A Kernel Approach for Semi-Supervised Clustering Framework for High Dimensional Data

M. Pavithra[1], Dr.R.M.S.Parvathi [2,]

Assistant Professor, Department of C.S.E, Jansons Institute of Technology, Coimbatore, India[1].

Dean- PG Studies, Sri Ramakrishna Institute of Technology, Coimbatore, India[2].

ABSTRACT

Clustering of high dimensionality data which can be seen in almost all fields these days is becoming very tedious process. The key disadvantage of high dimensional data which we can pen down is curse of dimensionality. As the magnitude of datasets grows the data points become sparse and density of area becomes less making it difficult to cluster that data which further reduces the performance of traditional algorithms used for clustering. Semi-supervised clustering algorithms aim to improve clustering results using limited supervision. The supervision is generally given as pair wise constraints; such constraints are natural for graphs, yet most semi-supervised clustering algorithms are designed for data represented as vectors [2]. In this paper, we unify vector-based and graph-based approaches. We first show that a recently-proposed objective function for semi-supervised clustering based on Hidden Markov Random Fields, with squared Euclidean distance and a certain class of constraint penalty functions, can be expressed as a special case of the global kernel k-means objective [3]. A recent theoretical connection between global kernel k-means and several graph clustering objectives enables us to perform semi-supervised clustering of data. In particular, some methods have been proposed for semi supervised clustering based on pair wise similarity or dissimilarity information. In this paper, we propose a kernel approach for semi supervised clustering and present in detail two special cases of this kernel approach. The semi supervised clustering problem is thus formulated as an optimization problem for kernel learning [4]. An attractive property of the optimization problem is that it is convex and, hence, has no local optima. While a closed-form solution exists for the first special case, the second case is solved using an iterative majorization procedure to estimate the optimal solution asymptotically. Experimental results based on both synthetic and real-world data show that this new kernel approach is promising for semi supervised clustering [5]. We consider the problem of clustering a given dataset into k clusters subject to an additional set of constraints on relative distance comparisons between the data items. The additional constraints are meant to reflect side-information that is not expressed in the feature vectors, directly. Relative comparisons can express structures at finer level of detail than must-link (ML) and cannot-link (CL) constraints that are commonly used for semi-supervised clustering [6]. Relative comparisons are particularly useful in settings where giving an ML or a CL constraint is difficult because the granularity of the true clustering is unknown. Our main contribution is an efficient algorithm for learning a kernel matrix using the log determinant divergence (a variant of the Bregman divergence) subject to a set of relative distance constraints. Given the learned kernel matrix, a clustering can be obtained by any suitable algorithm, such as kernel k-means. We show empirically that kernels found by our algorithm yield clustering's of higher quality than existing approaches that either use ML/CL constraints or a different means to implement the supervision using relative comparisons [7]. The proposed algorithm detects arbitrary shaped clusters in the dataset and also improves the performance of clustering by minimizing the intra-cluster distance and maximizing the

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

inter-cluster distance which improves the cluster quality.

KEYWORDS: Data Mining, Clustering, Semi-Supervised Clustering, High Dimensional Data, Pairwise Constraints, Kernel K-means, Kernel Matrix, Hidden Markov Random Fields, Similarity.

I. INTRODUCTION

The aim of cluster analysis is to identify either a grouping into a specified number of clusters or a hierarchy of nested partitions. Recently, kernel methods, such as kernel PCA [5], kernel LDA [6] and kernel ICA [7], have been introduced to extract features for recognition. However, kernel parameter selection is difficult. One method is by trial-and-error heuristics, which is easy to implement but not efficient and also causes overfitting problem. The second is using boosting method [8] to learn the combination of kernel functions with different kernel types or different kernel parameters. In [9] one transformed kernel function is discussed, which is good in theory but cannot give an easy and efficient way to obtain the transformation matrix [4]. It is to give an efficient and convenient approach to extract features for high-dimensional data classification problems by generalizing the Gaussian kernel function. We analyze the NN classifier for high-dimensional data classification problems. To obtain better performance, we generalize the Gaussian Kernel to the so-called Data Dependent kernel, which can be easier to calculate compared with the invariant kernel in [9] and obtain better performance than conventional Gaussian kernel and Bayesian Face Matching method [10]. Semi Supervised clustering for classification can be formalized as the problem of inferring a function f(x) from a set of n training samples xi ∈RJ and their corresponding class labels yi. The model developed in this paper is aimed at multi-category classification problems. Of particular interest is classification of high dimensional data, where each sample is defined by hundreds or thousands of measurements, usually concurrently obtained? Such data arise in many application domains [2].

The proposed classifier performs classification of high dimensional data without any pre-processing steps to reduce the number of variables. RKHS methods allow for nonlinear generalization of linear classifiers by implicitly mapping the classification problem into a high dimensional feature space where the data is thought to be linearly separable [3]. Kernel methods were first introduced into statistical learning by (1) and later reintroduced by (2) who constructed the Support Vector Machine, a generalization of the optimal hyper plane algorithm for binary classification. Bayesian treatments of this popular deterministic statistical learning method were motivated by the need to overcome the problem of quantifying uncertainty of SVM predictions, as Bayesian framework allows for probabilistic outputs to be obtained from the predictive distribution [5]. Statistical learning models usually have complex structure and contain parameters that need to be tuned, which is often done via cross-validation. In can be argued, see for example (3), that the Bayesian framework is a natural setting for statistical learning algorithms, as decisions on the complexity of structure and parameter settings can be approached by specifying prior distributions, which formalizes the prior beliefs about which inputs are relevant, what a distribution of a parameter is or how smooth a function is [7].

We propose an adaptive Semi-supervised Clustering Kernel Method based on Metric learning (SCKMM) to mitigate the above problems. Specifically, we first construct an objective function from pair wise constraints to automatically estimate the parameter of the Gaussian kernel [3]. Then, we use pair wise constraint-based K-means approach to solve the violation issue of constraints and to cluster the data. Furthermore, we introduce metric learning into nonlinear semi-supervised clustering to improve separability of the data for clustering [6]. Finally, we perform clustering and metric learning simultaneously. Experimental results on a number of real-world data sets validate the effectiveness of

the proposed method. Semi-supervised clustering, which employs both supervised and unsupervised data for clustering, has received significant amount of attention in recent studies on data mining and machine learning communities [8].

