Optimal Neighborhood Kernel approach towards Incomplete Multi-View Clustering (OK-IMVC)

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology in Computer Science

by

Arnab Ray

[Roll : CS1903]

under the guidance of

Dr. Swagatam Das
Associate Professor
Electronics and Communication Sciences Unit (ECSU)

Indian Statistical Institute
Kolkata- 700108, India
July 2021

Certificate

This is to certify that the dissertation entitled "**Optimal Neighborhood Kernel approach towards Incomplete Multi-View Clustering (OK-IMVC)**" submitted by **Arnab Ray** to Indian Statistical Institute, Kolkata, in partial fulfillment for the award of the degree of **Master of Technology in Computer Science** is a bonafide record of research carried out by him under my supervision and guidance. The dissertation work has met all the requirements as per the regulations and norms of the institute and, in my opinion, has reached the standard needed for submission.

Dr. Swagatam Das
Associate Professor,
Electronics and Communication Sciences Unit,
Indian Statistical Institute,
Kolkata-700108, INDIA

Acknowledgements

I would like to show my highest gratitude to my advisor, Dr. Swagatam Das, Associate Professor, Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, for his invaluable guidance and continuous support and encouragement. He has essentially taught me how to conduct a good research alongwith the technicalities involved in learning specific topics. He as also motivated me with great insights and innovative ideas in order to not lose focus while conducting research work.

My deepest thanks to all the professors of the Indian Statistical Institute, for their valuable suggestions and discussions which has added a vital dimension to my research.

Finally, I'm very much obliged to my parents and my family for their everlasting supports.

Last but not the least, I would also like to thank all of my friends for their help and support to shape this work. I would also like to extend my heartfelt thanks to all those, whom I have missed out from the above list.

Arnab Ray

. **Arnab Ray**
. Indian Statistical Institute
. Kolkata - 700108, India

Abstract

Incomplete multi-view clustering (IMVC) has become one of the most prominent area of research in the recent past. The objective of IMVC is to integrate a set of pre-specified incomplete views in order to improve clustering performance. Among various excellent solutions already proposed in literature, multiple kernel k-means with incomplete kernels (MKKM-IK) [1] has been one of the benchmark research works, which formulates the incomplete multi-view clustering problem as a joint optimization problem framework whereby the imputation and clustering paradigms are integrated effortlessly. Both the processes are performed alternately in an iterative fashion to make used of the advantages of clustering in the subsequent imputation process and vice-versa. However, the computationally intensive and associated storage requirements demanded more efficient methods to be devised. These include the incomplete multi-view clustering with late fusion and the efficient and effective way proposed by Liu et al [2]. However, all of the above mentioned algorithms initialize the consensus clustering matrix, considering the unified kernel as a strict convex combination of the incomplete base kernels. This bold assumption suppresses the selectivity and representation capability of the unified kernel.

In order to find a solution to the above problem, we propose a novel method called Optimal Neighborhood Kernel approach towards Incomplete Multi-View Clustering (OK-IMVC) which takes into account the representability of the unified or the optimal kernel. The consensus clustering matrix is continually updated via kernel k-means on the optimal neighborhood kernel, which is in turn computed based on the clustering results at the previous iteration. Specifically, our algorithm jointly learns a consensus clustering matrix, imputes each incomplete base matrix,learns the optimal neighborhood kernel and optimizes the corresponding alignment matrices. Further, we conduct comprehensive experiments to study the proposed OK-IMVC in terms of Normalized Mutual Information (NMI) index, purity score and running time. As indicated, our proposed method significantly and consistently outperforms some of the state-of-the-art algorithms with much less running time and memory.

Contents

Chapter 1

Introduction

1.1 Introduction

The task of clustering data is still a booming field of research in the context of unsupervised learning. Data comes in various forms in real life, ranging from feature-extracted mode to kernel form. Often, we encounter a basic problem in the proces of clustering data. This pertains to the non-separability of observations in the data space. As a remedy towards handling such data and cluster them via linear boundaries, the concept of kernel k-means came into being [3]. Kernel k-means maps the linearly non-separable data in input space to a higher dimensional reproducing kernel Hilbert space (RKHS) where the inherent data clusters become distinguishable, via a nonlinear transformation and thereafter performs k-means in the feature space. An important property of kernel k-means is its close association with the spectral clustering method as demonstrated by Dhillon et al [4].

Real-world applications involve collection of data from diverse domains or various feature descriptors. For instance, specific news articles are reported by a wide range of media houses, expression of the same semantic meaning in multiple languages or depiction of a time-varying signal in temporal and frequency domains. These multi-faceted representation of data is termed as *multi-view* data. They exhibit heterogenous properties, coming from diverse domains, but hold an underlying similarity amongst themselves. Therefore, a way to exploit this information, in order to uncover the potential values of multi-view data, is very important in big data research.

Clustering data with kernel fusion, referred to as Multiple-Kernel Clustering (MKC), is an emerging topic in machine learning. There are a wide range of algorithms that have been proposed in support of clustering multi-view data. A few of the classical MKC algorithms in literature include multiple kernel fuzzy c-means (MKFC) by Huang et al [11], the multiple kernel learning algorithms due to Gonen and Almpaydin [5] and the work by Zhao et al [6] that aims at finding the maximum margin hyperplane and the optimal kernel setup to cluster the data. Liu et al [7] came up with the novel idea of facilitating the process of clustering by carefully considering the correlation among different views. They have utilised the view-specific weight vector to optimally weigh the correlation coefficients between the corresponding views as well. However, they have made use of the assumption that all the views are *complete* , i.e , all the data points have been observed across all views. Herein comes the realm of incomplete multi-view clustering techniques.

Incomplete multi-view clustering techniques fall into two categories - *two-stage* category, where the clustering is performed separately after the data imputation process and the *single-stage* technique, wherein the processes of imputation and clustering are intertwined to take advantage of the effect of imputation on clustering process. One of the first works in the two-stage category was proposed by Trivedi et al [8] in 2010 where they've used kernel canonical correlation analysis (CCA) to impute the missing entries of the kernel matrices. Liu [1] proposed a one-stage method where the incomplete kernels are used as auxiliary variables to be optimized jointly alongwith the clustering procedure. However, the number of missing entries in the kernel matrices and hence the variables to be optimized were huge and of the order $O((n - n_v)(n + n_v + 1))$ in the v^{th} view, where n signifies the total number of data points and n_v denotes the number of observed data points in the v^{th} view. Thus, the overall computational and storage complexity were high.

To deal with this problem, Liu [2] came up with the idea of efficient and effective incomplete multi-view clustering by treating the base clustering matrices per view as the variables to be optimized alongwith the consensus clustering matrix. This strategy effectively reduces the number of imputation variables to the order of $O((n-n_v)k)$ in the v^{th} view, where k is the number of clusters.

The efficient and effective incomplete multi-view clustering procedure doesn't take into account the generalizability of the unified kernel matrix. Hence, the idea of optimality of the kernel matrix can be imposed to provide robustness to the clustering results. We first justify that Efficient and effective incomplete multi-view k-means clustering can be reformulated into the optimal kernel selection strategy suggested by J. Liu [9]. The optimal kernel selection procedure aids in learning the kernel parameters in a generalized manner, reducing the chances of overfitting which might come into play if the unified kernel is chosen strictly from the subspace spanned by the component kernels.

1.2 Our contribution

Our contributions are summarized as follows -

1. We have proposed that the Efficient and Effective Incomplete Multi-View k-means (EE-IMVC) can be reformulated into the *optimal neighborhood kernel selection* framework and thus developed the Optimal Neighborhood Kernel-based Incomplete multi-View k-means algorithm.

2. We have proposed a new measure of Hilbert-Schmidt independence criterion (HSIC) to determine the correlation amongst the incomplete kernels instead of the well-known Pearson's correlation coefficient.

3. We have also provided a performance comparison of our algorithm with state-of-the-art methods such as Efficient and Effective Incomplete Multi-View k-means [2] and Late Fusion Incomplete Multi-View k-means (LF-IMVC) [10]. We have mainly evaluated our method on benchmark multi-view datasets. We have used normalized mutual information (NMI) score and Purity index to evaluate the performance of our proposed method.

4. We've also performed a complexity analysis as well as the storage complexity involved in our algorithm.

1.3 Thesis Outline

The structure of the thesis is organized in the following manner. In Chapter 2, we briefly discuss about the preliminaries and the multi-view clustering paradigm including the Multi-kernel k-means with matrix-induced regularization scheme. In Chapter 3, we discuss about the background related to our work including . In chapter 4, we describe the detailed construction and the optimization problem involved in our scheme. In Chapter 5, we give a detailed performance analysis of our proposed method. In Chapter 6, we summarize the work done and discuss about the future directions related to our work.

