Subspace Metric Ensembles for Semi-supervised Clustering of High Dimensional Data

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. A critical problem in clustering research is the definition of a proper metric to measure distances between points. Semi-supervised clustering uses the information provided by the user, usually defined in terms of constraints, to guide the search of clusters. Learning effective metrics using constraints in high dimensional spaces remains an open challenge. This is because the number of parameters to be estimated is quadratic in the number of dimensions, and we seldom have enough side-information to achieve accurate estimates. In this paper, we address the high dimensionality problem by learning an ensemble of subspace metrics. This is achieved by projecting the data and the constraints in multiple subspaces, and by learning positive semi-definite similarity matrices therein. This methodology allows leveraging the given side-information while solving lower dimensional problems. We demonstrate experimentally using high dimensional data (e.g., microarray data) the superior accuracy achieved by our method with respect to competitive approaches.

1 Introduction

Clustering is the subject of active research in several fields such as statistics, pattern recognition, and machine learning. The clustering problem concerns the discovery of homogeneous groups of data according to a certain similarity measure, such that data in a cluster are more similar to each other than data assigned to different clusters. The definition of a proper similarity measure is a difficult problem that lies at the core of the field of machine learning. The structure of the groups discovered in the data by a given clustering technique strongly depends on the similarity measure used. Data mining adds to clustering the complication of large data sets with high dimensionality. Large amounts of unlabeled data are available in real-life data mining tasks, e.g., unlabeled messages in an automated email classification system, or genes of unknown functions in microarray data. This imposes unique computational requirements on clustering algorithms. Furthermore, the sparsity of the data in high dimensional spaces can severely compromise the ability of discovering meaningful clustering solutions.

Recently, semi-supervised clustering has become a topic of significant research interest. While labeled data are often limited and expensive to generate,

in many cases it is relatively easy for the user to provide pairs of similar or dissimilar examples. Semi-supervised clustering uses a small amount of supervised data, usually under the form of pairwise constraints on some instances, to aid unsupervised learning. The main approaches for semi-supervised clustering can be basically categorized into two general methods: constrained-based [22, 23, 3, 4] and metric-based [24, 8, 2]. The work in [5, 6] combines constraints with a distance metric. However, when facing high dimensional data, learning an effective metric with limited supervision remains an open challenge. The reason is that the number of parameters to be estimated is quadratic in the number of dimensions, and we seldom have enough side-information to achieve accurate estimates. For example, in our experiments we deal with microarray data with 4026 dimensions and less that 100 samples (a typical scenario with these kinds of data). Learning a similarity metric with a limited number of constraints becomes a very hard problem under these conditions due to the large parameter space to be searched.

In this paper, we address the high dimensionality problem by learning an ensemble of subspace metrics. This is achieved by projecting the data and the pairwise constraints in multiple subspaces, and by learning positive semi-definite similarity matrices therein [24]. This methodology allows leveraging the given side-information while solving lower dimensional problems. The diverse clusterings discovered within the subspaces are then combined by means of a graph-based consensus function that leverages the common structure shared by the multiple clustering results [9]. Our experimental results show the superior accuracy achieved by our method with respect to competitive approaches, which learn the metric in the full dimensional space.

2 Background

2.1 Distance Metric Learning

In the context of 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 more realistic to assume that pairwise constraints are available.

Suppose we have a set of points $X = \{\mathbf{x}_i\}_{i=1}^{N} \subseteq \Re^D$, and 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 learn a distance metric that brings the must-link points close to each other and moves the cannot-link points far away from each other. Consider learning a distance metric of the form: $d_A(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T A(\mathbf{x}-\mathbf{y})}$. To ensure that d_A is a metric (i.e., satisfies non-negativity and the triangle inequality), A is required to be positive semi-definite, i.e., $A \succeq 0$. In general terms, A parameterizes a family of Mahalanobis distances over $\Re^D$. When $A = I$, d_A gives the standard Euclidean distance; if A is diagonal, learning A corresponds to assigning different *weights* to features. In general, learning such a distance metric is equivalent to finding a transformation of the data that substitutes each point $\mathbf{x}$ with $\sqrt{A}\mathbf{x}$.