Generally, existing methods for semi-supervised clustering can be grouped into two categories. The first category is the linear method including both metric-based and constraint-based approaches, which either aims to guide clustering process towards a more appropriate data partitioning by making use of pair wise instance constraints [2] or initializes cluster centroids by those labelled instances [4]. Specifically, the idea behind the linear constraint-based approach is to modify the objective function of existing unsupervised clustering so as to satisfy pair wise constraints. The metric-based approach learns a distance metric from pair wise constraints, and then utilizes an existing clustering algorithm to learn the similarity between data by using the learned distance metric [1]. In practice, many real-world applications may involve data along with nonlinear patterns, which may not be effectively dealt with by those linear methods. The second one is the nonlinear method or kernel method, which is recently presented and proved powerful. These methods map the data into the feature space implicitly through a mapping induced by a kernel function such that a cluster assignment is performed with the help of the nonlinear boundary in the original space [3].

Clustering a weighted undirected graph by subsequently removing edges with low weights may be hindered by chaining nodes. Like in a single linkage agglomerative clustering a chain of adjacent nodes may connect two distant clusters and thus hinder a splitting of spatially separated clusters [5]. We enhance the elimination of these nodes by applying a pre-processing based on random walks. In the current semi-supervised learning methods, the selection of initial seeds for the clustering algorithm is done randomly by the user, which may cause the data to be selected not from all the clusters; as a result, achieving a clustering model with a high degree of accuracy is not possible [6]. Innovation in this paper is the more accurate way of selecting the initial seeds for the constrained k-means algorithm which results in increased accuracy of the algorithm [7].

II. RELATED WORK

In recent years various partitional cluster algorithms were adapted to make use of this kind of background information either by constraining the search process or by modifying the underlying metric [1]. It has been shown that including background knowledge might improve the accuracy of cluster results, i.e. the computed clusters better match a given classification of the data. Semi-supervised clustering approaches have been shown to outperform their linear competitors [1] for real world tasks. However, existing nonlinear approaches have the following two disadvantages: 1) they do not necessarily improve the separability of the data for clustering; 2) they cannot effectively solve the violation issue of pair wise constraints. In addition, the selection of the kernel parameters is left to manually tuning due to the fact that no sufficient supervision is provided [4]. In practice, it is well-known that the chosen values of the kernel parameters can largely affect the quality of the clustering results [3]. Therefore, it is necessary to overcome the difficulties described above for both conforming to the user's preferences and improving the performance of semi-supervised clustering. The key challenge in improving separability of the data is how we can solve the violation issue of pair wise constraints such that the unlabeled data can still capture the available cluster information [5]. To this end, we present a new adaptive semi-supervised clustering kernel method based on metric learning, called SCKMM, which simultaneously performs clustering and metric learning by studying several important issues.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

For the linear method for semi-supervised clustering, [2] proposed the constrained K-means algorithm by adjusting the cluster memberships to be consistent with the pair wise constraints. In [7], the authors presented probabilistic models for semi-supervised clustering where the pair wise constraints are incorporated into the clustering algorithms through the Bayesian priors [4]. It proposed a seeded K-means which tries to get better initial cluster centroids from the labelled instances and restricts the clustering process to be consistent with the constraints. It combined the gradient descent method and the iterative projections together as a convex optimization to learn a Mahalanobis distance for the K-means clustering. It proposed the relevant component analysis algorithm which learned a Mahalanobis distance by making use of the must-link constraints only this method is able to learn individual metrics for each cluster, which permits of different shapes [5]. However, the violation of pair wise constraints is not effectively solved in the clustering process. It provided a way to improve the semi-supervised clustering for high-dimensional data by the constraint-guided feature projection instead of the metric learning [6].

Another approach to the semi-supervised clustering is to cluster the data in terms of the kernel function. Kulis et al. It is to a kernel-based semi-supervised clustering. Instead of adding penalty term for pair wise constraints violated, a reward was given for the satisfaction of the constraints in this method [4]. Analogously, it presented an adaptive kernel learning method for semi-supervised clustering which kernelizes the objective function of Basu's method [6] in the input space. Different from Kulis' method that the setting of the kernel parameter was left to manually tuning, Yan's method optimized the parameter of a Gaussian Kernel iteratively during the clustering process [2]. The above two nonlinear methods, Bayesian method for clustering points using a kernel matrix determinant based measure of similarity between data points. It is nonparametric in that prior mass is assigned to all possible partitions of the data. The proposed method explores this neglected aspect by introducing weights to the views which are learned automatically [3].

From the viewpoint of how the view kernels are combined under our framework, semi supervised multiple kernel learning can also be considered as related work. Kernel k-means is an extension of the standard -means clustering algorithm that identifies nonlinearly separable clusters [2]. In order to overcome the cluster initialization problem associated with this method, we propose the global kernel k-means algorithm, a deterministic and incremental approach to kernel-based clustering [4]. Our method adds one cluster at each stage, through a global search procedure consisting of several executions of kernel -means from suitable initializations [5]. This algorithm does not depend on cluster initialization, identifies nonlinearly separable clusters, and, due to its incremental nature and search procedure, locates near-optimal solutions avoiding poor local minima [6].

Furthermore, two modifications are developed to reduce the computational cost that do not significantly affect the solution quality. The proposed methods are extended to handle a weighted data point, which enables their application to graph partitioning. We experiment with several data sets and the proposed approach compares favourably to kernel k-means with random restarts [1]. Kernel k-means [3] is an extension of the standard k-means algorithm that maps data points from the input space to a feature space through a nonlinear transformation and minimizes the clustering error in feature space. Thus, nonlinearly separated clusters in input space are obtained, overcoming the second limitation of k-means. This algorithm suffers from two serious limitations [2]. First the solution depends heavily on the initial positions of the cluster centres, resulting in poor minima, and second it can only find linearly separable clusters [4]. A simple but very popular workaround for the first limitation is the use of multiple restarts where the centres of the clusters are randomly placed to different initial positions and thus better local minima can be found. The number of restarts and also we are never sure if the initializations tried are good so as to obtain a near optimal minimum [5].