Chapter 2

Preliminaries

2.1 Introduction to k-means clustering

The realm of clustering data falls under the category of unsupervised learning and is a very important machine learning tool in literature. The main objective of clustering data is to partition a given group of unlabeled observations into disjoint subsets, so that data points that belong to the same cluster are very similar to each other and as dissimilar as possible to those residing in any other cluster. One of the primitive clustering methods is the Lloyd's k-Means heuristic algorithm. It involves an iterative process of cluster assignment, each data point being assigned to the closest of the k cluster centers and a center recalculation step, where the cluster centers are updated to the mean of all data samples assigned to that cluster. The initial cluster assignment is done arbitrarily. The process continues till a state of convergence is reached, i.e , until there are no changes in the membership of the clusters or after a certain number of iterations. However, the main limitation of this iterative approach is that the boundaries separating the clusters can only be hyperplanes, whih are necessarily affine. If the data points are so scattered that the clusters cannot be separated via hyperplanes, the standard k-Means algorithm will not be able to produce good results. For instance, Lloyd's k-means algorithm cannot cluster the data pertaining to two concentric circles efficiently, where the data points corresponding to the two circles are inherently from different clusters.

2.2 A look into kernel k-means clustering

The limitation of k-means algorithm stems from the fact that the method actually aims to perform clustering by forming hyperplanes between the clusters to separate them. However, if the data becomes linearly non-separable, the Lloyd's k-means algorithm no longer can cluster the data efficiently. Herein comes the idea of *kernel k-means* algorithm. The basic idea behind kernel methods is to map the data points into a higher-dimensional space known as the Reproducing Kernel Hilbert Space (*RKHS*). It is thus possible for a linear separator which partitions the data points in that higher-dimensional space, to have a non-linear projection back in the original data space, which helps us to solve the non-linear separability problem.

The *kernel trick* helps us to circumvent the actual feature projection of the data points into the high-dimensional space ,i.e, we don't explicitly need to know the mapping function or the actual projection o0f the data points into the higher-dimensional space. It uses a kernel function $\phi(x)$ to implicitly calculate the dot products of the vectors corresponding to the projection of data point in the feature space. If $\phi(x_i)$ and $\phi(x_j)$ are the corresponding projections of the data points x_i and x_j in the feature space, then $\kappa(x_i, x_j) = \phi(x_i)^T\phi(x_j)$ gives us the kernel function pertaining to those data points $\forall$ i and j.

The Kernel k-Means algorithm due to Scholkopf et al [3] is an extension of the Lloyd's k-Means clustering algorithm. It involves a nice application of the kernel trick, projecting the data onto a higher-dimensional Hilbert space and computes Euclidean distances between data points in that space. Let there be k clusters given by C_j , $j = 1, 2, \ldots, k$ and data points x_i ; $x_i \in \mathrm{R}^d$ $i = 1, 2, \ldots, n$. The data is linearly non-separable in the original space and hence a mapping function $\phi(.)$:$R^d \longrightarrow R^l$ is used to project them into a high-dimensional space. With each data point being assigned to a cluster, each cluster C_j has a centroid m_j in the high-dimensional space given by :

$$m_j = \frac{\sum_{x_i \in C_j} \phi(x_i)}{|C_j|}$$

where $|C_j|$ is the cardinality of the cluster C_j.

The squared distance of the data point x_i mapped into the feature space from the cluster centroid m_j (effectively the distance of the point x_i from cluster C_j in feature space) is thus given by :

$$D(x_i, m_j) = \|\phi(x_i) - m_j\|^2$$
$$\text{or } D(x_i, m_j) = \phi(x_i)^T\phi(x_i) - 2\phi(x_i)^T m_j + m_j^T m_j$$

Substituting the expression for m_j, we get

$$D(x_i, m_j) = \kappa(x_i, x_i) \text{ - } 2\frac{\sum_{x_k \in C_j} \kappa(x_i, x_k)}{|C_j|} + \frac{\sum_{x_l \in C_j}\sum_{x_k \in C_j} \kappa(x_l, x_k)}{|C_j|^2}$$

The distance of data point x_i from each of the k cluster centers is measured thereafter and the data sample is then reassigned to that cluster C_j with the minimum distance $D(x_i, m_j)$. This is an iterative process, in which the distances of the data points from each of the cluster centers are computed and then they're reassigned to the nearest cluster , until there's no more change in the cluster assignments or the maximum number of iterations have been completed. The initial cluster entry assignments are usually completely random.

2.3 Multi-view learning

In most real-life machine learning applications in video surveillance, social computing and the likes, data is collected from diverse domains or different feature descriptors which exhibit heterogeneous properties. These various forms of depicting the same data or the same object with respect to different domain of data collection is known as multi-view data and each such variable group is referred to as a particular view. The different views for a particular data acquired can take on varied forms, for instance, in image data can be described via - colour feature descriptor, local binary patterns (LBPs), local shape descriptor and spatial/temporal context captured by multiple cameras for activity recognition using sparse camera networks. Text data, on the other hand, can be described using views like : words in documents, information describing documents and the co-citation network graphs for scientific document management tasks.

Traditional machine learning algorithms that are applied on multi-view data, such as kernel-based classifiers and spectral clustering, concatenate all component views into one single unified view to adapt to the learning setting. This technique of agglomerating all the information of the different views into one single compact structure leads to overfitting in case of small data samples and is not meaningful either because each view describes a fixed and predefined statistical property. In contrast to single view learning, learning in multi-view setting introduces a single function to model a particular view and jointly optimizes all the functions pertaining to all views to exploit the information extracted from the redundant views of the same input data, thereby improving the learning performance. Thus, the need to learn more sophisticated representation of the data in a multi-view setting is necessary. As a result, multi-view learning has received enormous attention over the past decade or so and the existing algorithms can be subtly categorised into three groups:

- Co-training
- Multiple-kernel learning
- Subspace-based learning

2.4 Multiple kernel k-means clustering techniques

Many methods have already been proposed to tackle the issue of multiple kernel clustering considering a wide range of problems ([11]; [12]; [13]). The existing methods in literature pertaining to multiple kernel clustering can be roughly subdivided into two categories. The first category aims to learn a consensus matrix via low-rank optimization ([11]; [12]; [13]). [11] proposes to build a transition probability matrix for each view. These view-specific matrices are then used to unearth a shared low-rank transition probability matrix which would serves as a vital input to standard

Markov chain clustering method. [12] proposes to learn the structured kernel noise for each view which would in turn help in building a shared kernel structure acting as the consensus. In [13], a similarity matrix modification is proposed by learning the clustering results in one view and assigning data points to clusters in another view. The other category of algorithms optimizes a group of kernel coefficients, and uses the combined kernel for clustering. ([14], [15], [12],[16]). The work in [14] suggests a multiple kernel k-means clustering algorithm using data fusion. Gonen and Margolin [15] have proposed a multiple kernel k-means algorithm where the kernels are combined together in a localized fashion to better simulate the sample-adaptive characteristics of data.

2.4.1 Multipe kernel k-means with matrix-induced regularization

The data in a multiple-kernel framework can be represented by a group of feature maps $\{\phi_i(.)\}_{i=1}^{V}$, one for each view. Each data sample x_j can be represented in a multi-view setting via : $[\phi_1(x_j), \phi_2(x_j), \ldots, \phi_V(x_j)]$; where V depicts the total number of views and $\phi_r(x_j)$ corresponds to the representation of data point x_j in the r^{th} view. The view-wise kernel weight vector is given by $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_V]^T$. Thus, the combined kernel matrix that is used for clustering is $K_\mu = \mu_v^2 K_v$, where K_v corresponds to the kernel matrix for the v^{th} view. Hence, $K_\mu(x_i, x_j) = \sum_{v=1}^{V} \mu_v^2 K_v(x_i, x_j)$ gives us the entry of the unified kernel function corresponding to the data points x_i and x_j. The kernel function is usually chosen in such a manner that the component kernel matrices and hence the unified kernel matrix all are symmetric and positive semi-definite. The kernel functions that are most commonly used in these methods are the Gaussian (RBF) kernel, polynomial kernel or *tanh* kernel.