The problem of learning A can be formulated as a convex optimization problem [24]: $\min_A \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in ML} \|\mathbf{x}_i - \mathbf{x}_j\|_A^2$, such that $\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in CL} \|\mathbf{x}_i - \mathbf{x}_j\|_A \geq 1$, and $A \succeq 0$. If we restrict A to be diagonal, i.e., $A = diag\{A_{11}, ..., A_{DD}\}$, the problem can be solved using the Newton-Raphson method and by minimizing the following function: $G(A) = G(A_{11}, \cdots, A_{DD}) = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in ML} \|\mathbf{x}_i - \mathbf{x}_j\|_A^2 - \log\big(\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in CL} \|\mathbf{x}_i - \mathbf{x}_j\|_A\big)$. The Newton-Raphson method computes the Hessian matrix (of size $D \times D$) in each iteration to determine the new search direction. When learning a full matrix A, the Newton-Raphson method requires $O(D^6)$ time to invert the Hessian over D^2 parameters. Clearly, in high dimensional spaces this computation becomes prohibitively expensive.

2.2 Cluster Ensembles

In an effort to achieve improved classifier accuracy, extensive research has been conducted in classifier ensembles. Recently, cluster ensembles have emerged. Cluster ensembles offer a solution to challenges inherent to clustering arising from its ill-posed nature. In fact, it is well known that off-the-shelf clustering methods may discover very different structures in a given set of data. This is because each clustering algorithm has its own bias resulting from the optimization of different criteria. Cluster ensembles can provide robust and stable solutions by leveraging the consensus across multiple clustering results, while averaging out spurious structures due to the various biases to which each participating algorithm is tuned. In the following we formally define the clustering ensemble problem.

Consider a set of data $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$. A clustering ensemble is a collection of S clustering solutions: $C = \{C_1, C_2, \cdots, C_S\}$. Each clustering solution C_l, for $l = 1, \cdots, S$, is a partition of the set X, i.e. $C_l = \{C_l^1, C_l^2, \cdots, C_l^{K_l}\}$, where $\bigcup_K C_l^K = X$. Given a collection of clustering solutions C and the desired number of clusters K, the objective is to combine the different clustering solutions and compute a new partition of X into K disjoint clusters.

Different methods have been introduced in the literature to solve the clustering ensemble problem. The techniques presented in [10] compute a matrix of similarities between pairs of points, and then perform agglomerative clustering to generate a final clustering solution. In [20, 21] the authors introduce new features to describe the data, and apply K-means and EM to output the final clustering solutions. Recently, several approaches have modeled the clustering ensemble problem as a graph partitioning problem [17, 21]. In the following, we provide the necessary definitions of graph partitioning.

A graph partitioning problem takes in input a weighted graph G and an integer K. A weighted graph G is defined as a pair $G = (V, E)$, where V is a set of vertices and E is a $|V| \times |V|$ similarity matrix. Each element E_{ij} of E captures the similarity between vertices V_i and V_j, with $E_{ij} = E_{ji}$ and $E_{ij} \geq 0 \ \forall \ i, j$. Given G and K, the problem of partitioning G into K subgraphs consists in computing a partition of V into K groups of vertices $V = \{V_1, V_2, \cdots, V_K\}$. The sum of the weights (i.e., similarity values) of the crossed edges is defined as the cut of the

partition. In general, we want to find a K-way partition that minimizes the cut. In [9], the authors propose a method (Hybrid-Bipartite-Graph-Formulation, or HBGF), which considers both the similarity of instances and the similarity of clusters when producing the final clustering solution. Specifically, given a cluster ensemble $C = \{C_1, C_2, \cdots, C_S\}$, HBGF constructs a bipartite graph $G = (V, E)$ as follows. $V = V^C \cup V^I$, where each vertex in V^C represents a cluster of the ensemble C, and V^I contains N vertices each representing an instance of the data set X. If both vertices i and j represent clusters or instances, $E_{ij} = 0$; otherwise, if instance i belongs to cluster j, $E_{ij} = E_{ji} = 1$, and 0 otherwise. The authors in [9] use a multi-way spectral graph partitioning algorithm [15] to find a K-way partition of the resulting bipartite graph.

