is the total number of data points and d is the feature dimension. The given dataset X has m different number of clusters or classes. A data point $\mathbf{x_i}$ of the dataset can be connected by all the data points by edges with weight s_{ij} and from these weighted edges we can form a matrix $S \in \mathbb{R}^{n \times n}$ called similarity matrix. In spectral analysis there is another matrix, $L = \left(D - \frac{S+S^T}{2}\right) \in \mathbb{R}^{n \times n}$ called the Laplacian matrix where $D \in \mathbb{R}^{n \times n}$ is called degree matrix. D is a diagonal matrix and its i^{th} diagonal element is $\frac{\sum_j (s_{ij}+s_{ji})}{2}$. Now if the initial similarity matrix of the dataset X is S, then the optimal similarity matrix can be learned by minimizing the following problem:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \|X - XS\|_F^2 + \lambda \|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0. \end{aligned} \tag{2.1}$$

where $\lambda > 0$ is a trade-off parameter. This equation is in sample space. To incorporate the kernel method in the framework, Eq. (2.1) is converted into kernel space by using a kernel mapping function ϕ. Now Eq. (2.1) can be written as:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \|\phi(x) - \phi(x)S\|_F^2 + \lambda \|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0. \end{aligned} \tag{2.2}$$

It is known that the kernel trick is $K(x,y) = \phi(x)^T\phi(y)$. $K \in \mathbb{R}^{n\times n}$ is a positive semidefinite kernel matrix or Gram matrix. Now using this kernel trick (See Appendix A.1), Eq. (2.2) can be written as:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & Tr(K - 2KS + S^TKS) + \lambda\|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0. \end{aligned} \tag{2.3}$$

If there are m number of clusters or classes in the dataset, a graph with exactly m connected components is obtained by solving the minimization problem mentioned in Eq. (2.3). This model is known as kernel based graph learning (KGL). But the performance of this model largely depends on the choice of the kernel K.

2.2.2 Self-weighted Multi-view Multiple Kernel Learning

There is a given dataset X with q number of views denoted by $(X_1, X_2, \cdots, X_v, \cdots, X_q)$ where, $X_v \in \mathbb{R}^{d^v\times n}$, n is the total number of data points and d^v is the feature dimension of the v^{th} view X_v. For each view of the dataset, u number of kernels are used. The use of multiple kernels instead of a single kernel solves the issue of choice of the kernel faced in KGL method. Also for better performances of the proposed framework, multiple views of the dataset with different partial information but same cluster indicator matrix are used. This proposed learning framework that uses multiple views and multiple kernels for each view is known as self-weighted multi-view multiple kernel graph-based learning (SMVMKL). One of the issues to use multiple kernels for each view is to assign proper weight to each kernel of each view. To solve this weight assignment issue and to learn the optimal kernel from those multiple kernels, the proposed SMVMKL framework is formulated depending on the the assumptions that the kernel closer to the optimal kernel are assigned a larger value, and the contribution of a view influences the weight assigned to the kernel of that view. Using these assumptions, the optimal kernel learning is formulated (see Appendix A.2) as:

$$\underset{K}{\text{minimize}} \quad \sum_{v=1}^{q}\sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K\|_F^2 \tag{2.4}$$

where,

$$Z_{(p,v)} = \frac{1}{2\| \ H^{(p,v)} - K\|_F} \tag{2.5}$$

Here $H^{(p,v)}$ is the p^{th} kernel of the v^{th} view of the given dataset X and $Z_{(p,v)}$ is the weight assigned to $H^{(p,v)}$. From Eqs. (2.4) and (2.5), it is observed that the weight $Z_{(p,v)}$ depends on the target variable K which is unavailable directly. Therefore, at every iteration, K is calculated first, and then $Z_{(p,v)}$ is updated by Eq. (2.5). And instead of making the optimal kernel to be a linear combination of candidate kernels, Eq. (2.4) allows the most suitable kernel to be in some kernel's neighborhood [56]. It is also noted that no additional parameter has been introduced for the weight assignment. The optimal weight $Z_{(p,v)}$ for p^{th} kernel of v^{th} view is calculated automatically according to the kernel matrices. Now combining Eqs. (2.3), (2.4), and (2.5), the SMVMKL framework can be formulated as:

$$\begin{aligned} \underset{S,K}{\text{minimize}} \quad & Tr\left(K - 2KS + S^T KS\right) + \lambda\|S\|_F^2 \\ & + \beta \sum_{v=1}^{q}\sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K\|_F^2 \\ \text{subject to} \quad & S \geqslant 0 \end{aligned} \tag{2.6}$$

The similarity graph matrix S follows the following important property [57].

Theorem 2.1 *The multiplicity m of the eigenvalue 0 of the Laplacian matrix L is equal to the number of connected components in the graph associated with S.*

This property imposes an extra constraint on S and that constraint is $rank(\mathrm{L}) =$

$n-m$. After incorporating this constraint, Eq. (2.6) can now be written as:

$$\begin{aligned} \underset{S,K}{\text{minimize}} \quad & Tr\left(K - 2KS + S^T KS\right) + \lambda \|S\|_F^2 \\ & + \beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K\|_F^2 \\ \text{subject to} \quad & S \geqslant 0, \quad rank(L) = n - m \end{aligned} \tag{2.7}$$

The constraint in Eq. (2.7) will be ensured if $\sum_{i=1}^{m} \sigma_i(L) = 0$ where, σ_i is the i^{th} eigenvalue of L. Now defining a hyper-parameter α, Eq. (2.7) can be rewritten as:

$$\begin{aligned} \underset{S,K}{\text{minimize}} \quad & Tr\left(K - 2KS + S^T KS\right) + \lambda \|S\|_F^2 \\ & + \alpha \sum_{i=1}^{m} \sigma_i(L) \\ & + \beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K\|_F^2 \\ \text{subject to} \quad & S \geqslant 0 \end{aligned} \tag{2.8}$$

According to the Ky Fan's Theorem [58]:

$$\underset{P^T P = I}{\text{minimize}} \quad Tr\left(P^T LP\right) = \sum_{i=1}^{m} \sigma_i(L) \tag{2.9}$$

where $P \in \mathbb{R}^{n \times m}$ is the label or cluster indicator matrix. Using Eq. (2.9), Eq. (2.8) can be rewritten as:

$$\begin{aligned} \underset{S,P,K}{\text{minimize}} \quad & Tr\left(K - 2KS + S^T KS\right) + \lambda \|S\|_F^2 \\ & + \alpha Tr\left(P^T LP\right) \\ & + \beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K\|_F^2 \\ \text{subject to} \quad & S \geqslant 0, \ \ P^T P = I \end{aligned} \tag{2.10}$$

This model formulates the SMVMKL framework. This framework negotiates between the process of label learning and the process of optimal kernel learning and the performance of this framework can be improved repeatedly by iterative updating.

2.2.3 Robust Kernelized Graph-based Learning

There is a given the dataset $X \in \mathbb{R}^{d \times n}$ where $X = [x_1, x_2, x_3, \cdots, x_i, \cdots, x_n]$, $\mathbf{x_i} \in \mathbb{R}^{d \times 1}$, n is the total number of data points and d is the feature dimension of each data point. If the initial similarity matrix of the given dataset X is S, then the optimal similarity matrix can be learned by minimizing the problem given in Eq. (2.1). But the optimal similarity matrix may get affected by the presence of outliers in the dataset. To solve this issue a robust framework is needed and the robustness is obtained by using the $l_{2,1}$-norm instead of the Frobenius norm used in Eq. (2.1). Now Eq. (2.1) becomes:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \|X - XS\|_{2,1} + \lambda \|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0. \end{aligned} \tag{2.11}$$

where $\lambda > 0$ is a trade-off parameter. Using the kernel mapping function ϕ, Eq. (2.11) can be written as:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \|\phi(X) - \phi(X)S\|_{2,1} + \lambda \|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0. \end{aligned} \tag{2.12}$$

Using the kernel trick, Eq. (2.12) can be rewritten (See Appendix A.3) in the kernel space as:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \sum_{i=1}^{n} \sqrt{k_{ii} - 2k^i s_i + s_i^T K s_i} + \lambda \|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0. \end{aligned} \tag{2.13}$$

where, $s_i \in \mathbb{R}^{n \times 1}$ is the i^{th} column of S, $k^i \in \mathbb{R}^{1 \times n}$ is the i^{th} row of K and k_{ii} is the i^{th} diagonal element of K. This framework is known as robust kernelized graph-based

learning.

2.2.4 Robust Self-weighted Multi-view Multiple Kernel Learning

If the given dataset X has q number of views denoted by $(X_1, X_2, \cdots, X_v, \cdots, X_q)$ where, $X_v \in \mathbb{R}^{\mathrm{d^v \times n}}$, n is the total number of data points and d^v is the feature dimension of the v^{th} view X_v. For each view of the dataset, u number of kernel are used. Now following the same procedure that formulates the SMVMKL framework and using Eq. (2.13), the robust self-weighted multi-view multiple kernel learning framework (RSMVMKL) framework can be formulated as:

$$\begin{aligned} \underset{S,P,K}{\text{minimize}} \quad & \sum_{i=1}^{n} \sqrt{k_{ii} - 2k^i s_i + s_i^T K s_i} + \lambda \|S\|_F^2 \\ & + \alpha Tr(P^T L P) \\ & + \beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K \|_F^2 \\ \text{subject to} \quad & S \geqslant 0, \ \ P^T P = I \end{aligned} \tag{2.14}$$

Eq.(2.14) can be simplified into:

$$\begin{aligned} \underset{S,P,K}{\text{minimize}} \quad & \sum_{i=1}^{n} d_{ii} \left(k_{ii} - 2k^i s_i + s_i^T K s_i\right) + \lambda \|S\|_F^2 \\ & + \alpha Tr \left(P^T L P\right) \\ & + \beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K \|_F^2 \\ \text{subject to} \quad & S \geqslant 0, \ \ P^T P = I \end{aligned} \tag{2.15}$$

where,

$$d_{ii} = \frac{1}{\sqrt{k_{ii} - 2k^i s_i + s_i^T K s_i}} \tag{2.16}$$

2.3 Optimization

Proposed SMVMKL and RSMVMKL frameworks have been stated in Eq. (2.10) and Eq. (2.15) respectively. From both the equations it is observed that both the proposed frameworks negotiate between the process of label learning and optimal kernel learning. Therefore, to solve the problem sated in Eq. (2.10) and Eq. (2.15) two iterative algorithms have been stated where S, P, and K are updated in an iterative manner. Both the proposed algorithms serve two purpose: clustering and semi supervised classification.

2.3.1 SMVMKL

2.3.1.1 Clustering

For simplicity, the problem in Eq. (2.10) has been divided into three sub-problems with respect to every variable which can be solved by an alternating and iterative algorithm. For every sub-problem only one variable is updated and other variables are considered as constant.

(i) To update S, keeping P and K constant, Eq. (2.10) becomes:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & Tr\left(K - 2KS + S^T KS\right) + \lambda \|S\|_F^2 \\ & + \alpha Tr\left(P^T LP\right) \\ \text{subject to} \quad & S \geqslant 0 \end{aligned} \tag{2.17}$$

An elementary but very important equation of spectral analysis is:

$$\sum_{i,j} \frac{1}{2} \|P_{i,:} - P_{j,:}\|_2^2 s_{ij} = Tr(P^T LP) \tag{2.18}$$

By using Eq. (2.18), it can be shown that the problem in Eq. (2.17) is column wise

independent and it can be written for each i^{th} column of S as:

$$-2k^i s_i + s_i^T K s_i + \lambda s_i^T s_i + \frac{\alpha}{2} g_i^T s_i \tag{2.19}$$

where $g_i = [g_{i1}, g_{i2}, \cdots, g_{in}]^T \in \mathbb{R}^n$ and $g_{ij} = \|P_{i,:} - P_{j,:}\|_2^2$.
Setting the first derivative of Eq. (2.19) with respect to s_i to zero, we get:

$$-2k^i + 2Ks_i + 2\lambda s_i + \frac{\alpha}{2} g_i = 0 \tag{2.20}$$

From Eq. (2.20) each column of S is obtained as:

$$\mathbf{s_i} = (\lambda I + K)^{-1}(k^i - \frac{\alpha}{4} g_i) \tag{2.21}$$

Thus from Eq. (2.21) it is easily observed that each i^{th} column s_i can be obtained parallelly.

(ii) To update kernel matrix K, keeping S and F fixed, the problem stated in Eq. (2.10) can be written as:

$$\begin{aligned} \underset{K}{\text{minimize}} \quad & Tr\left(K - 2KS + S^T K S\right) \\ & + \beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K\|_F^2 \end{aligned} \tag{2.22}$$

By setting the first derivative of Eq. (2.22) with respect to K to be zero, we get

$$I - 2S^T + SS^T + 2\beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} \left(K - H^{(p,v)}\right) = 0 \tag{2.23}$$

By solving Eq. (2.23), we get the optimal kernel matrix as:

$$K = \frac{2S^T - SS^T - I + 2\beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} H^{(p,v)}}{2\beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)}} \tag{2.24}$$

From Eqs. (2.21), and (2.24), it is observed that S and K are dependent on each other and can negotiate between themselves for a better result.

(iii) To update the label indicator matrix P keeping S and K fixed, the problem in Eq. (2.10) becomes as following:

$$\begin{aligned} \underset{P}{\text{minimize}} \quad & \alpha Tr\left(P^T L P\right) \\ \text{subject to} \quad & P^T P = I \end{aligned} \tag{2.25}$$

If there are m number of classes, then the optimal solution of P is the m eigenvectors of L corresponding to the m smallest eigenvalues.

2.3.1.2 Semi-supervised Classification

Two fundamental stages of semi-supervised classification are graph construction and label inference. These stages are unified in the proposed SMVMKL framework. Using the labelled samples, the graph is constructed and then by label inference, the unknown labels are predicted.

The SMVMKL framework for semi-supervised classification is given as:

$$\begin{aligned} \underset{S,P,K}{\text{minimize}} \quad & Tr\left(K - 2KS + S^T K S\right) + \lambda \|S\|_F^2 \\ & + \alpha Tr(P^T L P) \\ & + \beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K \|_F^2 \\ \text{subject to} \quad & S \geqslant 0, \ \ P_l = Y_l \end{aligned} \tag{2.26}$$

where l is the total number of labelled points, and $Y_l = [y_1, y_2, y_3, \cdots, y_i, \cdots, y_l]^T \in \mathbb{R}^{\mathrm{l} \times \mathrm{m}}$ is the labelled indicator matrix where $y_i \in R^{m \times 1}$ is the labelled indicator vector of the i^{th} sample. When $y_{ij} = 1$, the i^{th} sample belongs to the j^{th} class. Without loss of generality, the data points are divided in such a way that the first l data points are labelled and rest of the u data points are unlabelled, such that $l + u = n$ and n is

the total number of data points. Now the Laplacian matrix L and and class indicator matrix P can be written as a block matrix:

$$L = \begin{pmatrix} L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{pmatrix} \text{ and } P = \begin{pmatrix} P_l \\ P_u \end{pmatrix} \tag{2.27}$$

where, P_l is the labelled indicator matrix and P_u is the unlabelled indicator matrix. Now Eq. (2.26) can be solved using the same iterative procedure used to solve Eq. (2.10), but the difference lies in updating P. The optimal class indicator matrix P is obtained by solving the following minimization problem:

$$\begin{aligned} &\underset{P}{\text{minimize}} \quad Tr(P^T L P) \\ &\text{subject to} \quad P_l = Y_l \end{aligned} \tag{2.28}$$

By setting the first derivative of Eq. (2.28) with respect to P equal to zero, we get:

$$\begin{pmatrix} L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{pmatrix} \begin{pmatrix} Y_l \\ P_u \end{pmatrix} = 0 \tag{2.29}$$

By solving Eq. (2.29) we get,

$$P_u = -L_{uu}^{-1} L_{ul} Y_l \tag{2.30}$$

If ij^{th} element of P is p_{ij}, then the final class labels of the unlabelled data points can be assigned by the following decision rule:

$$y_i = \arg\max p_{ij} \tag{2.31}$$

$$\forall i = l+1, l+2, ..., n. \quad \forall j = 1, 2, 3, ..., m$$

The proposed SMVMKL algorithm is summarized and stated in Algorithm 1.

Algorithm 1 : Proposed SMVMKL framework

Input: Kernel matrices for each view: $\{H^{(p,v)}\}$, parameters α, β and λ.

Output:
Clustering: Similarity matrix S with exact m connected components and optimal kernel matrix K.
Classification: The label matrix P for all data points.

Initialization: S, K, $Z_{(p,v)}$.

Repeat:

- update the i^{th} column of S as per Eq. (2.21)
- calculate K using Eq. (2.24)
- update $Z_{(p,v)}$ by Eq. (2.5)
- Clustering: calculate P by solving Eq. (2.25) as the m smallest eigenvector, correspond to the m smallest eigenvalues of the Laplacian matrix L.

Until stopping criterion is met.

- Classification: assign the class label to the unlabelled points by Eq. (2.31)

2.3.2 RSMVMKL

A robust self-weighted multi-view multiple kernel graph-based learning algorithm has been proposed to solve the RSMVMKL framework stated in Eq. (2.15). This proposed algorithm also serves two purposes: clustering and semi supervised classification.

2.3.2.1 Clustering

For simplicity, the problem defined in Eq. (2.15) has also been divided into three different sub-problems for with respect to each variable which can be solved by an iterative algorithm. For every sub-problem only one variable is updated and other variables are considered as constant.

(i) To update S keeping F and K fixed, Eq. (2.15) can be written as:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \sum_{i=1}^{n} d_{ii}\left(-2k^i s_i + s_i^T K s_i\right) + \lambda \|S\|_F^2 \\ & + \alpha Tr\left(P^T L P\right) \\ \text{subject to} \quad & S \geqslant 0 \end{aligned} \tag{2.32}$$

Using the same elementary equation of spectral clustering as stated in Eq. (2.18), it is observed that the problem stated in Eq. (2.32) is column wise independent for each i^{th} column $\mathbf{s_i}$ of S and it can be stated as:

$$d_{ii}(-2k^i s_i + s_i^T K s_i) + \lambda s_i^T s_i + \frac{\alpha}{2} g_i^T s_i \tag{2.33}$$

Now setting the first derivative of Eq. (2.33) with respect to $\mathbf{s_i}$ to be zero, each column of the similarity matrix is obtained as:

$$\mathbf{s_i} = (\lambda I + d_{ii} K)^{-1}(d_{ii} k^i - \frac{\alpha}{4} g_i) \tag{2.34}$$

Thus each i^{th} column $\mathbf{s_i}$ can be computed parallelly.

(ii) To get the optimal kernel matrix K keeping S and P fixed, Eq.(2.15) can be written as:

$$\begin{aligned} \underset{K}{\text{minimize}} \quad & \sum_{i=1}^{n} d_{ii}\left(k_{ii} - 2k^i s_i + s_i^T K s_i\right) \\ & + \beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K\|_F^2 \end{aligned} \tag{2.35}$$

By setting the first derivative of Eq.(2.35) with respect to K to be zero, the optimal kernel matrix is obtained as:

$$K = \frac{2\beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} H^{(p,v)} - \sum_{i=1}^{n} d_{ii}\left(E_i - 2\hat{S}_i + \mathbf{s_i}^T \mathbf{s_i}\right)}{2\beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)}} \tag{2.36}$$

where, $\hat{S}_i = [0, 0, \cdots, s_i, \cdots, 0]^T \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$ and $E_i \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$ whose all elements are zero except the $(ii)^{th}$ element.

(iii) For updating the class indicator matrix P keeping S and K fixed, the problem in Eq.(2.15) can be restated as:

$$\begin{aligned} \underset{P}{\text{minimize}} \quad & \alpha Tr(P^T L P) \\ \text{subject to} \quad & P^T P = I \end{aligned} \tag{2.37}$$

If there are m number of classes, then the optimal solution of P is the m eigenvectors of L corresponding to the m smallest eigenvalues.

2.3.2.2 Semi-supervised Classification

Following the same procedure adopted in semi-supervised classification using SMVMKL, the RSMVMKL framework can be written as:

$$\begin{aligned} \underset{S,P,K}{\text{minimize}} \quad & \sum_{i=1}^{n} d_{ii}\left(k_{ii} - 2k^i s_i + s_i^T K s_i\right) + \lambda \|S\|_F^2 \\ & + \alpha Tr\left(P^T L P\right) \\ & + \beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K\|_F^2 \\ \text{subject to} \quad & S \geqslant 0, P_l = Y_l \end{aligned} \tag{2.38}$$

where l is the total number of the labelled points.
Eq.(2.38) can be solved using the iterative procedure used to solve Eq. (2.15), only differing in the updation method of P. To solve for P, the steps for solving Eq.(2.26) have been followed and the unlabelled indicator matrix P_u is obtained from Eq. (2.30). The final class labels of the unlabelled data points can be assigned by Eq. (2.31). The proposed RSMVMKL algorithm is summarized and stated in Algorithm 2.

Algorithm 2 : Proposed RSMVMKL framework

Input: Kernel matrices for each view: $\{H^{(p,v)}\}$, parameters α, β and λ.

Output:
Clustering: Similarity matrix S with exact m connected components and optimal kernel matrix K.
Classification: The label matrix P for all data points.

Initialization: d_{ii}, E_i, S, K.

Repeat:

- update the i^{th} column of S as per Eq.(2.34)
- calculate K using Eq.(2.36)
- update d_{ii} by Eq.(2.16)
- update $Z_{(p,v)}$ by Eq.(2.5)
- Clustering: calculate P by solving Eq.(2.25) as the eigenvectors, corresponds to the m smallest eigenvalues of the Laplacian matrix, L.

Until stopping criterion is met.

- Classification: assign the class label to the unlabelled points by Eq.(2.31)

2.4 Experiment

Various experiments are performed on different real-world datasets to validate the performances of the proposed SMVMKL and RSMVMKL frameworks. For each framework, multiple views of a dataset are used and multiple kernels are assigned to each view of the data. Here twelve kernels are designed for each view of a given dataset. Of which, seven are Gaussian kernel of the form: $K(y, z) = exp(-\| y - z\|_2^2/(nd_{max}^2))$, where the maximum distance between samples is d_{max} and n varies over the set [0.01,0.05.0.1,1,10,50,100]. The 8^{th} kernel is of linear form: $K(y, z) = y^T z$. And the last four are polynomial kernel of the form: $K(y, z) = (a + y^T z)^b$ where a and b vary between {0, 1} and {2, 4} respectively. For semi-supervised classification, the percentage of labeled data is changed to observe the performances of both the SMVMKL and RSMVMKL algorithm while different number of labeled data samples are vailable. For the exiperiments, 10%, 30%, and 50% of labelled data are considered.