The optimization procedure involved in single-view kernel k-means is given by:

$$\min_H Tr(K(I_n - HH^T)) \text{ s.t } H \in \mathrm{R}^{n\times k} \ ; \ H^T H = I_k$$

where K refers to the kernel matrix corresponding to the single view and H is the clustering matrix we need to determine. Number of data points is given by n and there are k clusters. I_k refers to the identity matrix of dimension k.

Similarly, the clustering procedure in multi-view setup can be reduced to the above framework by replacing the kernel K with our unified kernel matrix K_μ, which is nothing but the linear combination of the base kernels. Hence, the optimization framework in multi-kernel k-means can be devised as follows :

$$\min_H Tr(K_\mu(I_n - HH^T)) \text{ s.t } H \in \mathrm{R}^{n\times k} \ ; \ H^T H = I_k \ ; \ \boldsymbol{\mu}^T \mathbf{1}_V = 1$$

which is equivalent to the following -

$$\min_H \sum_{i=1}^{V} \mu_i^2 Tr(K_i(I_n - HH^T)) \text{ s.t } H \in \mathrm{R}^{n\times k} \text{ ; } H^T H = I_k \text{ ; } \sum_{i=1}^{V} \mu_i = 1$$

From the above discussion, it is clear that the relative value of μ_v is only dependent on H and the corresponding v^{th} kernel K_v. This certifies the fact that state-of-the-art multiple kernel k-means (MKKM) algorithms do not adequately account for the mutual influence of these kernels in the process of updating the view coefficients. The algorithm proposed by X. Liu et al [19] considers the mutual dependence of the component kernels, thereby selecting and weighing them accordingly. The kernels with high correlation with each other would be selected together and assigned to similar weights. This particular approach results in a sparse framework for the selection of the component kernels. Thus, with a view to reduce the redundancy and enhance the diversity of the unified kernel matrix, a regularization term is proposed in [19], which necessarily takes into the account the correlation amongst the given base kernels.

A new criteria $\boldsymbol{M}(.,.)$ is introduced to capture the correlation between each pair of base kernels. The quantity $M(K_i, K_j)$ gives us the correlation between the kernel matrices K_i and K_j. In order to incorporate high selectivity of non-correlated kernels , the regularization term corresponding to two correlated kernels K_i and K_j is given by : $M(K_i, K_j) = M_{ij}\ \mu_i \mu_j M(K_i, K_j)$. The view-specific weights are thus modified during optimization process in such a way that almost the total weight is distributed among the non-correlated kernels, which are the ones desired to be picked. This regularization term enhances the chances of the weights μ_i and μ_j to assume high values if the kernels K_i and K_j are less correlated. Based on these observations, the regularization term incorporated is given by :

$$min_{\mu \in R_+^V} \sum_{i,j=1}^{V} \mu_i \mu_j M_{ij} = \boldsymbol{\mu}^T \boldsymbol{M} \boldsymbol{\mu} \text{ such that } \boldsymbol{\mu}^T \mathbf{1}_V = 1$$

Incorporating the regularization term in the framework, the optimization problem looks like :

$$\min_{\boldsymbol{H} \in R^{n\times k}, \boldsymbol{\mu} \in R_+^V} Tr(\boldsymbol{K}_\mu(\boldsymbol{I}_n - \boldsymbol{H}\boldsymbol{H}^T)) + \frac{\lambda}{2} \boldsymbol{\mu}^T \boldsymbol{M} \boldsymbol{\mu}$$

The alternate optimization procedure involved in solving the above problem consists of to steps -

- Update the clustering matrix $\boldsymbol{H}$ by performing a simple kernel k-means on the unified kernel K_μ with $\boldsymbol{\mu}$ fixed.
- Update the view-specific weight vector $\boldsymbol{\mu}$ by solving the following quadratic programming problem with $\boldsymbol{H}$ fixed.

$$\min_{\mu\in R_{+}^{V}} \frac{1}{2}\boldsymbol{\mu}^{T}(2\boldsymbol{T}+\lambda\boldsymbol{M}\boldsymbol{\mu} \text{ s.t } \boldsymbol{\mu}^{T}\mathbf{1}_{V}=1$$

where $T = diag([Tr(K_1(I_n - HH^T)], Tr(K_2(I_n - HH^T), \ldots, Tr(K_V(I_n - HH^T)]]$ is a $V \times V$ diagonal matrix.

Chapter 3

Related work

3.1 Past works on incomplete multi-kernel clustering

Different approaches have been proposed in order to handle missing entries in multi-view data for clustering purposes. We'll do a brief survey of the already suggested methods to cluster data in a multi-view setup for both complete and incomplete views.

Liu et al[7] proposed to incorporate a matrix-induced regularization technique to take into account the correlation among different views in order to perform clustering on kernels which are more diverse and less redundant. They have used a convex combination of all the kernels as the unified kernel, and used the same weight vector for the kernels to take into account the correlations between the corresponding kernels. Similarly, Liu et al [17] suggested to use the concept of an optimal neighborhood kernel matrix, which significantly enhances the representation capability of the unified kernel so chosen. This framework aims at finding the best kernel matrix G, which closely resembles the convex combination of the component kernels in positive semi-definite (P.S.D) space, but possesses a more general representation.

The kernel-based clustering methods pertaining to incomplete views have become popular in recent years. The presence of incomplete base kernel matrices makes it more challenging to utilize the information of all views for clustering. Most of the real-world datasets comes in the form of inaccuracies involving missing observations due to sensor component failure or noisy environment. The approach followed to cluster such incomplete multi-view data requires extra consideration. The research models proposed along this line is termed as - incomplete multi-view clustering (IMVC). There are two broad categories into which the proposed methods in this particular field can be classified. The first category deals with the imputation and clustering tasks separately , performing traditional clustering on the data in post-imputation phase. Thus, these are also termed as *"two-stage"* incomplete multi-view clustering algorithms. The second category of algorithms aims at preserving the basic relationship that exists between the two processes.

The most popular imputation techniques in practice include zero-filling, mean value filling, k-nearest-neighbor filling and expectation-maximization (EM) filling [18]. Some advanced methods have also been proposed to perform matrix-based imputation such as [15], [19], [20], [21]. The first work to deal with incomplete view clustering was proposed in [8]. It uses the kernel representation of a particular view as the similarity matrix and employs Laplace's regularization to impute missing entries of the other views. This method, however, constraints the fact that one view should be completely observed with all the data points. The "one-stage" algorithms aim at unifying imputation and clustering into one global optimization procedure and instantiate a clustering algorithm termed as multiple kernel k-means with incomplete kernels (MKKM-IK) algorithm proposed by Liu et al.[1] . The clustering results at the previous iteration is used to guide the imputation procedure of absent kernel entries, and the latter is used in turn to conduct the subsequent clustering. By this way, these two procedures are effortlessly interleaved, with the objective of better clustering performance.

3.2 Multi-view clustering using optimal neighborhood kernel approach

3.2.1 Selection and tuning of optimal neighborhood kernel in clustering

The approach of optimal neighborhood kernel learning is guided by the fact that in most of the multi-view clustering algorithms, the unified kernel is taken as a convex combination of the component kernels. This reduces the variability of the unified kernel and does not consider the effect of clustering on the learning process of the optimal kernel. The optimal neighborhood kernel clustering framework was suggested by Liu [17] on *complete* multi-view data. The algorithm effectively enlarges the space from which an optimal kernel can be selected and thus, incorporates variability in the clustering procedure. One of the first works in the realm of optimal kernel selection was given by J. Liu [9] for the task of classification via SVM. This approach was extended to handle multi-view clustering framework pertaining to complete kernels by X. Liu . One cool advantage of the optimal neighborhood kernel is that it is iterativey updated according to the clustering results at every iteration, keeping it in *"close"* proximity with the linearly combined base kernels. The optimal neighborhood kernel clustering (ONKC) framework suggested by X. Liu is given by -

$$\min_{G,\alpha,H} Tr(G(I_n - HH^T)) + \frac{\rho}{2}\|G - K_\alpha\|_F^2 + \frac{\lambda}{2}\alpha^T M\alpha$$

where G is the optimal kernel which is required to be positive semi-definite and the distance between G and $K_\alpha = \sum_{i=1}^{m} \alpha_i K_i$ (K_i being the i^{th} component kernel and α denotes the view-specific weight vector ; number of views = m) is expressed as the squared Frobenius norm of their matrix difference, $\|G - K_\alpha\|_F^2$. This particular algorithm effectively widens the region from which an optimal kernel can be selected, which allows it to be in a better position than the traditional methods to identify a robust kernel for clustering. The learning process of the optimal kernel G and the clustering matrix H are essentially coupled with each other to extract the very essence of one procedure aiding the other and vice-versa.

