Proceedings of the 21st International Conference on Machine Learning,
(ICML-2004), pp. 81-88, Banff, Canada, July, 2004

Integrating Constraints and Metric Learning in Semi-Supervised Clustering

Mikhail Bilenko MBILENKO@CS.UTEXAS.EDU
Sugato Basu SUGATO@CS.UTEXAS.EDU
Raymond J. Mooney MOONEY@CS.UTEXAS.EDU
Department of Computer Sciences, University of Texas at Austin, Austin, TX 78712 USA

Abstract

Semi-supervised clustering employs a small amount of labeled data to aid unsupervised learning. Previous work in the area has utilized supervised data in one of two approaches: 1) constraint-based methods that guide the clustering algorithm towards a better grouping of the data, and 2) distance-function learning methods that adapt the underlying similarity metric used by the clustering algorithm. This paper provides new methods for the two approaches as well as presents a new semi-supervised clustering algorithm that integrates *both* of these techniques in a uniform, principled framework. Experimental results demonstrate that the unified approach produces better clusters than both individual approaches as well as previously proposed semi-supervised clustering algorithms.

1. Introduction

In many learning tasks, unlabeled data is plentiful but labeled data is limited and expensive to generate. Consequently, *semi-supervised learning*, which employs both labeled and unlabeled data, has become a topic of significant interest. More specifically, *semi-supervised clustering*, the use of class labels or pairwise constraints on some examples to aid unsupervised clustering, has been the focus of several recent projects (Wagstaff et al., 2001; Basu et al., 2002; Klein et al., 2002; Xing et al., 2003; Bar-Hillel et al., 2003; Segal et al., 2003).

Existing methods for semi-supervised clustering fall into two general approaches we call *constraint-based* and *metric-based*. In constraint-based approaches, the clustering algorithm itself is modified so that user-provided labels or pairwise constraints are used to guide the algorithm towards a more appropriate data partitioning. This is done by modifying the clustering objective function so that it includes satisfaction of constraints (Demiriz et al., 1999), enforcing constraints during the clustering process (Wagstaff et al., 2001), or initializing and constraining clustering based on labeled examples (Basu et al., 2002). In metric-based approaches, an existing clustering algorithm that uses a distance metric is employed; however, the metric is first trained to satisfy the labels or constraints in the supervised data. Several distance measures have been used for metric-based semi-supervised clustering including Euclidean distance trained by a shortest-path algorithm (Klein et al., 2002), string-edit distance learned using Expectation Maximization (EM) (Bilenko & Mooney, 2003), KL divergence adapted using gradient descent (Cohn et al., 2003), and Mahalanobis distances trained using convex optimization (Xing et al., 2003; Bar-Hillel et al., 2003).

Previous metric-based semi-supervised clustering algorithms exclude unlabeled data from the metric training step, as well as separate metric learning from the clustering process. Also, existing metric-based methods use a single distance metric for all clusters, forcing them to have similar shapes. We propose a new semi-supervised clustering algorithm derived from K-Means, MPCK-MEANS, that incorporates *both* metric learning and the use of pairwise constraints in a principled manner. MPCK-MEANS performs distance-metric training with each clustering iteration, utilizing both unlabeled data and pairwise constraints. The algorithm is able to learn individual metrics for each cluster, which permits clusters of different shapes. MPCK-MEANS also allows violation of constraints if it leads to a more cohesive clustering, whereas earlier constraint-based methods forced satisfaction of all constraints, leaving them vulnerable to noisy supervision.

By ablating the metric-based and constraint-based components of our unified method, we present experimental results comparing and combining the two approaches on multiple datasets. The two methods for semi-supervision individually improve clustering accuracy, and our unified approach integrates their strengths. Finally, we demonstrate that the semi-supervised metric learning in our approach outperforms previously proposed methods that learn metrics prior to clustering, and that learning multiple cluster-specific metrics can lead to better results.

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004. Copyright 2004 by the authors.

2. Problem Formulation

2.1. Clustering with K-Means

K-Means is a clustering algorithm based on iterative relocation that partitions a dataset into K clusters, locally minimizing the total squared Euclidean distance between the data points and the cluster centroids. Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}, \mathbf{x}_i \in \mathbb{R}^m$ be a set of data points, x_{id} be the d-th component of $\mathbf{x}_i$, $\{\boldsymbol{\mu}_h\}_{h=1}^{K}$ represent the K cluster centroids, and l_i be the cluster assignment of a point $\mathbf{x}_i$, where $l_i \in \{1, \ldots, K\}$. The Euclidean K-Means algorithm creates a K-partitioning $\{\mathcal{X}_h\}_{h=1}^{K}$ of $\mathcal{X}$ so that the objective function $\sum_{\mathbf{x}_i \in \mathcal{X}} \|\mathbf{x}_i - \boldsymbol{\mu}_{l_i}\|^2$ is locally minimized.

It can be shown that the K-Means algorithm is essentially an EM algorithm on a mixture of K Gaussians under assumptions of identity covariance of the Gaussians, uniform mixture component priors and expectation under a particular type of conditional distribution (Basu et al., 2002). In the Euclidean K-Means formulation, the squared L_2-norm $\|\mathbf{x}_i - \boldsymbol{\mu}_{l_i}\|^2 = (\mathbf{x}_i - \boldsymbol{\mu}_{l_i})^T(\mathbf{x}_i - \boldsymbol{\mu}_{l_i})$ between a point $\mathbf{x}_i$ and its corresponding cluster centroid $\boldsymbol{\mu}_{l_i}$ is used as the distance measure, which is a direct consequence of the identity covariance assumption of the underlying Gaussians.