2.4.1 Dataset

To show the better learning performances of both the SMVMKL and RSMVMKL, various experiments are performed on different real-world benchmark datasets. To perform the learning task, Animal with Attributes [37], MSRC-v1 [40], Jaffe [43], Reuters Multilingual Data [45] and Caltech101 [46] datasets are used for both the SMVKL and RSMVMKL framework. As, both the SMVMKL and RSMVMKL algorithm based on multi-view, different views are extracted from those datasets. The details of the datasets are stated in section 1.5. All the related information of the datasets are given in Table 2.1.

Table 2.1: Statistics of the datasets used for the experiment

dataset	number of views	Instances	Classes
Animal with Attributes	3	100	10
MSRC-V1	3	210	7
Jaffe	3	200	10
REUTERS	5	180	6
Caltech101	6	241	5

2.4.2 Comparison Methods

Comparisons with the following state-of-the-art methods have been carried out to observe the effectiveness of the proposed SMVMKL and RSMVMKL algorithms.

- **MMSC** [41]: Multi-modal spectral clustering method learns a Laplacian matrix that is shared commonly by each and every modal of the dataset. Non-negative relaxation is also used to improve clustering performances.
- **Co-train MVSC** [20]: In a co-training approach for multi-view spectral clustering method, a graph is learned for each view and spectral clustering is performed on each graph. The clustering in one view helps to improve the performances of other views and vice-versa.
- **Co-reg MVSC** [21]: In a co-regularized multi-view spectral clustering ap-

proach, a cluster indicator matrix is learned where each sample belongs to the same cluster for each view and it is done by co-regularizing the different clustering hypothesis.

- **SwMC** [23]: In self-weighted multi-view clustering with multiple graphs method, a common Laplacian rank constrained graph is obtained for each and every view and the proper weight assignment to each view is done automatically.
- **MVCSK** [59]: In this kernelized multi-view clustering method, a kernel is used to consider the nonlinearity present in the dataset and the use of kernel improves the clustering performances.
- **GFSC** [60]: In multi-graph fusion for multi-view spectral clustering, graph fusion is performed where the fusion graph approximates the original graph of each view and maintains a proper structure of the cluster. Simultaneously spectral clustering is also performed.
- **RGC** [7]: The method of robust graph learning from noisy datais a robust version of manifold regularized robust principal component analysis (RPCA). It uses the enhanced low-rank recovery by exploiting the graph smoothness assumption for better graph learning performance.
- **AMGL** [22]: Auto-weighted multiple graph learning method learns an optimal graph for each view and automatic weight assignment to each view is also done. The objective function of AMGL for semi-supervised classification is a convex function thus obtaining the globally optimal result.
- **MLAN** [27]: Multi-view learning with adaptive neighbor method learns an optimal graph for each view by learning the local structure of the graph and the optimal graph can be partitioned into specific clusters. Here also the ideal weight assignment to each view is done automatically.
- **AMMSS** [48]: Adaptive multi-modal semi-supervised classification approach learns a class indicator matrix that is shared commonly by each modal of the

dataset and the proper weights to different models of the dataset are assigned.

Spectral clustering (**SC**) [2] and semi-supervised learning using Gaussian fields and harmonic functions (**GFHF**) [61] have been performed on each view of each dataset and are considered as the baselines for clustering and semi-supervised task respectively.

2.5 Result

The clustering and semi supervised classification performances of the proposed SMVMKL and RSMVMKL frameworks have been compared with various other existing methods mentioned here.

Table 2.2: Clustering performances on different datasets for SMVMKL and RSMVMKL framework

	Animal with Attributes			MSRC-v1			Jaffe			Reuters			Caltech-101		
	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
SC(1)	0.2780	0.2381	0.2955	0.2821	0.1848	0.3298	0.4366	0.4905	0.5915	0.3467	0.1621	0.3817	0.3506	0.1889	0.3651
SC(2)	0.2435	0.1902	0.2645	0.4354	0.3826	0.5048	0.4977	0.6063	0.6526	0.3522	0.2202	0.3997	0.3614	0.1086	0.4073
SC(3)	0.2285	0.1612	0.2550	0.3728	0.3343	0.4351	0.5482	0.5766	0.6338	0.3344	0.2234	0.3778	0.4064	0.2102	0.4313
SC(4)										0.3253	0.1468	0.3586	0.5438	0.3637	0.5454
SC(5)										0.3769	0.2094	0.3997	0.5247	0.2229	0.5301
SC(6)													0.3494	0.1175	0.3512
RGC	0.2600	0.2058	0.2700	0.5305	0.4253	0.5390	0.7371	0.7940	0.7512	0.3756	0.1756	0.5867	0.0.5519	0.4525	0.5889
MMSC	0.2787	0.2479	0.3677	0.6269	0.5951	0.6395	0.8751	0.9067	0.9437	0.4478	0.3170	0.4933	0.5232	0.4028	0.5245
Co-train MVSC	0.2850	0.2534	0.3480	0.5924	0.5150	0.6433	0.9117	0.9378	0.9594	0.4167	0.2716	0.5111	0.6091	0.4488	**0.6680**
Co-reg MVSC	0.2570	0.2033	0.5190	0.4336	0.3753	0.5169	0.9145	0.9440	0.9678	0.4561	0.2857	0.5067	0.5855	0.4570	0.6502
SwMC	0.2200	0.1854	0.2200	0.6238	0.5669	0.6476	0.6850	0.7287	0.6700	0.2167	0.1152	0.2389	0.4979	0.2910	0.5104
MVCSK	0.2552	0.1969	0.5247	0.6429	0.5851	0.7238	0.9631	0.9512	0.9674	0.4556	0.2975	0.5611	0.5270	0.2996	0.5602
GFSC	0.2784	0.2314	0.4270	0.6976	0.6109	0.7271	0.9425	0.9652	0.9681	0.4420	0.2855	0.5956	0.6089	**0.5060**	0.6575
SMVMKL	0.2900	0.2637	**0.5500**	0.7000	0.6604	0.7312	0.9765	0.9654	0.9765	0.4788	0.3207	0.5667	0.6183	0.4756	0.6598
RSMVMKL	**0.3100**	**0.2880**	0.4400	**0.7238**	**0.6622**	**0.7714**	**0.9800**	**0.9719**	**0.9800**	**0.4833**	**0.3347**	**0.6000**	**0.6266**	0.4836	**0.6680**

2.5.1 Performance Evaluation

Table 2.2 and Table 2.3 show the clustering and the semi-supervised learning performances of SMVMKL and RSMVMKL respectively. Fig. 2.1 shows that the clustering performances of both SMVMKL and RSMVMKL methods in terms of ACC, NMI and

Table 2.3: Semi-supervised classification performances on different datasets for SMVMKL and RSMVMKL framework

	Animal with Attributes			MSRC-v1			Reuters		
	ACC			ACC			ACC		
Labelled Rate	0.1	0.3	0.5	0.1	0.3	0.5	0.1	0.3	0.5
GFHF(1)	0.1633	0.1740	0.1757	0.5100	0.5809	0.6247	0.2148	0.2444	0.2444
GFHF(2)	0.1411	0.1386	0.1640	0.5400	0.6123	0.6571	0.2277	0.2288	0.2371
GFHF(3)	0.1744	0.1900	0.2020	0.5247	0.5963	0.6377	0.2487	0.2651	0.2822
GFHF(4)							0.2161	0.2286	0.2356
GFHF(5)							0.2432	0.2571	0.2689
RGC	0.1222	0.2038	0.2183	0.7725	0.8371	0.8857	0.3710	0.6460	0.7033
AMMSS	0.1656	0.1929	0.2200	0.4434	0.7503	0.8219	0.5519	0.6373	0.6856
AMGL	0.1556	0.2000	0.2400	0.7606	0.8980	0.9238	0.3148	0.3968	0.444
MLAN	0.1613	0.2011	0.2328	0.7794	0.8611	0.8807	**0.6296**	0.7158	0.7444
SMVMKL	0.1833	0.2314	**0**.3000	0.8196	**0.9286**	**0.9867**	0.6191	0.6910	**0.8422**
RSMVMKL	**0.1937**	**0.2743**	**3080**	**0.8323**	0.9116	0.9772	0.6200	**0.7215**	0.8358

PUR in comparison with other existing graph-based methods that used only a single view are better. Also, Fig. 2.2 shows that both the proposed methods perform better than various multi-view spectral clustering methods. Considering the MMSC [41] method, where there exists a parameter that affects the creation of the Laplacian matrix during clustering, it is noticed in Fig. 2.3 that the performance of the clustering task using MMSC algorithm changes with the penalty parameter considered. But SMVMKL and RSMVMKL methods are free from such a parameter. Also, in the co-reg MVSC [21] method, there exists a weight parameter that affects the clustering performance and is shown in Fig. 2.4. But being self-weighted, SMVMKL and RSMVMKL methods are free from this weight parameter. Previously it has also been stated that for the learning task, 12 different kernels have been used and an optimal kernel is learnt from those given kernels. Now in Fig. 2.5 it has been shown how the learning performance gets affected while using different numbers of kernel and it is clearly observed from the figure that the learning performance is best while using 10 or 12 number of kernels and the accuracy of the performance gets reduced while using a very small number of kernel.

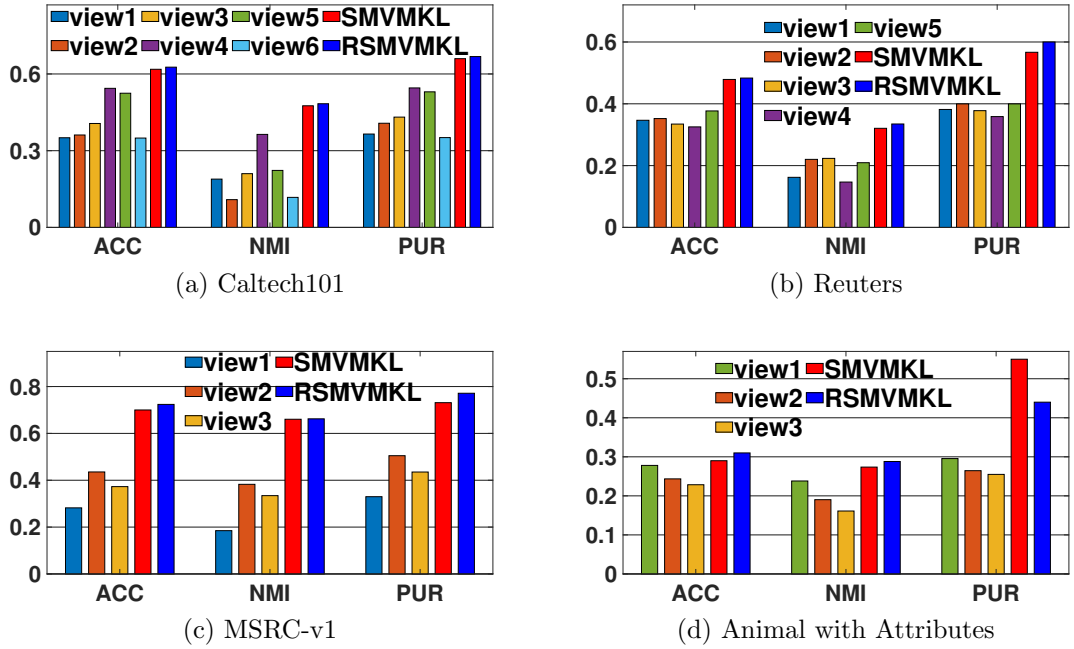
2.5.2 Convergence Analysis

The convergence of the proposed algorithms are described in this section. The equations defined in Eq. (2.10) and Eq. (2.15) are not jointly convex and the variables are coupled with each other making it difficult to optimize the whole function at once. Therefore those objective functions are divided into three subproblems. Eq. (2.10) is split into Eq. (2.17), Eq. (2.22), and Eq. (2.25). Eq. (2.15) is divided into (2.32), Eq. (2.35) and Eq. (2.37). The first two sub-problems for both the SMVMKL and the RSMVMKL algorithms are individually convex with respect to one variable. Therefore, the optimal solution is obtained by solving each sub-problem which converges to a global solution. But though Eq. (2.25) and Eq. (2.37) is not convex, it has a closed form optimal solution. In Fig. 2.6 and Fig. 2.14, it is shown how the value of the objective function of SMVMKL algorithm converges after a few iterations on different datasets in the case of clustering and semi-supervised classification task and the convergence of the proposed RSMVMKL algorithm has been shown in Fig. 2.10 and in Fig. 2.17. As, the performance of the clustering largely depends on the value of the learned similarity matrix, it has been shown in Fig. 2.7, 2.11, 2.15 and 2.18, that how the rate of updation of the similarity matrix with each iteration also converges and after certain numbers of iteration, the learned similarity matrix almost remains same.

2.5.3 Parameter Tuning and Sensitivity

The proposed algorithms contain three regularization parameters: α, β and λ. In order to find the right combination of the regularization parameters for the algorithms to give the best performance, a grid search has been performed. The parameters are observed to lie in the range: $\alpha \in [1e-6, 1e-3]$, $\beta \in [100, 1000]$ and $\lambda \in [1, 25]$.

The clustering performances of the proposed frameworks are shown in Fig. 2.8, Fig. 2.9, Fig. 2.12 and Fig. 2.13 for different values of α, β and λ. The performances of semi-supervised classification task of both SMVMKL and RSMVMKL method have



(a) Caltech101 (b) Reuters

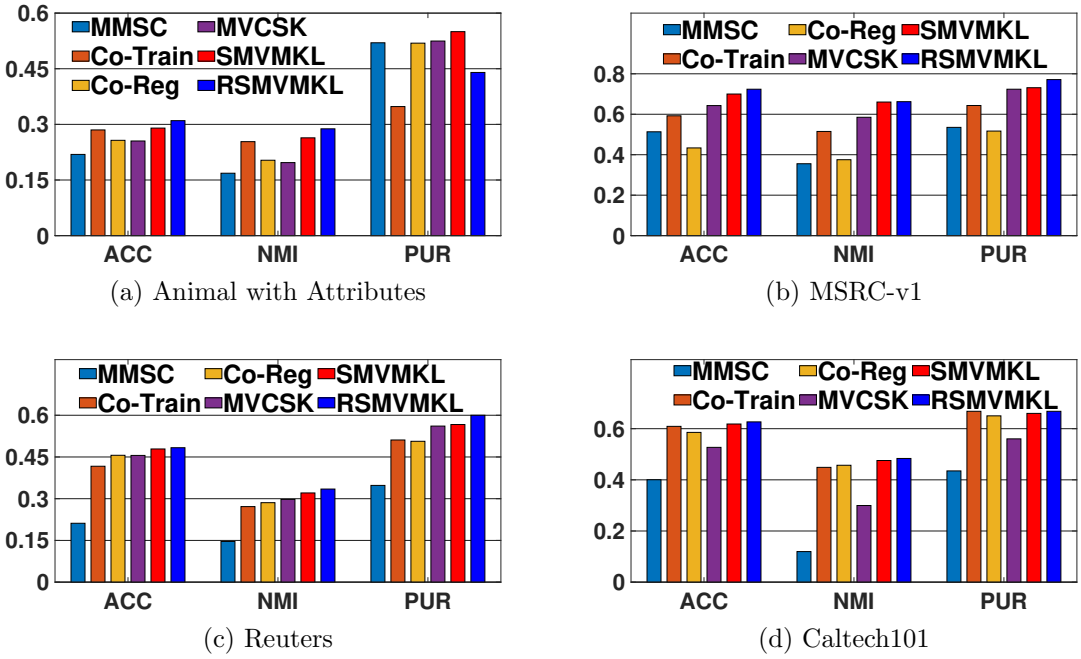
(c) MSRC-v1 (d) Animal with Attributes

Figure 2.1: Clustering performance between SMVMKL, RSMVMKL and SC (uses only one view) on Caltech, Reuters, MSRC-v1 and Animal with Attributes dataset.

been shown in Fig. 2.16 and Fig. 2.19 respectively for different values of α and β and λ.

2.5.4 Computational Complexity

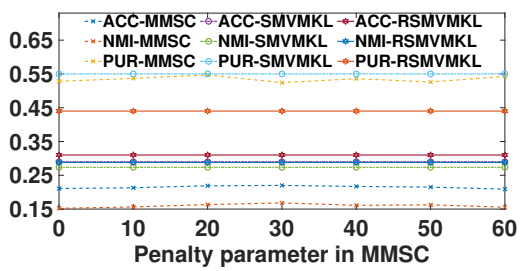
The computational complexity of both SMVMKL and RSMVMKL framework are discussed here. If there are n number of instances then the computational complexity of Algorithm 1 and Algorithm 2 is $\mathcal{O}(n^3)$. If the given dataset has v number of views and p number of kernel for each view then the complexity of kernel construction is $\mathcal{O}(n^2pv)$. But generally we have $pv << n$ thus $n^2pv << n^3$. Therefor, overall complexity of both the Algorithm 1 and Algorithm 2 is $\mathcal{O}(n^3)$.



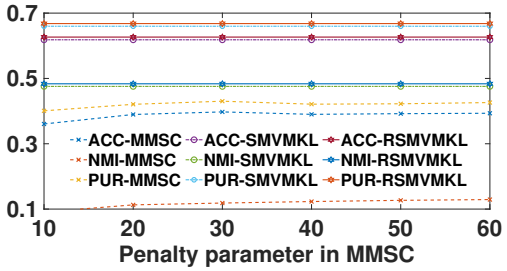
(a) Animal with Attributes (b) MSRC-v1

(c) Reuters (d) Caltech101

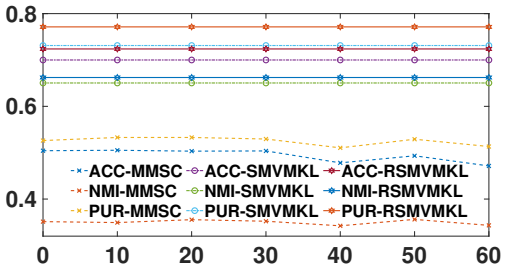
Figure 2.2: Clustering performance between SMVMKL, RSMVMKL and MVSC (uses multiple views) on Animal with Attributes, MSRC-v1, Reuters and Caltech dataset.



(a) Animal with Attributes

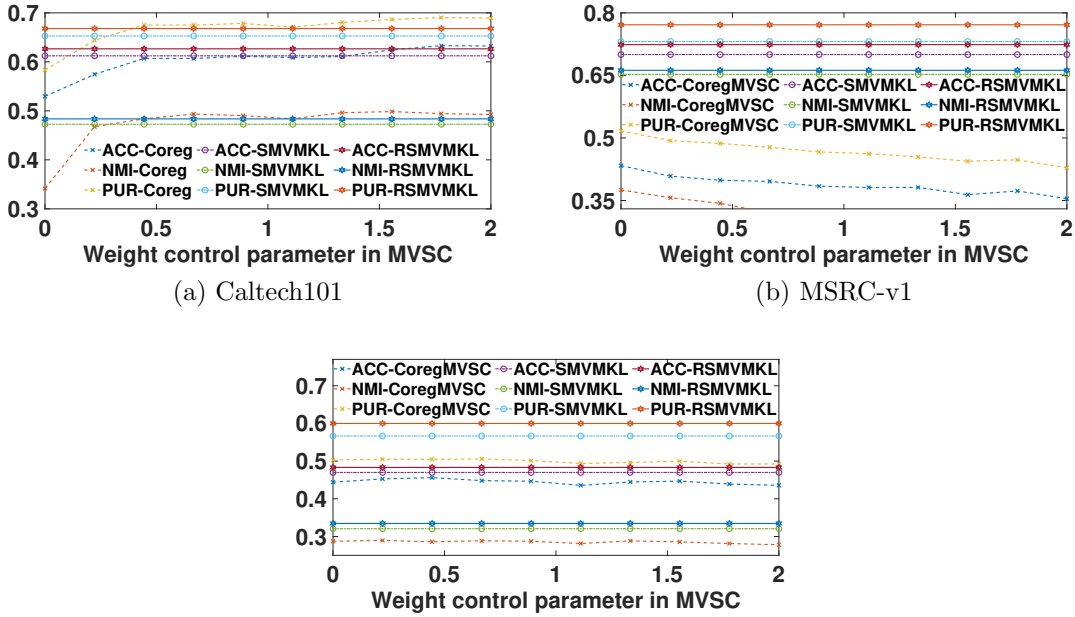


(b) Caltech



(c) MSRC-v1

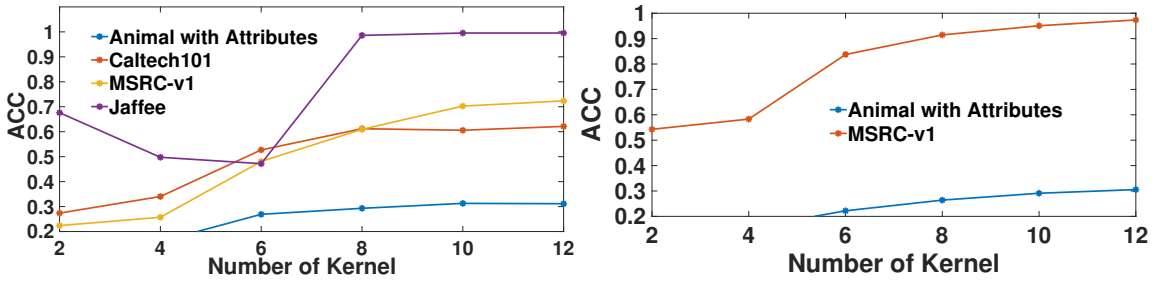
Figure 2.3: Comparison among the proposed SMVMKL and RSMVMKL methods and MMSC on Animal with Attributes, Caltech101 and MSRC-v1 dataset.