Algorithm for multi-view clustering using optimal neighborhood kernel

The algoritm described in [17] is described as follows -

Input : Component kernel matrices $\{K_i\}_{i=1}^m$, regularization parameters ρ and λ , number of clusters : k, threshold : ϵ
Output : Optimal neighborhood kernel matrix G , clustering matrix H , view-specific weight vector α

- Initialize $\alpha^{(0)} = \mathbf{1}_m/m$, $G^{(0)} = K_\alpha^{(0)}$ and $z = 0$.
- Iterate until convergence :
  - **1**. Determine the unified kernel matrix at the z^{th} iteration by weighing the component kernels according to the vie-specific weights , $K_{\alpha^{(z)}} = \sum_{i=1}^m \alpha_i^{(z)} K_i$
  - **2**. Update the clustering matrix $H^{(z)}$ by application of kernel k-means on the optimal neighborhood kernel G at the previous iteration. The optimization procedure involved is given by -

    $$\min_H \; Tr(G(I_n - HH^T)) \text{ such that } H \in \mathrm{R}^{nxk} \; , \; H^T H = I_k$$

  - **3**. Update the optimal neighborhood kernel matrix by minimizing the objective w.r.t G , which can be re-written as :

    $$\min_G \frac{1}{2}\|G - (K_\alpha - \frac{1}{\rho}(I_n - HH^T))\|_F^2 \text{ such that } G \text{ is PSD.}$$

    This optimization problem involves minimizing the squared Frobenius norm of the matrix difference of G and $B = K_\alpha - \frac{1}{\rho}(I_n - HH^T)$ such that G is positive semi-definite (PSD). This surmounts to nothing but to project the matrix B onto the PSD space. This can be done by performing singular-value decomposition (SVD) on B and extracting the non-negative singular values of the same in order to reconstruct it. The optimal solution can thus be written as $G = US^+V^T$, where $B = USV^T$ is the SVD decomposition of the matrix B. S^+ is obtained from the diagonal matrix S of singular values of B, by replacing the negative singular values with zeros and keeping the remaining ones intact.

- **4**. Update the view-specific weight vector $\alpha^{(z)}$ at the z^{th} iteration by solving the quadratic programming involving α -

  $$\min_\alpha \frac{\rho+\lambda}{2}\alpha^T M\alpha - b^T\alpha \text{ such that } \alpha^T\mathbf{1}_m = 1$$

  where M is the mxm matrix capturing the correlation between the component views with $M_{ij} = Tr(K_i^T K_j)$ and b is a m-vector where the r^{th} element is given by $b_r = \rho Tr(GK_r)$
- Increment the iteration count $z = z + 1$

3.3 Multi-view clustering with incomplete views

Most of the traditional multi-view clustering methods in practice as well as the above specified algorithms make an assumption that all the views available are complete, i.e. , all the data points have been observed across all the views. However, in practice, due to inaccuracies in data acquisition or sensor faults. It is a common occurrence to observe that some views of a sample are absent in practical applications such as prediction of Alzheimer's disease or cardiac disease discrimination. The literature covering this realm of clustering data with incomplete views is termed as Incomplete Multi-View Clustering (IMVC). These methods can be broadly classified into two categories : *two-stage* and *single-stage* algorithms. The former is computation-heavy and doesn't take into account the relation between imputation and clustering steps. The latter set of algorithms aim at imputing the data and performing the clustering procedure in an interleaved manner. Two of the most famous *single-stage* methods proposed in literature are : **Late Fusion Incomplete Multi-View Clustering** by X. Liu et al [10] and **Efficient and Effective Incomplete Multi-View Clustering** by X. Liu et al [2].

3.3.1 Late Fusion Incomplete Multi-View Clustering (LF-IMVC)

LF-IMVC aims to simultaneously perform clustering and imputation of the incomplete base clustering matrices $H_v^a \ _{v=1}^{m}$ of the component views instead of directly filling the missing entries of the incomplete kernel matrices. It aims to learn a consensus H across all views from the complete base clustering matrices $H_v^o \ _{v=1}^{m}$ and in turn imputes the missing part from the learned consensus. Thus, the two processes are made to negotiate with each other to achieve better performance. The mathematical model involved can be proposed as follows :

$$\begin{gathered} max_{(H,\ H_v \ _{v=1}^{m},\ W_v \ _{v=1}^{m})}\ Tr(H^T(\textstyle\sum_{j=1}^m H_j W_j)) \\ \text{s.t } H \in \mathrm{R}^{n\times k} \ ;\ H^T H = I_k \\ W_v \in \mathrm{R}^{k\times k} \ ;\ W_v^T W_v = I_k \\ H_v \in \mathrm{R}^{n\times k} \ ;\ H_v(s_v, :) = H_v^{(0)} \ ;\ H_v^T H_v = I_k \end{gathered}$$

where H_v is the v^{th} base clustering matrix such that $H_v(s_v, :)$ refers to the rows corresponding to the observed data entries, s_v being the indices corresponding to the observed data points for the v^{th} view.

Algorithm for Late Fusion Incomplete Multi-View Clustering (LF-IMVC)

The algoritm described in [22] is described as follows -

Input : Incomplete component kernel matrices $\{K_i\}_{i=1}^m$, regularization parameters ρ and λ , number of clusters : k , threshold : ϵ
Output : Consensus clustering matrix H

- Initialize $W_v^{(0)}\,{}_{v=1}^{m} = I_k$, $H_v^{(0)}\,{}_{v=1}^{m} = \mathbf{0}_{n*k}$ and iteration count $z = 0$
- Iterate until convergence
  - 1. Update the consensus clustering matrix $H^{(z)}$ using base clustering matrices $\{H_v\}_{v=1}^m$ andbase clustering alignment matrices $\{W_v\}_{v=1}^m$, which is essentially a SVD problem. The equivalent optimization procedure is given by :

    $$\max_H Tr(H^T P) \text{ s.t } H \in \mathrm{R}^{n \times k} \text{ ; } H^T H = I_k$$

    where $P = \sum_v H_v W_v$. The optimal solution to the above optimization problem is given by $H = UV^T$, such that $P = USV^T$ is the SVD decomposition of P. This is a single SVD optimization problem and its complexity is given by $O(nk^2)$.

  - 2. Update the base clustering alignment matrices $\{W_v\}_{v=1}^m$ using base clustering matrices $\{H_v\}_{v=1}^m$ and the consensus H, which is also a SVD problem. The equivalent optimization procedure is given by :

    $$\max_{W_v} Tr(W_v^T Q_v) \text{ s.t } W_v \in \mathrm{R}^{k \times k} \text{ ; } W_v^T W_v = I_k$$

    where $Q_v = \sum_v H_v^T H$. The optimal solution to the above optimization problem is given by $W_v = UV^T$, such that $Q_v = USV^T$ is the SVD decomposition of Q_v. This particular optimization process involves m SVD problems in order to determine the alignment matrices for all the views. Each SVD optimization sub-problem takes $O(k^3)$ time.

  - 3. Update the base clustering matrices $\{H_v\}_{v=1}^m$ using base clustering alignment matrices $\{W_v\}_{v=1}^m$ and the consensus H, which is also a SVD problem to solve. The equivalent optimization procedure is given by :

    $$\max_{H_v} Tr(H_v^T R_v) \text{ s.t } H_v \in \mathrm{R}^{n \times k} \text{ ; } H_v^T H_v = I_k$$

    where $R_v = \sum_v H W_v^T$. The optimal solution to the above optimization problem is given by $H_v = UV^T$, such that $R_v = USV^T$ is the SVD decomposition of R_v. This particular optimization process involves m SVD problems in order to determine the alignment matrices for all the views. Each SVD optimization sub-problem incurs $O(nk^2)$ time.