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

III. KERNEL APPROACH FOR SEMI SUPERVISED CLUSTERING

A key advantage to this approach is that the algorithm assumes a similarity matrix as input. As given in the derivation, the similarity matrix is the matrix of vector inner products (i.e. the Gram matrix), but one can easily generalize this by applying a kernel function on the vector data to form the similarity matrix [1]. The constraint information gives us important clues about the cluster structure, and this information may be incorporated into the initialization step of the algorithm [3]. After this step, we generate initial clusters by using a farthest-first algorithm. We compute the connected components and then choose k of these as the k initial clusters [4].

The farthest first algorithm selects the largest connected component, and then iteratively chooses the cluster farthest away from the currently chosen cluster, where the distance between clusters is measured by the total pair wise distance between the points in these clusters [5]. Once these k initial clusters are chosen, we initialize all points by placing them in their closest cluster. Note that all the distance computations in this procedure can be performed efficiently in kernel space. We must show that each of these objectives can be written as a special case of the weighted kernel k -means objective function [6]. n other words, we explicitly assume that the distance function δ is in fact the Euclidean distance in some unknown vector space. This is equivalent to assume that the evaluators base their distance-comparison decisions on some implicit features, even if they might not be able to quantify these explicitly [8].

IV. KERNEL APPROACH FOR HIGH DIMENSIONAL DATA

The high dimensional data contains more number of attributes, in which some attributes are more important for representing the data points. In order to identify the important attributes in the dataset, the Kernel Principal Component Analysis is used. The kernel principal components are used for defining the kernel function [3]. By using the kernel function [6], i.e., an appropriate non-linear mapping from the original input space to a higher dimensional feature space, clusters that are non-linearly separable in input space can be extracted. The number of clusters is determined automatically using kernel score of all points. The initial clusters are formed using kernel principal components and the kernel scores [2]. The number of clusters and the initial clusters are used as input parameters for kernel clustering algorithm. This algorithm is expected to offer improvement by providing higher inter-cluster distance and lower intra-cluster distance. Since kernel mapping is applied, the algorithm detects arbitrary shaped clusters [4]. After mapping the samples into a higher feature space by a nonlinear mapping function ϕ, the samples in the feature space are observed as Φ. However, once the kernel function is known, we can easily deal with the nonlinear mapping problem by replacing the mapping functions by the kernel functions [7].

V. PROPOSED WORK

V a. KERNEL MULTIVARIATE ANALYSIS (KMVA)

The framework of kernel MVA (kMVA) algorithms is aimed at extracting nonlinear projections while actually working with linear algebra. Let us first consider a function φ : Rd → F that maps input data into a Hilbert feature space F [1]. The new mapped data set is defined as Φ = [φ(x1), ⋯, φ (xl)]>, and the features extracted from the input data will now be given by Φ0 = ΦU, where matrix U is of size dim (F) ×nf [2]. The direct application of this idea suffers from serious practical limitations when the dimension of F is very large, which is typically the case. To implement practical kernel MVA

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

algorithms we need to rewrite the equations in the first half of Table II in terms of inner products in F only [3]. For doing so, we rely on the availability of a kernel matrix Kx = ΦΦ> of dimension l×l, and on the Representer's Theorem [7], which states that the projection vectors can be written as a linear combination of the training samples, i.e, U = Φ>A, matrix A = [α1,...,αnf] . The same σ has been used for all methods, so that features are extracted from the same mapping of the input data [4]. We can see that the non-linear mapping improves class separability.

$$\text{kPCA}: \quad \mathbf{K}_x\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}$$

$$\text{kPLS}: \quad \begin{pmatrix} \mathbf{0} & \mathbf{K}_x\mathbf{Y} \\ \mathbf{Y}\mathbf{K}_x & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \mathbf{v} \end{pmatrix} = \lambda \begin{pmatrix} \boldsymbol{\alpha} \\ \mathbf{v} \end{pmatrix}$$

$$\text{kOPLS}: \quad \mathbf{K}_x\mathbf{Y}\mathbf{Y}^\top\mathbf{K}_x\boldsymbol{\alpha} = \lambda\mathbf{K}_x\mathbf{K}_x\boldsymbol{\alpha}$$

$$\text{kCCA}: \quad \begin{pmatrix} \mathbf{0} & \mathbf{K}_x\mathbf{Y} \\ \mathbf{Y}\mathbf{K}_x & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \mathbf{v} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{K}_x\mathbf{K}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_y \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \mathbf{v} \end{pmatrix}$$

V b. SEMI-SUPERVISED CLUSTERING USING HIDDEN MARKOV RANDOM FIELDS (SSCHMRF)

Semi-supervised clustering is to use a small number of labelled data to aid the clustering of unlabeled data. We propose a novel iterative algorithm to realize semi-supervised clustering considering local constraints [5]. It considers a labelling problem as a Markov process, where each intermediates state stands for a distribution of labels over data points. The goal is to preserve the locality, namely, local constraints as much as possible in the final clusters [6]. Topologically speaking, the clustering process creates a projection, which brings a certain data point from the configuration space to a graph space. It has been proved that the local embedding relations to k nearest neighbours are preserved for such projection [8].

V b1. ALGORITHM FOR SEMI-SUPERVISED CLUSTERING USING HIDDEN MARKOV RANDOM FIELDS (SSCHMRF)

1: Proposed Semi-supervised clustering

Input:

Set of data points X ←{x1, x2... xN}, xi ∈<d,
Number of clusters C,
Initially labelled data set S = {s1...sc...sC},
Models for each cluster Mc.

Output:

A partitional clustering [16] result {X1...Xc}, where ∪C c Xc = X;
While # (unclustered points) > 0 and! Stop do
1 Denote the set of labelled data: ˆ X;
2 Get set of nearest neighbour's Λ ⊂ X \ {ˆ X} of ˆ X;
3 for each unlabeled point xu ∈ Λ do
4 Knnu ←k-nearest-neighbour (xu, k, X);
5 each labelled point in Knnu votes for the cluster
6 of xu using its own label;
7. Update the set for labelled data ˆ X;

V b2. PROPERTIES OF SEMI-SUPERVISED CLUSTERING USING HIDDEN MARKOV RANDOM FIELDS (SSCHMRF)

1) The distances definition between any pair of points

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

2) The number of nearest neighbours k that should be considered for k-nearest-neighbour ( ). The selection of k depends on the expected sparseness of the target dataset.
3) A stop condition for the cluster. E.g. when the distance to the nearest unlabeled point is greater than a rational threshold, the cluster model should be considered as stable, hence the clustering should not extend the current cluster further.