3 Subspace Metric Cluster Ensemble Algorithm

A limited number of pairwise constraints may not be effective for learning a distance metric in high dimensional spaces due to the large parameter space to be searched. We tackle this issue by reducing the given high dimensional problem with fixed supervision into a number of *smaller* problems, for which the dimensionality is reduced while the amount of supervision is unchanged. To achieve this goal, we utilize and leverage the paradigm of learning with ensembles. It is well known that the effectiveness of an ensemble of learners depends on both the accuracy and diversity of the individual components [12]. A good accuracy-diversity trade-off must be achieved to obtain a consensus solution that is superior to the components. Our method generates accurate learners by assigning each of them a problem of lower dimensionality, and, at the same time, by providing each of them the entire amount of constraints. Furthermore, diversity is guaranteed by providing the learners different *views* (or projections) of the data. Since such views are generated randomly from a (typically) large pool of dimensions, it is highly likely that each learner receives a different perspective of the data, which leads to the discovery of diverse (and complementary) structures within the data. The experimental results presented in this paper corroborate the motivation behind our approach. The details of our subspace metric ensemble algorithm follow.

We are given a set X of data in the D dimensional space, a set of must-link constraints ML, and a set of cannot-link constraints CL. We assume that the desired number of clusters to be discovered in X is fixed to K. We reduce a D dimensional semi-supervised clustering problem into a number (S) of semi-supervised clustering problems of reduced dimensionality F. To this end we draw S random samples of F features from the original D dimensional feature space. Moreover, for each must-link constraint $(\mathbf{x}_i, \mathbf{x}_j) \in ML$, we generate the projected must-link constraints $(\mathbf{x}_i, \mathbf{x}_j)_{F_l}$, for $l = 1, \cdots, S$. This gives new S sets of must-link constraints: $ML_{F_1}, \cdots, ML_{F_S}$. Similarly, for each cannot-link constraint $(\mathbf{x}_i, \mathbf{x}_j) \in CL$, we generate the projected cannot-link constraints $(\mathbf{x}_i, \mathbf{x}_j)_{F_l}$, for $l = 1, \cdots, S$. This results in S new sets of cannot-link constraints: $CL_{F_1}, \cdots, CL_{F_S}$.

Algorithm: *Subspace metric cluster ensemble*
Input: X, ML, CL, K, number of features F, ensemble size S
Output: Partition of X into K clusters
Method:

1. Generate S subspaces F_1, F_2, $\cdots$, F_S by random sampling F features without replacement from the D-dimensional original space.
2. For each constraint $(\mathbf{x}_i, \mathbf{x}_j) \in ML$, generate the projected constraints $(\mathbf{x}_i, \mathbf{x}_j)_{F_l}$, for $l = 1, \cdots, S$; this gives the new S sets $ML_{F_1}, \cdots, ML_{F_S}$. Likewise, generate new S sets of cannot-link constraints $CL_{F_1}, \cdots, CL_{F_S}$.
3. Learn matrix A_l in subspace F_l, using the corresponding sets of constraints ML_{F_l} and CL_{F_l}, for $l = 1, \cdots, S$, according to the method presented in [24].
4. Cluster data X in each subspace F_l with K-means, using the metric d_{A_l} and the number of clusters fixed to K. This gives an ensemble of S clusterings.
5. Use the HBGF algorithm [9] to construct the bipartite graph $G = (V, E)$ from the resulting S clusterings.
6. Use spectral graph partitioning to obtain the final clustering result.

Fig. 1. Subspace Metric Cluster Ensemble Algorithm

We have now reduced the original problem into S smaller problems, each of dimensionality F, where we can assume $F \ll D$. We proceed by learning the matrices A_l in each subspace F_l, using the corresponding sets of constraints ML_{F_l} and CL_{F_l}, with $l = 1, \cdots, S$, according to the method presented in [24] (as described in Section 2.1). Note that the dimensionality of each matrix A_l is now reduced to $F \times F$. We then cluster the data X in each subspace F_l, with K-means, using the corresponding distance metrics d_{A_l}, for $l = 1, \ldots, S$, and with the number of clusters fixed to K. This gives an ensemble of S clusterings. We finally use the HBGF algorithm [9] to construct the bipartite graph $G = (V, E)$ from the resulting S clusterings as described in Section 2.2, and apply spectral graph partitioning to obtain the final clustering result. The algorithm is summarized in Figure 1. In our experiments, we refer to this algorithm as *KMeans-Metric-S*.