- Increment iteration count $z = z + 1$

3.3.2 Efficient and Effective Incomplete Multi-View Clustering (EE-IMVC)

EE-IMVC aims to simultaneously perform clustering and imputation of the incomplete base clustering matrices $\ H_v^a \ _{v=1}^m$ of the component views instead of directly filling the missing entries of the incomplete kernel matrices. It aims to learn a consensus H across all views from the complete base clustering matrices $\ H_v^o \ _{v=1}^m$ and in turn imputes the missing part from the learned consensus. The two processes are made to negotiate with each other to achieve better performance. However, in this particular method, the alignment of the consensus with the convex-weighted combination of the base clustering matrices is maximized as part of the optimization procedure. The updation of the weights of the corresponding base clustering matrices adds up as an extra step in the framework as compared to LF-IMVC. The mathematical model involved can be proposed as follows :

$$\begin{gathered} max_{(H,\ H_v \ _{v=1}^m,\ W_v \ _{v=1}^m),\boldsymbol{\alpha}}\ Tr(H^T(\textstyle\sum_{j=1}^m \alpha_j H_j W_j)) \\ \text{s.t } H \in \mathrm{R}^{n\times k} \ ; \ H^T H = I_k \\ W_v \in \mathrm{R}^{k\times k} \ ; \ W_v^T W_v = I_k \\ H_v \in \mathrm{R}^{n\times k} \ ; \ H_v(s_v,:) = H_v^{(0)} \ ; \ H_v^T H_v = I_k \\ \boldsymbol{\alpha} \in \mathrm{R}^m \ ; \ \boldsymbol{\alpha^T} \mathbf{1}_m = 1 \end{gathered}$$

where H_v is the v^{th} base clustering matrix such that $H_v(s_v,:)$ refers to the rows corresponding to the observed data entries, while the other entries are to be imputed. s_v are the indices corresponding to the observed data points for the v^{th} view.

Algorithm for Efficient and Effective Incomplete Multi-View Clustering (EE-IMVC)

The algoritm described in [22] is described as follows -

Input : Incomplete component kernel matrices $\{K_i\}_{i=1}^m$, regularization parameters ρ and λ , number of clusters : k , threshold : ϵ
Output : Consensus clustering matrix H

- Initialize $\ W_v^{(0)} \ _{v=1}^m = I_k$, $\ H_v^{(0)} \ _{v=1}^m = \mathbf{0}_{n*k}$ and iteration count $z = 0$
- Iterate until convergence

– 1. Update the consensus clustering matrix $H^{(z)}$ using base clustering matrices $\{H_v\}_{v=1}^m$, base clustering alignment matrices $\{W_v\}_{v=1}^m$ and the view-wise weight vector $\boldsymbol{\alpha}$, which is essentially a SVD problem. The equivalent optimization procedure is given by :

$$\max_H \; Tr(H^T P) \text{ s.t } H \in \mathrm{R}^{n\times k} \; ; \; H^T H = I_k$$

where $P = \sum_v \alpha_j H_v W_v$. The optimal solution to the above optimization problem is given by $H = UV^T$, such that $P = USV^T$ is the SVD decomposition of P. This is a single SVD optimization problem and its complexity is given by $O(nk^2)$.

– 2. Update the base clustering alignment matrices $\{W_v\}_{v=1}^m$ using base clustering matrices $\{H_v\}_{v=1}^m$, view weight vector α and the consensus H, which is also another SVD problem. The equivalent optimization procedure is given by :

$$\max_{W_v} \; Tr(W_v^T Q_v) \text{ s.t } W_v \in \mathrm{R}^{k\times k} \; ; \; W_v^T W_v = I_k$$

where $Q_v = \sum_v H_v^T H$. The optimal solution to this optimization problem is given by $W_v = UV^T$, where $Q_v = USV^T$ is the SVD decomposition of Q_v. This particular optimization process involves m SVD problems in order to determine the alignment matrices for all the views. Each SVD optimization sub-problem takes $O(k^3)$ time.

– 3. Update the base clustering matrices $\{H_v\}_{v=1}^m$ using previously updated base clustering alignment matrices $\{W_v\}_{v=1}^m$ and the consensus H, which is also a SVD problem to solve. The equivalent optimization procedure is given by :

$$\max_{H_v} \; Tr(H_v^T R_v) \text{ s.t } H_v \in \mathrm{R}^{n\times k} \; ; \; H_v^T H_v = I_k$$

where $R_v = \sum_v H W_v^T$. The optimal solution to the above optimization problem is given by $H_v = UV^T$, such that $R_v = USV^T$ is the SVD decomposition of R_v. This particular optimization process involves m SVD problems in order to determine the alignment matrices for all the views. Each SVD optimization sub-problem incurs $O(nk^2)$ time.

– 4. Update the view-specific weight vector $\boldsymbol{\alpha}$ by solving the following optimization problem :

$$\begin{gathered} \max_\alpha \; f^T \boldsymbol{\alpha} \text{ s.t } \boldsymbol{\alpha} \in \mathrm{R}^m \\ \sum_v \alpha_v^2 = 1 \\ \text{where } \boldsymbol{f} = [f_1, f_2, \ldots, f_m] \; ; \; f_r = Tr(H^T H_r W_r) \end{gathered}$$

The optimal solution to this problem is given by : $\boldsymbol{\alpha} = \frac{\boldsymbol{f}}{\|\boldsymbol{f}\|}$

– Increment iteration count $z = z + 1$

Chapter 4

Method

4.1 Notations

The previous notations defined in Section 2.4.1 for Multiple kernel k-means using matrix-induced regularization remains the same for our proposed method.

Notations	Description
n	The number of observations
V	The number of views
$\boldsymbol{K}$	Dataset in the form of a kernel matrix
K_i	The kernel matrix corresponding to i^{th} view
$\boldsymbol{C}_k$	The k^{th} cluster center/ The k^{th} cluster
n_k	Number of observations in cluster C_k
t_v	Indices for the observed data points for the v^{th} view
n_v	Number of data points that could be observed in v^{th} view
L	Optimal neighborhood kernel matrix
H	Consensus clustering matrix across all views
H_v	Base clustering matrix for v^{th} view
W_v	Base clustering alignment matrix for v^{th} view
$H_v^{(a)}$	Entries of H_v that are to be imputed
M	View-specific correlation matrix
γ	View-specific weight vector

Table 4.1: Notations used in our method

4.2 Introduction to imputation-based optimal kernel clustering for incomplete multi-view data

Traditional multi-view clustering algorithms on incomplete views fall into two categories : the "two-stage" algorithms , wherein the missing entries in the incomplete views are filled by an imputation scheme like zero-filling , mean substitution, k-nearest neighbor filling , expectation-maximization (EM) filling , stochastic regression filling and many other advanced techniques. Thereafter , a standard multi-view clustering scheme is deployed to satisfactorily partition the data into disjoint groups. The second class of algorithms tries to unify the two processes involved - imputation of incomplete views and subsequent clustering into a single optimization problem. The two processes are alternately performed , i.e., the clustering result at the previous iteration guides the imputation procedure at the next iteration and so on. The clustering scheme used in both classes of algorithms involves selection of a unified kernel matrix , which is essentially considered to be a linear combination of all the kernel matrices pertaining to the different views. This strong assumption puts a hard constraint on the space in which the unified kernel matrix resides, which significantly supresses the representability of the unified/optimal kernel.

4.3 Selection of optimal neighborhood kernel for clustering incomplete multi-view data

Learning a compact representation of the unified kernel in case of multi-view data for incomplete views require careful handling of the incomplete component kernels and also the parameters involving the consensus clustering matrix. In this method, we incorporate the idea of selecting an optimal neighborhood kernel to combine the influence of the incomplete base kernels which are given as input and the view-wise base clustering matrices which are continually updated on the run. This reformulation of the incomplete multi-view clustering into the mould of optimal kernel strategy also gives us an idea of the overall location of the data samples which are missing in most of the views.

The optimal neighborhood kernel corresponding to the incomplete views should belong to the space of positive semi-definite kernels. It necessarily gives us

4.4 The proposed formulation

In this section, we describe the scheme of imputation-based clustering using optimal kernel selection strategy for incomplete multi-view data (OK-IMVC). Here, we assume that the pre-specified kernel matrices K_i ; $i = 1, 2, ...V$ are only a noisy observation of the ”ideal” kernel matrix L , where V is the number of views. The mathematical model of the incomplete multi-view clustering scheme is given below :

$$\begin{gathered}
\min_{H,\gamma,K_p;p=1,2...V} \mathbf{Tr}(L(I_n - HH^T)) + \|L - K_\gamma\|_F^2 \\
\text{s.t } H \in R^{nxk} \; ; \; H^T H = I_k \\
\gamma^T \mathbf{1}_V = 1 \; ; \; \gamma_p \geq 0 \\
K_\gamma = \sum_{j=1}^{V} K_j \\
K_v(t_v, t_v) = K_v^{(cc)} \; ; \; K_v \text{ is P.S.D } \forall v
\end{gathered}$$

where t_v denotes the sample indices for which the v^{th} view is present and $K_v^{(cc)}$ is the corresponding submatrix generated from these samples. G corresponds to the optimal kernel matrix which should be , according to Frobenius norm , as close to the linear combination of the view-specific kernel matrices as possible.