V c. MAXIMUM-LIKELIHOOD KERNEL DENSITY ESTIMATION (MLKDE)

KDEs estimate the probability density function of a D dimensional dataset X, consisting of N independent and identically distributed samples x1... xN with the sum [9]. This formulation shows that the density estimated with the KDE is non-parametric, since no parametric distribution is imposed on the estimate; instead the estimated distribution is defined by the sum of the kernel functions centred on the data points. KDEs thus require the selection of two design parameters, namely the parametric form of the kernel function and the bandwidth matrix [10]. It has been shown that the efficiencies of kernels with respect to the mean squared error between the true and estimated distribution do not differ significantly and that the choice of kernel function should rather be based on the mathematical properties of the kernel function, since the estimated density function inherits the smoothness properties of the kernel function [11]. Based on the formulation of the leave-one-out ML objective function. We derive a new kernel bandwidth estimator named the minimum leave-one-out entropy (MLE) estimator [12]. (To our knowledge, this is the first attempt where partial derivatives are used to derive variable bandwidths in a closed form solution).

$$p_{\mathbf{H}}(x_i) = \frac{1}{N}\sum_{j=1}^{N} K_{\mathbf{H}_j}\left(\mathbf{x}_i - \mathbf{x}_j|\mathbf{H}_j\right)$$

$$\frac{\partial}{\partial \mathbf{H}_k}\left(l_{\mathbf{H}}(X)\right) = \sum_{i=1}^{N} \frac{\frac{\partial}{\partial \mathbf{H}_k}\left[\frac{1}{N-1}\sum_{j\neq i} K_{\mathbf{H}_j}\left(\mathbf{x}_i - \mathbf{x}_j|\mathbf{H}_j\right)\right]}{p_{\mathbf{H}(-i)}(\mathbf{x}_i)}$$

$$\frac{\partial}{\partial \mathbf{H}_k}\left(l_{MLE}(X)\right) = \sum_{i=1}^{N} \frac{\frac{1}{N-1}\frac{\partial}{\partial \mathbf{H}_k}\left[K_{\mathbf{H}_k}\left(\mathbf{x}_i - \mathbf{x}_k|\mathbf{H}_k\right)\right]}{p_{\mathbf{H}(-i)}(\mathbf{x}_i)}$$

V d. ADAPTIVE-SEMI SUPERVISED-KERNEL-KMEANS (ASSKKM)

The problem of setting kernel's parameters, and of finding in general a proper mapping in feature space, is even more di□cult when no labelled data are provided, and all we have available is a set of pair wise constraints [2]. In this paper we utilize the given constraints to derive an optimization criterion to automatically estimate the optimal kernel's parameters. Our approaches integrate the constraints into the clustering objective function, and optimize the kernel's parameters iteratively while discovering the clustering structure [3]. Specifically, we steer the search for optimal parameter values by measuring the amount of must link and cannot-link constraint violations in feature space. Following the method proposed in [4], we scale the penalty terms by the distances of points that violate the constraints, in feature space. That is, for violation of a must-link constraint (xi,xj), the larger the distance between the two points xi and xj in feature space, the larger the penalty; for violation of a cannot-link constraint (xi,xj), the smaller the distance between the two points xi and xj in feature space, the larger the penalty [5].

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

$$J_{obj} = \sum_{c=1}^{k} \sum_{\mathbf{x}_i, \mathbf{x}_j \in \pi_c} \frac{K(\mathbf{x}_i, \mathbf{x}_i) + K(\mathbf{x}_j, \mathbf{x}_j) - 2K(\mathbf{x}_i, \mathbf{x}_j)}{2|\pi_c|}$$
$$+ \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in ML, l_i \neq l_j} w_{ij}(K(\mathbf{x}_i, \mathbf{x}_i) + K(\mathbf{x}_j, \mathbf{x}_j) - 2K(\mathbf{x}_i, \mathbf{x}_j))$$
$$+ \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in CL, l_i = l_j} \overline{w}_{ij}(K(\mathbf{x}', \mathbf{x}') + K(\mathbf{x}'', \mathbf{x}'') - 2K(\mathbf{x}', \mathbf{x}''))$$
$$- K(\mathbf{x}_i, \mathbf{x}_i) - K(\mathbf{x}_j, \mathbf{x}_j) + 2K(\mathbf{x}_i, \mathbf{x}_j))$$

V d1. ALGORITHM FOR ADAPTIVE-SEMI SUPERVISED-KERNEL-KMEANS

Input:

– Set of data points X = {xi}N i=1
– Set of must-link constraints ML
– Set of cannot-link constraints CL
– Number of clusters k
– Constraint violation costs wij and wij

Output:

– Partition of X into k clusters

Method:

1. Initialize clusters {π (0) c} k c=1 using the given constraints; set t = 0.
2. Repeat Step3 - Step6 until convergence.
3. E-step: Assign each data point xi to a cluster π (t) c so that Jkernel−obj is minimized.
4. M-step (1): Re-compute B (t) cc, forc =1, 2, ⋯, k.
5. M-step (2): Optimize the kernel parameter σ using gradient descent according to the rule: σ (new) = σ (old) −ρ∂Jkernel−obj ∂σ.
6. Increment t by 1.