The authors in [24] also provide clustering results obtained by performing *constrained* K-means combined with the learned matrix A. During the assignment of points to clusters, each pair of points in the set of must-link constraints is assigned to the same cluster (only must-link constraints are used in [24] in this case). To compare our subspace approach with the constrained K-means clustering in full space, we also perform constrained K-means using the learned matrices A_l in each of the generated subspaces. In our experiments, we refer to the resulting technique as *KMeans-Metric-S-cst*. We take advantage of both cannot-link and must-link constraints, and compare the clustering results (in full and reduced spaces) under the same conditions.

In order to implement constrained K-means, once we have learned the matrix A, in full space and in each of the subspaces, we rescale the data according to

$\mathbf{x} \rightarrow \sqrt{A}\mathbf{x}$, and minimize the objective function [4, 5, 11]:

$$J_{obj} = \sum_{C_k} \sum_{\mathbf{x}_i \in C_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in ML} w_{ij} 1[C_i \neq C_j] + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in CL} \overline{w}_{ij} 1[C_i = C_j]$$

where C_k, $k = 1, \cdots, K$, indexes the clusters, $\boldsymbol{\mu}_k$ is the centroid of cluster C_k, C_i and C_j are the clusters to which points $\mathbf{x}_i$ and $\mathbf{x}_j$ are assigned, w_{ij} and $\overline{w}_{ij}$ are the constraint violation costs between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $1[\cdot]$ is an indicator function with a value of 1 if its argument is true, and 0 otherwise. w_{ij} and $\overline{w}_{ij}$ are parameters whose values can be provided as part of the supervision, or can be chosen by the user according to the degree of confidence in the constraints. Here, our concern is to perform a fair comparison between the subspace metric learning approach and the metric learning in full space under the same conditions. Thus, following [4, 5], we set each w equal to the average distance between pairs of points in the data. Such constant provides a scaling for the constraint violation costs which is comparable to the within-cluster scatter measure (first term in the equation above). The same constraint information is provided to K-means performed in full space and in each of the subspaces. We observe that the approaches in [6, 5] can easily be extented to incorporate our concept of subspace metric cluster ensemble.

4 Experimental Design

To demonstrate the effectiveness of our subspace metric cluster ensemble, we consider two microarray data sets which reflect the challenges discussed above: the NC160 [16] and Lymphoma [1] data sets. For the NC160 data set, cDNA microarrays were used to examine the variation in 1155 gene expression values among the 61 cell lines from the National Center Institutes anticancer drug screen. The data set contains 8 classes. To deal with missing values, we deleted genes that have more than 50 missing values, and used the K-nearest neighbor method to impute the remaining missing values. For a gene with missing values, the K nearest neighbors are identified from the subset of genes that have complete expression values ($K = 7$ in our experiments). The average of the neighbors' values is used to substitute a missing value. The Lymphoma data set [1] contains 96 samples from patients, each with 4026 gene expression values. The samples are categorized into 9 classes according to the type of mRNA sample studied. Classes that have less than 5 samples are removed from the experiments, and hence 6 classes and 88 samples remained. We also perform experiments on two UCI data sets: Wine (N=178, D=13, K=3) and Breast-Cancer (N=569, D=30, K=2).

We compare two variants of the proposed subspace cluster ensemble approach: the *KMeans-Metric-S* and the *KMeans-Metric-S-cst* algorithms, as described in Section 3. Constrained K-means is performed by minimizing the objective function provided in Section 3, where the weights of the constraint violation costs are set to the average distance between pairs of points in the data set. We also compare the corresponding variants for metric learning in full feature space,

which we call *KMeans-Metric-F* and *KMeans-Metric-F-cst*, respectively, as well as constrained K-Means (*KMeans-cst*), and *K-Means* with no supervision.