We also define the base clustering matrices for the different views as-

$$\boldsymbol{H}_v = [H_v^{(o)^T}, H_v^{(a)^T}]^T \ ,$$

where the v^{th} base clustering matrix $H_v^{(o)} \in R^{n_v x k}$ can be obtained by solving the kernel k-means problem on the v^{th} incomplete base kernel matrix $K_v(t_v, t_v) = K_v^{(cc)}$. $(v = 1, 2, ...V)$.
$H_v^{(a)} \in R^{(n-n_v)xk}$ denote the absent part of the v^{th} base clustering matrix H_v. The main crux of our algorithm is to simultaneously perform clustering and imputation of the present base clustering matrices $H_v^{(a)}$ directly while keeping the portion $H_v^{(o)}$ unchanged during the learning course. Also, we need to construct a consensus clustering matrix $H \in R^{nxk}$ which should reflect the agreement between the V base clustering matrices for the different views. These two processes are seamlessly integrated so as to utilise the result of clustering at the previous iteration to help impute the base clustering matrices.At every iteration, we shall update the optimal kernel matrix $G \in R^{nxn}$ which shall be in the neighborhood of K_γ , with respect to the updated kernel weight vector $\gamma \in R^V$; $\gamma^T \mathbf{1}_V = 1$. This kernel weight vector plays an important role in the selection of different base kernels in order to determine the optimal kernel matrix. We need to sufficiently consider the mutual correlation between the different kernel matrices, where kernels corresponding to complementary information are the most likely ones to be picked owing to the sparsity (l_1 -norm) constraint applicable to the kernel weights. This enhances the diversity of the unified kernel. We account for this predicament by adding a matrix-induced regularization term to the resultant objective.

The consensus clustering matrix H is updated following a simple kernel k-means over the optimal neighborhood kernel matrix L. We want to maintain a semblance of agreement of the individual base clustering matrices and the consensus H, using a *matrix dot product* form, given by $Tr(H^T \sum_{v=1}^{V} \gamma_v [H_v^{(o)^T}, H_v^{(a)^T}]^T W_v)$. The weighted incomplete base clustering matrices $[H_v^{(o)^T}, H_v^{(a)^T}]^T$ are aligned with respect to the consensus via orthogonal matrix W_v, pertinent to the v^{th} view and so on. Keeping H fixed, the incomplete base clustering matrices alongwith the alignment matrices W_v are updated with a view to maximize the *agreement* between them and the consensus. The optimal kernel L is in turn updated using the clustering results and the view-specific weights at the previous iteration.

The above idea can thus be fulfilled using two optimization schemes as follows :-

$$\begin{gathered}\min_{H,W_v,\gamma,L} Tr(\mathbf{L}(I_n - HH^T)) + \tfrac{\rho}{2}\|\mathbf{L} - \mathbf{K}_\gamma\|_F^2 + \tfrac{\lambda}{2}\gamma^T\mathbf{M}\gamma \ , \\ \max_{{H_v}^{(a)}} Tr(\mathbf{H}^T \textstyle\sum_{v=1}^{V} \gamma_v[\mathbf{H}_v^{(o)^T}, \mathbf{H}_v^{(a)^T}]^T\mathbf{W}_v) \\ \text{s.t } \mathbf{H} \in R^{nxk} \ , \mathbf{H}^T\mathbf{H} = I_k.\end{gathered}$$

where L is the optimal kernel for the incomplete base kernels. L is required to be positive semi-definite (P.S.D) and the distance between L and $K_\gamma = \sum_{i=1}^{V} K_i$, which is the linearly weighted combination of the incomplete kernels, is expressed in terms of the squared Frobenius norm of their matrix difference. The correlation between the views in the form of Hilbert-Schmidt norm is expressed as- $M \in R^{VxV}$ with the element $M_{ij} = Tr(K_iQK_jQ)$; $Q = \frac{1}{n}(\mathbf{I}_n - \mathbf{1}_n\mathbf{1}_n^T)$ capturing the correlation between the i^{th} and j^{th} kernel matrices K_i and K_j respectively. H and ${H_v}^{(a)}$ are the consensus clustering matrix and the absent portion of the v^{th} base clustering matrix which needs to be imputed , W_v is the v^{th} permutation matrix to optimally align H_v with H. $\gamma = [\gamma_1, \gamma_2, ...\gamma_V]^T$, which corresponds to the weight vector for the V base clustering matrices as well as the base kernel matrices in order to seek the optimal kernel L.

So the total objective to be realised here involves a sort of minimax optimization over the consensus clustering indicator matrix H , as the selection of the optimal kernel L based on the linearly combined incomplete kernels involves a minimization procedure and the alignment criterion of the V base clustering matrices H_v for $v = 1, 2, \ldots, V$ with the global consensus involves a trace maximization operation. The two objectives need to be tuned seamlessly by the corresponding regularization parameters in such a way that they agree upon a common clustering setup.

4.5 Initialization of H_v and W_v

In our proposed method, the consensus clustering matrix is obtained by performing a simple kernel k-means procedure on the optimal neighborhood kernel L, where L is initially composed of the linearly weighted combination of the observed portion of the component kernel matrices $K_v(t_v, t_v)$ $\in$R$^{n_v \times n_v}$, n_v being the number of data points observed in the v^{th} view. The incomplete portions of the base clustering matrices as well as the kernel alignment matrices are initialised with zeros. The entries of the base clustering matrices corresponding to the observed data samples for each view are initialized by performing a simple kernel k-means on the kernel sub-matrix formed by those entries for each view. The base clustering alignment matrices W_v are initialized to the identity matrix of order k for each view. This initialization has well demonstrated good clustering performance via our proposed OK-IMVC framework.

4.6 Alternate optimization

Jointly optimizing all the unknowns is difficult. So we propose a four-step alternate optimization scheme to solve our dual objective problem efficiently. The view-specific weight vector γ is uniformly initialized as follows : $\gamma_1 = \gamma_2 = ... = \gamma_V = \frac{1}{\sqrt{V}}$

i) **Optimizing H with fixed γ , L ,$(W_v)_{v=1}^V$ and $(H_v)_{v=1}^V$:**

The equivalent optimization boils down to the following :

$$\min_H Tr(\mathbf{L}(I_n - HH^T)) \text{ s.t } H \in R^{nxk} \text{ and } H^T H = I_k$$

This is effectively a kernel k-means problem with L as the kernel matrix and k being the number of clusters. This problem can be optimally solved to obtain the consensus matrix H, by stacking the k eigenvectors of the optimal neighborood kernel matrix L, corresponding to the top k eigenvalues.

ii) **Optimizing $(W_v)_{v=1}^V$ with fixed γ , L ,H and $(H_v)_{v=1}^V$:**

The equivalent optimization boils down to the following :

$$\max_H Tr(W_v^T Y) \text{ s.t } W_v \in R^{kxk} \text{ and } W_v^T W_v = I_k$$
$$\text{where } Y = H_v^T H$$

This is effectively a singular value decomposition problem and can effectively be solved in $O(k^3)$, k being the number of clusters. Let the singular value decomposition of Y be given by : $Y = USV^T$, where S is the diagonal-like matrix comprising the singular values of X. Let B =$S^{\frac{1}{2}}$. then the optimization problem can be expressed as : $\max_{W_v : W_v^T W_v = I_k} Tr(W_v^T UB^2 V^T) = Tr((W_v^T UB)(VB)^T)$ = $\langle W_v^T UB, BV \rangle$.

iii) **Optimizing γ with fixed $(H_v^{(a)})_{v=1}^V$, L ,H and $(W_v)_{v=1}^V$**

The equivalent optimization is formulated as follows :

$$\min_\gamma \frac{(\rho+\lambda)}{2}\gamma^T M \gamma - b^T \gamma$$
$$\text{s.t } \gamma^T 1_V = 1$$
$$\text{where } \mathbf{b} = [b_1, b_2, ...b_V]^T \text{ with } b_m = \rho Tr(LK_m) + Tr(H^T H_m W_m)$$

This is effectively a quadratic programming problem with linear constraints.

iv) **Optimizing L with fixed $(H_v^{(a)})_{v=1}^V$, γ ,H and $(W_v)_{v=1}^V$**

The optimization problem for determination of the optimal kernel matrix G is given below:

$$\min_L \tfrac{1}{2}\|L - B\|_F^2 \text{ s.t } L \text{ is positive semi-definite.}$$
$$\text{where } B = K_\gamma - \tfrac{1}{\rho}(I_n - HH^T)$$

The objective of this problem is to find the projection of B in P.S.D space. According th Theorem 2 in (Zhou et al. 2015),it's optimal solution can be readily written as $G = U_B S_B^+ V_B^T$ where $B = U_B S_B V_B^T$ and S_B^+ is a diagonal matrix keeping the positive singular values of S_B and setting the others to zeroes.

v) **Optimizing $(H_v^{(a)})_{v=1}^V$ with fixed γ , L ,H and $(W_v)_{v=1}^V$:**

The equivalent optimization boils down to the following :

$$\max_{H_v^{(a)}} Tr(H_v^{(a)T} Z) \text{ s.t } H_v^{(a)} \in R^{(n-n_v)xk} \text{ and } H_v^{(a)T} H_v^{(a)} = I_k$$
$$\text{where } Z = H(t_v, :)W_v^T$$

where t_v denotes the indices for which the v^{th} view is absent or missing. This is effectively a singular value decomposition problem and can effectively be solved in $O((n - n_v)k^2)$, k being the number of clusters.

