An Adaptive Kernel Method for Semi-Supervised Clustering

Bojun Yan and Carlotta Domeniconi

Department of Information and Software Engineering
George Mason University
Fairfax, Virginia 22030, USA
byan@gmu.edu, carlotta@ise.gmu.edu

Abstract. Semi-supervised clustering uses the limited background knowledge to aid unsupervised clustering algorithms. Recently, a kernel method for semi-supervised clustering has been introduced, which has been shown to outperform previous semi-supervised clustering approaches. However, the setting of the kernel's parameter is left to manual tuning, and the chosen value can largely affect the quality of the results. Thus, the selection of kernel's parameters remains a critical and open problem when only limited supervision, provided in terms of pairwise constraints, is available. In this paper, we derive a new optimization criterion to automatically determine the optimal parameter of an RBF kernel, directly from the data and the given constraints. Our approach integrates the constraints into the clustering objective function, and optimizes the parameter of a Gaussian kernel iteratively during the clustering process. Our experimental comparisons and results with simulated and real data clearly demonstrate the effectiveness and advantages of the proposed algorithm.

1 Introduction

As a recent emerging technique, semi-supervised clustering has attracted significant research interest. Compared to traditional clustering algorithms, which only use unlabeled data, semi-supervised clustering employs both unlabeled and supervised data to obtain a partitioning that conforms more closely with the user's preferences. Several recent papers have discussed this problem [16, 8, 1, 18, 2, 12].

In semi-supervised clustering, limited supervision is provided as input. The supervision can have the form of labeled data or pairwise constraints. In many applications it is natural to assume that pairwise constraints are available [1, 16]. For example, in protein interaction and gene expression data [13], pairwise constraints can be derived from the background domain knowledge. Similarly, in information and image retrieval, it is easy for the user to provide feedback concerning a qualitative measure of similarity or dissimilarity between pairs of objects. Thus, in these cases, although class labels may be unknown, a user can still specify whether pairs of points belong to the same cluster or to different

ones. Furthermore, a set of classified points implies an equivalent set of pairwise constraints, but not vice versa.

Recently, a kernel method for semi-supervised clustering has been introduced [12]. This technique extends semi-supervised clustering to a kernel space, thus enabling the discovery of clusters with non-linear boundaries in input space. While a powerful technique, the applicability of a kernel-based semi-supervised clustering approach is limited in practice, due to the critical settings of kernel's parameters. In fact, the chosen parameter values can largely affect the quality of the results. While solutions have been proposed in supervised learning to estimate the optimal kernel's parameters, the problem presents open challenges when no labeled data are provided, and all we have available is a set of pairwise constraints.

In this paper, we derive a new optimization criterion to automatically estimate the optimal parameter of a Gaussian kernel, directly from the data and the given constraints. Our approach integrates the constraints into the clustering objective function, and optimizes the parameter of a Gaussian kernel iteratively during the clustering process. As a result, our technique is able to automatically embed, during the clustering process, the optimal non-linear similarity within the feature space. This makes our adaptive technique capable of discovering clusters with non-linear boundaries in input space with high accuracy, as demonstrated in our experiments. Our proposed method enables the practical utilization of powerful kernel-based semi-supervised clustering approaches by providing a mechanism to automatically set the involved critical parameters.

The rest of the paper is organized as follows. Section 2 provides the necessary background on kernel-based clustering and semi-supervised clustering. Section 3 motivates our approach, and discusses the details of our algorithm. Section 4 describes our experimental settings and results. Section 5 discusses the related work, and finally we provide conclusions and future research directions in Section 6.

2 Background

This section introduces the necessary background on kernel-based clustering and semi-supervised clustering.