(a) Caltech101 (b) MSRC-v1

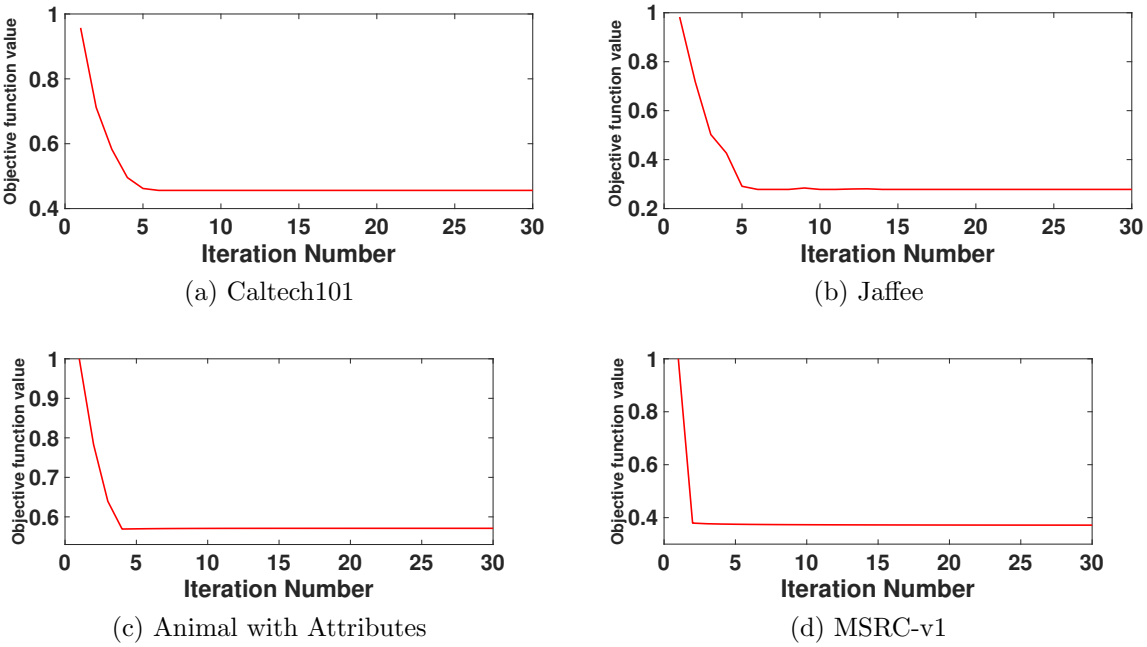
(c) Reuters

Figure 2.4: Comparison among the proposed SMVMKL and RSMVMKL methods and Co-reg MVSC on Caltech, MSRC-v1 and Reuters dataset.



(a) Clustering (b) SSC

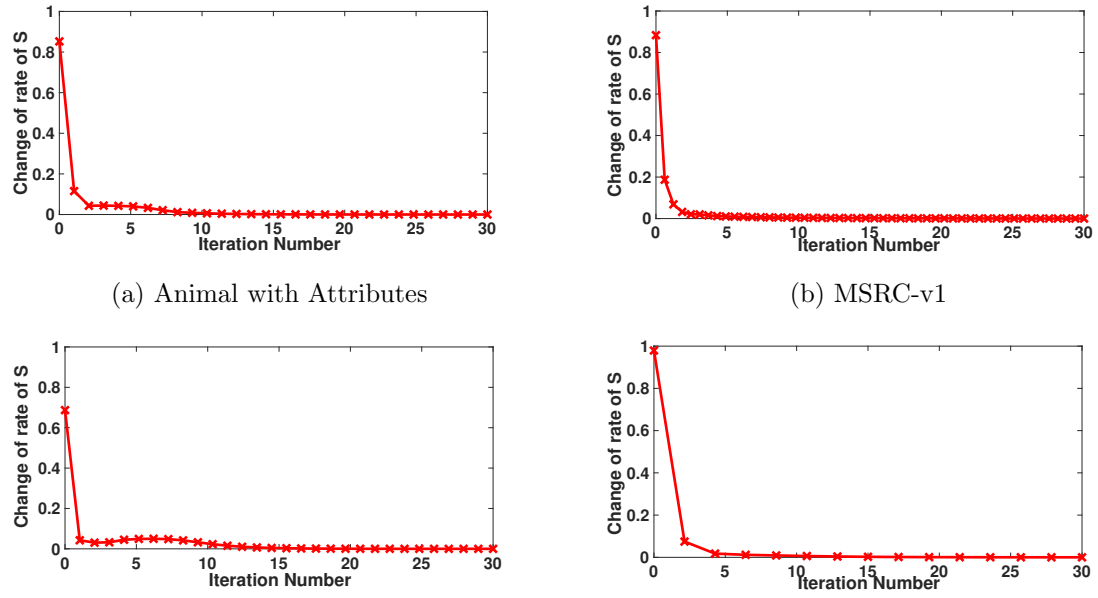
Figure 2.5: Change of the performance of RSMVMKL with different numberos of kernel while performing the clustering and semi-supervised classification task on different datasets.



(a) Caltech101 (b) Jaffee

(c) Animal with Attributes (d) MSRC-v1

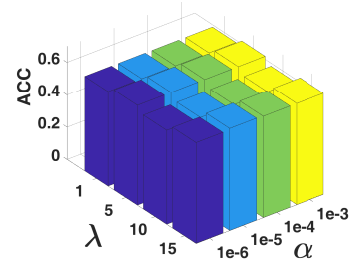
Figure 2.6: Clustering convergence of SMVMKL framework on different datasets.



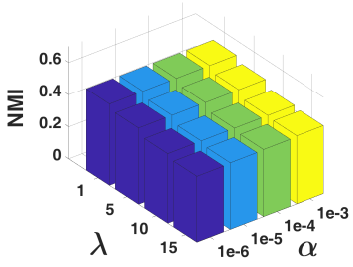
(a) Animal with Attributes (b) MSRC-v1

(c) Reuters (d) Caltech101

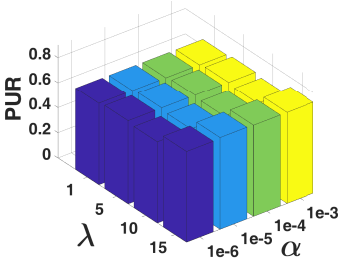
Figure 2.7: Rate of change of similarity matrix for clustering task using SMVMKL.



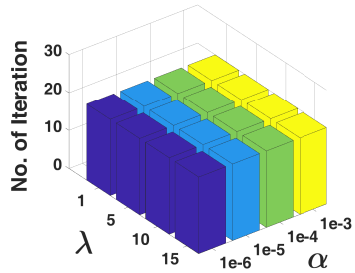
(a) ACC



(b) NMI

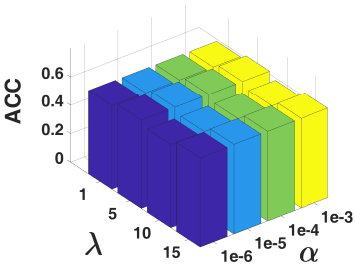


(c) PUR

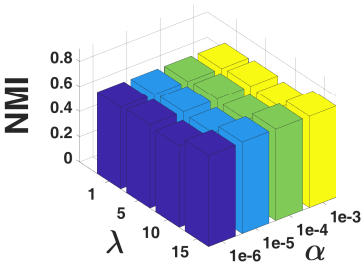


(d) No. of iteration

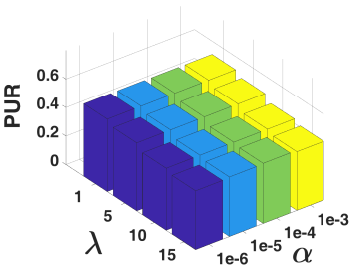
Figure 2.8: Clustering performance of SMVMKL framework on Caltech101 dataset for different values of α and λ while $\beta = 100$.



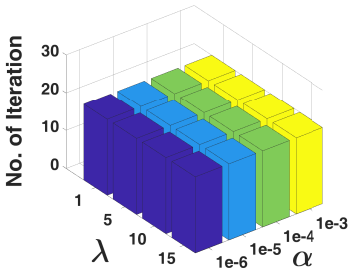
(a) ACC



(b) NMI

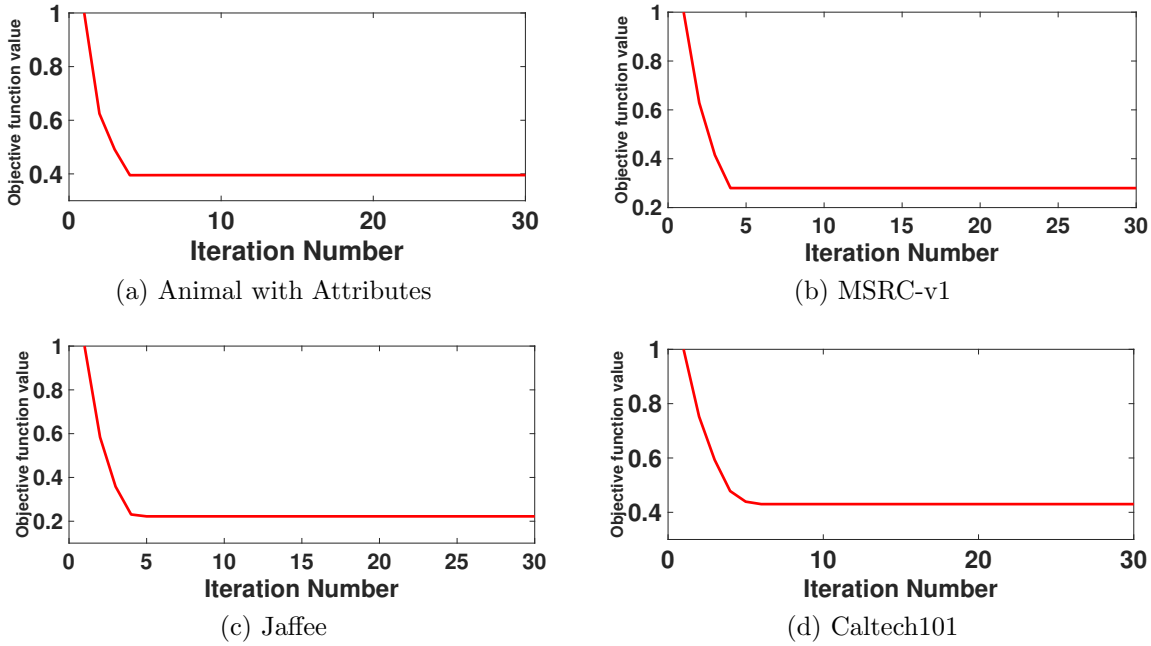


(c) PUR



(d) No. of iteration

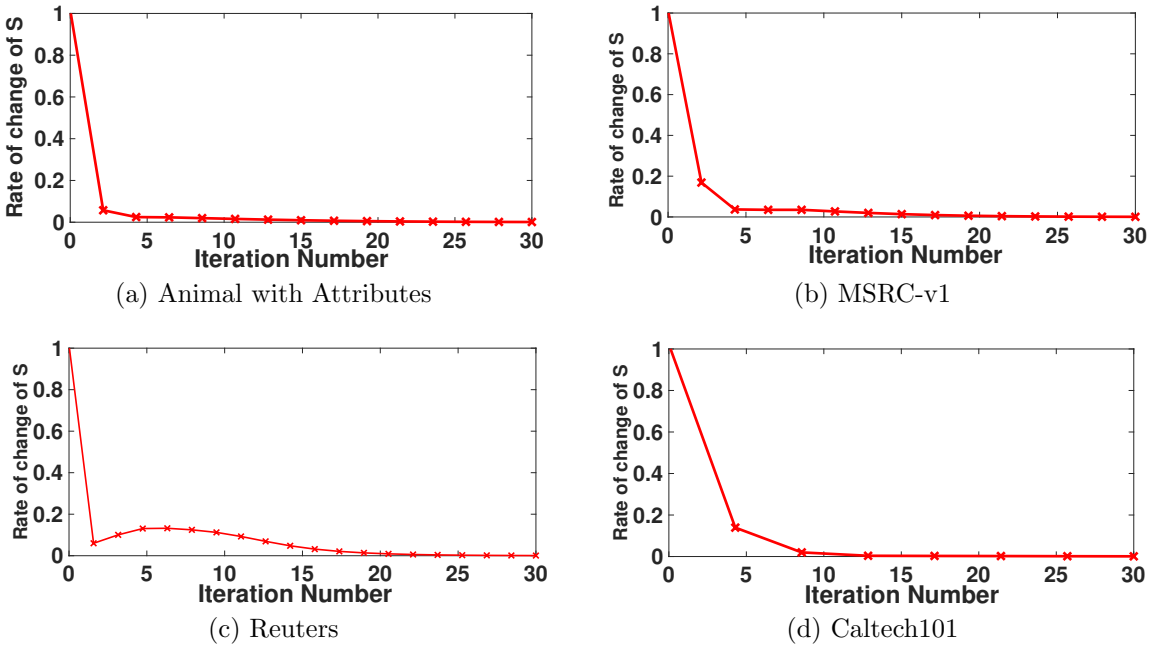
Figure 2.9: Clustering performance of SMVMKL framework on Caltech101 dataset for different values of α and λ while $\beta = 1000$.



(a) Animal with Attributes (b) MSRC-v1

(c) Jaffee (d) Caltech101

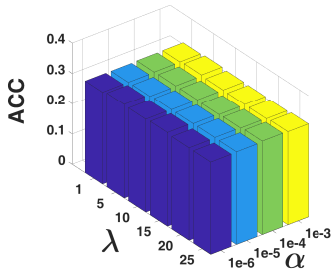
Figure 2.10: Clustering convergence of RSMVMKL framework on different datasets.



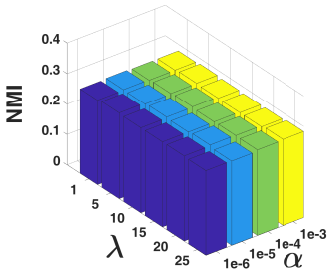
(a) Animal with Attributes (b) MSRC-v1

(c) Reuters (d) Caltech101

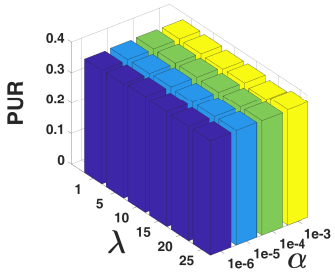
Figure 2.11: Rate of change of similarity matrix for clustering task using RSMVMKL



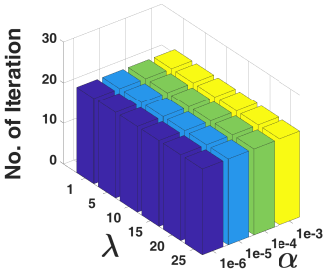
(a) ACC



(b) NMI

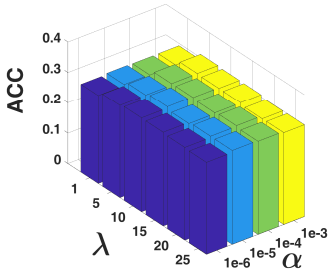


(c) PUR

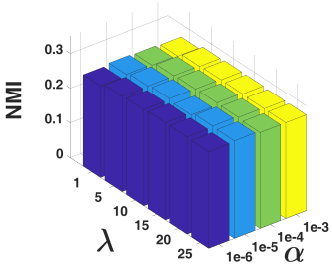


(d) No. of iteration

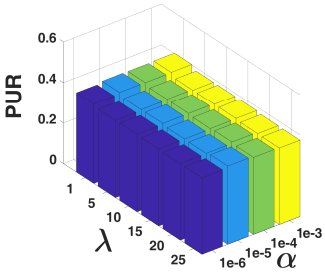
Figure 2.12: Clustering performance of RSMVMKL framework on Animal with Attributes dataset for different values of α and λ while $\beta = 100$.



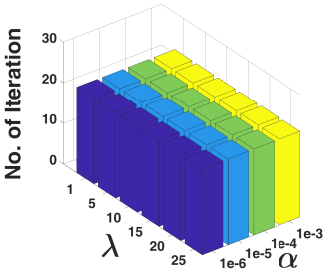
(a) ACC



(b) NMI

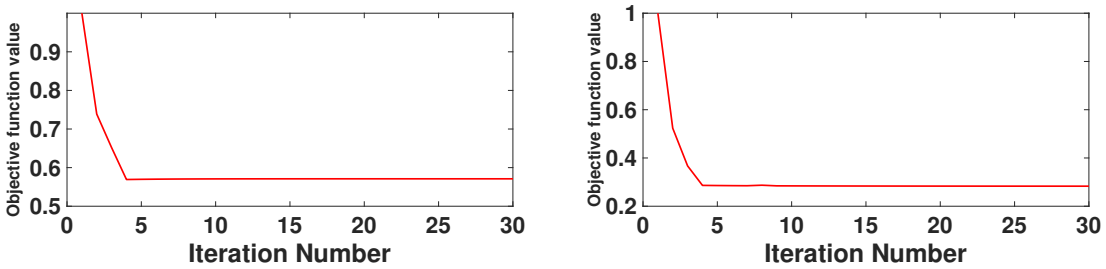


(c) PUR



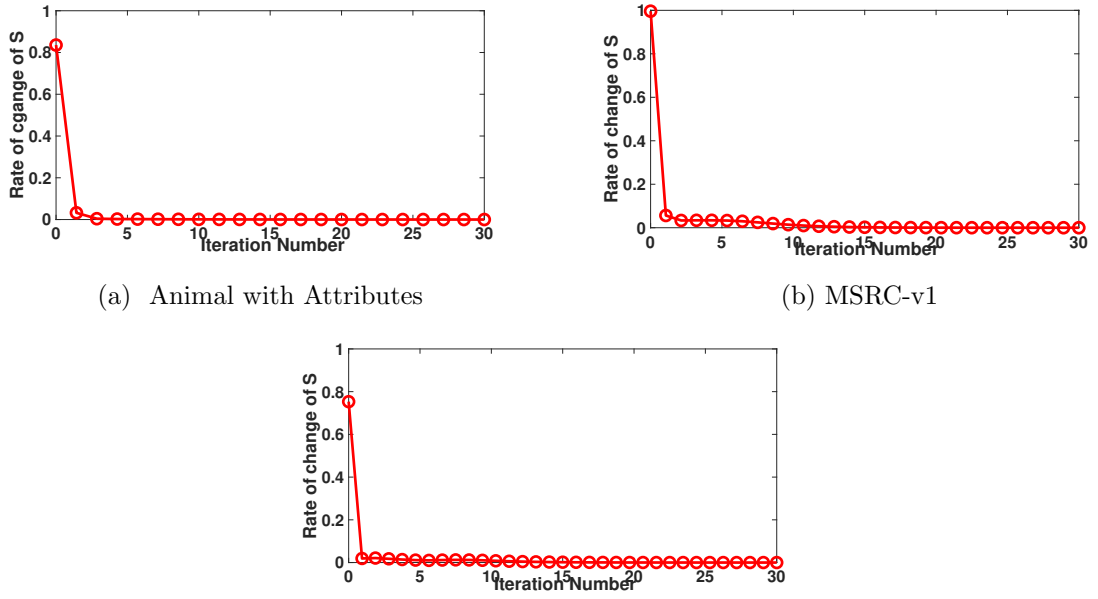
(d) No. of iteration

Figure 2.13: Clustering performance of RSMVMKL framework on Animal with Attributes dataset for different values of α and λ while $\beta = 1000$.



(a) Animal with Attributes (b) MSRC-v1

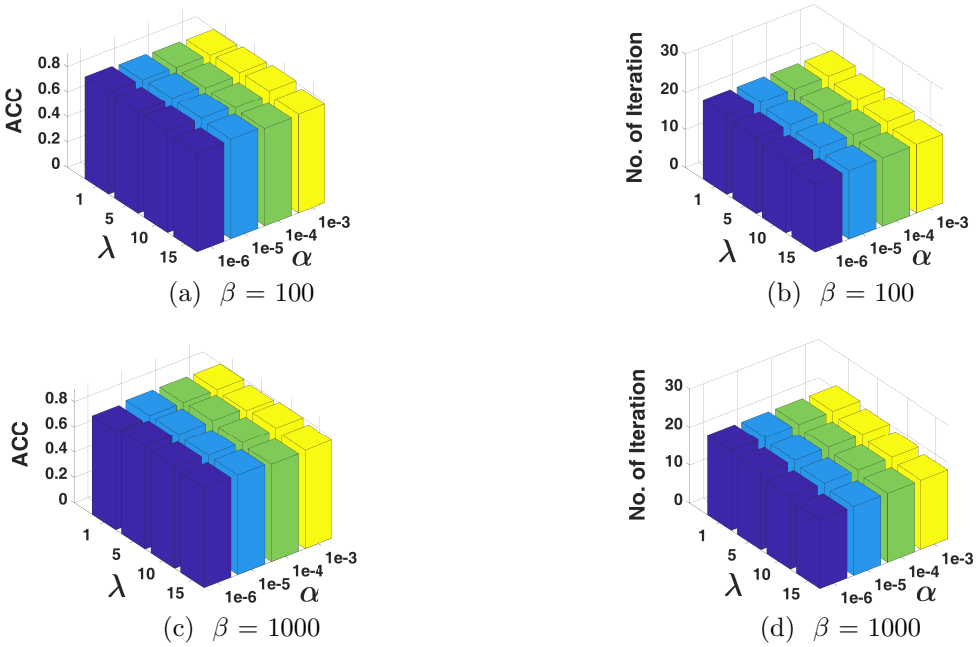
Figure 2.14: Semi-supervised classification (when 50% labelled data is available) convergence for proposed SMVMKL method on different datasets.



(a) Animal with Attributes (b) MSRC-v1

(c) Reuters

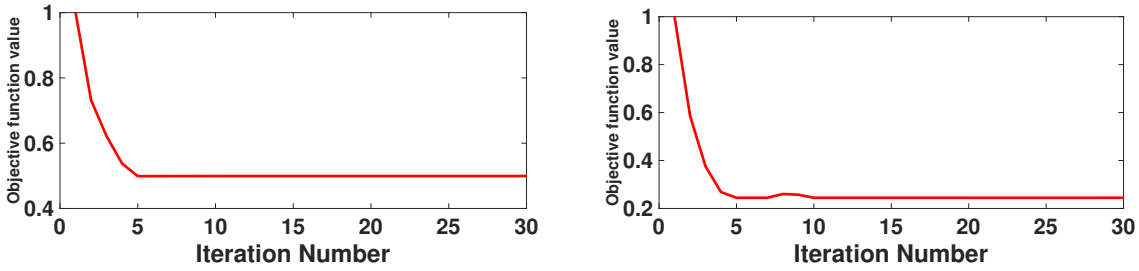
Figure 2.15: Rate of change of similarity matrix while performing semi-supervised classification task using SMVMKL framework.



(a) $\beta = 100$ (b) $\beta = 100$

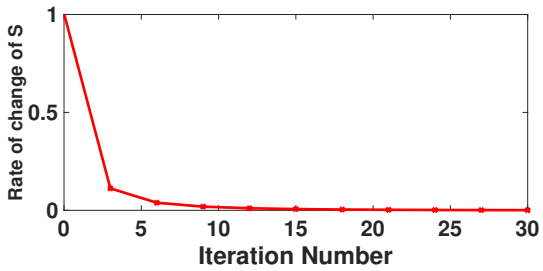
(c) $\beta = 1000$ (d) $\beta = 1000$

Figure 2.16: Semi-supervised classification performance of SMVMKL framework on MSRC-v1 dataset for different values of α and λ while $\beta = 100$ and $\beta = 1000$.