The selection of the regularization coefficients λ and ρ should be done with utmost care so that the two objectives involved in the optimization process mutually agree towards an optimal solution. We have observed that the second objective which aims at maximizing the alignment of the base clustering matrices with the consensus is involved directly in a remarkable tug of war with the second term $\|L - K_\gamma\|_F^2$ of the first objective , which aims at selection of the optimal kernel matrix L. The two terms involved get stuck in a race against each other , one decreases slightly when the other increases and vice-versa towards reaching a point of convergence.

4.7 The algorithm

```
Algorithm 1: OK-IMVC Algorithm
Data: $\{K_i\}_{i=1}^{V}$: Kernel matrices for $v$ views, $\{t_i\}_{i=1}^{V}$ : List of indices of the
      data samples unobserved in each view, $\lambda$ , $\rho$ ,
Result: Consensus clustering matrix $H$, view-specific weight vector $\gamma$ ,
        optimal neighborhood kernel $L$ , Base clustering matrices $\{H_i\}_{i=1}^{V}$ ,
        Base clustering alignment matrices $\{W_i\}_{i=1}^{V}$
Calculate view-wise correlation matrix $M$
for $i$=1 to $V$ do
    (a) Initialize the $i^{th}$ base clustering matrix $H_i(t_i,:)$ by performing kernel
    $k$-means on the sub-matrix $K_i(t_i,t_i)$ and fill the other entries with zeros
    (b) Initialize the $i^{th}$ base clustering alignment matrix : $W_i = I_k$
end
Initialize the consensus clustering matrix $H_{(0)}$ by performing kernel
  $k$-means over the averaged incomplete base kernel $K_0 = \sum_{i=1}^{V} \frac{1}{V} K_i$;
Calculate the optimal neighborhood kernel $L$ for the first iteration by
  performing SVD on $(K_0 - \frac{1}{\rho}(I_n - H_{(0)}H_{(0)}^T))$ and selecting the non-negative
  singular values ;
$t = 0$;
while ($obj^{(t)}$-$obj^{(t-1)} \leq \epsilon obj^{(t-1)}$) do
    Perform kernel $k$-means on the optimal kernel $L$ to obtain the consensus
      clustering matrix $H$
    for $i$=1 to $v$ do
        Calculate the $i^{th}$ base clustering alignment matrix $W_i$ by performing
          an SVD operation on $H_i^T H$.
    end
    Determine the view-specific weight vector $\boldsymbol{\gamma}$ by performing a quadratic
      programming problem with linear constraints given by :

                  $\min_{\gamma} \frac{(\rho+\lambda)}{2}\gamma^T M \gamma - b^T \gamma$
                              s.t $\gamma^T 1_V = 1$
        where $\mathbf{b} = [b_1, b_2, .....b_V]^T$ with $b_m = \rho Tr(LK_m) + Tr(H^T H_m W_m)$

    With the newly updated eight vector, compute $K_\gamma = \sum_{i=1}^{V} \gamma_i K_i$;
    Update the optimal neighborhood kernel $L$ by performing an SVD over
    $(K_\gamma - \frac{1}{\rho}(I_n - HH^T))$ and rejecting the negative singular values.
    for $i$=1 to $v$ do
        Calculate the $i^{th}$ base clustering matrix $H_i$ by performing an SVD
        operation on $HW_i^T$.
    end
    Determine the cluster membership from the consensus $H$ by the
    following procedure : $C(i) = argmax_p H(i,p)$ ;
    Update iteration count $t = t + 1$;
end
```

Chapter 5

Performance analysis of our proposed method

5.1 Complexity Analysis

The computational complexity of classical kernel k-means procedure is $O(n^2k)$, where n is the number of data points and k is the number of clusters. The $EE-IMVC$ algorithm proposed by Liu et al. has a total running time of $O(nk^2+Vk^3+\sum_{v=1}^{V}(n-n_v)k^2)$, where n , V and k are the number of samples, views and clusters respectively. The number of observable data points in the v^{th} view is given by n_v. So, the proposed $OK-IMVC$ framework will understandably incur a larger running time due to the incorporation of two subtle aspects - considering the correlation between the different views and also the estimation of the optimal neighborhood kernel matrix.

The optimization process involved in the $OK-IMVC$ algorithm comprises a kernel k-means structure for estimating the consensus clustering matrix. This involves a running time of $O(n^2k)$. This is followed by solving two SVD problems, one for determining the view-specific base clustering matrices $(H_v)_{v=1}^{V}$, which can be done in $O(\sum_{v=1}^{V}(n-n_v)k^2)$ time ; while the other aims at computing the cluster alignment matrices $(W_v)_{v=1}^{V}$, which costs a running time of $O(Vk^3)$. Thereafter, the optimal kernel L is computed as a modified SVD problem, which further requires $O(n^3)$ time.

5.2 Experimental Results

In this section, we evaluate the performance of our *OK-IMVC* algorithm on a few benchmark multi-view datasets. Our algorithm has been implemented in *Python 3.7* and the results obtained have been performed on a *Windows* machine with a 4-core 2.30 GHz processor and 8 GB RAM.

5.2.1 Parameter Settings

The $OK - IMVC$ algorithm is characterized by two tunable parameters : λ and ρ. The regularization term λ is used to take into account the parity between the overall objective and the quadratic term $\gamma^T M \gamma$, which necessarily relates to the weighted view-specific correlation terms. The parameter ρ, on the other hand, ensures that the optimal neighborhood kernel L is in close *vicinity* of the weighted combination of incomplete kernels.

Our experimental observations show that the results are satisfactory for values of λ taken from the grid of $\{2^{-5}, 2^{-4.9}, ..., 2^{-2..5}\}$. Likewise, the range of values for ρ can be taken from the set $\{2^{-7}, 2^{-6.5}, ..., 2^{-4}\}$ to achieve the best possible clustering results. Thus, we can observe that tuning these hyperparameters is of utmost importance to obtain desirable results.

Also, we need to fix the value of the threshold parameter ϵ; anything in the interval $[10^{-4}, 10^{-2}]$ works fine .

5.3 Description of the multi-view datasets in our study

Dataset Name	**Samples** n	**Number of views** V	K
Flowers17	1360	7	17
ProteinFold	694	12	27
Caltech102-30	3060	48	102
Digital	2000	3	10
Flowers102	8189	4	102

Table 5.1: Description of benchmark multi-view kernel datasets

5.4 Evaluation on the real-world multi-view datasets

We apply the OK-IMVC algorithm with parameters set in the range given by : λ is chosen from the set $\{2^{-5}, 2^{-4.9}, \ldots 2^{-2..5}\}$ while ρ is chosen from the given set $\{2^{-7}, 2^{-6.5}, ..., 2^{-4}\}$ on the different multi-view datasets described above, with incomplete views. The algorithm is given as input the missing indices of the data points corresponding to every view. The results of the clustering are collected in terms of Normalized Mutual Information (NMI) score and clustering Purity index. These indicators are thereafter used to compare our proposed OK-IMVC algorithm with the state-of-the-art methods in incomplete multi-view clustering including the following :

i) Efficient and Effective Incomplete Multi-View Clustering (EE-IMVC) [2]
ii) Late Fusion Incomplete Multi-View Clustering (LF-IMVC) [10]
iii) Localized Incomplete Multiple Kernel k-means (LI-MKKM) [22]
iv) Multiple Kernel k-means with alignment-maximization filling (MKKM+AF) [23]
v) Multiple Kernel k-means with mean-value filling (MKKM+MF)
vi) Multiple Kernel k-means with zero-filling (MKKM+ZF) -*

Method	Flowers-17	ProteinFold	Caltech102-30	UCI-Digital	Flowers-102
OK-IMVC	**50.35**	**45.21**	**54.58**	62.45	*50.14*
LF-IMVC	*45.89*	34.83	*53.37*	*68.99*	49.90
EE-IMVC	45.19	*36.91*	52.9	**69.76**	**50.80**
MKKM-IK	43.76	34.60	40.4	46.87	39.60
LI-MKKM	44.62	35.40	45.6	45.35	41.70
MKKM+AF	40.30	31.25	39.1	46.98	37.80
MKKM+MF	36.46	30.72	37.7	40.01	37.40
MKKM+ZF	37.0	30.60	37.7	41.77	37.40

Table 5.2: Performance comparison in terms of NMI values of OK-IMVC with state-of-the-art incomplete multi-view clustering algortihms on real-world multi-view datasets

In both the tables, the best values for a particular dataset is highlighted in bold. The second-best values obtained are italicized. We can observe that our algorithm achieves best performance in most of the cases involving both clustering measures.