2.1 Kernel KMeans

Let X be a dataset of N samples and D dimensions, $X = \{\mathbf{x}_i\}_{i=1}^{N} \subseteq \Re^D$. Let $\phi : \Re^D \rightarrow \Re^{D'}$ be a non-linear mapping function, which maps data from the input (D dimensional) space to a feature space (D' dimensional), with $D' > D$. The Kernel KMeans algorithm generates a partition $\{\pi_c\}_{c=1}^{k}$ of X (π_c represents the c^{th} cluster) so that the objective function $\sum_{c=1}^{k} \sum_{\mathbf{x}_i \in \pi_c} \|\phi(\mathbf{x}_i) - \mathbf{m}_c^{\phi}\|$ is minimized, where $\mathbf{m}_c^{\phi} = \frac{1}{|\pi_c|} \sum_{\mathbf{x}_i \in \pi_c} \phi(\mathbf{x}_i)$ represents the centroid of cluster π_c in feature space. The key issue of Kernel-KMeans is the computation of distances in feature space. The distance of a point $\mathbf{x}_i$ from $\mathbf{m}_c^{\phi}$ in feature space can be

expressed as: $\|\phi(\mathbf{x}_i) - \mathbf{m}_c^\phi\| = A_{ii} + B_{cc} - D_{ic}$, where $A_{ii} = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_i)$, $D_{ic} = \frac{2}{|\pi_c|} \sum_{\mathbf{x}_j \in \pi_c} \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$, and $B_{cc} = \frac{1}{|\pi_c|^2} \sum_{\mathbf{x}_j, \mathbf{x}_{j'} \in \pi_c} \phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}_{j'})$.

Following the standard SVM method, we can represent the dot product of points in kernel space using an appropriate Mercer kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ [15]. Since data points always appear in the form of dot products, the terms for distance computation can be rewritten using the kernel trick: $A_{ii} = K(\mathbf{x}_i, \mathbf{x}_j)$, $D_{ic} = \frac{2}{|\pi_c|} \sum_{\mathbf{x}_j \in \pi_c} K(\mathbf{x}_i, \mathbf{x}_j)$, and $B_{cc} = \frac{1}{|\pi_c|^2} \sum_{\mathbf{x}_j, \mathbf{x}_{j'} \in \pi_c} K(\mathbf{x}_j, \mathbf{x}_{j'})$. We note that A_{ii} is common to every cluster, thus we can avoid calculating it, while B_{cc} must be calculated once in each iteration.

2.2 HMRF Model and Kernel-based Semi-supervised Clustering

In semi-supervised clustering, we are given a set of pairwise constraints: must-link $ML = \{(\mathbf{x}_i, \mathbf{x}_j)\}$ and cannot-link $CL = \{(\mathbf{x}_i, \mathbf{x}_j)\}$. The goal is to partition the data into k clusters so that a given measure of distorsion between each point and the corresponding cluster representative is minimized, and, at the same time, the smallest number of constraint violation is achieved. Basu et al. (2004) [2] proposed a framework for semi-supervised clustering based on Hidden Markov Random Fields (HMRFs). Considering the squared Euclidean distance as a measure of cluster distortion, and the generalized Potts potential as constraint violation potential, the semi-supervised clustering objective can be expressed as [2]:

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

where $\mathbf{m}_c$ is the centroid of cluster π_c, ML is the set of must-link constraints, CL is the set of cannot-link constraints, w_{ij} and $\overline{w}_{ij}$ are the penalty costs for violating a must-link and a cannot-link constraint respectively, and l_i represents the cluster label of $\mathbf{x}_i$.

Kulis et al. (2005) [12] extended this framework to a kernel-based semi-supervised clustering. Instead of adding a penalty term for a must-link violation, a reward is given for the satisfaction of the constraint. This is achieved by subtracting the corresponding penalty term from the objective:

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

The algorithm derived in [12] (called SS-Kernel-KMeans), when combined with the Gaussian kernel, is shown to outperform the HMRF-KMeans approach [2], and SS-Kernel-KMeans combined with a linear kernel. However, the setting of the kernel's parameter is left to manual tuning, and the chosen value can largely affect the quality of the results. Thus, the selection of kernel's parameters remains a critical and open problem when only limited supervision is available. This leads to the motivation of our approach discussed in the next Section.