2.2. Semi-supervised Clustering with Constraints

In *semi-supervised clustering*, a small amount of labeled data is available to aid the clustering process. Our framework uses both must-link and cannot-link constraints between pairs of instances (Wagstaff et al., 2001), with an associated cost for violating each constraint. In many unsupervised-learning applications, e.g., clustering for speaker identification in a conversation (Bar-Hillel et al., 2003), or clustering GPS data for lane-finding (Wagstaff et al., 2001), considering supervision in the form of constraints is more realistic than providing class labels. While class labels may be unknown, a user can still specify whether pairs of points belong to same or different clusters. Constraint-based supervision is also more general than class labels: a set of classified points implies an equivalent set of pairwise constraints, but not vice versa.

Since K-Means cannot directly handle pairwise constraints, we formulate the goal of pairwise constrained clustering as minimizing a combined objective function, defined as the sum of the total squared distances between the points and their cluster centroids, and the cost incurred by violating any pairwise constraints. Let $\mathcal{M}$ be a set of must-link pairs where $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}$ implies $\mathbf{x}_i$ and $\mathbf{x}_j$ should be in the same cluster, and $\mathcal{C}$ be a set of cannot-link pairs where $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}$ implies $\mathbf{x}_i$ and $\mathbf{x}_j$ should be in different clusters. Let $W = \{w_{ij}\}$ and $\overline{W} = \{\overline{w}_{ij}\}$ be penalty costs for violating the constraints in $\mathcal{M}$ and $\mathcal{C}$ respectively. Therefore, the goal of pairwise constrained K-Means is to minimize the following objective function, where point $\mathbf{x}_i$ is assigned to the partition $\mathcal{X}_{l_i}$ with centroid $\boldsymbol{\mu}_{l_i}$:

$$\begin{aligned} J_{\text{pckmeans}} = & \sum_{\mathbf{x}_i \in \mathcal{X}} \|\mathbf{x}_i - \boldsymbol{\mu}_{l_i}\|^2 + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}} w_{ij} \mathbb{1}[l_i \neq l_j] \\ & + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}} \overline{w}_{ij} \mathbb{1}[l_i = l_j] \end{aligned} \quad (1)$$

where $\mathbb{1}$ is the indicator function, $\mathbb{1}[true]$ = 1 and $\mathbb{1}[false]$ = 0. This mathematical formulation is motivated by the *metric labeling* problem with the *generalized Potts* model (Kleinberg & Tardos, 1999).

2.3. Semi-supervised Clustering with Metric Learning

While pairwise constraints can guide a clustering algorithm towards a better grouping, they can also be used to adapt the underlying distance metric. Pairwise constraints effectively represent the user's view of similarity in the domain. Since the original data representation may not specify a space where clusters are sufficiently separated, modifying the distance metric warps the space to minimize distances between same-cluster objects, while maximizing distances between different-cluster objects. As a result, clusters discovered using learned metrics adhere more closely to the notion of similarity embodied in the supervision.

We parameterize Euclidean distance using a symmetric positive-definite matrix $\mathbf{A}$ as follows: $\|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}} = \sqrt{(\mathbf{x}_i - \boldsymbol{\mu}_{l_i})^T \mathbf{A} (\mathbf{x}_i - \boldsymbol{\mu}_{l_i})}$; the same parameterization was previously used by Xing et al. (2003) and Bar-Hillel et al. (2003). If $\mathbf{A}$ is restricted to a diagonal matrix, it scales each dimension by a different weight and corresponds to feature weighting; otherwise new features are created that are linear combinations of the original ones.

In previous work on adaptive metrics for clustering (Cohn et al., 2003; Xing et al., 2003; Bar-Hillel et al., 2003), metric weights are trained to simultaneously minimize the distance between must-linked instances and maximize the distance between cannot-linked instances. A fundamental limitation of these approaches is that they assume a single metric for all clusters, preventing them from having different shapes. We allow a separate weight matrix for each cluster, denoted $\mathbf{A}_h$ for cluster h. This is equivalent to a generalized version of the K-Means model described in section 2.1, where cluster h is generated by a Gaussian with covariance matrix $\mathbf{A}_h^{-1}$ (Bilmes, 1997). It can be shown that maximizing the complete data log-likelihood under this generalized K-Means model is equivalent to minimizing the objective function:

$$J_{\text{mkmeans}} = \sum_{\mathbf{x}_i \in \mathcal{X}} \left(\|\mathbf{x}_i - \boldsymbol{\mu}_{l_i}\|^2_{\mathbf{A}_{l_i}} - \log(\det(\mathbf{A}_{l_i})) \right) \quad (2)$$

where the second term arises due to the normalizing constant of l_i-th Gaussian with covariance matrix $\mathbf{A}_{l_i}^{-1}$.

2.4. Integrating Constraints and Metric Learning

Combining Eqns.(1) and (2) leads to the following objective function that minimizes cluster dispersion under the learned metrics while reducing constraint violations:

$$\mathcal{J}_{\text{combined}} = \sum_{\mathbf{x}_i \in \mathcal{X}} \left(\|\mathbf{x}_i - \boldsymbol{\mu}_{l_i}\|^2_{\mathbf{A}_{l_i}} - \log(\det(\mathbf{A}_{l_i})) \right) + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}} w_{ij} \mathbb{1}[l_i \neq l_j] + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} \overline{w}_{ij} \mathbb{1}[l_i = l_j] \quad (3)$$

If we assume uniform constraint costs w_{ij} and $\overline{w}_{ij}$, all constraint violations are treated equally. However, the penalty for violating a must-link constraint between *distant* points should be higher than that between *nearby* points. Intuitively, this captures the fact that if two must-linked points are far apart according to the current metric, the metric is grossly inadequate and needs severe modification. Since two clusters are involved in a must-link violation, the corresponding penalty should affect the metrics for both clusters. This can be accomplished via multiplying the penalty in the second summation of Eqn.(3) by the following function:

$$f_M(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2_{\mathbf{A}_{l_i}} + \frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2_{\mathbf{A}_{l_j}} \quad (4)$$