Method	Flowers-17	ProteinFold	Caltech102-30	UCI-Digital	Flowers-102
OK-IMVC	**53.78**	**49.86**	*34.38*	*68.73*	39.37
LF-IMVC	*53.40*	36.70	**34.85**	**79.80**	**49.90**
EE-IMVC	47.72	*41.35*	*34.38*	62.51	*41.80*
MKKM-IK	45.90	29.82	18.60	50.75	26.20
LI-MKKM	48.80	28.34	23.40	48.60	28.10
MKKM+AF	42.20	27.52	16.90	50.39	17.00
MKKM+MF	38.20	27.16	15.30	43.26	16.90
MKKM+ZF	38.40	27.22	15.30	44.64	15.30

Table 5.3: Performance comparison in terms of Purity values of OK-IMVC with state-of-the-art incomplete multi-view clustering algortihms on real-world multi-view datasets

Chapter 6

Conclusion

Clustering of multi-view data is challenging due to the diverse representation of data according to different views. In our study, which is inspired by the literature of incomplete multi-view clustering (IMVC), we proposed a novel optimal neighborhood kernel-based approach to handle incomplete views with an additional regularization term. Based on the framework, we have developed an efficient method to solve the problem and named it as Optimal Neighborhood Kernel-based Incomplete Multi-View k-means Clustering (OK-IMVC) algorithm. Our algorithm is based on the optimal neighborhood kernel strategy for k-means clustering in the works proposed by J. Liu et al [9] and X. Liu et al [17].

The experimental results obtained in section 5.4 confirmed the outperformance of our approach over other approaches like the Efficient and Effective IMVC (EE-IMVC) [2] and Late Fusion IMVC (LF-IMVC) [10] in most of the cases. We have mainly worked with the benchmark multi-view kernel datasets. Our algorithm outperforms both of the above-mentioned algorithms in most of the cases. Even for the UCI Digital and FLowers-102 datasets, our method has outperformed many of the existing state-of-the-art techniques in the realm of incomplete multi-view clustering. In the cases where the other two algorithms have produced better clustering indexes such as NMI score or purity index, our scheme is only marginally behind. For our comparisons, we have used the original $Matlab$ codes by X. Liu et al [2],[24], [10] for EE-IMVC and LF-IMVC respectively. The code for our implementation of the Optimal Neighborhood kernel-based incomplete multi-view k-means algorithm is available at $\boldsymbol{LINK}$.

6.1 Extension of our method and scope for future work

Despite the good performance of our proposed method over a wide range of cases, it poses a few limitations and there is definitely scope for improvement. The performance of clustering is dependent meticulously on the selection of regularization parameters λ and ρ, which needs to be selected with care and cannot be auto-tuned. The values of the coefficients have been tuned via grid search in order to promote good clustering results. Moreover, our proposed algorithm doesn't adequately take into account the influence of the incomplete base clustering matrices on one another. The correlation between the incomplete component kernels can be replaced over iteration, by a regularization term which captures the correlation between the base clustering matrices instead. This correlation term can be expressed in terms of various divergence measures like KL-divergence or Itakura-Saito distance , and other metrics like Hilbert-Schmidt Independence Criterion (HSIC) or other forms of Bregman divergence metrics.

This work can be further extended to the realm of clustering multi-view data streams. The paradigm of multi-view data stream clustering is a relatively less explored field of research. Incomplete multi-view data stream poses a novel research problem to tackle, given that all the information regarding the data needs to be stored and subsequently updated in the form of specific statistical measures. The dynamism involved in the data acquisition process requires smarter algorithms to counter data *incompleteness* problems.

Bibliography

[1] X. Liu, X. Zhu, M. Li, L. Wang, E. Zhu, T. Liu, M. Kloft, D. Shen, J. Yin, and W. Gao, “Multiple kernel k k-means with incomplete kernels,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 5, pp. 1191–1204, 2019.

[2] X. Liu, X. Zhu, M. Li, C. Tang, E. Zhu, J. Yin, and W. Gao, “Efficient and effective incomplete multi-view clustering,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 4392–4399, 2019.

[3] B. Schölkopf, A. Smola, and K.-R. Müller, “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[4] 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*, pp. 551–556, 2004.

[5] M. Gönen and E. Alpaydın, “Multiple kernel learning algorithms,” *The Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[6] B. Zhao, J. T. Kwok, and C. Zhang, “Multiple kernel clustering,” in *Proceedings of the 2009 SIAM International Conference on Data Mining*, pp. 638–649, SIAM, 2009.

[7] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, “Multiple kernel k-means clustering with matrix-induced regularization,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[8] P. Rai, A. Trivedi, H. Daumé III, and S. L. DuVall, “Multiview clustering with incomplete views,” in *Proceedings of the NIPS Workshop on Machine Learning for Social Computing*, Citeseer, 2010.

[9] X. Liu, S. Zhou, Y. Wang, M. Li, Y. Dou, E. Zhu, and J. Yin, “Optimal neighborhood kernel clustering with multiple kernels,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.

[10] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, and W. Gao, “Late fusion incomplete multi-view clustering,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 10, pp. 2410–2423, 2018.

[11] R. Xia, Y. Pan, L. Du, and J. Yin, “Robust multi-view spectral clustering via low-rank and sparse decomposition,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 28, 2014.

[12] 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.

[13] 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)*, pp. 393–400, 2011.

[14] S. Yu, L. Tranchevent, X. Liu, W. Glanzel, J. A. Suykens, B. De Moor, and Y. Moreau, "Optimized data fusion for kernel k-means clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 1031–1039, 2011.

[15] M. Gönen and A. A. Margolin, "Localized data fusion for kernel k-means clustering with application to cancer biology," *Advances in Neural Information Processing Systems*, vol. 27, pp. 1305–1313, 2014.

[16] Y. Lu, L. Wang, J. Lu, J. Yang, and C. Shen, "Multiple kernel clustering based on centered kernel alignment," *Pattern Recognition*, vol. 47, no. 11, pp. 3656–3664, 2014.

[17] J. Liu, X. Liu, J. Xiong, Q. Liao, S. Zhou, S. Wang, and Y. Yang, "Optimal neighborhood multiple kernel clustering with adaptive local kernels," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[18] Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an em approach," in *Advances in neural information processing systems*, pp. 120–127, 1994.

[19] C. Xu, D. Tao, and C. Xu, "Multi-view learning with incomplete views," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5812–5825, 2015.

[20] W. Shao, L. He, and S. Y. Philip, "Multiple incomplete views clustering via weighted nonnegative matrix factorization with $l_{2,1}$ regularization," in *Joint European conference on machine learning and knowledge discovery in databases*, pp. 318–334, Springer, 2015.

[21] S. Bhadra, S. Kaski, and J. Rousu, "Multi-view kernel completion," *Machine Learning*, vol. 106, no. 5, pp. 713–739, 2017.

[22] X. Zhu, X. Liu, M. Li, E. Zhu, L. Liu, Z. Cai, J. Yin, and W. Gao, "Localized incomplete multiple kernel k-means.," in *IJCAI*, pp. 3271–3277, 2018.

[23] M. Li, X. Liu, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel clustering with local kernel alignment maximization," 2016.

[24] X. Liu, M. Li, C. Tang, J. Xia, J. Xiong, L. Liu, M. Kloft, and E. Zhu, "Efficient and effective regularized incomplete multi-view clustering," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

Chapter 7

Appendix-I

Provide a convergence proof here if needed with plots of the objective function over iteration.

Clustering results with plots for missing ratio = 0.1-0.9