3 Adaptive Kernel-based Semi-supervised Clustering

In kernel-based learning algorithms it is important that the kernel function in use conforms with the learning target. For classification, the distribution of data in feature space should be correlated to the label distribution. Similarly, in semi-supervised clustering, one wishes to learn a kernel that maps pairs of points subject to a must-link constraint close to each other in feature space, and maps points subject to a cannot-link constraint far apart in feature space.

The authors in [9] introduce the concept of *kernel alignment* to measure the correlation between the groups of data in feature space and the labeling to be learned. In [17], a Fisher discriminant rule is used to estimate the optimal spread parameter of a Gaussian kernel. The selection of kernel's parameters is indeed a critical problem. For example, empirical results in the literature have shown that the value of the spread parameter σ of a Gaussian kernel can strongly affect the generalization performance of an SVM. Values of σ which are too small or too large lead to poor generalization capabilities. When $\sigma \to 0$, the kernel matrix becomes the identity matrix. In this case, the resulting optimization problem gives Lagrangians which are all 1s, and therefore every point becomes a support vector. On the other hand, when $\sigma \to \infty$, the kernel matrix has entries all equal to 1, and thus each point in feature space is maximally similar to each other. In both cases, the machine will generalize very poorly.

The problem of setting kernel's parameters, and of finding in general a proper mapping in feature space, is even more difficult when no labeled data are provided, and all we have available is a set of pairwise constraints. In this paper we utilize the given constraints to derive an optimization criterion to automatically estimate the optimal kernel's parameters. Our approach integrates the constraints into the clustering objective function, and optimize the kernel's parameters iteratively while discovering the clustering structure. 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 [2, 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 $(\mathbf{x}_i, \mathbf{x}_j)$, the larger the distance between the two points $\mathbf{x}_i$ and $\mathbf{x}_j$ in feature space, the larger the penalty; for violation of a cannot-link constraint $(\mathbf{x}_i, \mathbf{x}_j)$, the smaller the distance between the two points $\mathbf{x}_i$ and $\mathbf{x}_j$ in feature space, the larger the penalty. According to these rules, we can formulate the penalty terms as follows:

$$P_{ML}(\mathbf{x}_i, \mathbf{x}_j) = w_{ij} \|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 1(l_i \neq l_j) \tag{1}$$

$$P_{CL}(\mathbf{x}_i, \mathbf{x}_j) = \overline{w}_{ij}((D^{\phi}_{max})^2 - \|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2) 1(l_i \neq l_j) \tag{2}$$

D^{ϕ}_{max} is the maximum distance between any pair of points in feature space; it ensures that the penalty for violated cannot-link constraints is non-negative. By combining these two penalty terms with the objective function of Kernel KMeans, we obtain the objective function for our adaptive semi-supervised ker-

nel KMeans (Adaptive-SS-Kernel-KMeans) approach:

$$J_{obj} = \sum_{c=1}^{k} \sum_{\mathbf{x}_i \in \pi_c} (\|\phi(\mathbf{x}_i) - \mathbf{m}_c^{\phi}\|^2) + \sum_{(\mathbf{x}_i,\mathbf{x}_j) \in ML, l_i \neq l_j} w_{ij} \|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 + \sum_{(\mathbf{x}_i,\mathbf{x}_j) \in CL, l_i = l_j} \overline{w}_{ij} ((D_{max}^{\phi})^2 - \|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2) \quad (3)$$

Suppose $\mathbf{x}^{'}$ and $\mathbf{x}^{''}$ are the farthest points in feature space. We use the equality $\sum_{c=1}^{k} \sum_{\mathbf{x}_i \in \pi_c} \|\mathbf{x}_i - \mathbf{m}_c\|^2 = \sum_{c=1}^{k} \sum_{\mathbf{x}_i, \mathbf{x}_j \in \pi_c} \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2|\pi_c|}$ to re-formulate Equation (3) as follows:

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

By expanding the distance computation in feature space $\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2$, and using the kernel trick $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$, we obtain:

$$\begin{aligned} J_{obj} = & \sum_{c=1}^{k} \sum_{\mathbf{x}_i, \mathbf{x}_j \in \pi_c} \frac{K(\mathbf{x}_i, \mathbf{x}_i) + K(\mathbf{x}_j, \mathbf{x}_j) - 2K(\mathbf{x}_i, \mathbf{x}_j)}{2|\pi_c|} \\ & + \sum_{(\mathbf{x}_i,\mathbf{x}_j) \in ML, l_i \neq l_j} w_{ij} (K(\mathbf{x}_i, \mathbf{x}_i) + K(\mathbf{x}_j, \mathbf{x}_j) - 2K(\mathbf{x}_i, \mathbf{x}_j)) \\ & + \sum_{(\mathbf{x}_i,\mathbf{x}_j) \in CL, l_i = l_j} \overline{w}_{ij} (K(\mathbf{x}', \mathbf{x}') + K(\mathbf{x}'', \mathbf{x}'') - 2K(\mathbf{x}', \mathbf{x}'') \\ & - K(\mathbf{x}_i, \mathbf{x}_i) - K(\mathbf{x}_j, \mathbf{x}_j) + 2K(\mathbf{x}_i, \mathbf{x}_j)) \end{aligned} \quad (4)$$

Let us consider the Gaussian kernel function: $K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2))$. (From now on we utilize the Gaussian kernel to derive our algorithm, since it has excellent learning properties. Other kernel functions can be used as well.) We want to minimize J_{obj} with respect to the kernel parameter σ. As observed earlier, when $\sigma \to \infty$, all points in feature space are maximally similar to each other, and the objective function (4) is trivially minimized. To avoid this degenerate case, we add the following constraint:

$$\sum_{\mathbf{x}_i \in X} \|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_r)\|^2 \geq Const \quad (5)$$

where $\mathbf{x}_r$ is a point randomly selected from X. By incorporating constraint (5) into the objective function, and applying the kernel trick for distance computation in feature space, we finally obtain:

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

Given $\frac{\partial K(\mathbf{x}_i,\mathbf{x}_j)}{\partial \sigma} = exp(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^3}$, we compute $\frac{\partial J_{kernel-obj}}{\partial \sigma}$:

$$\begin{aligned}
\frac{\partial J_{kernel-obj}}{\partial \sigma} = & \sum_{c=1}^{k} \sum_{(\mathbf{x}_i,\mathbf{x}_j) \in \pi_c} -\frac{1}{|\pi_c|} exp(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^3} \\
& - \sum_{(\mathbf{x}_i,\mathbf{x}_j) \in ML, l_i \neq l_j} 2w_{ij} exp(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^3}) \\
& + \sum_{(\mathbf{x}_i,\mathbf{x}_j) \in ML, l_i \neq l_j} 2\overline{w}_{ij}(exp(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^3} \\
& - exp(\frac{-\|\mathbf{x}' - \mathbf{x}''\|^2}{2\sigma^2})\frac{\|\mathbf{x}' - \mathbf{x}''\|^2}{\sigma^3}) \\
& + \sum_{\mathbf{x}_i \in X} 2exp(\frac{-\|\mathbf{x}_i - \mathbf{x}_r\|^2}{2\sigma^2})\frac{\|\mathbf{x}_i - \mathbf{x}_r\|^2}{\sigma^3}
\end{aligned} \tag{6}$$

In the following we derive an EM-based strategy to optimize $J_{kernel-obj}$ by gradient descent.

3.1 EM-based Strategy

To minimize the objective function $J_{kernel-obj}$, we use an EM-based strategy. We initialize the clusters utilizing the mechanism proposed in [12]: we take the transitive closure of the constraints to form neighborhoods, and then perform a farthest-first traversal on these neighborhoods to get the k initial clusters. We ensure that the same set of constraints is given to the competitive algorithm in our experiments.