For K-Means, KMeans-Metric-F and KMeans-Metric-S, we randomly initialize the clusters, and set the number of clusters K to the actual number of classes in the data. For KMeans-cst, KMeans-Metric-F-cst and KMeans-Metric-S-cst, the clusters are initialized using the approach presented in [5, 11]: 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 constraint information is given to each competitive algorithm. For all four methods KMeans-Metric-S, KMeans-metric-F, KMeans-metric-F-cst, and KMeans-metric-F-cst we learn a distance metric d_A with diagonal matrix A.

5 Experimental Results

To evaluate clustering results, we use the Rand Statistic index [19, 24, 22]. Figures 2-3 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. For studying the effect of constraints in clustering, 30% of the data is randomly drawn as the training set at any particular fold, and the constraints are generated only using the training set. We observe that, since folds are randomly generated, there is no guarantee that all classes are represented within the training data. The clustering algorithm was run on the whole data set, but we calculate the Rand Statistic only on the test set. Each point on the learning curve is an average of results over 20 runs.

We learn the metrics in 120 subspaces separately for the NCI60 and Lymphoma data sets, each of which is produced by randomly selecting 60 features (i.e., $S = 120$ and $F = 60$). For the Wine and Breast-Cancer data sets, the metrics are learned separately in 30 subspaces ($S = 30$), each of which is produced by randomly selecting 8 features ($F = 8$) for Wine data and 10 features ($F = 10$) for Breast-Cancer data.

From Figures 2-3, we can clearly appreciate the benefits of using our subspace metric ensemble techniques. For the two high dimensional data, NCI60 and Lymphoma, when a small number of constraints is used, the KMeans-Metric-S and KMeans-Metric-S-cst algorithms show large clustering quality improvements with respect to the competing techniques. In fact, for these two data sets, the algorithms KMeans-Metric-S and KMeans-Metric-S-cst leverage the given side-information while solving much lower dimensional problems (from 1155 dimensions down to 60, and from 4026 down to 60, respectively).

For the Wine and Breast-Cancer data, the improvement of the clustering quality for KMeans-Metric-S and KMeans-Metric-S-cst is more gradual throughout the increasing of the number of constraints. The dimensionalities of these two data sets is much lower (13 and 30 respectively), and the dimensionalities of the subspaces are 8 and 10, respectively. The overall gap in quality improvement

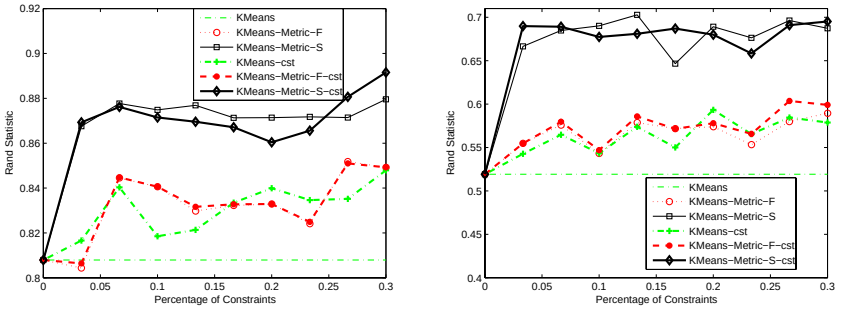

Fig. 2. Clustering results (*left*) on NCI60 data and (*right*) on Lymphoma data

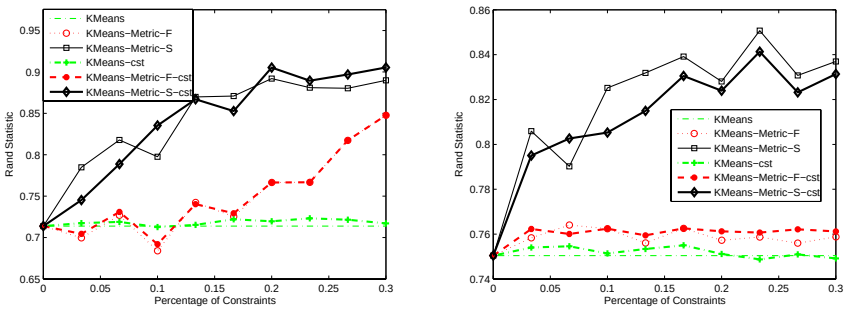

Fig. 3. Clustering results (*left*) on Wine data and (*right*) on Breast-Cancer data