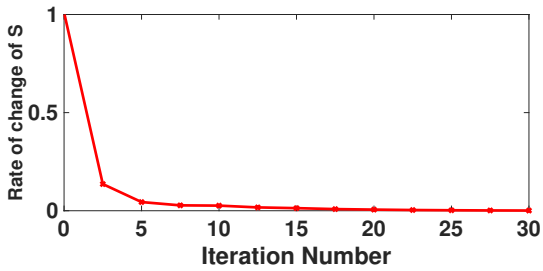


(a) Animal with Attributes (b) MSRC-v1

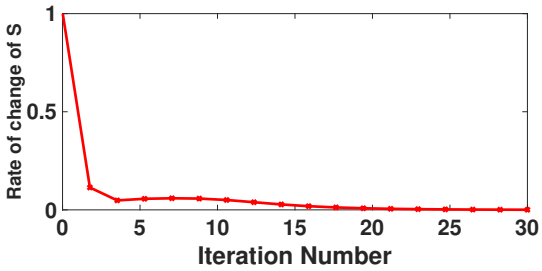
Figure 2.17: Semi-supervised classification (when 50 labelled data is available) convergence for proposed RSMVMKL method on different datasets.



(a) Animal with Attributes

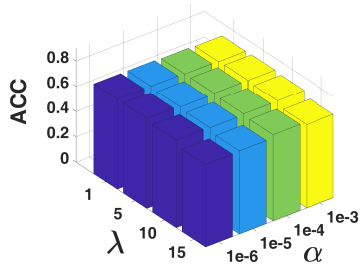


(b) MSRC-v1

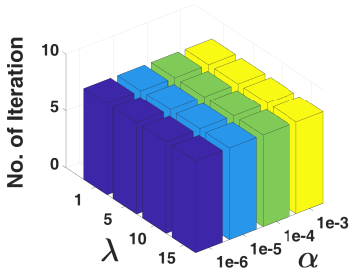


(c) Reuters

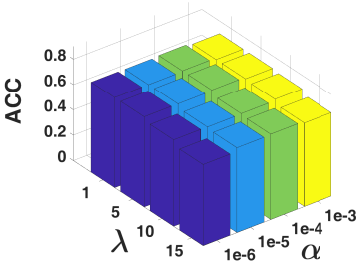
Figure 2.18: Rate of change of similarity matrix while performing semi-supervised classification task using RSMVMKL framework.



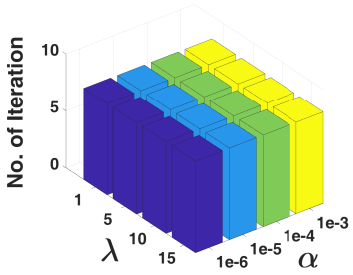
(a) $\beta = 100$



(b) $\beta = 100$



(c) $\beta = 1000$



(d) $\beta = 1000$

Figure 2.19: Semi-supervised classification performance of RSMVMKL framework on Reuters dataset for different values of α and λ while $\beta = 100$ and $\beta = 1000$.

2.6 Summary

In this chapter a novel SMVMKL framework has been proposed which performs the unsupervised and semi supervised learning task based on the graph representation of the dataset by using multiple kernels on multiple views. The weight assignment to each kernel of each view is also done automatically without using any additional weight assignment parameter. But this SMVMKL framework faces some problem due to the presence of noises and data outliers in the dataset and it may degrade the learning performances. Later, this issue is solved by using another novel framework named RSMVKL. Various extensive experiments on different real-world datasets show the effective and better performances of the proposed SMVMKL and RSMVMKL framework than the performances of other existing methods.

Chapter 3

Low-rank Kernelized Graph-based Clustering on Multiple Views

3.1 Introduction

Graph-based clustering [6] [62] is one of the most important learning tasks in the field of machine learning and pattern recognition. To achieve a better clustering performance, nonlinearity present in the dataset is considered while doing the clustering task and it is done by introducing various kernel methods [50][8]. Later multiple kernels learning [12][11] has been introduced in graph-based clustering to solve the issue of appropriate kernel selection. But those kernel learning methods don't consider the noise present in the dataset and get affected by that noise. To solve this issue, low-rank kernel optimization is incorporated in the proposed low-rank multi-view multi-kernel graph-based clustering (LRMVMKC) framework that uses low-rank kernel optimization using multiple kernels on multiple views. The main contributions of this paper are the use of low-rank kernel optimization which makes the framework less sensitive to the noise present in the dataset and the use of multiple kernels on multiple views instead of a single view that provides more information about the

dataset thus improving the clustering performances. The main contributions of the proposed methods have been summarized as:

- a novel low-rank kernel optimization using multiple kernels on multiple views and an appropriate weight assignment method to each kernel.
- Integration of kernel learning, graph construction and label learning thus achieving the optimal performances by negotiating between each other.

Several experiments have been performed on various real-world benchmark datasets using the proposed LRMVMKC framework and it has been observed that the performances of the proposed framework are better than other existing state-of-the-art methods.

3.2 Methodology

Multiple features are extracted from a given dataset and each feature is considered as a view. For each view multiple kernels are allocated. Now by using low-rank kernel optimization, the optimal kernel and the optimal similarity matrix of the graph are obtained. Proper weight assignment to each kernel of each view is also done by a self-weighted algorithm.

3.2.1 Low-rank Kernelized Graph-based Clustering

Let $X \in \mathbb{R}^{d \times n}$ is a given data matrix where the total number of available data samples and the feature dimension are denoted by n and d respectively. S is the initial similarity matrix or adjacency matrix of the given dataset. Now the optimal solution of S can be learned from the following minimization problem:

$$\begin{aligned} &\underset{S}{\text{minimize}} \quad \|X - XS\|_F^2 + \lambda \|S\|_F^2 \\ &\text{subject to} \quad S \geqslant 0. \end{aligned} \tag{3.1}$$

where λ is the trade-off parameter. To incorporate the kernel in the proposed framework, we use the kernel trick, $K(x,y) = \phi(x)^T\phi(y)$. Now Eq.(3.1) can be stated as:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \|\phi(x) - \phi(x)S\|_F^2 + \lambda\|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0. \end{aligned} \tag{3.2}$$

For any given matrix A, $\|A\|_F^2 = Tr(A^TA)$. Now using this property, Eq.(3.2) can be written as:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & Tr\left(K - 2KS + S^TKS\right) + \lambda\|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0. \end{aligned} \tag{3.3}$$

where, $K \in \mathbb{R}^{n\times n}$ is the kernel matrix. To make the proposed framework less sensitive to the noise and data outliers present in the dataset, low-rank minimization of kernel has been incorporated in the framework. Now Eq.(3.3) can be stated as following:

$$\begin{aligned} \underset{S,K}{\text{minimize}} \quad & Tr\left(K - 2KS + S^TKS\right) + \lambda\|S\|_F^2 + \zeta\|K\|_\star \\ \text{subject to} \quad & S \geqslant 0. \end{aligned} \tag{3.4}$$

where, $\|K\|_\star$ is the nuclear norm of the kernel matrix K and ζ is a regularization parameter. But the solution to this problem stated in Eq.(3.4) suffers from the choice of kernel and also it uses only a single view of the dataset.

3.2.2 Low-rank Multi-view Multi-kernel Graph-based Clustering

To solve this, multiple kernels on multiple views have been integrated with our proposed method. The multiple views of a given dataset X with m number of clusters are denoted by $[X_1, X_2, \cdots, X_v, \cdots, X_q]$ where $X_v \in \mathbb{R}^{d^v\times n}$ and total number of available data samples and the feature dimension of v^{th} view X_v are denoted by n and d^v respectively. To solve the issue of kernel selection, u number of kernels have been assigned to each view. The proposed LRMVMKC method is framed depending on

the following presumptions:
(i) each kernel of each view is a perturbed version of the consensus kernel
(ii) the closer the kernel to the consensus kernel, the larger weight will be assigned to that kernel.

Using these assumptions, the optimal kernel from multiple kernels is learned from the following minimization problem:

$$\underset{K}{\text{minimize}} \quad \sum_{v=1}^{q}\sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K\|_F^2 \tag{3.5}$$

where,

$$Z_{(p,v)} = \frac{1}{2\| \ H^{(p,v)} - K\|_F} \tag{3.6}$$

Here $H^{(p,v)}$ is the p^{th} kernel of the v^{th} view of the given dataset and $Z_{(p,v)}$ is the weight assigned to $H^{(p,v)}$. It can be observed from Eq.(3.5) and Eq.(3.6) that no extra parameter has been used for proper weight allotment of kernels. The proper weight can be allocated to each kernel of each view by Eq.(3.6). Now uniting Eq.(3.4), Eq.(3.5) and Eq.(3.6), the proposed LRMVMKC framework can be stated as:

$$\begin{aligned} \underset{S,K}{\text{minimize}} \quad & Tr\left(K - 2KS + S^T KS\right) + \lambda\|S\|_F^2 + \zeta\|K\|_\star \\ & + \beta\sum_{v=1}^{q}\sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K\|_F^2 \\ \text{subject to} \quad & S \geqslant 0 \end{aligned} \tag{3.7}$$

where, β, λ and ζ are three regularization parameters.
The similarity matrix S has an important property [57]:
"The multiplicity m of the eigenvalue 0 of the Laplacian matrix L is equal to the number of connected components in the graph associated with S."
This property provides a constraint on S and that constraint is $rank(\mathrm{L}) = n - m$.

Now Eq.(3.7) can no be rewritten as:

$$\begin{aligned} \underset{S,K}{\text{minimize}} \quad & Tr\left(K - 2KS + S^T KS\right) + \lambda\|S\|_F^2 + \zeta\|K\|_\star \\ & + \beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K\|_F^2 \\ \text{subject to} \quad & S \geqslant 0, \quad rank(L) = n - m \end{aligned} \tag{3.8}$$

Let i^{th} smallest eigenvalue of L is σ_i(L). Because of the positive semi-definiteness property of L, σ_i(L) $\geqslant 0$. Therefore $\sum_{i=1}^{m} \sigma_i(L) = 0$ will guarantee the constraint $rank$(L) $= n - m$. As stated in Ky Fan's Theorem [58]:

$$\min_{P^T P = I} \quad Tr(P^T LP) = \sum_{i=1}^{m} \sigma_i(L) \tag{3.9}$$

where $P \in \mathbb{R}^{n \times m}$ denotes the label or cluster indicator matrix. Now using Eq.(3.9), Eq.(3.8) can be rewritten as:

$$\begin{aligned} \underset{S,P,K}{\text{minimize}} \quad & Tr\left(K - 2KS + S^T KS\right) + \lambda\|S\|_F^2 + \zeta\|K\|_\star \\ & + \alpha Tr\left(P^T LP\right) \\ & + \beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K\|_F^2 \\ \text{subject to} \quad & S \geqslant 0, \quad P^T P = I \end{aligned} \tag{3.10}$$

Where α is a regularization parameter. This model formulates the proposed LR-MVMKC framework.

3.3 Optimization

The proposed framework stated by Eq.(3.10) is solved by an iterative algorithm. In this iterative procedure, when one variable is updated then the rest of the variables

are treated as a constant. One auxiliary variable has been introduced to make the variable separable from each other and using that Eq.(3.10) can be stated as following:

$$\begin{aligned} \underset{S,P,K}{\text{minimize}} \quad & Tr\left(K - 2KS + S^T KS\right) + \lambda \|S\|_F^2 + \zeta \|W\|_\star \\ & + \alpha Tr\left(P^T LP\right) \\ & + \beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K\|_F^2 \\ \text{subject to} \quad & S \geqslant 0, \quad P^T P = I, \quad W = K \end{aligned} \tag{3.11}$$

The corresponding augmented Laglangian function can be written as:

$$\begin{aligned} \mathcal{L}(S, K, P, W, Y) = & Tr\left(K - 2KS + S^T KS\right) + \lambda \|S\|_F^2 \\ & + \zeta \|W\|_\star + \alpha Tr\left(P^T LP\right) \\ & + \beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K\|_F^2 \\ & + \frac{\mu}{2} \|W - K + \frac{Y}{\mu}\|_F^2 \end{aligned} \tag{3.12}$$

where μ denotes a penalty parameter and Y is the Lagrangian multipliers. Now all the variables stated in Eq.(3.12) can be updated one by one alternatively.
(i) For updating S, Eq.(3.12) is written as:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & Tr\left(K - 2KS + S^T KS\right) + \lambda \|S\|_F^2 \\ & + \alpha Tr\left(P^T LP\right) \\ \text{subject to} \quad & S \geqslant 0 \end{aligned} \tag{3.13}$$

An important equation of spectral analysis is:

$$\sum_{i,j} \frac{1}{2} \|P_{i,:} - P_{j,:}\|_2^2 s_{ij} = Tr(P^T LP) \tag{3.14}$$

Now the problem stated in Eq.(3.13) can be solved for each i^{th} column of S as:

$$-2k^i s_i + {s_i}^T K s_i + \lambda s_i^T s_i + \frac{\alpha}{2} g_i^T s_i, \tag{3.15}$$

where $g_i = [g_{i1}, g_{i2}, \cdots, g_{in}]^T \in \mathbb{R}^n$ and $g_{ij} = \|P_{i,:} - P_{j,:}\|_2^2$. By setting the first derivative of Eq.(3.15) with respect to $\mathbf{s_i}$ to zero,each sample of S is calculated by:

$$\mathbf{s_i} = (\lambda I + K)^{-1}(k^i - \frac{\alpha}{4} g_i) \tag{3.16}$$

(ii) To obtain the optimal kernel matrix K, Eq.(3.12) can be written as:

$$\begin{aligned} \underset{K}{\text{minimize}}\, & Tr\left(K - 2KS + S^T K S\right) \\ & + \beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} \| \ H^{(p,v)} - K\|_F^2 \\ & + \frac{\mu}{2} \|W - K + \frac{Y}{\mu}\|_F^2 \end{aligned} \tag{3.17}$$

From Eq.(3.17) the kernel matrix K is updated as:

$$K = \frac{2S^T - SS^T - I + 2\beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} H^{(p,v)} + \mu W + Y}{2\beta \sum_{v=1}^{q} \sum_{p=1}^{u} Z_{(p,v)} + \mu} \tag{3.18}$$

(iii) For updating W, Eq.(3.12) becomes:

$$\underset{W}{\text{minimize}} \ \ \zeta \|W\|_\star + \frac{\mu}{2} \|W - K + \frac{Y}{\mu}\|_F^2 \tag{3.19}$$

Let $B = \left(K - \frac{Y}{\mu}\right) \in \mathbb{R}^{n \times n}$ and $SVD(B) = U.diag(\sigma).V^T$. Now the optimal W can be obtained by:

$$W = U.diag\left(max(\sigma - \frac{\zeta}{\mu}, 0)\right).V^T \tag{3.20}$$

(iv) For Updating the label matrix P, the problem becomes:

$$\begin{aligned} \underset{P}{\text{minimize}} \quad & \alpha Tr(P^T L P) \\ \text{subject to} \quad & P^T P = I \end{aligned} \tag{3.21}$$

If there are m number of classes, then m eigenvectors of L corresponding to the m smallest eigenvalues will be the optimal solution of P.

The proposed algorithm for LRMVMKC framewrok has been stated in Algorithm 3.

Algorithm 3 : Proposed LRMVMKC algorithm

Input: Kernel for each view: $\{H^{(p,v)}\}$, parameters α, β, λ, ζ and μ.
Output: Similarity matrix S with exact m connected components and optimal kernel matrix K.
Initialization: S, K, $W = K$, $Z_{p,v}$ and Y
Repeat:

- update the i^{th} column of S as per Eq.(3.16)
- calculate K using Eq.(3.18)
- calculate W using Eq.(3.20)
- update $Z_{(p,v)}$ by Eq.(3.6)
- update Lagrangian multiplier Y as: $Y = Y + \mu(W - K)$
- calculate P by solving Eq.(3.21) as the m smallest eigenvectors, correspond to the m smallest eigenvalues of the Laplacian matrix L.

Until stopping criterion is met.

3.4 Experiment

To perform the proposed clustering task multiple kernels are needed. Twelve different kernels consist of seven Gaussian kernels, one linear kernel and four polynomial kernels are created for each view of a given dataset. Gaussian kernels are created by using: $K(y, z) = exp(-\| y-z\|_2^2/(td_{max}^2))$, where maximum distance between data samples is denoted by d_{max} and the values of t is chosen from the set: [0.01,0.05.0.1,1,10,50,100]. The 8^{th} The linear kernel is of the form: $K(y, z) = y^T z$. And the polynomial kernels are of the form: $K(y, z) = (a + y^T z)^b$ where a and b vary between {0, 1} and {2, 4}

respectively.

3.4.1 Dataset

To show the excellent clustering task performed by the LRMVMKC method, different real-world benchmark datasets are used on which the clustering task of the LRMVMKC framework is performed. Here, Animal with Attributes [37], MSRC-v1 [40], Yale [44], Reuters Multilingual Data [45] and Caltech101 [46] datasets are used. As the LRMVMKC algorithm based on multi-view, different multiple views are extracted from all the datasets. All the detailed information of the datasets and their views have been stated in section 1.5. And all the related information of the datasets also stated in Table 3.1.

Table 3.1: Statistics of the datasets used for the experiment

dataset	number of views	Instances	Classes
Animal with Attributes	3	100	10
MSRC-V1	3	210	7
Yale	3	165	15
REUTERS	5	180	6
Caltech101	6	241	5

3.4.2 Comparison Methods

The clustering performances of the proposed LRMVMKC framework on the above mentioned datasets have been compared with different existing multi-view methods and kernelized methods.

- **KKM** [8]: Kernel K-means method establishes the connection between the K-means and spectral clustering algorithm and then perform the clustering task using kernlized method thus incorporating the nonlinearity into the model. Here, the KKM method is considered as the baseline for clustering performances.
- **Co-train MVSC** [20]: In a co-training approach for multi-view spectral clus-

tering method, a graph is learned for each view and spectral clustering is performed on each graph. The clustering in one view helps to improve the performances of other views and vice-versa.

- **Co-reg MVSC** [21]: In a co-regularized multi-view spectral clustering approach, a cluster indicator matrix is learned where each sample belongs to the same cluster for each view and it is done by co-regularizing the different clustering hypothesis.
- **SwMC** [23]: In self-weighted multi-view clustering with multiple graphs method, a common Laplacian rank constrained graph is obtained for each and every view and the proper weight assignment to each view is done automatically.
- **AMVL** [63]: Auto-weighted multi-view learning method using multiple views and the weight assignment to each view is done automatically. It performs the clustering task and local structure learning task simultaneously and obtains a optimal graph that can be directly partitioned into different clusters.

3.5 Result

The clustering performances of the proposed LRMVMKC have been performed on different real-world datasets and they have been compared with other existing methods.

3.5.1 Performance Evaluation

Three evaluation metrics that have been used for evaluating the clustering performances of the proposed LRMVMKC framework are accuracy (ACC), Normalized Mutual Information (NMI) and Purity (PUR). The comparison of clustering performances of the proposed LRMVMKC method with respect to other existing multi-view and kernelized methods have been shown in Table 3.2. As clustering performances of KKM method is considered as the baseline here, the clustering performances of

Table 3.2: Clustering performance on different datasets for LRMVMKC framework

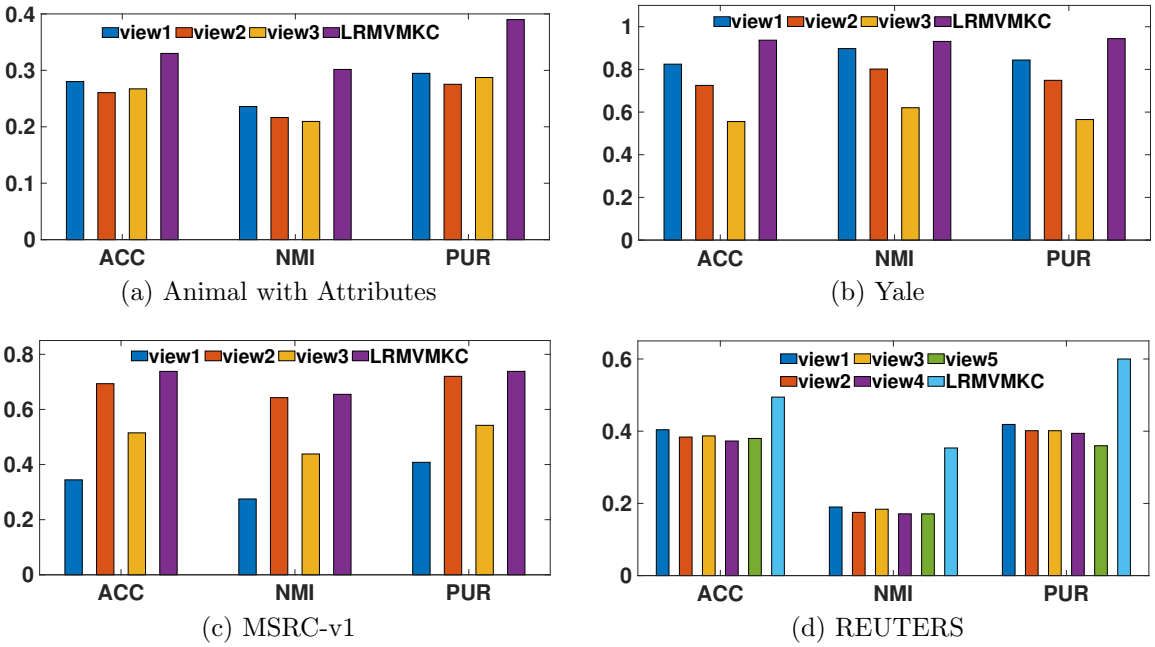
	Animal with Attributes			MSRC-v1			Yale			Reuters			Caltech-101		
	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
KKM(1)	0.2801	0.2360	0.2947	0.3445	0.2751	0.4081	0.8246	0.8975	0.8440	0.4040	0.1900	0.4187	0.4408	0.2123	0.4751
KKM(2)	0.2605	0.2164	0.2753	0.6934	0.6427	0.7201	0.7250	0.8017	0.7488	0.3837	0.1751	0.4013	0.3637	0.1313	0.4086
KKM(3)	0.2672	0.2095	0.2873	0.5150	0.4383	0.5423	0.5554	0.6204	0.5650	0.3869	0.1839	0.4013	0.5173	0.2608	0.5331
KKM(4)										0.3728	0.1711	0.3940	0.5539	0.4045	0.5836
KKM(5)										0.3799	0.1710	0.3597	0.6111	0.4485	0.6375
KKM(6)													0.5478	0.3211	0.5623
Co-train MVSC	0.2850	0.2534	0.3480	0.5924	0.5150	0.6433	0.9117	0.9378	0.9594	0.4167	0.2716	0.5111	0.6091	0.4488	0.6680
Co-reg MVSC	0.2570	0.2033	**0.5190**	0.4336	0.3753	0.5169	0.9145	0.9440	0.9678	0.4561	0.2857	0.5067	0.5855	0.4570	0.6502
MMSC	0.2787	0.2479	0.3677	0.6269	0.5951	0.6395	0.8751	0.9067	0.9437	0.4478	0.3170	0.4933	0.5232	0.4028	0.5245
SwMC	0.2200	0.1854	0.2200	0.6238	0.5669	0.6476	0.6850	0.7287	0.6700	0.2167	0.1152	0.2389	0.4979	0.2910	0.5104
AMVL	0.3040	0.2541	0.3110	0.7143	0.6957	0.7288	0.9631	0.9512	0.9674	0.4278	0.2794	0.4440	0.6266	0.4778	0.6473
LRMVMKC	**0.3300**	**0.3016**	0.3900	**0.7381**	**0.6548**	**0.7381**	**0.9800**	**0.9719**	**0.9800**	**0.4944**	**0.3535**	**0.6000**	**0.6349**	0.5218	**0.6929**

the LRMVMKC and the clustering performances of KKM method for each view of the dataset are plotted in Fig. 3.1. Also, the performances of LRMCMKC and that of others multi-view existing methods are plotted in Fig. 3.2. And from both the the figure Fig. 3.1 and Fig. 3.2, it is easily observed that in most of the cases the performances of the LRMVMKC are better than other existing methods. Now in Fig. 3.3 it has been shown how the learning performance gets affected while using different numbers of kernel and it is clearly observed from the figure that the learning performance is best while using 10 or 12 number of kernels and the accuracy of the performance gets reduced while using a very small number of kernel.