E-step: The algorithm assigns data points to clusters so that the objective function $J_{kernel-obj}$ is minimized. Since the objective function integrates the given must-link and cannot-link constraints, it is minimized by assigning each point to the cluster with the closest centroid (first term of $J_{kernel-obj}$) which causes a minimal penalty for violations of constraints (second and third term of $J_{kernel-obj}$). The fourth term of $J_{kernel-obj}$ is constant during the assignment of data points in each iteration. When updating the cluster assignment of a given point, the assignment for the other points is kept fixed [3, 19]. During each iteration, data points are re-ordered randomly. The process is repeated until no change in point assignment occurs.

M-step: The algorithm re-computes the cluster representatives. In practice, since we map data in kernel space and do not have access to the coordinates of cluster representatives, we re-compute the term B_{cc} (as discussed in Section 2.1), which will be used to re-assign points to clusters in the E-step. Constraints are not used in this step. Therefore, only the first term of $J_{kernel-obj}$ is minimized.

We note that all the steps so far are executed with respect to the current feature space. We now optimize the feature space by optimizing the kernel parameter σ. To this extent, we apply the gradient descent rule to update the parameter σ of the Gaussian kernel: $\sigma^{(new)} = \sigma^{(old)} - \rho \frac{\partial J_{kernel-obj}}{\partial \sigma}$, where ρ is a scalar step length parameter optimized via a line-search method. The expression for $\frac{\partial J_{kernel-obj}}{\partial \sigma}$ is given in Equation (6).

A description of the algorithm (Adaptive-SS-Kernel-KMeans) is provided in Figure 1.

Algorithm: *Adaptive-SS-Kernel-KMeans*
Input:

- Set of data points $X = \{\mathbf{x}_i\}_{i=1}^{N}$
- Set of must-link constraints ML
- Set of cannot-link constraints CL
- Number of clusters k
- Constraint violation costs w_{ij} and $\overline{w}_{ij}$

Output:

- Partition of X into k clusters

Method:

1. Initialize clusters $\{\pi_c^{(0)}\}_{c=1}^{k}$ using the given constraints; set $t = 0$.
2. Repeat Step3 - Step6 until convergence.
3. E-step: Assign each data point $\mathbf{x}_i$ to a cluster $\pi_c^{(t)}$ so that $J_{kernel-obj}$ is minimized.
4. M-step(1): Re-compute $B_{cc}^{(t)}$, for $c = 1, 2, \cdots, k$.
5. M-step(2): Optimize the kernel parameter σ using gradient descent according to the rule: $\sigma^{(new)} = \sigma^{(old)} - \rho \frac{\partial J_{kernel-obj}}{\partial \sigma}$.
6. Increment t by 1.

Fig. 1. Adaptive-SS-Kernel-KMeans

4 Experimental Evaluation

4.1 Datasets

We performed experiments on one simulated dataset and four real datasets. (1) The simulated dataset contains two clusters in two dimensions distributed as

concentric circles (See Figure 2(a)). Each cluster contains 200 points. (2) **Digits**: This dataset is the pendigits handwritten character recognition dataset from the UCI repository[1] [5]. 10% of the data were chosen randomly from the three classes $\{3, 8, 9\}$ as done in [12]. This results in 317 points and 16 dimensions. (3) **Spectf**: This dataset is also from the UCI repository [5]. It describes the diagnosis of cardiac Single Proton Emission Computed Tomography (SPECT) images. Each patient is classified into one of two categories: normal or abnormal. 267 SPECT image sets (patients) were processed to extract features that summarize the original SPECT images. As a result, 44 continuous features were created for each patient. (4) **Vowel**: This dataset concerns the recognition of eleven steady state vowels of British English, using a specified training set of lpc derived log area ratios[2]. Three class corresponding to the vowels "i", "I", and "E" were chosen, for a total of 126 points and 10 dimensions; (5) **Segmentation**: This dataset is from UCI repository [5]. It has 210 points and 19 dimensions. The instances were drawn randomly from a database of 7 outdoor images. The images were hand-segmented to create a classification for every pixel.