Analogously, the penalty for violating a cannot-link constraint between two points that are *nearby* according to the current metric should be higher than for two *distant* points. To reflect this intuition, the following penalty term can be used with violated cannot-link constraints that are assigned to the same cluster ($l_i = l_j$):

$$f_C(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}'_{l_i} - \mathbf{x}''_{l_i}\|^2_{\mathbf{A}_{l_i}} - \|\mathbf{x}_i - \mathbf{x}_j\|^2_{\mathbf{A}_{l_i}} \quad (5)$$

where $(\mathbf{x}'_{l_i}, \mathbf{x}''_{l_i})$ is the maximally separated pair of points in the dataset according to l_i-th metric. This form of f_C ensures that the penalty for violating a cannot-link constraint remains non-negative since the second term is never greater than the first. The combined objective function then becomes:

$$\mathcal{J}_{\text{mpckm}} = \sum_{\mathbf{x}_i \in \mathcal{X}} \left(\|\mathbf{x}_i - \boldsymbol{\mu}_{l_i}\|^2_{\mathbf{A}_{l_i}} - \log(\det(\mathbf{A}_{l_i})) \right) + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}} w_{ij} f_M(\mathbf{x}_i, \mathbf{x}_j) \mathbb{1}[l_i \neq l_j] + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} \overline{w}_{ij} f_C(\mathbf{x}_i, \mathbf{x}_j) \mathbb{1}[l_i = l_j] \quad (6)$$

Costs w_{ij} and $\overline{w}_{ij}$ provide a way of specifying the relative importance of the labeled versus unlabeled data while allowing individual constraint weights. The following section describes how $\mathcal{J}_{\text{mpckm}}$ can be greedily optimized by our proposed metric pairwise constrained K-Means (MPCK-MEANS) algorithm.

3. MPCK-MEANS Algorithm

Given a set of data points $\mathcal{X}$, a set of must-link constraints $\mathcal{M}$, a set of cannot-link constraints $\mathcal{C}$, corresponding cost sets W and $\overline{W}$, and the desired number of clusters K, MPCK-MEANS finds a disjoint K-partitioning $\{\mathcal{X}_h\}_{h=1}^K$ of $\mathcal{X}$ (with each cluster having a centroid $\boldsymbol{\mu}_h$ and a local weight matrix $\mathbf{A}_h$) such that $\mathcal{J}_{\text{mpckm}}$ is (locally) minimized. The algorithm integrates the use of constraints and metric learning. Constraints are utilized during cluster initialization and when assigning points to clusters, and the distance metric is adapted by re-estimating the weight matrices $\mathbf{A}_h$ during each iteration based on the current cluster assignments and constraint violations. Pseudocode for the algorithm is presented in Fig.1.

Algorithm: MPCK-Means
Input: Set of data points $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$,
set of *must-link* constraints $\mathcal{M} = \{(\mathbf{x}_i, \mathbf{x}_j)\}$,
set of *cannot-link* constraints $\mathcal{C} = \{(\mathbf{x}_i, \mathbf{x}_j)\}$,
number of clusters K, sets of constraint costs W and $\overline{W}$.
Output: Disjoint K-partitioning $\{\mathcal{X}_h\}_{h=1}^K$ of $\mathcal{X}$ such that objective function $\mathcal{J}_{\text{mpckm}}$ is (locally) minimized.
Method:
1. Initialize clusters:
1a. create the λ neighborhoods $\{N_p\}_{p=1}^{\lambda}$ from $\mathcal{M}$ and $\mathcal{C}$
1b. if $\lambda \geq K$
initialize $\{\boldsymbol{\mu}_h^{(0)}\}_{h=1}^K$ using weighted farthest-first traversal starting from the largest N_p
else if $\lambda < K$
initialize $\{\boldsymbol{\mu}_h^{(0)}\}_{h=1}^{\lambda}$ with centroids of $\{N_p\}_{p=1}^{\lambda}$
initialize remaining clusters at random
2. Repeat until *convergence*
2a. `assign_cluster`: Assign each data point $\mathbf{x}_i$ to cluster h^* (i.e. set $\mathcal{X}_{h^*}^{(t+1)}$), for $h^* = \arg\min_h \big(\|\mathbf{x}_i - \boldsymbol{\mu}_h^{(t)}\|^2_{\mathbf{A}_h} - \log(\det(\mathbf{A}_h)) + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}} w_{ij} f_M(\mathbf{x}_i, \mathbf{x}_j) \mathbb{1}[h \neq l_j] + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} \overline{w}_{ij} f_C(\mathbf{x}_i, \mathbf{x}_j) \mathbb{1}[h = l_j]\big)$
2b. `estimate_means`: $\{\boldsymbol{\mu}_h^{(t+1)}\}_{h=1}^K \leftarrow \{\frac{1}{|\mathcal{X}_h^{(t+1)}|} \sum_{\mathbf{x}\in\mathcal{X}_h^{(t+1)}} \mathbf{x}\}_{h=1}^K$
2c. `update_metrics`: $\mathbf{A}_h = |\mathcal{X}_h| \Big(\sum_{\mathbf{x}_i\in\mathcal{X}_h} (\mathbf{x}_i - \boldsymbol{\mu}_h)(\mathbf{x}_i - \boldsymbol{\mu}_h)^T + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}_h} \frac{1}{2} w_{ij} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \mathbb{1}[l_i \neq l_j] + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}_h} \overline{w}_{ij} \big((\mathbf{x}'_h - \mathbf{x}''_h)(\mathbf{x}'_h - \mathbf{x}''_h)^T - (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T\big) \mathbb{1}[l_i = l_j] \Big)^{-1}$
2d. $t \leftarrow (t+1)$