3.5.2 Convergence Analysis

One of the most important properties of an algorithm is its convergence. The convergence of the LRMVMKC algorithm is shown in this section. In section 3.2, it has been stated how the LRMVMKC framework has been built up and the mathematical representation of the LRMVMKC framework is stated in Eq. 3.10. But it is observed that Eq. 3.10 is not a jointly convex one. Therefore, the Eq. 3.10 is divided into four different sub-problems. Each sub-problem is stated in Eq. 3.13, 3.17, 3.19 and 3.21 respectively where each sub-problem depends only on one variable while others variables are considered as constant. These sub-problems are convex with respect to

the respected variable and optimal solution is obtained by solving each problem which converges to a global solution. It is also shown in Fig. 3.4 that for each dataset, the LRMVMKC method always converges after a certain number of iteration.



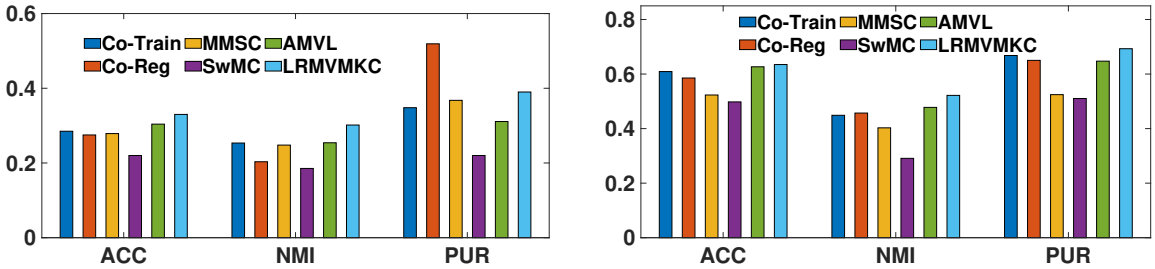
(a) Animal with Attributes (b) Yale

(c) MSRC-v1 (d) REUTERS

Figure 3.1: Clustering performance between LRMVMKC and KKM (uses only one view) on Animal with Attributes, Yale, MSRC-v1 and REUTERS dataset.

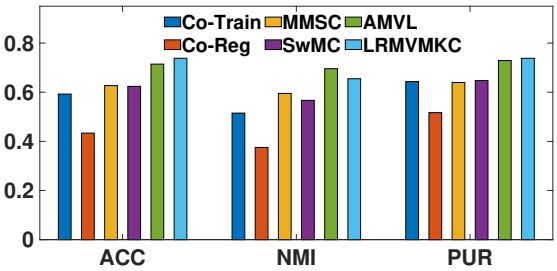
3.5.3 Parameter Tuning and Sensitivity

The mathematical representation of the LRMVMKC framework is stated in Eq. (3.10). To solve this problem, Lagrangian function is incorporated as per Eq. (3.12). From Eq. (3.12) it is easy to see that LRMVMKC algorithm contains five regularization parameters: α, β, γ, μ and ζ. Different values of these parameters may effect the learning performances of LRMVMKC framework differently.In order to find the appropriate values of those regularization parameters for which the algorithm will give the best learning performances, a grid search is performed where, $\alpha = [1e{-}5, 1e{-}4, 1e{-}3]$, $\beta = [10, 1000]$, $\gamma \in [1, 100]$, $\zeta = [0.01, 0.1, 1]$ and $\mu = [10, 100]$.

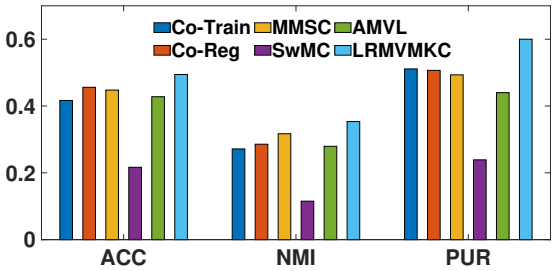


(a) Animal with Attributes

(b) Caltech101



(c) MSRC-v1



(d) REUTERS

Figure 3.2: Clustering performance between LRMVMKC and MVSC (uses multiple views) framework on Animal with Attributes, Caltech101, MSRC-v1 and REUTERS dataset.

In Fig. 3.5, clustering performances of LRMVMKC is shown for different values of α and ζ when $\beta = 10, \gamma = 1$ and $\mu = 10$. In Fig. 3.6, clustering performances of LRMVMKC is shown for different values of α and ζ when $\beta = 10, \gamma = 100$ and $\mu = 100$. From Fig. 3.5 and Fig. 3.6, it is observed that the clustering performances doesn't change much even if the values of γ and μ changes. In Fig. 3.7, clustering performances of LRMVMKC is shown for different values of α and ζ when $\beta = 1000, \gamma = 1$ and $\mu = 10$ and in Fig. 3.8, clustering performances of LRMVMKC is shown for different values of α and ζ when $\beta = 1000, \gamma = 100$ and $\mu = 100$. So from Fig. 3.5, Fig. 3.6, Fig. 3.7 and Fig. 3.8, it is also observed that the clustering performances of LRMVMKC doesn't change much for different values of β when γ and μ are kept constant. So from all these figures it can be concluded that the performances of the LRMVMKC framework is very less sensitive to those regularization parameters when their values are within a certain range.

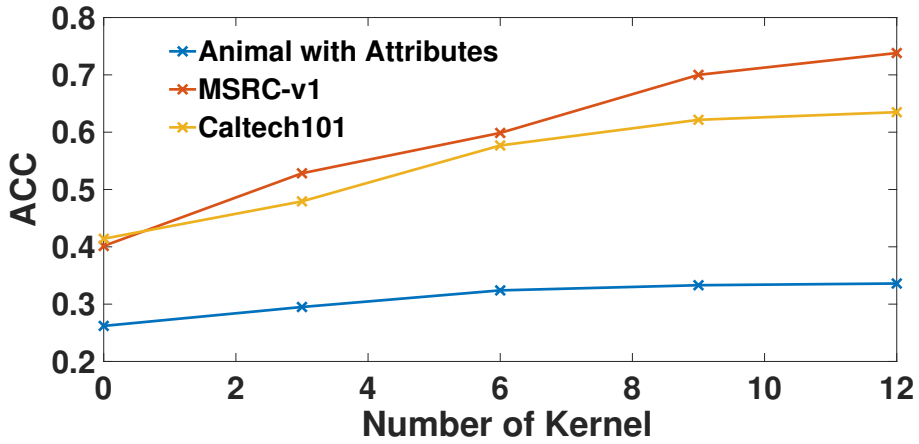
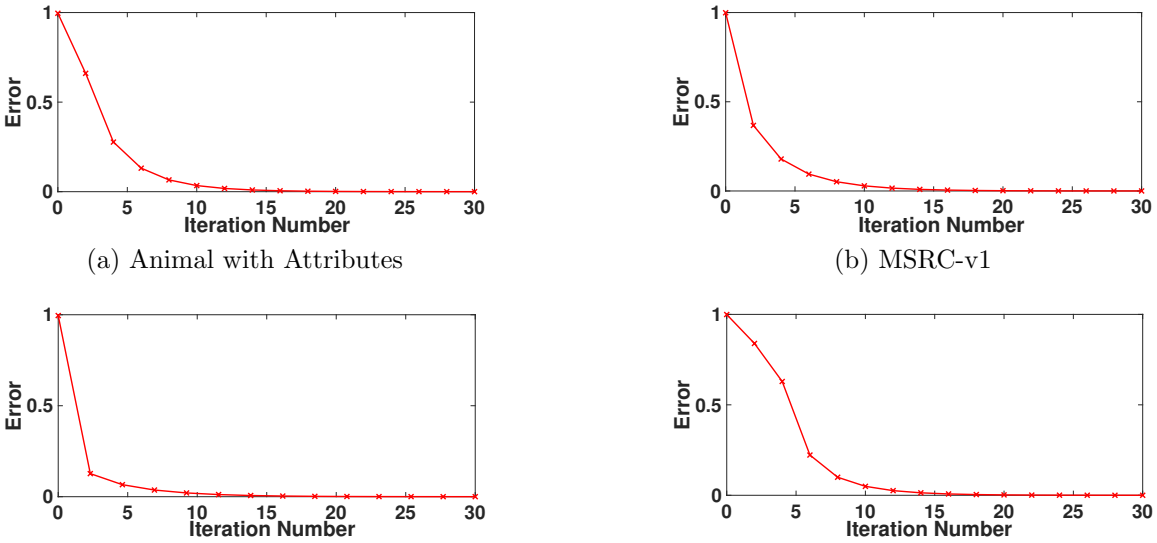


Figure 3.3: Change of the performance of LRMVMKC while performing the clustering and semi-supervised classification task on different data sets.



(a) Animal with Attributes

(b) MSRC-v1

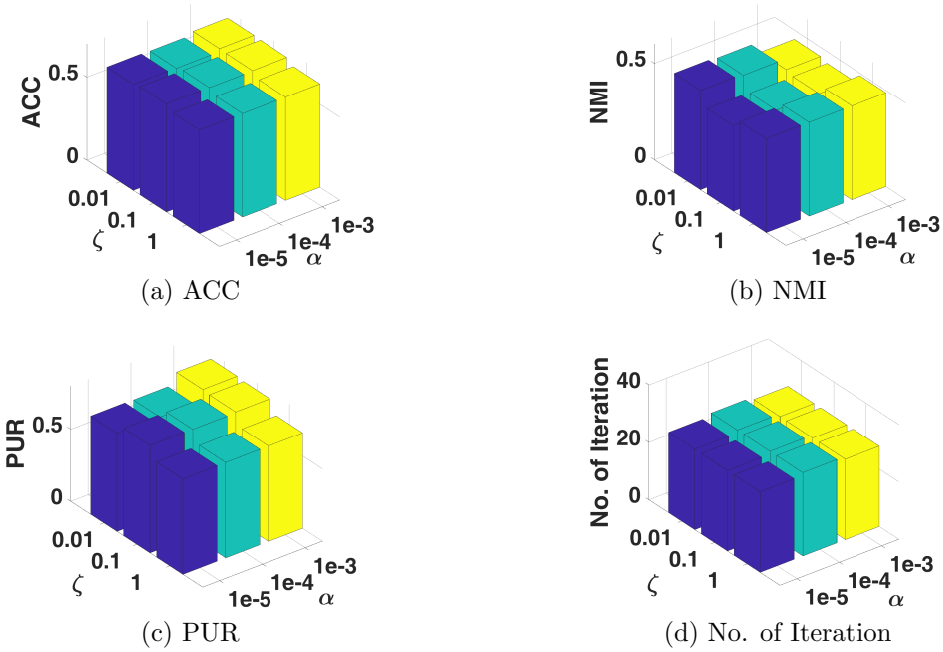
(c) REUTERS

(d) Caltech101

Figure 3.4: Clustering convergence of LRMVMKC framework on Animal with Attributes, MSRC-v1, Reuters and Caltech101 dataset.

3.5.4 Computational Complexity

The computational complexity of the LRMVMKC framework is discussed here. If a dataset has n number of samples, then the computational complexity of kernel construction for a view is $\mathcal{O}(n^2)$. Now if the given dataset has v number of views

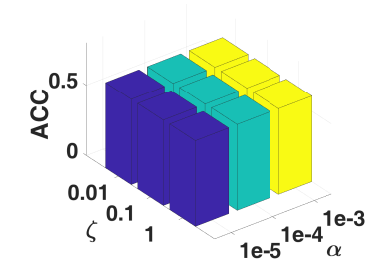


(a) ACC (b) NMI

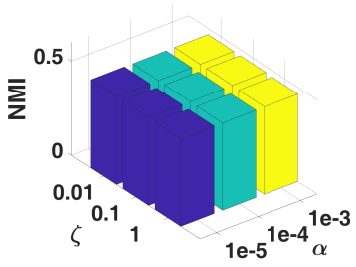
(c) PUR (d) No. of Iteration

Figure 3.5: Clustering performance of LRMVMKC framework on Caltech dataset for different values of α and ζ while $\beta = 10$, $\gamma = 1$ and $\mu = 10$.

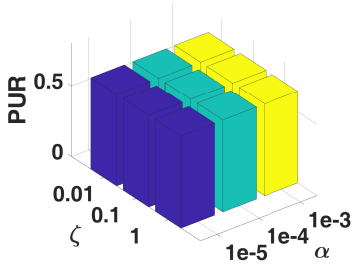
and p number of kernel for each view then the total complexity of kernel construction is $\mathcal{O}(n^2pv)$. The similarity matrix S and the kernel matrix K are updated as per Eq. 3.16 and Eq. 3.18 and the complexity of these are $\mathcal{O}(n^3)$. To update W as per Eq. 3.20, a SVD is performed and the complexity of SVD is $\mathcal{O}(n^3)$. The coplexity of updation of Y is $\mathcal{O}(n^2)$. But generally in real world datasets, we have $pv << n$. Therefor, overall complexity of the Algorithm 3 is $\mathcal{O}(n^3)$.



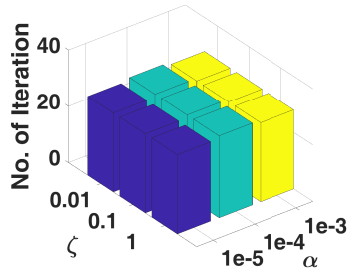
(a) ACC



(b) NMI

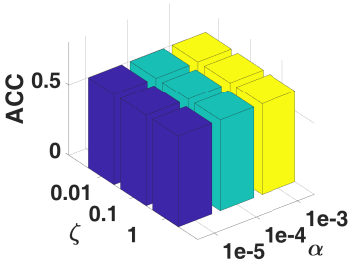


(c) PUR

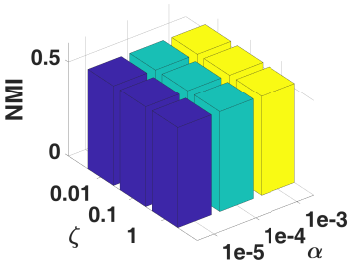


(d) No. of Iteration

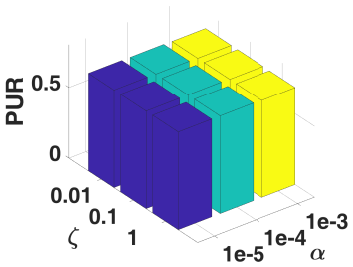
Figure 3.6: Clustering performance of LRMVMKC framework on Caltech dataset for different values of α and ζ while $\beta = 10$, $\gamma = 100$ and $\mu = 100$.



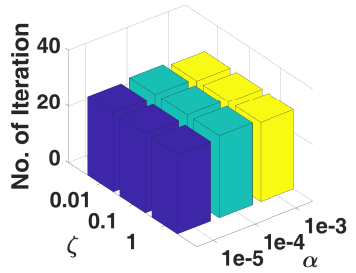
(a) ACC



(b) NMI

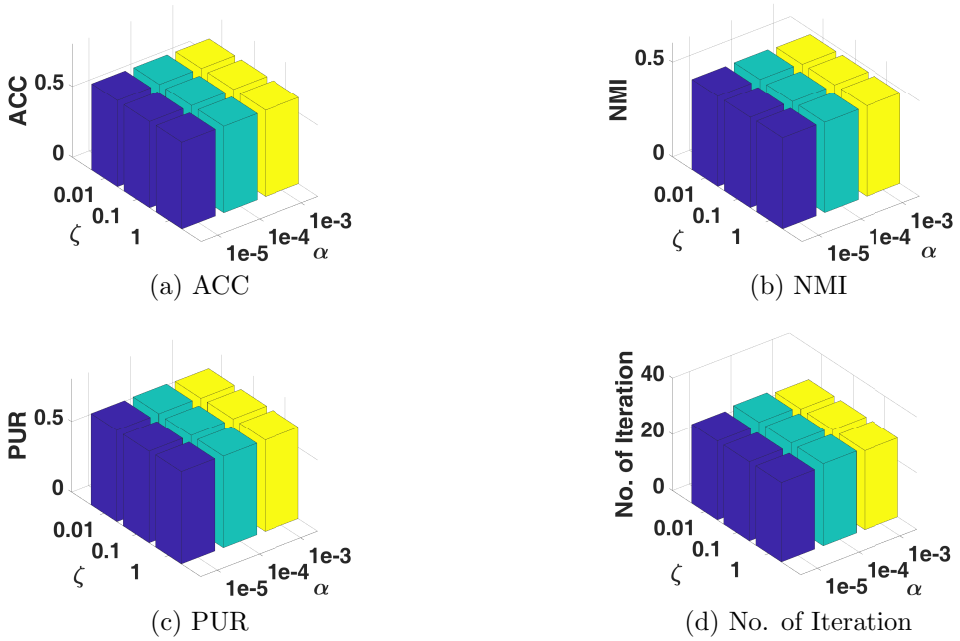


(c) PUR



(d) No. of Iteration

Figure 3.7: Clustering performance of LRMVMKC framework on Caltech dataset for different values of α and ζ while $\beta = 1000$, $\gamma = 1$ and $\mu = 10$.



(a) ACC (b) NMI

(c) PUR (d) No. of Iteration

Figure 3.8: Clustering performance of LRMVMKC framework on Caltech dataset for different values of α and ζ while $\beta = 1000$, $\gamma = 100$ and $\mu = 100$.

3.6 Summary

In this chapter a novel low-rank multi-view multi-kernel graph-based clustering framework is described which performs the clustering task based on the graph representation of the dataset by using multiple kernels on multiple views. One of the important task of this method is to assign proper weight to each kernel of each view. The weight assignment to each kernel of each view is also done automatically without using any additional weight assignment parameter. Then the optimal kernel is learned by low-rank kernel minimization method that makes the proposed method less sensitive to the noise present in the datasets. Various extensive experiments on different real-world datasets show the effective and better performances of the proposed LR-MVMKC framework than the performances of other existing methods.

Chapter 4

Kernelized Graph-based Multi-view Learning on High Dimensional Data

4.1 Introduction

There exists a large amount of data with high dimension and processing these high dimensional data is a challenge. The high dimensional data carries a lot of noise features which is detrimental for the clustering task. To solve this, some methods are proposed in [64], [65]. But in these methods, the nonlinear relationships between different data samples that may be present in the dataset are not considered. Also all these methods use only a single view of a dataset. But In many applications such as web page classification, video classification, surveillance systems, data and information are collected using multiple views. Each view provides different partial information about the dataset and a combination of these partial information helps to improve the clustering performances. In [66], multiple views have been used for the clustering of high dimensional data but here also the nonlinear relationships between

different data samples are not considered.

One of the ways to deal with the redundant features present in high dimensional dataset is to reduce the dimensionality of the data. There are various methods for dimensionality reduction [67], [68]. But these methods also don't consider the non-linear relationship between different data samples. And this issue is solved by kernel principal component analysis (KPCA) [69].

A novel multi-view kernelized graph-based clustering on high dimensional data (MVKGC) framework is presented in this chapter which performs the clustering task on the high dimensional dataset and simultaneously reduces the dimensionality of the dataset thus preventing the clustering performances to get affected from the redundant features present in the dataset. The main contribution of this chapter is the application of the kernel method while performing clustering task and dimension reduction on high dimensional data which enhances the clustering performances. In summary, the important contributions of this work are as follows:

- a novel kernelized graph-based clustering for high dimensional data using dimensionality reduction technique.
- use of multiple views and automatic weight assignments to each view as per their contribution.
- integration of dimensionality reduction, graph construction, kernel learning and label learning.

4.2 Methodology

To incorporate multiple views in the framework, multiple feature sets are extracted from the given dataset and each feature set is considered as a view. As the dimension of the data is high, the dimension reduction task is performed for each view by KPCA method which also includes the kernel trick into the proposed framework. The

proper weight assignment to a particular view is also learned automatically without introducing any weight assignment parameter.