V e. GLOBAL KERNEL K-MEANS (GKKM)

In this paper we propose the global kernel k-means algorithm for minimizing the clustering error in feature space, defined. Our method builds on the ideas of the global k-means and kernel k-means algorithms [6]. Global kernel kmeans maps the dataset points from input space to a higher dimensional feature space with the help of a kernel matrix as kernel k-means does. In this way nonlinearly separable clusters are found in input space [7]. Also global kernel k-means finds near optimal solutions to the M -clustering problem by incrementally and deterministically adding a new cluster centre at each stage and by applying kernel kmeans as a local search procedure instead of initializing all M clusters at the beginning of the execution [8]. Thus the problems of initializing the cluster centres and getting trapped in poor local minima are also avoided. In a nutshell global kernel k-means combines the advantages of both global kmeans (near optimal solutions) and kernel k-means (clustering in feature space).

$$b_k^n = \sum_{i=1}^{N} \max\left(d_{k-1}^i - \|\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{x}_i)\|^2, 0\right)$$
where $\|\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{x}_i)\|^2 = K_{nn} + K_{ii} - 2K_{ni}$

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

V e1. ALGORITHM FOR GLOBAL KERNEL K-MEANS (GKKM)

Input: Kernel matrix K, number of clusters M
Output: Final clusters $C_1, C_2, \dots, C_M$
// There is no need to solve for one cluster as the solution is trivial and optimal.$C_1^* = X$
1: for all k-clustering problems k=2 to M do
2: for all points $\boldsymbol{x}_n n = 1, \dots, N$ do // suppose $\boldsymbol{x}_n \in C_r^*$
3: Run Kernel k-Means with:
input $(K, k, C_1^*, \dots, C_r = C_r^* - \{\boldsymbol{x}_n\}, \dots, C_{k-1}^*, C_k = \{\boldsymbol{x}_n\})$
output $(C_1^n, \dots, C_1^k, E_k^n)$
4: end for
5: Find $E_k^* = min_n(E_k^n)$ and set $(C_1^*, \dots, C_k^*)$ to the Partitioning corresponding to E_k^*
//This is the solution with k clusters
6: end for
7: return$C_1 = C_1^*, \dots, C_M = C_M^*$ as output of the algorithm

V f. FAST GLOBAL KERNEL K –MEANS (FGKKM)

The fast global kernel k-means algorithm is a simple method for lowering the complexity of the original algorithm. We significantly reduce the complexity by overcoming the need to execute kernel -means times when solving the -clustering problem given the solution for the -clustering problem [9]. Specifically, kernel -means is employed only once and the kth cluster is initialized to include the point that guarantees the greatest reduction in clustering error. The above upper bound is derived from the following arguments [10]. First, when the kth cluster is initialized to include point. Second, since kernel k-means converges monotonically, we can be sure that the final error will never exceed our bound. When using this variant of the global kernel k-means algorithm to solve the M-clustering problem, we must execute kernel k-means M times instead of MN times [11]. In general, this reduction in complexity comes at the cost of finding solutions with higher clustering error than the original algorithm. However, as our experiments indicate, in several problems, the performance of the fast version is comparable to that of global kernel k-means, which makes it a good fast alternative [12].

$$L\left(\{q_j\}_{j=1}^N; \mathcal{X}\right) = \frac{1}{N}\sum_{i=1}^{N} \log\left[\sum_{j=1}^{N} q_j f_j(\mathbf{x}_i)\right]$$
$$= \frac{1}{N}\sum_{i=1}^{N} \log\left[\sum_{j=1}^{N} q_j e^{-\beta d_\varphi(\mathbf{x}_i, \mathbf{x}_j)}\right] + \text{const.}$$

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

V g. WEIGHTED KERNEL K-MEANS (WKKM)

If we associate a positive weight with each data point the weighted kernel k-means algorithm is derived [5]. The weights play a crucial role in proving an equivalence of clustering to graph partitioning, which is the reason we are interested in this version of kernel k-means. Again, suppose we want to solve the M-clustering problem [2]. The objective function is expressed as follows, where is the weight associated with data point. Algorithm can be applied with the slightest modification to get the weighted global kernel k-means algorithm. Specifically, we must include on the input the weights and run weighted kernel k-means instead of kernel k-means [3]. All other steps remain the same. The centre of the cluster in feature space is the weighted average of the points that belong to the cluster [4]. Once again, we can take advantage of the kernel trick and calculate the squared Euclidean distances.

$$\|\phi(\mathbf{x}_i) - \mathbf{m}_k\|^2 = K_{ii} - \frac{2\sum_{j-1}^{N} I(\mathbf{x}_j \in \mathcal{C}_k) w_j K_{ij}}{\sum_{j=1}^{N} I(\mathbf{x}_j \in \mathcal{C}_k) w_j} + \frac{\sum_{j=1}^{N}\sum_{l=1}^{N} I(\mathbf{x}_j \in \mathcal{C}_k) I(\mathbf{x}_l \in \mathcal{C}_k) w_j w_l K_{jl}}{\sum_{j-1}^{N}\sum_{l-1}^{N} I(\mathbf{x}_j \in \mathcal{C}_k) I(\mathbf{x}_l \in \mathcal{C}_k) w_j w_l}.$$

VI. EXPERIMENTS

In this section, we empirically demonstrate that our proposed semi-supervised clustering for kernel approach is both e□cient and e□ective.

VII a. DATASETS

The data sets used in our experiments include six UCI data sets1. Here is some basic information of those data sets. Table 5 summarizes the basic information of those data sets.

• Balance. This data set was generated to model psychological experimental results. There are totally 625 examples that can be classified as having the balance scale tip to the right, tip to the left, or be balanced.

• Iris. This data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.

• Ionosphere. It is a collection of the radar signals belonging to two classes. The data set contains 351 objects in total, which are all 34-dimensional.

• Soybean. It is collected from the Michalski's famous soybean disease databases, which contains 562 instances from 19 classes.

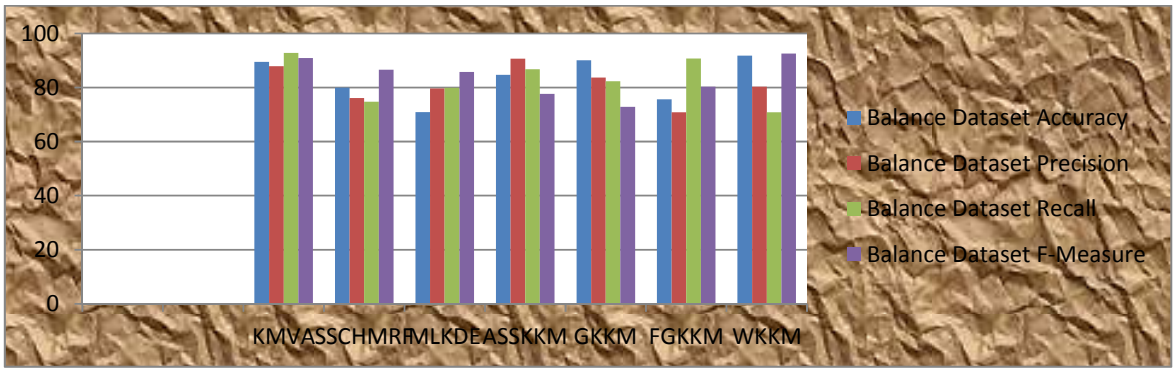
Datasets	Size	Classes	Dimensions
Balance	625	3	4
Iris	150	3	4
Ionosphere	351	2	34
Soybean	562	19	35