4.2 Evaluation Criterion

To evaluate the clustering results, we use the Rand Statistic index [14, 18, 16]. The Rand Statistic is an external cluster validity measure that estimates the quality of the clustering results with respect to the underlying classes of the data. Let P_1 be the partition of the data X after applying a clustering algorithm, and P_2 be the underlying class structure of the data. We refer to a pair of points $(\mathbf{x}_u, \mathbf{x}_v) \in X \times X$ from the data using the following terms:

- SS: if both points belong to the same cluster of P_1 and to the same group of the underlying class structure P_2.
- SD: if the two points belong to the same cluster of P_1 and to different groups of P_2.
- DS: if the two points belong to different clusters of P_1 and to the same group of P_2.
- DD: if both points belong to different clusters of P_1 and to different groups of P_2.

Assume now that N_{SS}, N_{SD}, N_{DS} and N_{DD} are the number of SS, SD, DS and DD pairs respectively, then $N_{SS} + N_{SD} + N_{DS} + N_{DD} = N_{Pair}$ which is the maximum number of all pairs in the data set[3]. The Rand Statistic index measures the degree of similarity between P_1 and P_2 as follows:

$$RandStatistic = (N_{SS} + N_{DD})/N_{Pair} \tag{7}$$

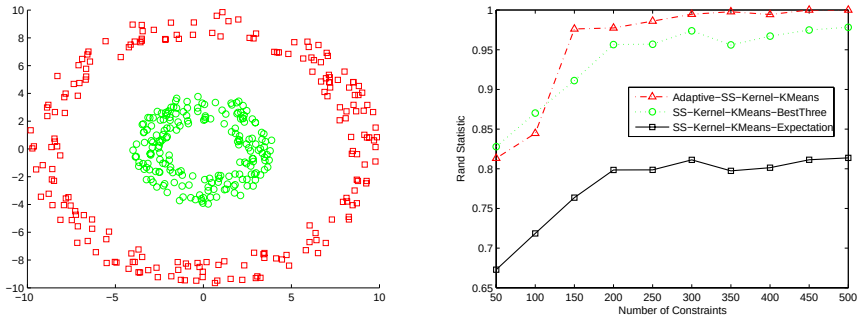
[1] http://www.ics.uci.edu/~mlearn/MLRepository.html
[2] http://www-stat-class.stanford.edu/~tibs/ElemStatLearn/
[3] $N_{Pair} = N(N-1)/2$, where N is the total number of points in the data set.

4.3 Results and Discussion

To evaluate the effectiveness of our proposed method Adaptive-SS-Kernel-KMeans we perform comparisons with SS-Kernel-KMeans [12]. As shown in [12], SS-Kernel-KMeans combined with a Gaussian kernel outperforms HMRF-KMeans and SS-Kernel-KMeans with a linear kernel. Therefore, the technique SS-Kernel-KMeans with Gaussian kernel was the proper choice for our empirical comparisons. SS-Kernel-KMeans requires in input a predefined value for the Gaussian kernel parameter σ. In absence of labeled data, parameters cannot be cross-validated; thus, we estimate the expected accuracy of SS-Kernel-KMeans by averaging the resulting clustering quality over multiple runs for different values of σ. Specifically, in our experiments, we test the SS-Kernel-KMeans algorithm with the values of σ^2: 0.1, 1, 10, 100, 1000, 10000. We report the average Rand Statistic achieved over the six σ values, as well as the average over the best three performances achieved, in order to show the advantage of our technique also in this latter case. The violation costs w_{ij} and $\overline{w}_{ij}$ are set to 1 in our experiments since we assume no a-priori knowledge on such costs. As value of k, we provide the actual number of classes in the data to both algorithms.