4.2.1 Kernelized Graph-based Clustering for High Dimensional Data

There is a given dataset $X \in \mathbb{R}^{d \times n}$ with m number of clusters where n is the total number of samples in the dataset and d is the feature dimension. If S is the initially given similarity matrix for the given dataset where $S = S^T \in \mathbb{R}^{n \times n}$ then the optimal similarity matrix can be learnt from the following minimization problem:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \|X - XS\|_F^2 + \lambda \|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0. \end{aligned} \tag{4.1}$$

where λ is a regularization parameter. Now if X is a high dimensional dataset, then there may exist many redundant features in the dataset which may degrade the learning performance. To solve this issue, KPCA [69] is incorporated in the proposed framework. By using KPCA, the high dimensional data X can be represented in a low dimensional space as $\hat{X}$ and it can be written as:

$$\hat{X} = W^T K \tag{4.2}$$

where, $K \in \mathbb{R}^{n \times n}$, $\hat{X} \in \mathbb{R}^{r \times n}$ and r ($r \ll d$) is the reduced dimension of X. K is the kernel matrix of the given dataset X and it can obtained by using the kernel trick $K(x, y) = \phi(x)^T \phi(y)$. $W \in \mathbb{R}^{n \times r}$ is the top r eigenvectors of K. Now using Eq. (4.2), Eq. (4.1) can be rewritten as:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \|W^T K - W^T K S\|_F^2 + \lambda \|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0. \end{aligned} \tag{4.3}$$

4.2.2 Multi-view Kernelized Graph-based Clustering for High Dimensional Data

If the given dataset X has multiple views denoted by $X_1, X_2, \cdots, X_v, \cdots X_q$, where $X_v \in \mathbb{R}^{d^v \times n}$ and total number of available data samples and the feature dimension of v^{th} view are denoted by n and d^v respectively. $K_v \in \mathbb{R}^{n \times n}$ is the kernel matrix for the v^{th} view, X_v. Now using multiple views instead of single view, Eq. (4.3) can be expressed as:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \sum_{v=1}^{q} (h_v)^{\gamma} \|W_v^T K_v - W_v^T K_v S\|_F^2 + \lambda \|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0, \;\; \hat{h_v^T} \mathbf{1} = 1, \;\; 0 \leqslant h_v \leqslant 1 \end{aligned} \tag{4.4}$$

where,$W_v \in \mathbb{R}^{r \times n}$ is the top l eigenvectors of K_v, h_v is the weight that has been assigned to the v^{th} view X_v and $\hat{h_v} = [h_1, h_2, \cdots, h_v, \cdots, h_q]^T \in \mathbb{R}^{q \times 1}$. There exists an extra parameter γ which is required for the smooth weight assignments to each view. But to make the weight assignment a parameter free method, a different framework of the following form has been proposed.

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \sum_{v=1}^{q} \|W_v^T K_v - W_v^T K_v S\|_F + \lambda \|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0 \end{aligned} \tag{4.5}$$

Where, no weight parameter ha been defined explicitly. The Lagrangian function of Eq. (4.5) can be expressed as:

$$\underset{S}{\text{minimize}} \quad \sum_{v=1}^{q} \|W_v^T K_v - W_v^T K_v S\|_F + \lambda \|S\|_F^2 + \mathcal{L}(\Lambda, S) \tag{4.6}$$

where $\mathcal{L}(\Lambda, S)$ has been obtained from the constraint term and Λ is the Lagrange multiplier. Now by equating the first derivative of Eq. (4.6) with respect to S to

zero:

$$\sum_{v=1}^{q} Z_v \frac{\delta \|W_v^T K_v - W_v^T K_v S\|_F^2}{\delta S} + \lambda \frac{\delta \|S\|_F^2}{\delta S} + \frac{\delta \mathcal{L}(\Lambda, S)}{\delta S} = 0 \tag{4.7}$$

Where,

$$Z_v = \frac{1}{2\|K - W_v^T K_v S\|_F} \tag{4.8}$$

As Z_v depends on S, direct solution of Eq. (4.7) is not possible. But when Z_v is considered as stationary then Eq. (4.7) can be written as:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \sum_{v=1}^{q} Z_v \|W_v^T K_v - W_v^T K_v S\|_F^2 + \lambda \|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0. \end{aligned} \tag{4.9}$$

In spectral analysis, $L = \left(D - \frac{S+S^T}{2}\right) \in \mathbb{R}^{n \times n}$ is called as Laplacian matrix where $D \in \mathbb{R}^{n \times n}$ is called degree matrix. D is a diagonal matrix and its i^{th} diagonal element is $\frac{\sum_j (s_{ij} + s_{ji})}{2}$. The similarity matrix S follows an important property [57]:

"The multiplicity m of the eigenvalue 0 of the Laplacian matrix L is equal to the number of connected components in the graph associated with S."

As the given dataset has m number of clusters, then the similarity matrix S will have m number of connected components. So the above mentioned property provides a constraint on S and that constraint is $rank(\mathrm{L}) = n - m$. Now Eq. (4.9) can be rewritten as:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \sum_{v=1}^{q} Z_v \|W_v^T K_v - W_v^T K_v S\|_F^2 + \lambda \|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0, rank(L) = n - m \end{aligned} \tag{4.10}$$

If $\sigma_i(\mathrm{L})$ is the i^{th} smallest eigenvalues of Laplacian matrix L then $\sigma_i(\mathrm{L}) \geqslant 0$ as L is positive semi-definite matrix. Therefore $\sum_{i=1}^{m} \sigma_i(L) = 0$ will guarantee the constraint $rank(\mathrm{L}) = n - m$. As per Ky Fan's Theorem [58]:

$$\min_{P^T P = I} \quad Tr(P^T L P) = \sum_{i=1}^{m} \sigma_i(L) \tag{4.11}$$

where, $P \in \mathbb{R}^{n \times m}$ is a label indicator matrix. Now Eq. (4.10) can be written as:

$$\begin{aligned} \underset{S,P}{\text{minimize}} \quad & \sum_{v=1}^{q} Z_v \|W_v^T K_v - W_v^T K_v S\|_F^2 + \lambda \|S\|_F^2 \\ & + \alpha Tr(P^T L P) \\ \text{subject to} \quad & S \geqslant 0, \quad P^T P = I \end{aligned} \tag{4.12}$$

where, α is a regularization parameter. We know that for any matrix N, $\|N\|_F^2 = Tr(N^T N)$. Using this Eq. (4.12) can be expressed as:

$$\begin{aligned} \underset{S,P}{\text{minimize}} \quad & \sum_{v=1}^{q} Z_v Tr\left(M^{(v)} - 2M^{(v)} S + S^T M^{(v)} S\right) \\ & + \lambda \|S\|_F^2 + \alpha Tr\left(P^T L P\right) \\ \text{subject to} \quad & S \geqslant 0, \quad P^T P = I \end{aligned} \tag{4.13}$$

where,

$$M^{(v)} = \left[\mathrm{m}_1^{(\mathrm{v})}, \mathrm{m}_2^{(\mathrm{v})}, \cdots \mathrm{m}_\mathrm{n}^{(\mathrm{v})}\right] = \left(\mathrm{W}_\mathrm{v}^\mathrm{T} \mathrm{K}_\mathrm{v}\right)^\mathrm{T} \left(\mathrm{W}_\mathrm{v}^\mathrm{T} \mathrm{K}_\mathrm{v}\right) \in \mathbb{R}^{\mathrm{n} \times \mathrm{n}} \tag{4.14}$$

$$Z_v = \frac{1}{2\sqrt{Tr\left(M^{(v)} - 2M^{(v)} S + S^T M^{(v)} S\right)}} \tag{4.15}$$

The proposed MVKGC framework is formulated by the above mentioned minimizing problem.

4.3 Optimization

An iterative algorithm is presented in this section to solve the optimization problem stated in Eq. (4.13). In this iterative procedure, one variable is updated while keeping other variable constant. The algorithm has two parts: clustering and semi supervised classification.

4.3.1 Clustering

For simplicity, the problem defined in Eq. (4.13) can be divided into two sub-problems and those two sub-problems can be solved iteratively. Every sub-problem is solved with respect to only one variable while the other variable is considered as a constant.

(i) while updating S, P is kept fixed and Eq. (4.13) can be written as:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \sum_{v=1}^{q} Z_v Tr\left(-2M^{(v)}S + S^T M^{(v)} S\right) \\ & + \lambda \|S\|_F^2 + \alpha Tr\left(P^T L P\right) \\ \text{subject to} \quad & S \geqslant 0 \end{aligned} \tag{4.16}$$

An important equation of spectral analysis is:

$$\sum_{ij} \frac{1}{2} \|P_{i,:} - P_{j,:}\|_2^2 s_{ij} = Tr\left(P^T L P\right) \tag{4.17}$$

Now the problem defined in Eq.(4.16) can be solved for each i^{th} column (s_i) of the similarity matrix S as:

$$\sum_{v=1}^{q} Z_v \left(-2M_i^{(v)} s_i + s_i^T M^{(v)} s_i\right) + \lambda s_i^T s_i + \frac{\alpha}{2} g_i^T s_i \tag{4.18}$$

where $g_i = [g_{i1}, g_{i2}, \cdots, g_{in}]^T \in \mathbb{R}^n$ and $g_{ij} = \|P_{i,:} - P_{j,:}\|_2^2$. Now setting the first derivative of Eq. (4.18) with respect to s_i to be zero, we get:

$$\mathbf{s_i} = \left(\sum_{v=1}^{q} Z_v M^{(v)} + \lambda I\right)^{-1} \left(\sum_{v=1}^{q} Z_v m_i^{(v)} - \frac{\alpha}{4} g_i\right) \tag{4.19}$$

Finally, we get the updated similarity matrix as:

$$S = [s_1, s_2, \cdots, s_n] \tag{4.20}$$

(ii) To update of P while keeping S fixed, Eq. (4.13) can be written as:

$$\begin{aligned} \underset{P}{\text{minimize}} \quad & \alpha Tr\left(P^T L P\right) \\ \text{subject to} \quad & P^T P = I \end{aligned} \tag{4.21}$$

If there are m number of clusters, then the solution P is m eigenvectors of L corresponding to the m smallest eigenvalues.
At each iteration, after updating S and P, Z_v is updated as per Eq. (4.15).

4.3.2 Semi-supervised Classification

Graph construction and label inference are two important stages of semi-supervised classification in graph-based learning. Both these stages are unified with dimension reduction and kernel method in the MVKGC framework. Using the labelled samples, the graph is constructed and then by label inference, the unknown labels are predicted.

The MVKGC framework for semi-supervised classification is given as:

$$\begin{aligned} \underset{S,P}{\text{minimize}} \quad & \sum_{v=1}^{q} Z_v Tr\left(M^{(v)} - 2M^{(v)}S + S^T M^{(v)} S\right) \\ & + \lambda \|S\|_F^2 + \alpha Tr\left(P^T L P\right) \\ \text{subject to} \quad & S \geqslant 0, \quad P^T P = I, \quad P_l = Y_l \end{aligned} \tag{4.22}$$

where l is the total number of labelled points, and $Y_l = [y_1, y_2, y_3, \cdots, y_i, \cdots, y_l]^T \in \mathbb{R}^{l \times c}$ is the labelled indicator matrix where $y_i \in \mathbb{R}^c$ is the labelled indicator vector of the i^{th} sample. When $y_{ij} = 1$, the i^{th} sample belongs to the j^{th} class. Without loss of generality, the data points are divided in such a way that the first l data points are labelled and rest of the u data points are unlabelled, such that $l + u = n$ and n is the total number of data points. Now the Laplacian matrix L and and class indicator matrix P can be written as a block matrix.:

$$L = \begin{pmatrix} L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{pmatrix} \text{ and } \quad P = \begin{pmatrix} P_l \\ P_u \end{pmatrix} \tag{4.23}$$

where, P_l is the labelled indicator matrix and P_u is the unlabelled indicator matrix. Now the updated S from Eq. (4.22) can be obtained using the same iterative procedure that is used to get optimal S from Eq. (4.13), but the difference lies in updating P. The updated class indicator matrix P is obtained by solving the following minimization problem:

$$\begin{aligned} \underset{P}{\text{minimize}} \quad & Tr(P^T L P) \\ \text{subject to} \quad & P_l = Y_l \end{aligned} \tag{4.24}$$

By setting the first derivative of Eq. (4.24) with respect to P equal to zero, we get:

$$\begin{pmatrix} L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{pmatrix} \begin{pmatrix} Y_l \\ P_u \end{pmatrix} = 0 \tag{4.25}$$

By solving Eq. (4.25) we get,

$$P_u = -L_{uu}^{-1} L_{ul} Y_l \tag{4.26}$$

If ij^{th} element of P is denoted as p_{ij}, then the final class labels of the unlabelled data points can be assigned by the following decision rule:

$$y_i = \arg\max_j p_{ij} \tag{4.27}$$

$$\forall i = l+1, l+2, ..., n. \quad \forall j = 1, 2, 3, ..., c$$

Algorithm 4 : Proposed MVKGC framework

Input: Kernel matrices for each view: $\{K_v\}_{v=1}^{q}$, parameters α and λ, Reduced feature dimension r.
Output:
Clustering: Similarity matrix S with exact m connected components and label indicator matrix P.
Classification: The label matrix P for all data points.
Initialization: S, Z_v
Repeat:

- update the i^{th} column of S as per Eq. (4.19)
- calculate P by solving Eq. (4.21) as the m smallest eigenvectors, correspond to the m smallest eigenvalues of the Laplacian matrix L.
- update Z_v by Eq. (4.15)

Until stopping criterion is met.

- Classification: assign the class label to the unlabelled points by Eq. (4.27)

4.4 Experiment

Several experiments are performed on the MVKGC framework to validate its clustering and semi-supervised classification performances. As it it is a multi-view based learning framework, different multiple features are extracted from the dataset at first. Then a kernel is assigned to each and every view. Here for the experiment a Gaussian kernel of the form: $K(y, z) = exp(-\| y - z\|_2^2/(d_{max}^2))$, where the maximum distance between samples is d_{max}, is assigned to each view. The semi-supervised classification performances of MVKGC is validated when different percentages of labelled data is available. For the experiment, 10%, 30%, and 50% of labelled data are considered.

4.4.1 Dataset

To show the excellent learning task performed by the MVKGC method, different real-world benchmark datasets are used on which the learning tasks of the MVKGC framework are performed. Here, Animal with Attributes [37], MSRC-v1 [40] and Caltech101 [46] datasets are used. From these datasets, multiple features which are high dimensional are extracted. All the details information of the datasets is already

Table 4.1: Statistics of the datasets used for the experiment

dataset	**Animal with Attributes**	**MSRC-v1**	**Caltech101**
Feature (Dimension)	Color Histogram (2688)	GIST (512)	CENTRIST (254)
	Local Self Similarity (2000)	HOG (35100)	HOG (1984)
	PyramidHOG (252)		GIST (512)
			LBP (928)
No. of Views	3	2	4
Instances	100	210	241
No. of Class	10	7	5

stated in section 1.5. And all the related feature information information of the datasets is also stated in Table 4.1.

4.4.2 Comparison Methods

The learning performances of the proposed MVKGC framework on the above mentioned datasets have been compared with different existing methods.

- **KKM** [8]: Kernel K-means method establishes the connection between the K-means and spectral clustering algorithm and then perform the clustering task using kernlized method thus incorporating the nonlinearity into the model. Here, the KKM method is considered as the baseline for clustering performances.
- **MMSC** [41]: Multi-modal spectral clustering method learns a Laplacian matrix that is shared commonly by each and every modal of the dataset. Non-negative relaxation is also used to improve clustering performances.
- **DEKM** [70] Discriminatively embedded K-means embeds the synchronous learning of multiple discriminative subspaces into multi- view K-Means clustering to construct a unified framework, and adaptively control the inter co-ordinations between these subspaces simultaneously.

- **RDEKM** [71] Re-weighted discriminatively embedded K-means is an unsupervised optimization scheme, which utilizes iterative re-weighted least squares to solve least absolute residual and adaptively controls the distribution of multiple weights in a re-weighted manner only based on its own low-dimensional subspaces and a common clustering indicator matrix.
- **RMKMC** [19] Robust multi-view K-means clustering is a novel robust large-scale multi-view K-means clustering approach, which can be easily parallelized and performed on multi-core processors for big visual data clustering.
- **LGC** [25] Learning with local and global consistency is a basic semi-supervised classification method which designs a classifying function which is sufficiently smooth with respect to the intrinsic structure collectively revealed by known labeled and unlabeled points. It is a single view method.
- **MLAN** [27]: Multi-view learning with adaptive neighbor method learns an optimal graph for each view by learning the local structure of the graph and the optimal graph can be partitioned into specific clusters. Here also the ideal weight assignment to each view is done automatically.
- **AMGL** [22]: Auto-weighted multiple graph learning method learns an optimal graph for each view and automatic weight assignment to each view is also done. The objective function of AMGL for semi-supervised classification is a convex function thus obtaining the globally optimal result.

4.5 Results

To verify the effectiveness of the proposed MVKGC framework, both the clustering and semi-supervised classification task are conducted on different datasets and the results are promising.

Table 4.2: Clustering performances on different datasets for MVKGC framework

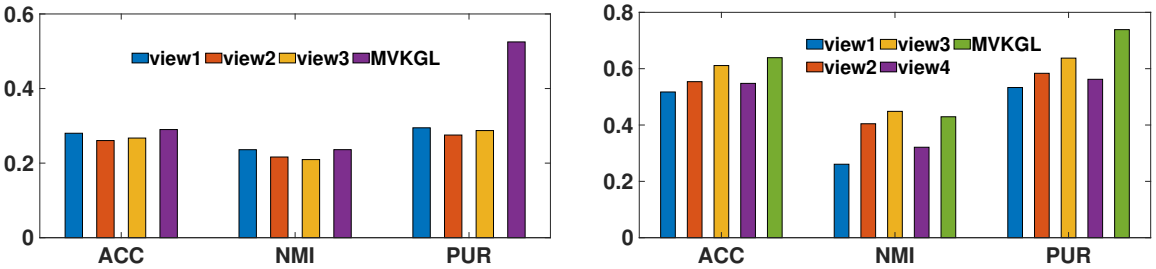
Methods	**MSRC-v1**			**Animal with Attributes**			**Caltech101**		
Evaluation Metric	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
KKM(1)	0.6974	0.6427	0.7201	0.2801	0.2360	0.2947	0.5173	0.2608	0.5331
KKM(2)	0.5150	0.4383	0.5423	0.2605	0.2164	0.2753	0.5539	0.4045	0.5836
KKM(3)				0.2672	0.2095	0.2873	0.6111	0.4485	0.6375
KKM(4)							0.5478	0.3211	0.5623
MMSC	0.5133	0.3554	0.5352	0.2190	0.1683	0.3847	0.4044	0.2295	0.4349
DEKM	0.5048	0.3687	0.5905	0.2100	0.1612	0.4856	0.5815	0.3481	0.7052
RDEKM	0.5619	0.4918	0.6439	0.2400	0.2017	0.3700	0.6017	0.3453	0.7055
RMKMC	0.6905	0.6293	0.7000	0.2517	0.2183	0.4000	0.5602	0.3804	0.6432
MVKGC	**0.7729**	**0.7070**	**0.8495**	**0.2900**	**0.2361**	**0.5950**	**0.6390**	**0.4293**	**0.7386**

Table 4.3: Semi-supervised classification performances on different datasets for MVKGC framework

dataset	**Animal with Attributes**			**MSRC-v1**			**Caltech101**		
Evaluation Metric	ACC			ACC			ACC		
Labelled Rate	0.1	0.3	0.5	0.1	0.3	0.5	0.1	0.3	0.5
LGC(1)	0.1839	0.2238	0.2413	0.7415	0.8174	0.8395	0.4770	0.5447	0.5675
LGC(2)	0.1849	0.2510	0.2811	0.5190	0.5952	0.6429	0.6222	0.6623	0.6870
LGC(3)	0.1682	0.1786	0.1849				0.6144	0.6451	0.6583
LGC(4)							0.5221	0.5535	0.5718
MLAN	0.2133	0.2571	0.2850	0.6508	0.7347	0.7810	0.6870	0.7295	0.7479
AMGL	0.1556	0.2057	0.2200	0.8071	0.8694	0.9095	0.6777	0.7297	0.7598
MVKGC	**0.2167**	**0.2629**	**0.3010**	**0.8135**	**0.8873**	**0.9229**	**0.6974**	**0.8116**	**0.8724**

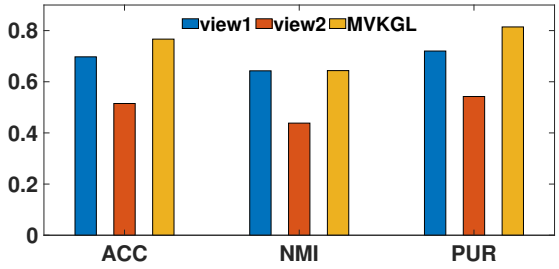
4.5.1 Performance Evaluation

Accuracy (ACC), Normalized Mutual Information (NMI) and Purity (PUR) are three evaluation metrics that are used for evaluating the performances of clustering and semi-supervised classification performances of the proposed MVKGC framework. The comparison of clustering performances and semi-supervised learning performances of the proposed MVKGC method with respect to other existing multi-view methods have been shown in Table 4.2 and Table 4.3 respectively. As clustering performances



(a) Animal with Attributes

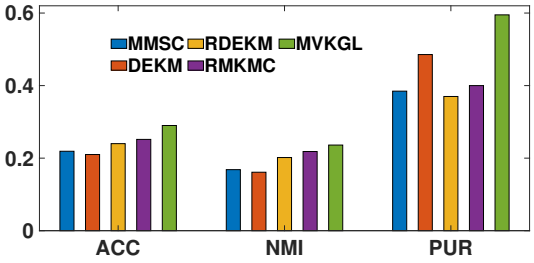
(b) Caletch101



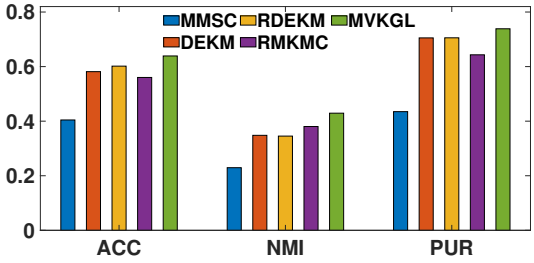
(c) MSRC-v1

Figure 4.1: Clustering performances between MVKGC and KKM (uses only one view) on Animal with Attributes, Caltech101 and MSRC-v1 dataset.

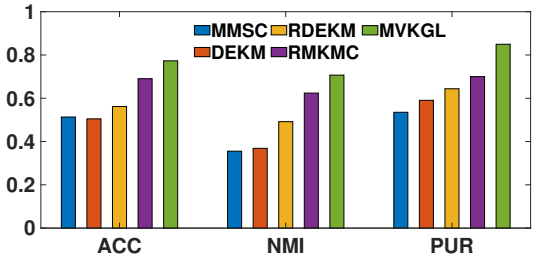
of KKM method is considered as the baseline here, the clustering performances of the MVKGC and the clustering performances of KKM method for each view of the dataset are plotted in Fig. 4.1. The clustering performances of the MVKGC framework and the clustering performances of some other existing multi-view methods are plotted in Fig. 4.2. It is easy to observe that the clustering performances of MVKGC framework is better than that of the KKM method and other multi-view methods. As the proposed MVKGC framework includes the dimension reduction of feature, the accuracy of its learning performances depends on the dimension of the selected features. It is observed from Fig. 4.3 and Fig. 4.4 that the clustering accuracy and semi-supervised classification accuracy starts to increase with the increasing dimension of feature but after a certain number of dimension the learning accuracy saturates or it starts to degrade.