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

VIII. EXPERIMENTAL RESULTS

VIII a. BALANCE DATASET RESULTS

Balance Dataset				
Algorithm	Accuracy	Precision	Recall	F-Measure
KMVA	89.45	87.91	92.77	90.89
SSCHMRF	79.91	76.08	74.78	86.56
MLKDE	70.92	79.67	79.89	85.78
ASSKKM	84.67	90.67	86.78	77.67
GKKM	90.07	83.66	82.33	72.88
FGKKM	75.66	70.89	90.75	80.34
WKKM	91.78	80.34	70.89	92.55



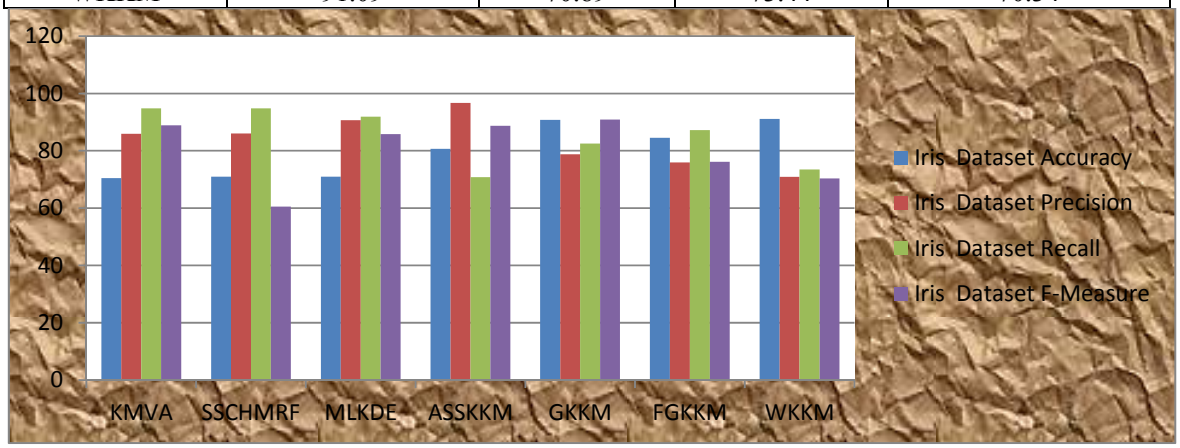
The above graph shows that performance of Balance dataset. The Accuracy of WKKM algorithm is 91.78 which is higher when compare to other six (KMVA, SSCHMRF, MLKDE, ASSKKM, GKKM, FGKKM) algorithms. The Precision of ASSKKM algorithm is 90.67 which is higher when compare to other six (KMVA, SSCHMRF, MLKDE, WKKM, GKKM, FGKKM) algorithms. The Recall of KMVA algorithm is 92.77 which is higher when compare to other six (WKKM, SSCHMRF, MLKDE, ASSKKM, GKKM, FGKKM) algorithms. The F-Measure of WKKM algorithm is 92.55 which is higher when compare to other six (KMVA, SSCHMRF, MLKDE, ASSKKM, GKKM, FGKKM) algorithms.

VIII b. IRIS DATASET RESULTS

Iris Dataset				
Algorithm	Accuracy	Precision	Recall	F-Measure
KMVA	70.45	85.91	94.79	88.89
SSCHMRF	70.91	86.08	94.78	60.56
MLKDE	70.92	90.67	91.89	85.78
ASSKKM	80.67	96.67	70.78	88.67
GKKM	90.78	78.76	82.54	90.89
FGKKM	84.56	75.9	87.23	76.12

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

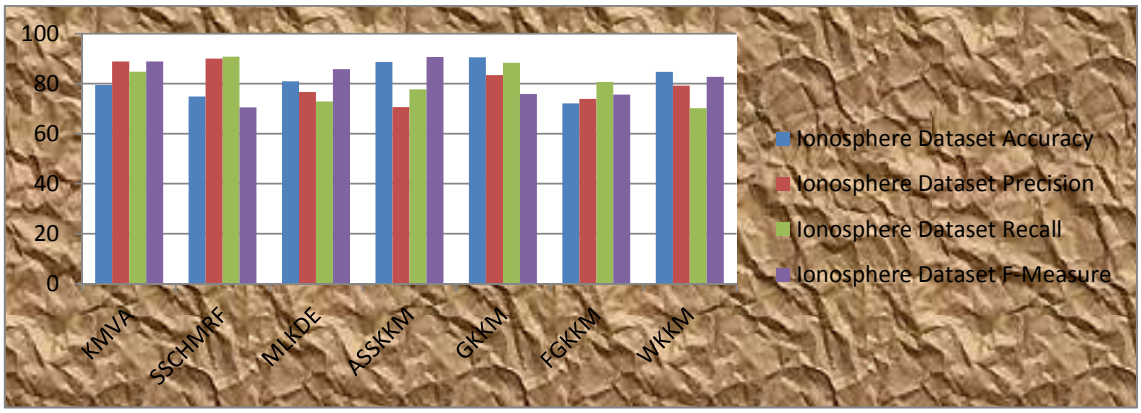
WKKM	91.09	70.89	73.44	70.34



The above graph shows that performance of Iris dataset. The Accuracy of WKKM algorithm is 91.09 which is higher when compare to other six (KMVA, SSCHMRF, MLKDE, ASSKKM, GKKM, FGKKM) algorithms. The Precision of ASSKKM algorithm is 96.67 which is higher when compare to other six (KMVA, SSCHMRF, MLKDE, WKKM, GKKM, FGKKM) algorithms. The Recall of KMVA algorithm is 94.78 which is higher when compare to other six (WKKM, SSCHMRF, MLKDE, ASSKKM, GKKM, FGKKM) algorithms. The F-Measure of GKKM algorithm is 90.89 which is higher when compare to other six (KMVA, SSCHMRF, MLKDE, ASSKKM, WKKM, FGKKM) algorithms.