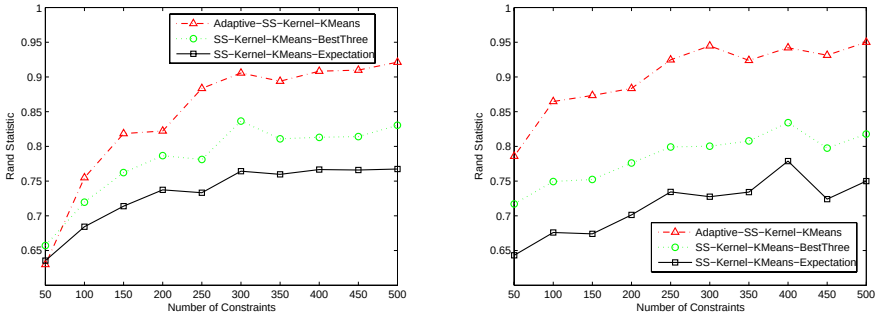
Figures 2-4 show the learning curves using 20 runs of 2-fold cross-validation for each data set (30% for training and 70% for testing). These plots show the improvement in clustering quality on the test set as a function of an increasing amount of pairwise constraints. To study the effect of constraints in clustering, 30% of the data was randomly drawn as the training set at any particular fold, and the constraints are generated only using the training set. The clustering algorithm was run on the whole data set, but we calculated the Rand Statistic only on the test set. Each point on the learning curve is an average of results over 20 runs.

The results shown in Figures 2-4 clearly demonstrate the effectiveness of our proposed technique Adaptive-SS-Kernel-KMeans. For all five datasets, the clustering quality achieved by our adaptive approach significantly outperforms the results provided by SS-Kernel-KMeans, averaged over the σ values tested. In most cases (TwoConcentric, Vowel, Digits, and Segmentation), the Adaptive-SS-Kernel-KMeans technique also outperforms the average top three performances of SS-Kernel-KMeans. For the Stectf data the two approaches show a similar trend. These results show that our adaptive technique is capable of estimating the optimal kernel parameter value from the given constraints. In particular, for the TwoConcentric data (see Figure 2(b)), the Adaptive-SS-Kernel-KMeans technique effectively uses the increased amount of constraints to learn a perfect separation of the two clusters. For the Digits, Spectf, and Segmentation data, the Adaptive-SS-Kernel-KMeans technique provides a clustering quality that is significantly higher than the one given by SS-Kernel-KMeans, even when a small amount of constraints is available. This behavior is very desirable since in practice only a limited amount of supervision might be available. We also emphasize that the cluster initialization mechanism employed in the EM-based strategy mitigates the sensitivity of the result at convergence from the starting point of the search.



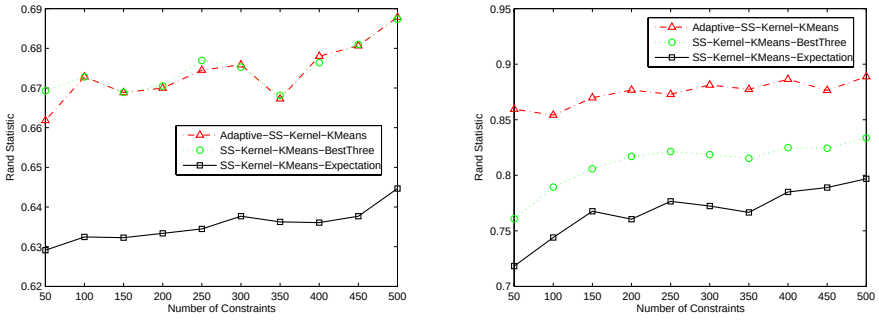
(a) (b)

Fig. 2. (a) TwoConcentric data (b) Clustering result on TwoConcentric data



(a) (b)

Fig. 3. (a) Clustering result on Vowel data (b) Clustering result on Digits data



(a) (b)

Fig. 4. (a) Clustering result on Spectf data (b) Clustering result on Segmentation data

5 Related Work

In the context of supervised learning, the work in [7] considers the problem of automatically tuning multiple parameters for a support vector machine. This is achieved by minimizing the estimated generalization error achieved by means of a gradient descent approach over the set of parameters. In [17], a Fisher discriminant rule is used to estimate the optimal spread parameter of a Gaussian kernel. The authors in [10] propose a new criterion to address the selection of kernel's parameters within a kernel Fisher discriminant analysis framework for face recognition. A new formulation is derived to optimize the parameters of a Gaussian kernel based on a gradient descent algorithm. This research makes use of labeled data to address classification problems. In contrast, our approach optimizes kernel's parameters based on unlabeled data and pairwise constraints, and aims at solving clustering problems.