(a) Animal with Attributes

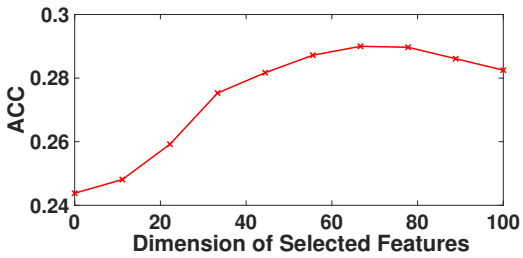


(b) Caletch101

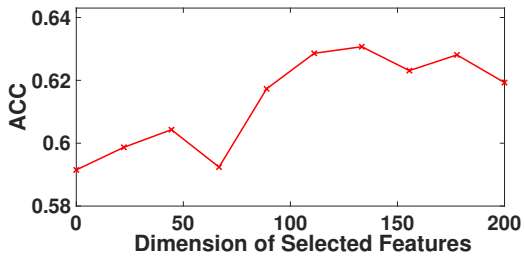


(c) MSRC-v1

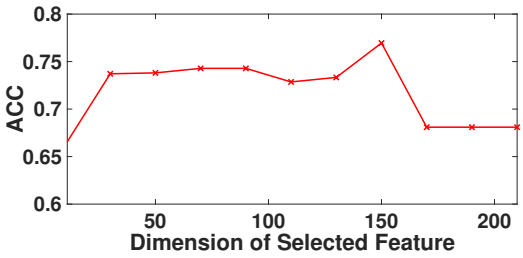
Figure 4.2: Clustering performances between MVKGC and different MVSC methods (uses multiple views) on Animal with Attributes, Caltech101 and MSRC-v1 dataset.



(a) Animal with Attributes

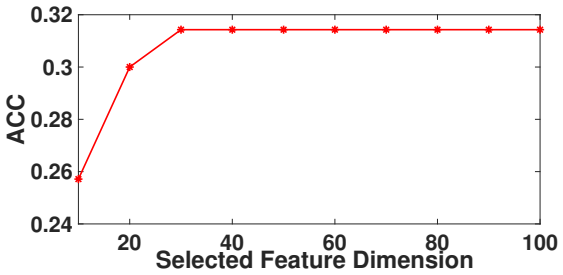


(b) Caletch101

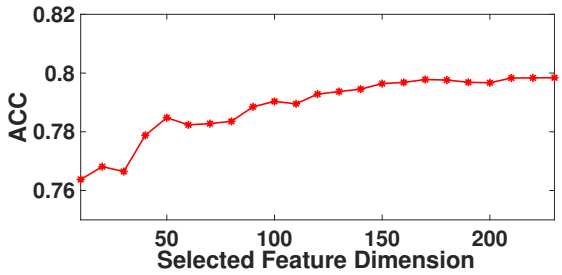


(c) MSRC-v1

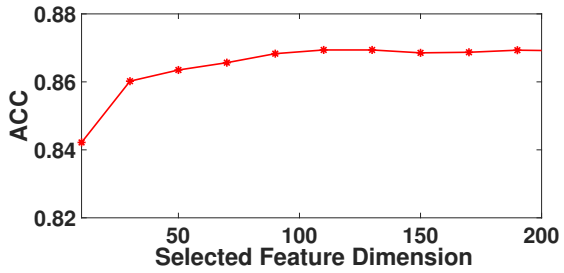
Figure 4.3: Clustering accuracy on Animal with Attributes, Caltech101 and MSRC-v1 datasets with different dimension of features.



(a) Animal with Attributes



(b) Caletch101

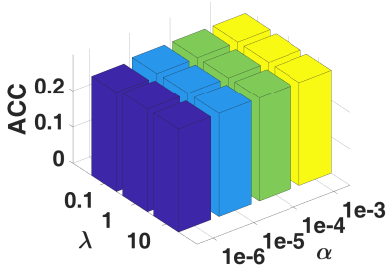


(c) MSRC-v1

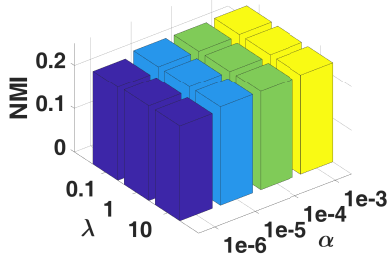
Figure 4.4: Semi-supervised classification accuracy on Animal with Attributes, Caltech101 and MSRC-v1 datasets with different dimension of features.

4.5.2 Parameter Tuning and Sensitivity

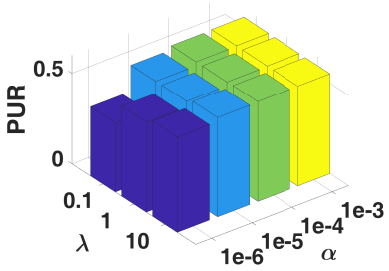
The proposed MVKGC algorithm contains two regularization parameters: α and λ. In order to find the right combinations of the regularization parameters for the algorithm to give the best performances, a grid search has been performed. The parameters are observed to lie in the range: $\alpha \in [1e-6, 1e-5, 1e-4, 1e-3]$ and $\lambda \in [0.1, 1, 10]$. The clustering performances and semi-supervised classification performances of the proposed MVKGC framework for different values of α and λ are shown in Fig. 4.5 and in Fig. 4.6 respectively. From both the figure it is easily observed that learning performances of MVKGC method is not much sensitive to different values of α and λ, when they are in the above mentioned range.



(a) ACC

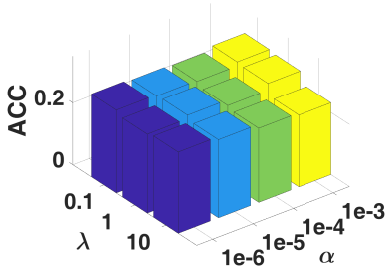


(b) NMI

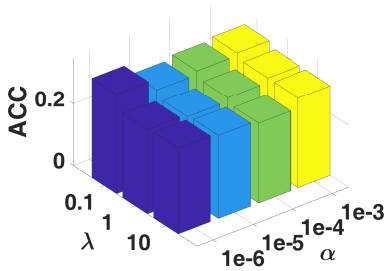


(c) PUR

Figure 4.5: Clustering performance of MVKGC framework on Animal with Attributes dataset for different values of α and λ.



(a) Labelled rate=30%



(b) Labelled rate=50%

Figure 4.6: Semi-supervised classification performance of MVKGC framework on Animal with Attributes dataset for different values of α and λ when 30% and 50% labelled data are available.

4.5.3 Convergence Analysis

The convergence of the proposed MVKGC algorithm is described in this section. The lemma, introduced in [72], has been used to prove the convergence of the proposed algorithm.

Lemma 4.1 *For any two positive numbers a and b, the following inequality holds:*

$$a - \frac{a^2}{2b} \leqslant b - \frac{b^2}{2b} \tag{4.28}$$

Theorem 1 In each iteration, the updated similarity matrix S will reduce the the objective value of the minimizing problem stated in (4.5) until convergence.

Proof: If $\widetilde{S}$ is the updated version of S at each iteration, then the following inequality can be written:

$$\begin{aligned} &\sum_v \frac{\|W_v^T K_v - W_v^T K_v \widetilde{S}\|_F^2}{2\|W_v^T K_v - W_v^T K_v S\|_F} + \|\widetilde{S}\|_F^2 \\ &\leqslant \sum_v \frac{\|W_v^T K_v - W_v^T K_v S\|_F^2}{2\|W_v^T K_v - W_v^T K_v S\|_F} + \|S\|_F^2 \end{aligned} \tag{4.29}$$

According to Lemma 4.1, the following can be written:

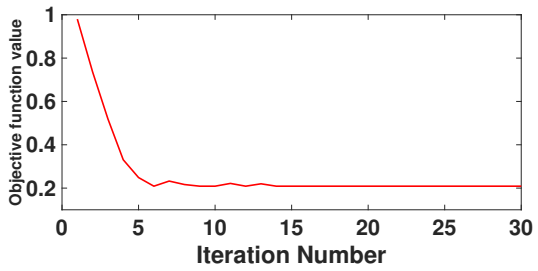
$$\begin{aligned} &\sum_v \|W_v^T K_v - W_v^T K_v \widetilde{S}\|_F - \sum_v \frac{\|W_v^T K_v - W_v^T K_v \widetilde{S}\|_F^2}{2\|W_v^T K_v - W_v^T K_v S\|_F} \\ &\leqslant \sum_v \|W_v^T K_v - W_v^T K_v S\|_F - \sum_v \frac{\|W_v^T K_v - W_v^T K_v S\|_F^2}{2\|W_v^T K_v - W_v^T K_v S\|_F} \end{aligned} \tag{4.30}$$

Now by adding (4.29) and (4.30) we get,

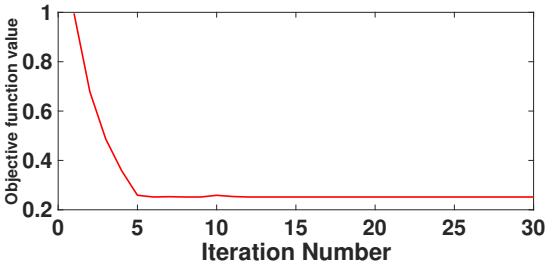
$$\begin{aligned} &\sum_v \|W_v^T K_v - W_v^T K_v \widetilde{S}\|_F + \|\widetilde{S}\|_F^2 \\ &\leqslant \sum_v \|W_v^T K_v - W_v^T K_v S\|_F + \|S\|_F^2 \end{aligned} \tag{4.31}$$

Thus, the proposed iterative algorithm will decrease the objective value of (4.5) monotonically.

It is shown in Fig. 4.7 and 4.9 that while running the proposed MVKGC algorithm, the value of the objective function is decreasing monotonically with each iteration while performing both the clustering and semi-supervised classification task.



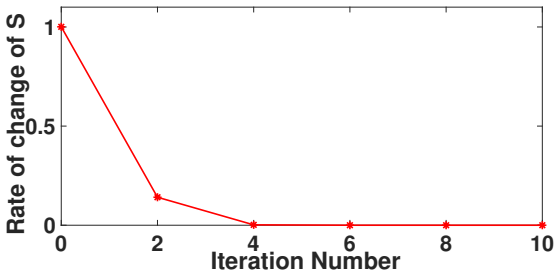
(a) MSRC-v1



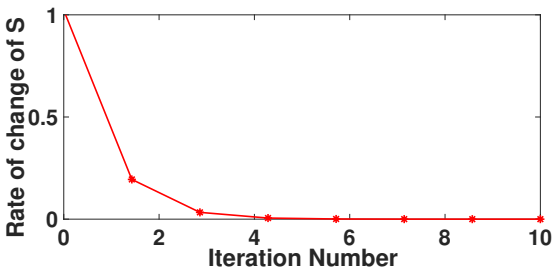
(b) Caletch101

Figure 4.7: Clustering convergence of MVKGC framework on MSRC-v1 and Caltech101 dataset.

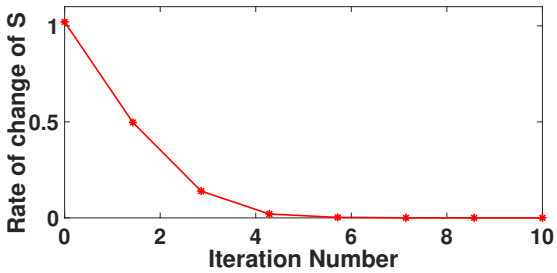
It is also shown in Fig. 4.8 and Fig. 4.10 that for both clustering and semi-supervised classification task, the rate of change of the simlilarity matrix (S) converges to a very small value after a certain number of iteration and after that even if the proposed algorithm is run, the performance remains almost same.



(a) Animal with Attributes

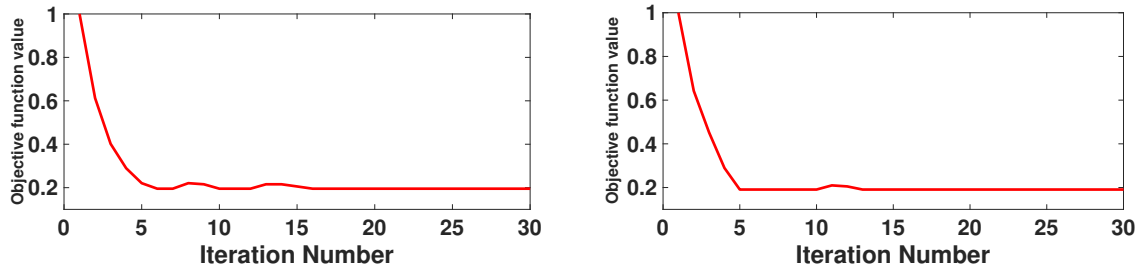


(b) Caletch101



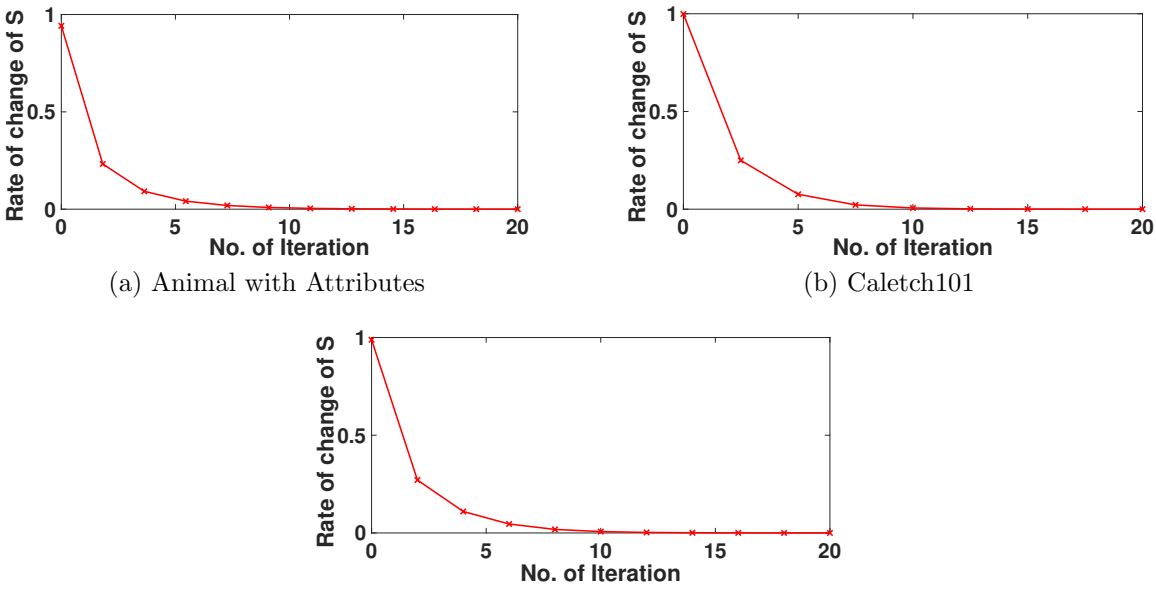
(c) MSRC-v1

Figure 4.8: Rate of change of similarity matrix for clustering task on different data sets.



(a) MSRC-v1 (b) Caletch101

Figure 4.9: Semi-supervised classification convergence of MVKGC framework on MSRC-v1 and Caltech101 dataset.



(a) Animal with Attributes (b) Caletch101

(c) MSRC-v1

Figure 4.10: Rate of change of similarity matrix for semi-supervised classification task on different data sets.

4.6 Summary

It is a challenging task to deal with high dimensional data along with multiple views. There may exist many redundant feature in high dimensional data which may affect the performances of the learning task. So it is important to get rid of those redundant features to achieve a better learning performances. This work proposes

a novel MVKGC framework that uses kernel principal component analysis (KPCA) to reduce the dimension of the high dimensional feature, thus reducing the effect of redundant feature in the learning task and also the kernel method considers the nonlinearity present in the dataset, thus improving the learning performances. The proposed MVKGC framework also uses multiple views where each view gives different partial information and combining them a better learning performances is achieved. One of the most important issues while using multiple views is proper weight assignment to each view according to their importance. It is also done automatically in the proposed method without introducing any extra weight assignment parameter. The extensive experiments on different real-world benchmark datasets show the efficient performances of the proposed MVKGC framework.

Chapter 5

Conclusion and Future Works

5.1 Conclusion

In this thesis, four novel graph-based learning frameworks have been proposed. One of the difficulties in graph-based learning is to use multiple views and multiple kernel together and their proper weight assignment. In chapter 2, a novel graph-based learning framework named SMVMKL has been proposed which incorporates the use of both the multiple views and multiple kernel and also the proper weight is assigned to each kernel of each view. And it is easily observed from various experiments on different datasets that the performance of the proposed SMVMKL framework is better than other existing methods that use only a single view. But SMVMKL framework suffers from the noise that may present in the data. To solve this issue, two different frameworks have been proposed. In chapter 2, one robust framework named RSMVMKL framework has been presented which uses $l_{2,1}$ norm to get rid of the noise present in the data. To improve the learning performance, both the use of multiple views and multiple kernel has been incorporated in the proposed framework and the proper weight assignment is done automatically. And it is observed from several experiments that the performance of RSMVMKL is better than the SMVMKL framework. In

chapter 3, another robust multi-view multiple kernel framework named LRMVMKC framework has been presented where instead of using $l_{2,1}$ norm, a low-rank kernel optimization method has been incorporated into the framework to overcome the issue of the noisy data. And the performance of LRMVMKC framework is even better than that of RSMVMKL. In chapter 4, a graph-based learning framework for high dimensional data has been explained. Presence of redundant features in high dimensional data affects the learning the performance of graph-based framework. To solve this issue of curse of dimensionality, a novel graph-based learning framework for high dimensional data named MVKGC has been proposed which incorporates the KPCA method into the framework thus improving the learning performance.

5.2 Future Works

In the proposed MVKGC framework, only a single kernel has been used. Later, the performance can be further improved by using multiple kernel instead of single kernel and the framework can also be made more robust to the noise.

Computational complexity of SMVMKL and RSMVMKL framework is $\mathcal{O}(n^3)$. If the dataset gets larger then the time required for the computation gets higher. So finding a way to reduce the computation time is a concern.

These works can also be shifted in graph neural network where neural network structure is applied on the graph structure of the data.

Appendix A

A.1 Incorporation of kernel trick

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \|\phi(X) - \phi(X)S\|_F^2 + \lambda\|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0. \end{aligned}$$

We know that, if $A \in \mathbb{R}^{m \times n}$ is a real valued matrix then $\|A\|_F^2 = Tr(A^T A)$. Using this relationship we can write:

$$\begin{aligned} \|\phi(X) - \phi(X)S\|_F^2 &= Tr\Big[\big(\phi(X) - \phi(X)S\big)^T\big(\phi(X) - \phi(X)S\big)\Big] \\ &= Tr\Big[\phi^T(X)\phi(X) - \phi^T(X)\phi(X)S - S^T\phi^T(X)\phi(X) + S^T\phi^T(X)\phi(X)S\Big] \\ &= Tr\Big[K - 2KS + S^T KS\Big] \end{aligned}$$

Now putting this value in the above equation we get,

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & Tr(K - 2KS + S^T KS) + \lambda\|S\|_F^2 \\ \text{subject to} \quad & S \geqslant 0. \end{aligned}$$

A.2 Self-weighted kernel learning algorithm from multiple kernel of multiple views

If we have total q number of views and for each view we have u number of kernel then we can learn the optimal kernel matrix as following:

$$\begin{aligned} \underset{K}{\text{minimize}} \quad & \sum_{v=1}^{q}\sum_{p=1}^{u} w_{(p,v)} \| H^{(p,v)} - K\|_F^2 + \gamma \| W\|_2^2 \\ \text{subject to} \quad & w_{(p,v)} \geq 0, (\mathbf{1_u})^T W (\mathbf{1_q}) = 1 \end{aligned}$$

where, $W \in \mathbb{R}^{\mathrm{u}\times\mathrm{q}}$ and the $(p,v)^{th}$ element of W is $w_{p,v}$ and $\mathbf{1_r}$ is a vector of size r with all of its element being 1.

So, the value of $w_{(p,v)}$ is heavily dependent on the value of γ. To get rid of this parameter, we follow [1] and present a formulation that induces a self-conducted weight learning. The proposed objective is:

$$\underset{K}{\text{minimize}} \quad \sum_{v=1}^{q}\sum_{p=1}^{u} \| H^{(p,v)} - K\|_F$$

Now, taking the derivative of this equation with respect to K and setting it to be zero:

$$\sum_{v=1}^{q} Z_{(p,v)} \frac{\delta H^{(p,v)} - K\|_F^2}{\delta K} = 0$$

where,

$$Z_{(p,v)} = \frac{1}{2\| H^{(p,v)} - K\|_F}$$

It is obvious that $w_{(p,v)}$ is dependent on K. But if we set $w_{(p,v)}$ stationary then we can write the minimization as following:

$$\underset{K}{\text{minimize}} \quad \sum_{v=1}^{q}\sum_{p=1}^{u} Z_{(p,v)} \| H^{(p,v)} - K\|_F^2$$

A.3 Cost function development of RSMVMKL

Given $X \in \mathbb{R}^{d \times n}$ and $S \in \mathbb{R}^{n \times n}$, we can write:

$$\left\| X - XS \right\|_{2,1} = \sum_{i=1}^{n} \left\| (X - XS)_i \right\|_2$$

where, $(X - XS)_i$ is the i^{th} column of $(X - XS)$.
Now assume that,

$$X = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \end{bmatrix} \quad \text{and} \quad S = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix}$$

Now we can write:

$$(X - XS) = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \end{bmatrix} - \begin{bmatrix} (x_{11}s_{11} + x_{12}s_{21}) & (x_{11}s_{12} + x_{12}s_{22}) \\ (x_{21}s_{11} + x_{22}s_{21}) & (x_{21}s_{12} + x_{22}s_{22}) \\ (x_{31}s_{11} + x_{32}s_{21}) & (x_{31}s_{22} + x_{32}s_{22}) \end{bmatrix}$$

$$(X - XS) = \begin{bmatrix} (x_{11} - x_{11}s_{11} - x_{12}s_{21}) & (x_{12} - x_{11}s_{12} - x_{12}s_{22}) \\ (x_{21} - x_{21}s_{11} - x_{22}s_{21}) & (x_{22} - x_{21}s_{12} - x_{22}s_{22}) \\ (x_{31} - x_{31}s_{11} - x_{32}s_{21}) & (x_{32} - x_{31}s_{22} - x_{32}s_{22}) \end{bmatrix}$$

Now taking the 1^{st} column of $(X - XS)$,

$$(X - XS)_1 = \begin{bmatrix} (x_{11} - x_{11}s_{11} - x_{12}s_{21}) \\ (x_{21} - x_{21}s_{11} - x_{22}s_{21}) \\ (x_{31} - x_{31}s_{11} - x_{32}s_{21}) \end{bmatrix}$$

$$
\begin{aligned}
(X - XS)_1 &= \begin{bmatrix} (x_{11} - x_{11}s_{11} - x_{12}s_{21}) \\ (x_{21} - x_{21}s_{11} - x_{22}s_{21}) \\ (x_{31} - x_{31}s_{11} - x_{32}s_{21}) \end{bmatrix} \\
&= \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \end{bmatrix} - \begin{bmatrix} (x_{11}s_{11} + x_{12}s_{21}) \\ (x_{21}s_{11} + x_{22}s_{21}) \\ (x_{31}s_{11} + x_{32}s_{21}) \end{bmatrix} \\
&= \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \end{bmatrix} - \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \end{bmatrix} \begin{bmatrix} s_{11} \\ s_{21} \end{bmatrix} \\
&= (X_1 - XS_1)
\end{aligned}
$$

Therefore, we can conclude that:

$$(X - XS)_i = (X_i - XS_i) \quad \text{for all } i$$

Now we can calculate:

$$\Big\|(X - XS)_i\Big\|_2^2 = (X - XS)_i^T (X - XS)_i = (X_i - XS_i)^T (X_i - XS_i)$$

Now using the kernel mapping function ϕ, we can write:

$$
\begin{aligned}
\Big\|(\phi(X) - \phi(X)S)_i\Big\|_2^2 &= \Big(\phi(X_i) - \phi(X)S_i\Big)^T \Big(\phi(X_i) - \phi(X)S_i\Big) \\
&= \phi(X_i)^T\phi(X_i) - \phi(X_i)^T\phi(X)S_i - S_i^T\phi(X)^T\phi(X_i) + S_i^T\phi(X)^T\phi(X)S_i \\
&= K_{ii} - 2K^iS_i + S_i^TKS_i
\end{aligned}
$$

where, K is gram matrix or kernel matrix, K^i is the i^{th} row of K.