Figure 1. MPCK-MEANS algorithm

3.1. Initialization

Good initial centroids are critical to the success of greedy clustering algorithms such as K-Means. To infer the initial clusters from the constraints, we take the transitive closure of the must-link constraints and augment the set $\mathcal{M}$ with these entailed constraints (assuming consistency of the constraints). Let λ be the number of connected components in the augmented set $\mathcal{M}$. These connected components are used to create λ neighborhood sets $\{N_p\}_{p=1}^{\lambda}$, where each neighborhood consists of points connected by must-links. For every pair of neighborhoods N_p and $N_{p'}$ that have at least one cannot-link between them, we add cannot-link constraints between every pair of points in N_p and $N_{p'}$ and augment the cannot-link set $\mathcal{C}$ with these entailed constraints. We will overload notation from this point and refer

to the augmented must-link and cannot-link sets as $\mathcal{M}$ and $\mathcal{C}$ respectively.

After this preprocessing step, we get λ neighborhood sets $\{N_p\}_{p=1}^{\lambda}$. These neighborhoods provide initial clusters for the MPCK-MEANS algorithm. If $\lambda \leq K$, we initialize λ cluster centers with the centroids of all the λ neighborhood sets. If $\lambda < K$, we initialize the remaining $K - \lambda$ clusters with points obtained by random perturbations of the global centroid of $\mathcal{X}$.

If $\lambda > K$, we select K neighborhood sets using a weighted variant of the farthest-first algorithm, which is a good heuristic for initialization in centroid-based clustering algorithms like K-Means. In weighted farthest-first traversal, the goal is to find K points which are maximally separated from each other in terms of a weighted distance. In our case, the points are the centroids of the λ neighborhoods, and the weight of each centroid is the size of its corresponding neighborhood. Thus, we bias farthest-first to select centroids which are relatively far apart but also represent large neighborhoods, in order to obtain good initial clusters.

In weighted farthest-first traversal, we maintain a set of traversed points at every step, and pick the following point having the farthest weighted distance from the traversed set (using the standard notion of distance from a set: $d(\mathbf{x}, S) = \min_{y \in S} d(\mathbf{x}, \mathbf{y})$), and so on. Finally, we initialize the K cluster centers with the centroids of the K neighborhoods chosen by weighted farthest-first traversal.

3.2. E-step

MPCK-MEANS alternates between cluster assignment in the E-step, and centroid estimation and metric learning in the M-step (see Step 2 in Fig.1). In the E-step, every point $\mathbf{x}$ is assigned to the cluster that minimizes the sum of the distance of $\mathbf{x}$ to the cluster centroid according to the local metric and the cost of any constraint violations incurred by this cluster assignment. Points are randomly re-ordered for each assignment sequence, and once a point $\mathbf{x}$ is assigned to a cluster, the subsequent points in the random ordering use the current cluster assignment of $\mathbf{x}$ to calculate possible constraint violations.

Note that this assignment step is order-dependent, since the subsets of $\mathcal{M}$ and $\mathcal{C}$ relevant to each cluster may change with the assignment of a point. We experimented with random ordering as well as a greedy strategy that first assigned instances that are closest to the cluster centroid and involved in a minimal number of constraints. These experiments showed that the order of assignment does not result in statistically significant differences in clustering quality; therefore, we used random ordering in our evaluation.

In the E-step, each point moves to a new cluster only if the component of $\mathcal{J}_{\text{mpckm}}$ contributed by this point decreases. So when all points are given their new assignment, $\mathcal{J}_{\text{mpckm}}$ will decrease or remain the same.

3.3. M-step

In the M-step, every cluster centroid $\boldsymbol{\mu}_h$ is first re-estimated using the points in corresponding $\mathcal{X}_h$. As a result, the contribution of each cluster to $\mathcal{J}_{\text{mpckm}}$ is minimized. The pairwise constraints do not take part in this centroid re-estimation step because the constraint violations only depend on cluster assignments, which do not change in this step. Thus, only the first term (the distance component) of $\mathcal{J}_{\text{mpckm}}$ is minimized. The centroid re-estimation step effectively remains the same as in K-Means.

The second part of the M-step performs metric learning, where the matrices $\{\mathbf{A}_h\}_{h=1}^{K}$ are re-estimated to decrease the objective function $\mathcal{J}_{\text{mpckm}}$. Each updated matrix of local weights $\mathbf{A}_h$ is obtained by taking the partial derivative $\frac{\partial \mathcal{J}_{\text{mpckm}}}{\partial \mathbf{A}_h}$ and setting it to zero, resulting in:

$$\begin{aligned}\mathbf{A}_h = |\mathcal{X}_h| \Bigg(& \sum_{\mathbf{x}_i \in \mathcal{X}_h} (\mathbf{x}_i - \boldsymbol{\mu}_h)(\mathbf{x}_i - \boldsymbol{\mu}_h)^T \\ & + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}_h} \frac{1}{2} w_{ij} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \mathbb{1}[l_i \neq l_j] \\ & + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}_h} \overline{w}_{ij} \big((\mathbf{x}'_h - \mathbf{x}''_h)(\mathbf{x}'_h - \mathbf{x}''_h)^T \\ & \qquad - (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \big) \, \mathbb{1}[l_i = l_j] \Bigg)^{-1}\end{aligned} \tag{7}$$

where $\mathcal{M}_h$ and $\mathcal{C}_h$ are subsets of must-link and cannot-link constraints respectively that contain points currently assigned to the h-th cluster.