In the context of semi-supervised clustering, the authors in [8] use a gradient descent approach combined with a weighted Jensen-Shannon divergence for EM clustering. The authors in [1] propose a method based on Redundant Component Analysis (RCA) that uses must-link constraints to learn a Mahalanobis distance. [18] utilizes both must-link and cannot-link constraints to formulate a convex optimization problem which is local-minima-free. [13] proposes a unified Markov network with constraints. [2] introduces a more general HMRF framework, that works with different clustering distortion measures, including Bregman divergences and directional similarity measures. All these techniques use the given constraints and an underlying (linear) distance metric for clustering points in input space. [12] extends the semi-supervised clustering framework to a non-linear kernel space. However, the setting of the kernel's parameter is left to manual tuning, and the chosen value can largely affect the results. The selection of kernel's parameters is a critical and open problem, which has been the driving force behind the work presented in this paper.

6 Conclusion and Future Work

We proposed a new adaptive semi-supervised Kernel-KMeans 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. 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. 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 different types of kernels.

Acknowledgements

This work was in part supported by NSF CAREER Award IIS-0447814.

References

1. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning distance functions using equivalence relations. International Conference on Machine Learning, pages 11-18, 2003.
2. Basu, S., Bilenko, M., Mooney, R.J.: A probabilistic framework for semi-supervised clustering. International Conference on Knowledge Discovery and Data Mining, 2004.
3. Besag, J.: On the statistical analysis of dirty pictures. Journal of the Royal Statistical Society, Series B (Methodological), 1986.
4. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and Metric Learning in semi-supervised clustering. International Conference on Machine Learning, 2004.
5. Blake, C.L., Merz, C.J.: UCI repository of machine learning databases. http://www.ics.uci.edu/˜mlearn/MLRepository.html, 1998.
6. Boykov, Y., Veksler, O., Zabih, R.: Markov Random fields with efficient approximations. IEEE Computer Vision and pattern Recognition Conference, 1998.
7. Chapelle, O., Vapnik, V.: Choosing Mutiple Parameters for Support Vector Machines. Machine Learning Vol.46, No. 1. pp.131-159, 2002.
8. Cohn, D., Caruana, R., McCallum, A.: Semi-supervised clustering with user feedback. TR2003-1892, Cornell University, 2003.
9. Cristianini, N., Shawe-Taylor, J., Elisseeff, A.: On Kernel-Target Alignment, Neural Information Processing Systems (NIPS), 2001.
10. Huang, J., Yuen, P.C., Chen, W.S., Lai, J. H.: Kernel Subspace LDA with optimized Kernel Parameters on Face Recognition. The sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004.
11. Kleinberg, J., Tardos, E.: Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov random fields. The 40th IEEE Symposium on Foundation of Computer Science, 1999.
12. Kulis, B., Basu, S., Dhillon, I., Moony, R.: Semi-supervised graph clustering: a kernel approach. International Conference on Machine Learning, 2005.
13. Segal, E., Wang, H., Koller, D.: Discovering molecular pathways from protein interaction and gene expression data. Bioinformatics, 2003.
14. Theodoridis, S., Koutroubas, K.: Pattern Recognition. Academic Press, 1999.
15. Vapnik., V.: The Nature of Statistical Learning Theory, Wiley, New York, USA, 1995.
16. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained K-Means clustering with background knowledge. International Conference on Machine Learning, pages 577-584, 2001.
17. Wang, W., Xu, Z., Lu W., Zhang, X.: Determination of the spread parameter in the Gaussian Kernel for classification and regression. Neurocomputing, Vol. 55, No. 3, 645, 2002.
18. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side-information. Advances in Neural Information Processing Systems 15, 2003.
19. Zhang, Y., Brady, M., Smith, S.: Hidden Markov random field model and segmentation of brain MR images. IEEE Transactions on Medical Imaging, 2001.