between the subspace metric ensemble methods and the remaining techniques on these data clearly shows the positive effect of using ensembles of clusterings: the accuracy and diversity of the individual clustering solutions allow to achieve an improved consensus clustering that is superior to the component ones. In fact, although the dimensionalities of the full space and the subspace do not differ greatly, the ensemble technique is capable of taking good advantage of the increased amount of supervision (this is particularly evident in Figure 3 for the Breast-Cancer data set). We observe that, in general, the trend for the algorithms that operate in full space is rather flat, showing that a limited amount of supervision does not have a high impact in high dimensional spaces.

No significant difference between KMeans-Metric-S and KMeans-Metric-S-cst was observed throughout the tested data sets. The same is true for the corresponding algorithms in full space.

Analysis of Diversity. We investigate here the trade-off between accuracy and diversity achieved by our subspace cluster ensembles. A Kappa-Error diagram [12] allows to visualize the diversity and the accuracy of an ensemble of classifiers. To analyze the diversity-quality trade-off of our subspace metric ensembles, we measure diversity using the *Normalized Mutual Information* (NMI) [18] between each pair of clustering solutions. In addition, we average the two

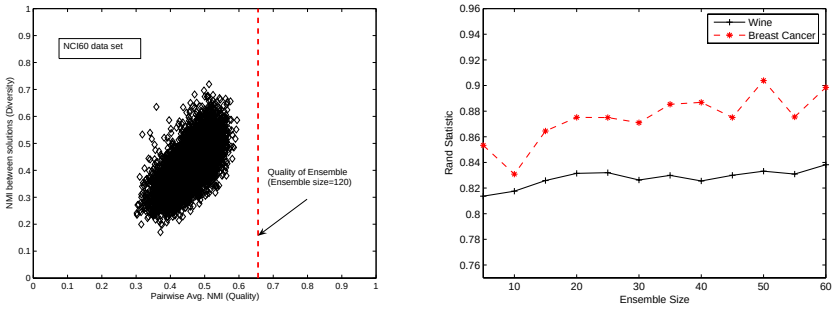

Fig. 4. (*left*) Diversity-Quality on NCI data; (*right*) Quality vs. Ensemble-Size on Wine and Breast-Cancer data

NMI values of the pair, each computed using the ground truth labels. Such average provides a single quality measure for each pair of clustering solutions.

We have plotted the diversity-quality diagrams based on the KMeans-Metric-S algorithm for each of the four data sets considered (the percentage of constraints is fixed to 12%). For lack of space we report here only the diagram for the NCI60 data set (Figure 4 (left)). For every data set, the quality of the ensemble is superior to that of the individual components, proving the effectivess of our subspace cluster ensembles. The vertical dashed line indicates the quality of the final clustering provided by the ensemble (measured in terms of NMI with respect to the underlying class structure). We note that when the NMI between two clustering solutions (shown on the y axis) is zero, the diversity is maximized. On the other hand, when the average NMI of each pair (shown on the x axis) is maximized, their quality is also maximized. Thus, ideally, the points would populate the right-hand bottom corner of the figure. Figure 4 (left) shows that the components of the ensemble have a relatively high diversity and quality, thus they achieve a good trade-off which is reflected in the increased quality of the ensemble with respect to the clustering components. This result emphasizes the fact that, while our technique reduces the curse-of-dimensionality effect, interestingly, it also takes advantage of the high dimensionality of the data to guarantee diversity among the ensemble components. As a consequence, it is most effective with high dimensional data.