Since each $\mathbf{A}_h$ is obtained by inverting the summation of covariance matrices in Eqn.(7), $\mathbf{A}_h^{-1}$, that summation must not be singular. If any of the obtained $\mathbf{A}_h^{-1}$ are singular, they can be conditioned via adding the identity matrix multiplied by a small fraction of the trace of $\mathbf{A}_h^{-1}$: $\mathbf{A}_h^{-1} = \mathbf{A}_h^{-1} + \epsilon \, \text{tr}(\mathbf{A}_h^{-1})\mathbf{I}$ (Saul & Roweis, 2003). If the $\mathbf{A}_h$ resulting from the inversion is negative definite, it is mended by projecting on the set $C = \{\mathbf{A} : \mathbf{A} \succeq 0\}$ of positive semi-definite matrices as described by Xing et al. (2003) to ensure that it parameterizes a distance metric.

For high-dimensional or large datasets, estimating the full matrix $\mathbf{A}_h$ can be computationally expensive. In such cases diagonal weight matrices can be used, which is equivalent to feature weighting, while using the full matrix corresponds to feature generation. In the case of diagonal $\mathbf{A}$, the d-th diagonal element, $a_{dd}^{(h)}$, corresponds to the weight of the d-th feature for the h-th cluster metric:

$$\begin{aligned}a_{dd}^{(h)} = |\mathcal{X}_h| \Bigg(& \sum_{\mathbf{x}_i \in \mathcal{X}_h} (\mathbf{x}_{id} - \boldsymbol{\mu}_{hd})^2 \\ & + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}_h} \frac{1}{2} w_{ij} (\mathbf{x}_{id} - \mathbf{x}_{jd})^2 \mathbb{1}[l_i \neq l_j] \\ & + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}_h} \overline{w}_{ij} \big((\mathbf{x}'_{hd} - \mathbf{x}''_{hd})^2 - (\mathbf{x}_{id} - \mathbf{x}_{jd})^2 \big) \mathbb{1}[l_i = l_j] \Bigg)^{-1}\end{aligned} \tag{8}$$

Intuitively, the first term in the sum, $\sum_{\mathbf{x}_i \in \mathcal{X}} (\mathbf{x}_{id} - \boldsymbol{\mu}_{hd})^2$, scales the weight of each feature proportionately to the feature's contribution to the overall cluster dispersion, analogously to scaling performed when computing unsupervised Mahalanobis distance. The last two terms that depend on constraint violations stretch each dimension attempting to mend the current violations. Thus, the metric weights are adjusted at each iteration in such a way that the contribution of different attributes to distance is variance-normalized, while constraint violations are minimized.

Instead of multiple metrics $\{\mathbf{A}_h\}_{h=1}^K$ the algorithm can use a single metric $\mathbf{A}$ for all clusters. The metric would be used and updated similarly to the description above, except that summations in Eqns.(7) and (8) would be over $\mathcal{X}$, $\mathcal{M}$, and $\mathcal{C}$ instead of $\mathcal{X}_h$, $\mathcal{M}_h$, and $\mathcal{C}_h$ respectively.

The objective function decreases after every cluster assignment, centroid re-estimation and metric learning step till convergence, implying that the MPCK-MEANS algorithm will converge to a local minima of $\mathcal{J}_{\text{mpckm}}$ as long as matrices $\{\mathbf{A}_h\}_{h=1}^K$ are obtained directly from Eqn.(7). If any $\mathbf{A}_h^{-1}$ is conditioned as described above to make it positive definite or if the maximally separated points $\{(x'_h, x''_h)\}_{h=1}^K$ change between iterations, convergence is no longer guaranteed theoretically; however, empirically this has not been a problem in our experience.

4. Experiments

4.1. Methodology and Datasets

Experiments were conducted on three datasets from the UCI repository: *Iris*, *Wine*, and *Ionosphere* (Blake & Merz, 1998); the *Protein* dataset used by Xing et al. (2003) and Bar-Hillel et al. (2003), and randomly sampled subsets from the *Digits* and *Letters* handwritten character recognition datasets, also from the UCI repository. For *Digits* and *Letters*, we chose two sets of three classes: {**I, J, L**} from *Letters* and {**3, 8, 9**} from *Digits*, sampling 10% of the data points from the original datasets randomly. These classes were chosen since they represent difficult visual discrimination problems. Table 1 summarizes the properties of the datasets: the number of instances N, the number of dimensions D, and the number of classes K.

Table 1. Datasets used in experimental evaluation

	Iris	*Wine*	*Ionosphere*	*Protein*	*Letters*	*Digits*
N	150	178	351	116	227	317
D	4	13	34	20	16	16
K	3	3	2	6	3	3

We have used pairwise F-Measure to evaluate the clustering results based on the underlying classes. F-Measure relies on the traditional information retrieval measures, adapted for evaluating clustering by considering same-cluster pairs:

$$Precision = \frac{\#PairsCorrectlyPredictedInSameCluster}{\#TotalPairsPredictedInSameCluster}$$

$$Recall = \frac{\#PairsCorrectlyPredictedInSameCluster}{\#TotalPairsInSameCluster}$$

$$F\text{–}Measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

We generated learning curves with 5-fold cross-validation for each dataset to determine the effect of utilizing the pairwise constraints. Each point on the learning curve represents a particular number of randomly selected pairwise constraints given as input to the algorithm. Unit constraint costs W and $\overline{W}$ were used for all constraints, original and inferred, since the datasets did not provide individual weights for the constraints. The clustering algorithm was run on the whole dataset, but the pairwise F-Measure was calculated only on the test set. Results were averaged over 50 runs of 5 folds.