VIII c. IONOSPHERE DATASET RESULTS

Ionosphere Dataset				
Algorithm	Accuracy	Precision	Recall	F-Measure
KMVA	79.45	88.91	84.77	88.89
SSCHMRF	74.91	90.08	90.78	70.56
MLKDE	80.98	76.67	72.89	85.78
ASSKKM	88.67	70.67	77.78	90.67
GKKM	90.56	83.45	88.34	75.89
FGKKM	72.12	73.9	80.67	75.66
WKKM	84.67	79.23	70.21	82.78

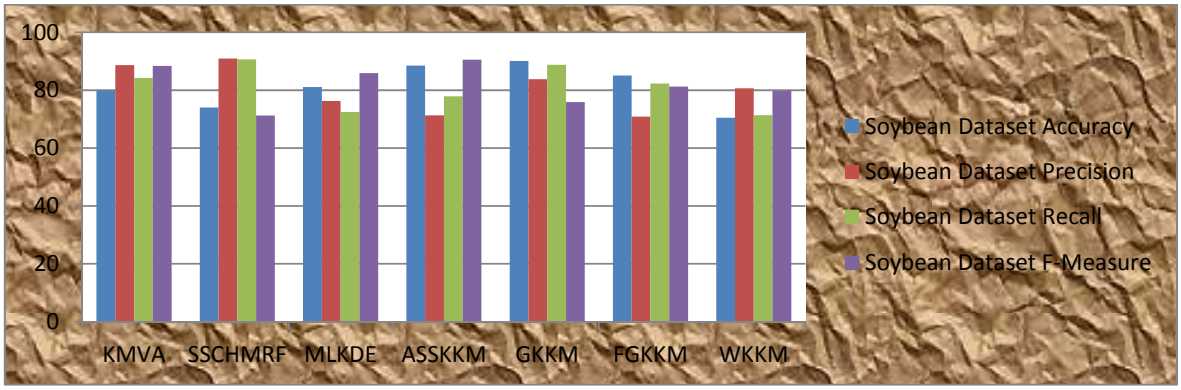


https://sites.google.com/site/ijcsis/
ISSN 1947-5500

The above graph shows that performance of Ionosphere dataset. The Accuracy of GKKM algorithm is 90.56 which is higher when compare to other six (KMVA, SSCHMRF, MLKDE, ASSKKM, WKKM, FGKKM) algorithms. The Precision of SSCHMRF algorithm is 90.08 which is higher when compare to other six (KMVA, WKKM, MLKDE, ASSKKM, GKKM, FGKKM) algorithms. The Recall of SSCHMRF algorithm is 90.78 which is higher when compare to other six (KMVA, WKKM, MLKDE, ASSKKM, GKKM, FGKKM) algorithms. The F-Measure of ASSKKM algorithm is 90.67 which is higher when compare to other six (KMVA, SSCHMRF, MLKDE, WKKM, GKKM, FGKKM) algorithms.

VIII d. SOYBEAN DATASET RESULTS

Soybean Dataset				
Algorithm	Accuracy	Precision	Recall	F-Measure
KMVA	79.89	88.65	84.23	88.34
SSCHMRF	74.03	90.89	90.67	71.23
MLKDE	81.08	76.32	72.45	85.9
ASSKKM	88.54	71.32	77.89	90.56
GKKM	90.08	83.78	88.78	75.9
FGKKM	85.09	70.89	82.33	81.23
WKKM	70.45	80.67	71.33	79.78



The above graph shows that performance of Soybean dataset. The Accuracy of GKKM algorithm is 90.08 which is higher when compare to other six (KMVA, WKKM, MLKDE, ASSKKM, WKKM, FGKKM) algorithms. The Precision of SSCHMRF algorithm is 90.89 which is higher when compare to other six (KMVA, WKKM, MLKDE, ASSKKM, GKKM, FGKKM) algorithms. The Recall of SSCHMRF algorithm is 90.67 which is higher when compare to other six (KMVA, WKKM, MLKDE, ASSKKM, GKKM, FGKKM) algorithms. The F-Measure of ASSKKM algorithm is 90.56 which is higher when compare to other six (KMVA, WKKM, MLKDE, SSCHMRF, GKKM, FGKKM) algorithms.

IX. CONCLUSION

We have shown that none of the ML estimators investigated performed optimally on all tasks, and based on theoretical motivations confirmed with empirical results it is clear that the optimal estimator depends on the degree of scale variation between features and the degree of changes in scale variation with features [2]. The results show that the full covariance ML-KBE and global MLE estimators (which estimate an identical full and diagonal covariance matrix respectively for each kernel)

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

performed optimally on three of the four datasets investigated, while the MLE estimator (which estimates a unique bandwidth for each kernel) performed optimally on only one dataset [3]. This is an interesting case of the bias-variance trade off: having fewer parameters, the full covariance ML-KBE and global MLE estimators are less flexible than the MLE and MLL estimators; however, those parameters can be estimated with greater reliability, leading to the best performance in many cases [4].

From the theoretical and empirical results in this work it is clear that the optimal estimator should somehow function like the full covariance ML-KBE estimator in regions with low spatial variability, and must function like the MLE estimator and be able to adapt bandwidths in regions with high spatial variability, especially for outliers [5]. We therefore believe that it would be interesting to investigate a hybrid kernel bandwidth estimator by first detecting and removing outliers. We have presented the global kernel k-means clustering algorithm, an algorithm that maps data points from input space to a higher dimensional feature space through the use of a kernel function and optimizes the clustering error in the feature space by locating near optimal minima [6]. The main advantages of this method are its deterministic nature, which makes it independent of cluster initialization, and the ability to identify nonlinearly separable clusters in input space. Another important feature of the proposed algorithm is that in order to solve the M -clustering problem all intermediate clustering problems, with 1… M clusters, are solved [7]. This may prove useful in problems where we seek the actual number of clusters. Moreover we developed the fast global kernel k-means algorithm which considerably reduces the computational cost of the original algorithm without degrading significantly the quality of the solution [8].