Therefore, we can express the $l_{2,1}$-norm component of the cost function as following:

$$\begin{aligned}\Big\|\phi(X) - \phi(X)S\Big\|_{2,1} &= \sum_{i=1}^{n} \Big\| \Big(\phi(X) - \phi(X)S\Big)_i \Big\|_2 \\ &= \sum_{i=1}^{n} \sqrt{K_{ii} - 2K^i S_i + S_i^T K S_i}\end{aligned}$$

A.4 What is nuclear norm of a matrix and why it is a convex envelope of the rank of the matrix?

For a given matrix $A \in \mathbb{R}^{m \times n}$, the nuclear norm of A is:

$$\|A\|_\star = trace\Big(\sqrt{A^H A}\Big) = \sum_{i=1}^{min(m,n)} \sigma_i(A)$$

where $\sigma_i(A)$ is the i^{th} singular value of A.
Nuclear norm of A is a convex envelope of the rank function $rank(A)$.

Theorem A.1 *On the set* $S = \Big\{X \in \mathbb{R}^{m \times n} \Big| \|X\| \leq 1\Big\}$, *the convex envelope of the function* $\phi(X) = Rank(X)$ *is* $\phi_{env}(X) = \|X\|_\star = \sum_{i=1}^{min(m,n)} \sigma_i(X)$.

Proof: Now, we follow [73] and prove the Theorem A.1 using the notion of conjugate function. The conjugate function $f^\star$ of a function $f : \mathcal{C} \to \mathbb{R}$, where $\mathcal{C} \subseteq \mathbb{R}^n$, is define as:

$$f^\star(y) = sup\Big\{\langle y, x\rangle - f(x) \Big| x \in \mathcal{C}\Big\}$$

where, $\langle y, x\rangle$ denotes the inner product in $\mathbb{R}^n$. The basic result of the convex analysis is that the conjugate of conjugate, $f^{\star\star}$, is the convex envelope of the function f given some technical conditions hold.

Part 1. *Computing* $\phi^\star$: The conjugate of rank function ϕ, on the set of matrices

with spectral norm less than or equal to one, is

$$\phi^{\star}(Y) = \sup_{\|X\| \leq 1} \Big(trace(Y^T X) - \phi(X) \Big)$$

where, $\langle y, x \rangle = trace(Y^T X)$ is the inner product in $\mathbb{R}^{m \times n}$. Let $q = min\{m, n\}$. By von Neumann's theorem [74],

$$trace(Y^T X) \leq \sum_{i=1}^{q} \sigma_i(X) \sigma_i(Y)$$

where, $\sigma_i(X)$ denotes the i^{th} largest singular values of X. Given X, the above inequality is achieved if U_X and V_X are chosen equal to U_Y and V_Y, respectively, where $X = U_X \Sigma_X V_X$ and $Y = U_Y \Sigma_Y V_Y$ are the SVDs of X and Y. Now, we can write:

$$\phi^{\star}(Y) = \sup_{\|X\| \leq 1} \Big(\sum_{i=1}^{q} \sigma_i(X) \sigma_i(Y) - Rank(X) \Big)$$

If, $X = 0$ then we have $\phi^{\star}(Y) = 0$ for all Y. If $Rank(X) = r$,$1 \leq r \leq q$ then $\phi^{\star}(Y)$ can be expressed as:

$$\phi^{\star}(Y) = max \Big\{ 0, \sigma_1(Y) - 1, \cdots, \sum_{i=1}^{r} \sigma_i(Y) - r, \cdots, \sum_{i=1}^{q} \sigma_i(Y) - q \Big\}$$

From the above expression, we can conclude that:

$$\phi^{\star}(Y) = \sum_{i=1}^{q} \Big(\sigma_i(Y) - 1 \Big)_+$$

where, a_+ denotes the positive part of a, i.e., $a_+ = max(a, 0)$.

Part 2. *Computing* $\phi^{\star\star}$: Now we find the conjugate of $\phi^{\star}$, defined as:

$$\phi^{\star\star}(Z) = \sup_{Y} \Big(trace(Z^T Y) - \phi^{\star}(Y) \Big)$$

for all $Z \in \mathcal{C}^{m \times n}$. As before, we choose $U_Y = U_Z$ and $V_Y = V_Z$ to get

$$\phi^{\star\star}(Z) = \sup_Y \Big(\sum_{i=1}^{q} \sigma_i(Z)\sigma_i(Y) - \phi^{\star}(Y) \Big)$$

Now if $\|Z\| \geq 1$ then we can choose $\sigma_1(Y)$ large enough so that $\phi^{\star\star}(Z) \to \infty$.
Now let, $\|Z\| \leq 1$. If $\|Y\| \leq 1$, then $\phi^{\star}(Y) = 0$ and the supremum is achieved for $\sigma_i = 1, i = 1, 2, \cdots, q$ and we get:

$$\phi^{\star\star}(Z) = \sum_{i=1}^{q} \sigma_i(Z) = \|Z\|_{\star}$$

Now, if $\|Y\| \geq 1$, we can show that the argument of the *sup* is always smaller than the value given above. By adding and subtracting the term $\sum_{i=1}^{q} \sigma_i(Z)$ and rearranging the terms, we get

$$\begin{aligned}
&\sum_{i=1}^{q} \sigma_i(Y)\sigma_i(Z) - \sum_{i=1}^{r} \Big(\sigma_i(Y) - 1 \Big) \\
&= \sum_{i=1}^{q} \sigma_i(Y)\sigma_i(Z) - \sum_{i=1}^{r} \Big(\sigma_i(Y) - 1 \Big) - \sum_{i=1}^{q} \sigma_i(Z) + \sum_{i=1}^{q} \sigma_i(Z) \\
&= \sum_{i=1}^{r} \Big(\sigma_i(Y) - 1 \Big) \Big(\sigma_i(Z) - 1 \Big) + \sum_{i=r+1}^{q} \Big(\sigma_i(Y) - 1 \Big) \Big(\sigma_i(Z) \Big) + \sum_{i=1}^{q} \sigma_i(Z)) \\
&< \sum_{i=1}^{q} \sigma_i(Z)
\end{aligned}$$

The last inequality holds because the first two terms of the third line always have a negative value.
In summary we can conclude that:

$$\phi^{\star\star}(Z) = \|Z\|_{\star}$$

over the set $\Big\{ Z \Big| \|Z\| \leq 1 \Big\}$. Thus over this set, $\|Z\|_{\star}$ is the convex envelope of the function $Rank(X)$.

A.5 Solution of a nuclear-norm minimization problem

In [75], the adaptive nuclear norm of a matrix $C \in \mathbb{R}^{p \times q}$ is defined as a weighted sum of its singular values:

$$\|C\|_{\star w} = \sum_{i=1}^{min(p,q)} w_i d_i(C)$$

where, the w_i are the non-negative weights.

Theorem A.2 *For any $\lambda > 0$, $0 \leq w_1 \leq w_2, \cdots \leq w_{min(p,q)}$ and $Y \in \mathbb{R}^{n \times q}$ with a singular value decomposition $Y = UDV^T$, a global optimal solution of the optimization problem*

$$\underset{C}{minimize} \quad \frac{1}{2}\| Y - C\|_F^2 + \lambda \| C\|_{\star w}$$

is $S_{\lambda w}(Y)$, where $S_{\lambda w}(Y) = US_{\lambda w}(D)V^T$ and
$S_{\lambda w}(D) = diag\big[\{d_i(Y) - \lambda w_i\}_+, i = 1, 2, \cdots, min(n, q)\big]$.

Proof: We first prove that $S_{\lambda w}(Y)$ is indeed a global solution of of the above equation. Assume, $h = min(n, q)$. Let, $g = \{g\}_{i=1}^h = d(C)$, which implies the entries of g are in non-decreasing order. Since the penalty term in the equation depends only on the singular values of C, then the above minimization problem can be written as:

$$\underset{g: g_1 \geq \cdots \geq g_h \geq 0}{\text{minimize}} \Big\{ \underset{C \in \mathbb{R}^{p \times q}, g = d(C)}{\text{minimize}} \quad \big(\frac{1}{2}\| Y - C\|_F^2 + \lambda \sum_{i=1}^{h} w_i g_i\big) \Big\}$$

For the inner minimization, we have the inequality

$$\begin{aligned}
\|Y - C\|_F^2 &= tr(YY^T) - 2tr(YC^T) + tr(CC^T) \\
&= \sum_{i=1}^{h} d_i^2(Y) - 2tr(YC^T) + \sum_{i=1}^{h} g_i^2 \\
&\geq \sum_{i=1}^{h} d_i^2(Y) - 2d(Y)^T g + \sum_{i=1}^{h} g_i^2
\end{aligned}$$

The last inequality due to von Neumann's trace inequality in [76],[74]. The equality holds when C admits the singular value decomposition $C = Udiag(g)V^T$, where U and V are the left and right singular matrices of Y. Then the optimization reduces to:

$$\underset{g:g_1\geq\cdots\geq g_h\geq 0}{\text{minimize}}\left(\sum_{i=1}^{h}\left[\frac{1}{2}g_i^2 - \{d_i(Y) - \lambda w_i\}g_i + \frac{1}{2}d_i^2(Y)\right]\right)$$

The objective function is completely separable and minimized only when $g_i = \{d_i(Y) - \lambda w_i\}_+$. Therefore, $S_{\lambda w}(Y) = Udiag\Big[\{d(Y) - \lambda w\}_+\Big]V^T$ is a global optimal solution.

Bibliography

[1] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

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

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

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

[5] J. Huang, F. Nie, and H. Huang, "A new simplex sparse learning model to measure data similarity for clustering," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[6] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained laplacian rank algorithm for graph-based clustering," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[7] Z. Kang, H. Pan, S. C. Hoi, and Z. Xu, "Robust graph learning from noisy data," *IEEE transactions on cybernetics*, 2019.

[8] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 551–556.

[9] C. Alzate and J. A. Suykens, "Image segmentation using a weighted kernel pca approach to spectral clustering," in *2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing*. IEEE, 2007, pp. 208–213.

[10] F. R. Bach, G. R. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 6.

[11] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "Simplemkl," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2491–2521, 2008.

[12] Z. Xu, R. Jin, H. Yang, I. King, and M. R. Lyu, "Simple and efficient multiple kernel learning by group lasso," in *Proceedings of the 27th international conference on machine learning (ICML-10)*. Citeseer, 2010, pp. 1175–1182.

[13] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, "Multiple kernel fuzzy clustering," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 120–134, 2011.

[14] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, and Y.-D. Shen, "Robust multiple kernel k-means using l21-norm," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[15] Y. Fang, R. Wang, B. Dai, and X. Wu, "Graph-based learning via auto-grouped sparse regularization and kernelized extension," *IEEE Transactions on knowledge and data engineering*, vol. 27, no. 1, pp. 142–154, 2014.

[16] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang, "Learning with l^1-graph for image analysis," *IEEE transactions on image processing*, vol. 19, no. 4, pp. 858–866, 2009.

[17] Z. Kang, L. Wen, W. Chen, and Z. Xu, "Low-rank kernel learning for graph-based clustering," *Knowledge-Based Systems*, vol. 163, pp. 510 – 517, 2019.

[18] Z. Kang, X. Lu, J. Yi, and Z. Xu, "Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification," *arXiv preprint arXiv:1806.07697*, 2018.

[19] X. Cai, F. Nie, and H. Huang, "Multi-view k-means clustering on big data," in *Twenty-Third International Joint conference on artificial intelligence*, 2013.

[20] A. Kumar and H. Daumé, "A co-training approach for multi-view spectral clustering," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 393–400.

[21] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *Advances in neural information processing systems*, 2011, pp. 1413–1421.

[22] F. Nie, J. Li, X. Li *et al.*, "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification." in *IJCAI*, 2016, pp. 1881–1887.

[23] ——, "Self-weighted multiview clustering with multiple graphs." in *IJCAI*, 2017, pp. 2564–2570.

[24] S. Huang, Z. Kang, and Z. Xu, "Self-weighted multi-view clustering with soft capped norm," *Knowledge-Based Systems*, vol. 158, pp. 1 – 8, 2018.

[25] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in neural information processing systems*, 2004, pp. 321–328.

[26] C.-G. Li, Z. Lin, H. Zhang, and J. Guo, "Learning semi-supervised representation towards a unified optimization framework for semi-supervised learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2767–2775.

[27] F. Nie, G. Cai, and X. Li, "Multi-view clustering and semi-supervised classification with adaptive neighbours," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[28] Z.-J. Zha, T. Mei, J. Wang, Z. Wang, and X.-S. Hua, "Graph-based semi-supervised learning with multiple labels," *Journal of Visual Communication and Image Representation*, vol. 20, no. 2, pp. 97 – 103, 2009, special issue on Emerging Techniques for Multimedia Content Sharing, Search and Understanding.

[29] X. Zhu and J. Lafferty, "Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 1052–1059.

[30] R. Johnson and T. Zhang, "Graph-based semi-supervised learning and spectral kernel design," *IEEE Transactions on Information Theory*, vol. 54, no. 1, pp. 275–288, 2008.

[31] A. Kapoor, H. Ahn, Y. Qi, and R. W. Picard, "Hyperparameter and kernel learning for graph based semi-supervised classification," in *Advances in neural information processing systems*, 2006, pp. 627–634.

[32] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, "Semi-supervised graph clustering: a kernel approach," *Machine learning*, vol. 74, no. 1, pp. 1–22, 2009.

[33] H. Tong, J. He, M. Li, C. Zhang, and W.-Y. Ma, "Graph based multi-modality learning," in *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, 2005, pp. 862–871.

[34] Y. Wang, J. Pei, X. Lin, Q. Zhang, and W. Zhang, "An iterative fusion approach to graph-based semi-supervised learning from multiple views," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2014, pp. 162–173.

[35] V. Sindhwani, P. Niyogi, and M. Belkin, "A co-regularization approach to semi-supervised learning with multiple views," in *Proceedings of ICML workshop on learning with multiple views*, vol. 2005. Citeseer, 2005, pp. 74–79.

[36] M. Karasuyama and H. Mamitsuka, "Multiple graph label propagation by sparse integration," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 12, pp. 1999–2012, 2013.

[37] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S.-F. Chang, "Designing category-level attributes for discriminative visual recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 771–778.

[38] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos." in *CVPR*, vol. 2. Minneapolis, MN, 2007, p. 3.

[39] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proceedings of the 6th ACM international conference on Image and video retrieval*. ACM, 2007, pp. 401–408.

[40] J. Winn and N. Jojic, "Locus: Learning object classes with unsupervised segmentation," in *null*. IEEE, 2005, pp. 756–763.

[41] X. Cai, F. Nie, H. Huang, and F. Kamangar, "Heterogeneous image feature integration via multi-modal spectral clustering," in *CVPR 2011*. IEEE, 2011, pp. 1977–1984.

[42] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[43] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *Proceedings Third IEEE international conference on automatic face and gesture recognition*. IEEE, 1998, pp. 200–205.

[44] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Generative models for recognition under variable pose and illumination," in *Proceedings fourth ieee international conference on automatic face and gesture recognition (cat. no. pr00580)*. IEEE, 2000, pp. 277–284.

[45] M. Amini, N. Usunier, and C. Goutte, "Learning from multiple partially observed views-an application to multilingual text categorization," in *Advances in neural information processing systems*, 2009, pp. 28–36.

[46] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *2004 Conference on Computer Vision and Pattern Recognition Workshop*. IEEE, 2004, pp. 178–178.

[47] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of machine learning research*, vol. 3, no. Dec, pp. 583–617, 2002.

[48] X. Cai, F. Nie, W. Cai, and H. Huang, "Heterogeneous image features integration via multi-modal semi-supervised learning model," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1737–1744.

[49] S. Günter and H. Bunke, "Validation indices for graph clustering," *Pattern Recognition Letters*, vol. 24, no. 8, pp. 1107 – 1113, 2003, graph-based Representations in Pattern Recognition.

[50] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The annals of statistics*, pp. 1171–1220, 2008.

[51] Z. Xu, R. Jin, I. King, and M. Lyu, "An extended level method for efficient multiple kernel learning," in *Advances in neural information processing systems*, 2009, pp. 1825–1832.

[52] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *Journal of machine learning research*, vol. 12, no. Jul, pp. 2211–2268, 2011.

[53] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, no. Jul, pp. 1531–1565, 2006.

[54] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *arXiv preprint arXiv:1304.5634*, 2013.

[55] 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*. ACM, 2009, pp. 129–136.

[56] J. Liu, J. Chen, S. Chen, and J. Ye, "Learning the optimal neighborhood kernel for classification," in *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[57] B. Mohar, Y. Alavi, G. Chartrand, and O. Oellermann, "The laplacian spectrum of graphs," *Graph theory, combinatorics, and applications*, vol. 2, no. 871-898, p. 12, 1991.

[58] K. Fan, "On a theorem of weyl concerning eigenvalues of linear transformations i," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 35, no. 11, p. 652, 1949.

[59] S. Huang, Z. Kang, I. W. Tsang, and Z. Xu, "Auto-weighted multi-view clustering via kernelized graph learning," *Pattern Recognition*, vol. 88, pp. 174–184, 2019.

[60] Z. Kang, G. Shi, S. Huang, W. Chen, X. Pu, J. T. Zhou, and Z. Xu, "Multi-graph fusion for multi-view spectral clustering," *Knowledge-Based Systems*, p. 105102, 2019.

[61] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International conference on Machine learning (ICML-03)*, 2003, pp. 912–919.

[62] W. Tang, Z. Lu, and I. S. Dhillon, "Clustering with multiple graphs," in *2009 Ninth IEEE International Conference on Data Mining*. IEEE, 2009, pp. 1016–1021.

[63] F. Nie, G. Cai, J. Li, and X. Li, "Auto-weighted multi-view learning for image clustering and semi-supervised classification," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1501–1511, 2017.

[64] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 793–804, 2013.

[65] F. Nie, W. Zhu, and X. Li, "Unsupervised feature selection with structured graph optimization," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[66] F. Yan, X.-d. Wang, Z.-q. Zeng, and C.-q. Hong, "Adaptive multi-view subspace clustering for high-dimensional data," *Pattern Recognition Letters*, 2019.

[67] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[68] P. Comon, "Independent component analysis, a new concept?" *Signal processing*, vol. 36, no. 3, pp. 287–314, 1994.

[69] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *International conference on artificial neural networks*. Springer, 1997, pp. 583–588.

[70] J. Xu, J. Han, and F. Nie, "Discriminatively embedded k-means for multi-view clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5356–5364.

[71] J. Xu, J. Han, F. Nie, and X. Li, "Re-weighted discriminatively embedded k-means for multi-view clustering," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 3016–3027, 2017.

[72] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint 2, 1-norms minimization," in *Advances in neural information processing systems*, 2010, pp. 1813–1821.

[73] M. Fazel, "Matrix rank minimization with applications," Ph.D. dissertation, PhD thesis, Stanford University, 2002.

[74] L. Mirsky, "A trace inequality of john von neumann," *Monatshefte für mathematik*, vol. 79, no. 4, pp. 303–306, 1975.

[75] K. Chen, H. Dong, and K.-S. Chan, "Reduced rank regression via adaptive nuclear norm penalization," *Biometrika*, vol. 100, no. 4, pp. 901–920, 2013.

[76] J. Von Neumann, *Some matrix-inequalities and metrization of matric space*, 1937.

List of Publications

Journal

1. S. Manna, J. R. Khonglah, A. Mukherjee, G.Saha "Robust kernelized graph-based learning," in *Pattern Recognition (2020)*, vol. 110, 107628.
https://www.sciencedirect.com/science/article/pii/S0031320320304313

Conferences

1. S. Manna, J. R. Khonglah, A. Mukherjee, G.Saha "Low-Rank Kernelized Graph-based Clustering using Multiple Views," in *NCC 2020 - 26th National Conference on Communications*, IIT Kharagpur, India.
https://ieeexplore.ieee.org/document/9056006
2. S. Manna, J. R. Khonglah, A. Mukherjee, G.Saha "Kernelized Graph-based Multi-view Clustering on High Dimensional Data," in *NCC 2020 - 26th National Conference on Communications*, IIT Kharagpur, India.
https://ieeexplore.ieee.org/document/9056029

Author's Biodata

Personal Information

Name	**Supratim Manna**
Date of Birth	13^{th} October 1993
Address	Garhbeta, West Midnapur, West Bengal, India-721127
Nationality	Indian
Email-ids	supratimmanna121@gmail.com

Educational Qualifications

since 2018 **MS (by Research)**, Indian Institute of Technology (IIT) Kharagpur.
Supervisor: Prof. Anirban Mukherjee

2011-2015 **B.E.**, Indian Institute of Engineering Science & Technology, Shibpur (formerly, Bengal Engineering and Science University, Shibpur).

Research Interest

Graph-based Learning, Graph Neural Network (GNN), Computer Vision., NLP