4.2. Results and Discussion

First, we compared constraint-based and metric-based semi-supervised clustering with the integrated framework as well as purely unsupervised and supervised approaches. Figs.2-7 show learning curves for the six datasets. For each dataset, we compared five clustering schemes:

- MPCK-MEANS clustering, which involves both seeding and metric learning in the unified framework described in Section 2.4; a single metric parameterized by a diagonal matrix is used for all clusters;
- MK-MEANS, which is K-Means clustering with the metric learning component described in Section 3.3, without utilizing constraints for initialization; a single metric parameterized by a diagonal matrix is used for all clusters;
- PCK-MEANS clustering, which utilizes constraints for seeding the initial clusters and directs the cluster assignments to respect the constraints without doing any metric learning, as outlined in Section 2.2;
- K-MEANS unsupervised clustering;
- SUPERVISED-MEANS, which performs assignment of points to nearest cluster centroids inferred from constraints, as described in Section 3.1. This algorithm provides a baseline for performance of pure supervised learning based on constraints.

On the presented datasets, the unified approach (MPCK-MEANS) outperforms individual seeding (PCK-MEANS) and metric learning (MK-MEANS). Superiority of semi-supervised over unsupervised clustering illustrates that providing pairwise constraints is beneficial to clustering quality. Improvements of semi-supervised clustering over SUPERVISED-MEANS indicate that iterative refinement of

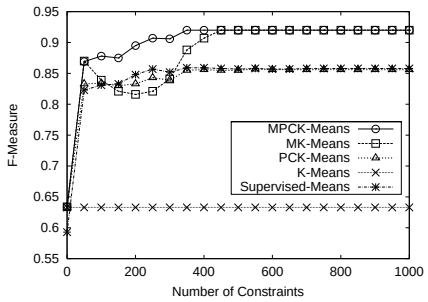

Figure 2. Iris: ablations

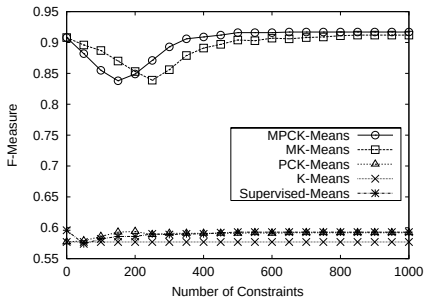

Figure 3. Wine: ablations

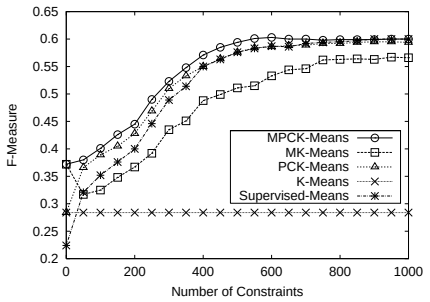

Figure 4. Protein: ablations

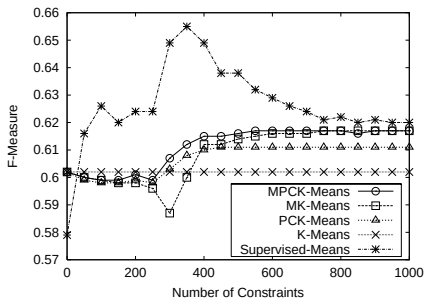

Figure 5. Ionosphere: ablations

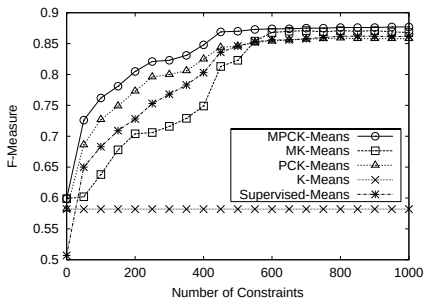

Figure 6. Digits-389: ablations

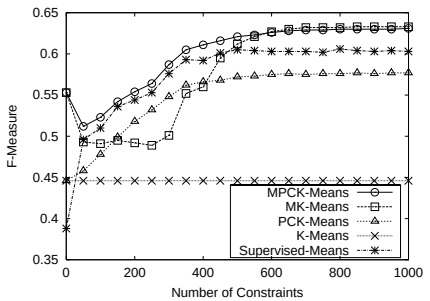

Figure 7. Letters-IJL: ablations

centroids using both constraints and unlabeled data outperforms purely supervised assignment based on neighborhoods inferred from constraints (for *Ionosphere*, MPCK-MEANS requires either the full weight matrix or individual cluster metrics to outperform SUPERVISED-MEANS, results for these experiments are shown on Fig.11).

For the *Wine*, *Protein*, and *Letter-IJL* datasets, the difference between methods that utilize metric learning (MPCK-MEANS and MK-MEANS) and those that do not (PCK-MEANS and regular K-Means) with no pairwise constraints indicates that even in the absence of constraints, weighting features by their variance (essentially using unsupervised Mahalanobis distance) improves clustering accuracy. For the *Wine* dataset, additional constraints provide an insubstantial improvement in cluster quality on this dataset, which shows that meaningful feature weights are obtained from scaling by variance using just the unlabeled data.

Some of the metric learning curves display a characteristic "dip", where clustering accuracy decreases when initial constraints are provided, but after a certain point starts to increase and eventually rises above the initial point on the learning curve. We conjecture that this phenomenon is due to the fact that metric parameters learned using few constraints are unreliable, and a significant number of constraints is required by the metric learning mechanism to estimate parameters accurately.

On the other hand, seeding the clusters with a small number of pairwise constraints has an immediate positive effect on the final cluster quality, while providing more pairwise constraints has diminishing returns, i.e., PCK-MEANS learning curves rise slowly. When both seeding and metric learning are utilized, the unified approach benefits from the individual strengths of the two methods, as can be seen from the MPCK-MEANS results.