Figure 4 (right) shows the quality of the ensemble as a function of different ensemble sizes for the Wine and Breast-Cancer data sets. Fixing the constraints (12%), each point in the graph is an average of results over 20 runs. While for the NCI60 and Lymphoma data (not included for lack of space) no significant increase in quality is observed within the range of sizes tested, for the Wine and Breast-Cancer data (Figure 4 (right)) we observe an increasing trend in quality as the ensemble size increases. It is reasonable to expect that increasing the ensemble size for the two data sets of lower dimensionality results in a better trade-off between accuracy and diversity of the components.

Time Complexity. The KMeans-Metric-F (or KMeans-Metric-F-cst) technique performs the computation of the distance metric d_A followed by the K-

Table 1. Running Times (measured in seconds)

Data set	Method	Learn. A	K-Means (1-run)	HBGF	S	T_{total}
NCI60	Full space	8.914	8.527	—	1	17.441
$(N = 61, D = 1155, K = 8)$	Subspace	0.086	0.058	0.191	120	17.471
Lymphoma	Full space	277.338	138.586	—	1	415.924
$(N = 88, D = 4026, K = 6)$	Subspace	0.136	0.067	0.158	120	24.518
Wine	Full space	0.027	0.045	—	1	0.072
$(N = 178, D = 13, K = 3)$	Subspace	0.021	0.033	0.032	30	1.652
Breast-Cancer	Full space	0.038	0.178	—	1	0.216
$(N = 569, D = 30, K = 2)$	Subspace	0.025	0.078	0.041	30	3.131

Means clustering. The first step (achieved via the Newton-Raphson method) requires the computation of $O(D^2)$ partial derivatives. The time-complexity of K-Means is $O(NKRD)$ [13], where N is the number of data, K is the number of clusters, R is the number of iterations, and D is the dimensionality of the data. When $D \gg NK$, the most costly step is the computation of the distance metric. The corresponding time complexities for the KMeans-Metric-S (or KMeans-Metric-S-cst) are $O(F^2 \times S)$ for the first step, and $O(N \times K \times r \times F \times S)$ for the second step, where F is the dimensionality of the subspaces (usually $F \ll D$), S is the ensemble size, and r is the number of iterations of K-Means. The subspace ensemble technique includes also the construction of the bipartite graph and the execution of a K-way graph partitioning (using spectral graph partitioning) whose cost is $O((\max\{N, K \times S\})^{3/2}K + rNK^2)$. The first term is due to the computation of K eigenvectors for a $N \times (K(S))$ matrix, and the second term corresponds to the complexity of K-Means in K dimensions. To compare the running times of the two approaches (full space vs. subspace ensemble), we fix the number of constraints and the ensemble size, and record the running times for each phase of the algorithms. We repeat the computation 20 times and report the average running times in Table 1. All the experiments are performed on a Linux machine with 2.8 GHz Pentium IV processor and 1 GB main memory. The total time for the approach in full space is $T_{total} = T_{metriclearning} + T_{KMeans}$; the total time for the subspace ensemble is $T_{total} = (T_{metriclearning} + T_{KMeans}) \times S + T_{HBGF}$.

The proposed subspace ensemble algorithm greatly reduce the running time for the Lymphoma data set. The full space and subspace approaches show comparable running times on the NCI60 data set. The ensemble approach has a larger running time for the Breast-Cancer and Wine data sets, since their dimensionalities are small. We emphasize that the computed running times are based on sequential executions for the ensemble components. Nevertheless, such computations can be easily run in parallel, allowing for further speed-up.

6 Related Work

The authors in [22] proposed the COP-KMeans algorithm, which assigns each point to the closest cluster that minimizes the violation of constraints. If no such cluster exists, it fails to assign the point. In [3], the authors utilized labeled

data to initialize the clusters. Constraints could be violated (Seeded-KMeans) in successive iterations, or could be strictly enforced (Constrainted-KMeans) throughout the algorithm. Moreover, [4] proposed the PCKMeans algorithm, which assigns weight parameters to the constraints. The work in [11] applies kernel methods to enable the use of both vector-based and graph-based data for semi-supervised clustering. However, the work in [11] does not learn a distance metric based on pairwise constraints.