We proposed a new adaptive semi-supervised Kernel K-Means algorithm. Our approach integrates the given constraints with the kernel function, and is able to automatically embed, during the clustering process, the optimal non-linear similarity within the feature space [9]. As a result, the proposed algorithm is capable of discovering clusters with non-linear boundaries in input space with high accuracy. Our technique enables the practical utilization of powerful kernel-based semi-supervised clustering approaches by providing a mechanism to automatically set the involved critical parameters [10]. In this approach, we observed that weighted kernel k-means updating of views according to their conveyed information resulting higher clustering accuracy, if the sparsity of the weights is appropriately moderated [11]. The threads concept we used in the algorithm for initializing clusters is working good and resulting in reduced run time and also provided us an interesting direction for developing and improving multi-view clustering algorithms [12].

X. FUTURE WORK

Clustering can then be performed on the remaining data and the full covariance ML-KBE bandwidth estimator can be used to estimate a unique full covariance kernel bandwidth for each cluster-each kernel will thus make use of the full covariance bandwidth matrix of the cluster to which it is assigned [1]. The MLE estimator can then be used to estimate a unique bandwidth for each outlier; since the MLE estimator can model scale variations, this will ensure that outliers have sufficiently wide bandwidths [2]. This proposed hybrid approach will thus generally function like the full covariance ML-KBE estimator in the clustered regions and has the added ability to change the direction of local scale variations between clusters [4]. This estimator will also be capable of making the bandwidths of kernel function centred on outliers sufficiently wide. We therefore propose to implement this hybrid ML kernel bandwidth estimator in future work and perform a comparative study between this

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

approach, the MLE, full covariance ML-KBE and the first hybrid approach proposed above [3]. As for future work a possible direction is the use of parallel processing to accelerate the global kernel k-means algorithm since the local search performed when solving the k clustering problem requires running kernel k-means N times and these executions are independent of each other [8]. Another important issue is the development of theoretical foundations behind the assumptions of the method. As already mentioned kernel k-means is closely related to spectral clustering. So extending the proposed algorithm by associating weights with each data point, following the ideas in [6], and using it to solve graph cut problems and comparing it to spectral methods is another possible research direction. Finally we plan to use the global kernel k-means in conjunction with criteria and techniques for estimating the optimal number of clusters [5].

As for future work, a possible direction is the use of parallel processing to accelerate the global kernel k-means algorithm, since the local search performed when solving the -clustering problem requires running kernel k-means times and these executions are independent of each other [7]. Another important issue is the development of theoretical results concerning the near-optimality of the obtained solutions [1]. Also, we plan to use global kernel k-means in conjunction with criteria and techniques for estimating the optimal number of clusters and integrate it with other exemplar-based techniques. Finally, the application of this algorithm to graph partitioning needs further investigation and a comparison with spectral methods and other graph clustering techniques is required [10]. In our future work we will consider active learning as a methodology to generate constraints which are most informative. We will also consider other kernel functions (e.g., polynomial) in our future experiments, as well as combinations of di erent types of kernels [9]. In future, we planned to parallelize the same algorithm using multi-core processing environment for reducing the run time to overcome the synchronization problem raised from thread concept. And we also planned to extend the algorithm for larger multi-modal Datasets (Big Data) [11].

There are a number of interesting potential avenues for future research in kernel methods for semi-supervised clustering. Along with learning the kernel matrix before the clustering, one could additionally incorporate kernel matrix learning in to the clustering iteration ,as was done [4] .One way to incorporate learning in to the clustering step is to devise away to learn the weights in weighted kernel k-means, by using the constraints [2]. Another possibility would be to explore the generalization of techniques in this paper beyond squared Euclidean distance, for unifying semi-supervised graph clustering with kernel-based clustering on an HMRF using other popular clustering distortion measures, e.g., KL-divergence, or cosine distance [12]. We would also like to extend the work of to explore techniques of active learning and model selection in the context of semi-supervised kernel clustering. To speed up the algorithm further, the multi-level methods of can be employed [13].

REFERENCES

[1] A.Likas, N.Vlassis ,and J.J.Verbeek, “The global k-means clustering algorithm,” Pattern Recognition., vol. 36, no. 2, pp. 451–461, Feb. 2013.
[2] I. S. Dhillon, Y. Guan, and B. Kulis, “Kernel k-means, spectral clustering and normalized cuts,” in Proc. 10th ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining, pp. 551–556, 2014.
[3] I.S.Dhillon, Y.Guan, and B.Kulis, “A unified view of kernel k-means, spectral clustering and graph cuts,”Univ.Texas, Austin, TX, Tech.Rep. TR-04-25, 2009.
[4] G. Tzortzis and A. Likas, “The global kernel -means clustering algorithm,” in Proc. IEEE Int. Joint Conf. Neural Netw., pp. 1977–1984, 2011.
[5] I. S. Dhillon, Y. Guan, and B. Kulis, “Weighted graph cuts without eigenvectors: a multilevel approach,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 29, no. 11, pp. 1944-1957, Nov. 2012.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

[6] R. Zhang and A. I. Rudnicky, “A large scale clustering scheme for kernel k-means,” in Proc. 16th Int. Conf. on Pattern Recognition, pp. 289-292, 2012.
[7] Boykov Y, Veksler O, Zabih R, “Markov Random fields with e□cient approximations”, IEEE Computer Vision and pattern Recognition Conference, 2010.
[8] Tzortzis, Grigorios, and Aristidis Likas. "Kernel Based Weighted Multi-view Clustering." In ICDM, pp. 675-684. 2012.
[9] Bilenko M, & Basu S,”A comparison of inference techniques for semi-supervised clustering with hidden Markov random fields”, In Proceedings of the ICML-2014 workshop on statistical relational learning and its connections to other fields (SRL-2001), Banff, Canada, 2014.
[10] Kulis B, Basu S, Dhillon I, & Mooney R,” Semi-supervised graph clustering: a kernel approach”, In Proceedings of the 22nd international conference on machine learning (pp. 457–464), 2011.
[11] Wagsta□ K., Cardie C, Rogers S, Schroedl S,” Constrained k-means clustering with background knowledge”, In ICML, Morgan Kaufmann, 577–584, 2010.
[12] Shrivastava A, Singh S, Gupta A,” Constrained semi-supervised clustering using attributes and comparative attributes”, In: ECCV,2012.
[13] Weston J, C. Leslie, D. Zhou, A. Elisseeff, and W. S. Noble ,”Cluster kernels for semi-supervised protein classification”, Advances in Neural Information Processing Systems 17,2011.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500