In another set of experiments, we evaluated the utility of using individual metrics for each cluster and the usefulness of learning a full weight matrix $\mathbf{A}$ (feature generation) as opposed to a diagonal matrix (feature weighting). We have also compared our methods with RCA, a semi-supervised clustering algorithm that performs metric learning separately from the clustering process (Bar-Hillel et al., 2003), and that has been shown to outperform a similar approach by Xing et al. (2003). Figs.8-13 show learning curves for the six datasets on the following clustering schemes:

- MPCK-MEANS-S-D, which is same as MPCK-MEANS on Figs.2-7 and involves both seeding and metric learning; a single metric (S) parameterized by a diagonal matrix (D) is used for all clusters;
- MPCK-MEANS-M-D, which involves both seeding and metric learning; multiple metrics (M) parameterized by diagonal matrices (D) are used;
- MPCK-MEANS-S-F, which involves both seeding and metric learning; a single metric (S) parameterized by a full matrix (F) is used for all clusters;
- MPCK-MEANS-M-F, which involves both seeding and metric learning; multiple metrics (M) parameterized by full matrices (F) are used;

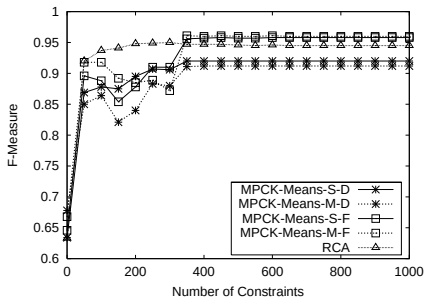

Figure 8. Iris: metric learning

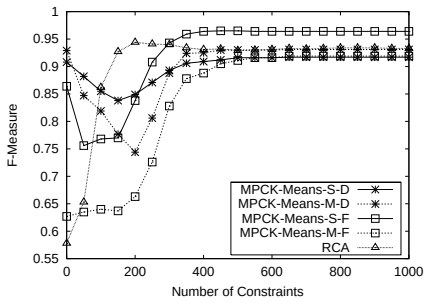

Figure 9. Wine: metric learning

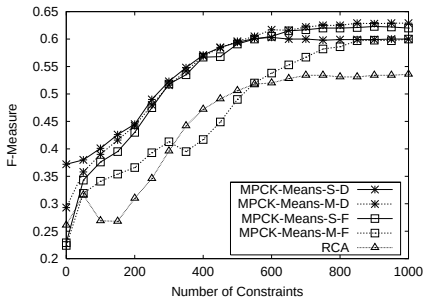

Figure 10. Protein: metric learning

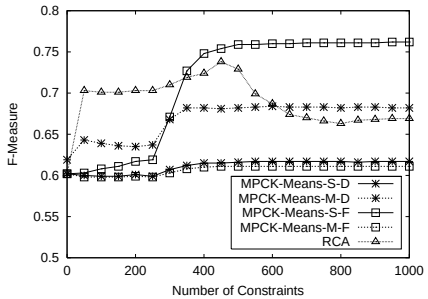

Figure 11. Ionosphere: metric learning

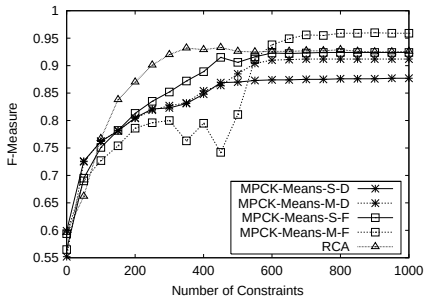

Figure 12. Digits-389: metric learning

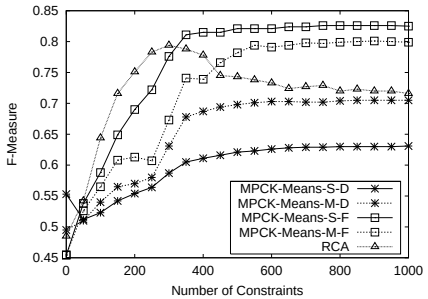

Figure 13. Letters-IJL: metric learning

- RCA clustering, which uses distance metric learning described in (Bar-Hillel et al., 2003) and initialization inferred from constraints as described in Section 3.1.

As can be seen from results, both full matrix parameterization and individual metrics for each cluster can lead to significant improvements in clustering quality. However, the relative usefulness of these two techniques varies between the datasets, e.g., multiple metrics are particularly beneficial for *Protein* and *Digits* datasets, while switching from a diagonal to a full weight matrix leads to large improvements on *Wine*, *Ionosphere*, and *Letters*. These results can be explained by the fact that the relative success of the two techniques depends on the properties of a particular dataset: using a full weight matrix helps when the attributes are highly correlated, while multiple metrics lead to improvements when clusters in the dataset are of different shapes or lie in different subspaces of the original space. A combination of the two techniques is most helpful when both of these requirements are satisfied, as for *Iris* and *Digits*, which was observed by visualizing these datasets. For other datasets, either multiple metrics or full weight matrix lead to maximum performance in isolation.

Comparing the performance of different variants of MPCK-MEANS with RCA, we can see that early on the learning curves, where few pairwise constraints are available, RCA leads to better metrics than MPCK-MEANS. However, as more training data is provided, the ability of MPCK-MEANS to learn from both supervised and unsupervised data as well as use individual metrics allows MPCK-MEANS to produce better clustering.

Overall, our results indicate that the integrated approach to utilizing pairwise constraints in clustering with individual metrics outperforms seeding and metric learning individually and leads to improvements in cluster quality. Extending the basic approach with a full parameterization matrix and individual metrics for each cluster can lead to significant improvements over the basic method.