In recent work on semi-supervised clustering with pairwise constraints, [8] used gradient descent combined with a weighted Jensen-Shannon divergence in the context of EM clustering. [2] proposed a Redundant Component Analysis (RCA) algorithm that uses must-link constraints to learn a Mahalanobis distance. [24] utilized both must-link and cannot-link constraints to formulate a convex optimization problem which is local-minima-free. [5, 6] proposed a method based on Hidden Markov Random Fields (HMRFs) which learns a metric during clustering to minimize an objective function which incorporates the constraints. This is equivalent to the minimization of the posterior energy of the HMRF.

7 Conclusions and Future Work

We have addressed the problem of learning effective metrics for clustering in high dimensional spaces when limited supervision is available. We have proposed an approach based on learning with ensembles that is capable of producing components which are both accurate and diverse. In our future work we will investigate the sensitivity of our approach with respect to the dimensionality of subspaces, and possibly define an heuristic to automatically estimate an "optimal" value for such parameter. Furthermore, we will explore alternative mechanisms to credit weights to features by utilizing the constraints; consequently we will bias the sampling in feature space to favor the estimated most relevant features.

Acknowledgments

This work was in part supported by NSF CAREER Award IIS-0447814. The authors would like to thank Xiaoli Zhang Fern for the code of the HBGF algorithm, and Brian Kulis for the code of the methods described in [11]. We are thankful to Hong Chai for helping us with the processing of the microarray data.

References

1. A. A. Alizadeh, M. B. Eisen, R. E. Davis, and C. Ma et al. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature, 403, pages 503-511, 2000.
2. A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. *International Conference on Machine Learning*, 2003.
3. S. Basu, A. Banerjee, and R. J. Mooney. Semi-supervised clustering by seeding. *International Conference on Machine Learning*, 2002.

4. S. Basu, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrainted clustering. *SIAM International conference on Data Mining*, 2004.
5. S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. *International Conference on Knowledge Discovery and Data Mining*, 2004.
6. M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and Metric Learning in semi-supervised clustering. *International Conference on Machine Learning*, 2004.
7. C. L. Blake and C. J. Merz. UCI repository of machine learning databases. http://www.ics.uci.edu/ mlearn/MLRepository.html, 1998.
8. D. Cohn, R. Caruana, and A. McCallum. Semi-supervised clustering with user feedback. TR2003-1892, Cornell University, 2003.
9. X. Z. Fern and C. E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. *International Conference on Machine Learning*, 2004.
10. A. L. N. Fred and A. K. Jain. Data clustering using evidence accumulation. *International Conference on Pattern Recognition*, 2002.
11. B. Kulis, S. Basu, and I. Dhillon, and R. Mooney. Semi-supervised graph clustering: a kernel approach. *International Conference on Machine Learning*, 2005.
12. D. D. Margineantu and T. G. Dietterich. Pruning adaptive boosting. *International Conference on Machine Learning*, 1997.
13. J. McQueen. Some Methods for Classification and Analysis of Multivariate Observation. L. Le Cam and J. Neyman (Eds.), *Berkeley Symposium on Mathematical Statistics and Probability*, pages 281-297, 1967.
14. S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. Machine Learning, 52, pages 91-118, 2003.
15. A Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems 14*, 2002.
16. D. T. Ross, U. Scherf, M. B. Eisen, and C. M. Perou et al. Systematic variation in gene expression patterns in human cancer cell lines. Nature Genetics, 24(3), pages 227-235, 2000.
17. A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Machine Learning Research, 3, pages 583-417, 2002.
18. A. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. *AAAI Workshop on Artificial Intelligence for Web Search*, 2000.
19. S. Theodoridis and K. Koutroubas. Pattern Recognition. Academic Press, 1999.
20. A. Topchy, A. K. Jain, and W. Punch. Combining multiple weak clusterings. *IEEE International Conference of Data Mining*, 2003.
21. A. Topchy, A. K. Jain, and W. Punch. A mixture model for clustering ensembles. *SIAM International Conference on Data Mining*, 2004
22. K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained K-Means clustering with background knowledge. *International Conference on Machine Learning*, 2001.
23. K. Wagstaff. Intelligent Clustering with Instance-Level Constraints. PhD thesis, Cornell University, 2002.
24. E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. *Advances in Neural Information Processing Systems 15*, 2003.