5. Related work

In previous work on constrained pairwise clustering, Wagstaff et al. (2001) proposed the COP-KMeans algorithm that has a heuristically motivated objective function. Our formulation, on the other hand, has an underlying generative model based on Hidden Markov Random Fields (see (Basu et al., 2004) for a detailed analysis). Bansal et al. (2002) also proposed a framework for pairwise constrained clustering, but their model performs clustering using only the constraints, whereas our formulation uses both constraints and an underlying distance metric between the points for clustering.

Schultz and Joachims (2004) recently introduced a method for learning distance metric parameters based on relative comparisons. In unsupervised clustering, Domeniconi (2002) proposed a variant of K-Means that incorporated learning individual Euclidean metric weights for each cluster; our approach is more general since it allows metric learning to utilize pairwise constraints along with unlabeled data.

In recent work on semi-supervised clustering with pairwise constraints, Cohn et al. (2003) used gradient descent for weighted Jensen-Shannon divergence in the context of EM clustering. Xing et al. (2003) utilized a combination of gradient descent and iterative projections to learn a Mahalanobis metric for K-Means clustering. Also, Bar-Hillel et al. (2003) proposed a Redundant Component Analysis (RCA) algorithm that uses only must-link constraints to learn a Mahalanobis metric using convex optimization. All these metric learning techniques for clustering train a single metric first using only supervised data, and then perform clustering on the unsupervised data. In contrast, our method integrates distance metric learning with the clustering process and utilizes both supervised and unsupervised data to learn multiple metrics, which experimentally leads to improved results. Finally, a unified objective function for semi-supervised clustering with constraints was recently proposed by Segal et al. (2003), however, it did not incorporate distance metric learning.

6. Conclusions and Future Work

This paper has presented MPCK-MEANS, a new approach to semi-supervised clustering that unifies the previous constraint-based and metric-based methods. It is based on a variation of the standard K-Means clustering algorithm and uses pairwise constraints along with unlabeled data for constraining the clustering and learning distance metrics. In contrast to previously proposed semi-supervised clustering algorithms, MPCK-MEANS also allows clusters to lie in different subspaces and have different shapes.

By ablating the individual components of our integrated approach, we have experimentally compared metric learning and constraints in isolation with the combined algorithm. Our results have shown that by unifying the advantages of both techniques, the integrated approach outperforms the two techniques individually. We have shown that using individual metrics for different clusters, as well as performing feature generation via a full weight matrix in contrast to feature weighting with a diagonal weight matrix, can lead to improvements over our basic algorithm.

Extending our approach to high-dimensional datasets, where Euclidean distance performs poorly, is the primary avenue for future research. Other interesting topics for future work include selection of most informative pairwise constraints that would facilitate accurate metric learning and obtaining good initial centroids, as well as methodology for handling noisy constraints and cluster initialization sensitive to constraint costs.

7. Acknowledgments

We would like to thank anonymous reviewers and Joel Tropp for insightful comments. This research was supported in part by NSF grants IIS-0325116 and IIS-0117308, and by a Faculty Fellowship from IBM Corp.

References

Bansal, N., Blum, A., & Chawla, S. (2002). Correlation clustering. *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science (FOCS-02)* (pp. 238–247).

Bar-Hillel, A., Hertz, T., Shental, N., & Weinshall, D. (2003). Learning distance functions using equivalence relations. *Proceedings of 20th International Conference on Machine Learning (ICML-2003)* (pp. 11–18).

Basu, S., Banerjee, A., & Mooney, R. J. (2002). Semi-supervised clustering by seeding. *Proceedings of 19th International Conference on Machine Learning (ICML-2002)* (pp. 19–26).

Basu, S., Bilenko, M., & Mooney, R. J. (2004). A probabilistic framework for semi-supervised clustering. In submission, available at http://www.cs.utexas.edu/~ml/publication.

Bilenko, M., & Mooney, R. J. (2003). Adaptive duplicate detection using learnable string similarity measures. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)* (pp. 39–48).

Bilmes, J. (1997). *A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models* (Tech. Report ICSI-TR-97-021). ICSI.

Blake, C. L., & Merz, C. J. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Cohn, D., Caruana, R., & McCallum, A. (2003). *Semi-supervised clustering with user feedback* (Tech. Report TR2003-1892). Cornell University.

Demiriz, A., Bennett, K. P., & Embrechts, M. J. (1999). Semi-supervised clustering using genetic algorithms. *Artificial Neural Networks in Engineering (ANNIE-99)* (pp. 809–814).

Domeniconi, C. (2002). *Locally adaptive techniques for pattern classification*. Doctoral dissertation, University of California, Riverside.

Klein, D., Kamvar, S. D., & Manning, C. (2002). From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. *Proceedings of the The Nineteenth International Conference on Machine Learning (ICML-2002)* (pp. 307–314).

Kleinberg, J., & Tardos, E. (1999). Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science (FOCS-99)* (pp. 14–23).

Saul, L., & Roweis, S. (2003). Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, *4*, 119–155.

Segal, E., Wang, H., & Koller, D. (2003). Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics*, *19*, i264–i272.

Schultz, M., and Joachims, T. (2004). Learning a distance metric from relative comparisons. *Advances in Neural Information Processing Systems 16*.

Wagstaff, K., Cardie, C., Rogers, S., & Schroedl, S. (2001). Constrained K-Means clustering with background knowledge. *Proceedings of 18th International Conference on Machine Learning (ICML-2001)* (pp. 577–584).

Xing, E. P., Ng, A. Y., Jordan, M. I., & Russell, S. (2003). Distance metric learning, with application to clustering with side-information. *Advances in Neural Information Processing Systems 15* (pp. 505–512).