Unifying Search-Based and Similarity-Based Approaches to Semi-Supervised Clustering

Sugato Basu, Mikhail Bilenko and Raymond J. Mooney
Department of Computer Sciences
University of Texas at Austin
Austin, TX 78712
{sugato, mbilenko, mooney}@cs.utexas.edu

Abstract

Semi-supervised clustering employs a small amount of labeled data to aid unsupervised learning. Previous work in the area has employed one of two approaches: 1) Search-based methods that utilize supervised data to guide the search for the best clustering, and 2) Similarity-based methods that use supervised data to adapt the underlying similarity metric used by the clustering algorithm. This paper presents a unified approach based on the K-Means clustering algorithm that incorporates both of these techniques. Experimental results demonstrate that the combined approach generally produces better clusters than either of the individual approaches.

1 Introduction

In many learning tasks, there is a large supply of unlabeled data but limited labeled data since it can be expensive to generate. Consequently, semi-supervised learning, learning from a combination of both labeled and unlabeled data, has become a topic of significant recent interest [4, 9, 12]. More specifically, semi-supervised clustering, the use of class labels or constraints\(^1\) on some examples to aid standard unsupervised clustering, has been the focus of several projects in the past few years [13, 1, 10, 14].

Existing methods for semi-supervised clustering fall into two general approaches that we call search-based and similarity-based. In search-based approaches, the clustering algorithm itself is modified so that user-provided labels or constraints are used to bias the search for an appropriate partition. This can be done by modifying the objective function for evaluating clusterings so that it includes satisfying constraints [8], enforcing constraints during the clustering process [13], and/or initializing clusters based on labeled examples [1]. In similarity-based approaches, an existing clustering algorithm that uses a similarity metric is employed; however, the similarity metric is first trained to satisfy the labels or constraints in the supervised data. Several similarity metrics have been used for similarity-based semi-supervised clustering including string-edit distance trained using EM [2], KL divergence trained using gradient descent [6], Euclidean distance modified by a shortest-path algorithm [10], or Mahalanobis distances trained using convex optimization [14]. Several clustering algorithms using trained similarity metrics have been

\(^1\)Constraints typically specify that two examples must be in the same class (must-link) or must be in different classes (cannot-link).
employ for semi-supervised clustering, including single-link \[2\] and complete-link \[10\] agglomerative clustering, EM \[6\], and K-Means \[14\].

Unfortunately, similarity-based and search-based approaches to semi-supervised clustering have not been adequately compared experimentally, so their relative strengths and weaknesses are largely unknown. Also, the two approaches are not incompatible, therefore, applying a search-based approach with a trained similarity metric is clearly an additional option which may have advantages over both existing approaches. In this paper, we present a new unified semi-supervised clustering algorithm derived from K-Means that incorporates both metric learning and using labeled data as seeds and/or constraints. By ablating the similarity-based and search-based components in this unified method, we present experimental results comparing and combining the two approaches. Both methods for semi-supervision individually can improve clustering accuracy, although metric learning requires sufficient labeled data to be beneficial. Finally, when metric learning is helpful, combining it with seeding and constraints results in even better performance than either approach alone.

2 Problem Formulation

2.1 Semi-supervised K-Means Clustering with Constraints

K-Means is a clustering algorithm based on iterative relocation that partitions a dataset into \( K \) clusters, locally minimizing the total 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, \( z_{id} \) be the \( d \)-th component of \( \mathbf{x}_i \), \( \{ \mu_k \}_{k=1}^K \) represent the \( K \) cluster centroids, and \( l_i \) be the cluster assignment of a point \( \mathbf{x}_i \), where \( l_i \in \mathcal{L} \) and \( \mathcal{L} = \{1, \ldots, K\} \). The Euclidean K-Means algorithm creates a \( K \)-partitioning \( \{ \mathcal{X}_k \}_{k=1}^K \) of \( \mathcal{X} \) so that the objective function \( \sum_{\mathbf{x}_i \in \mathcal{X}} || \mathbf{x}_i - \mu_{l_i} ||^2 \) 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 priors of the mixture components and expectation under a particular conditional distribution \[1\].

In a semi-supervised clustering setting, a small amount of labeled data is available to aid the unsupervised clustering process. For pairwise constrained clustering, we consider a framework that has pairwise must-link and cannot-link constraints (with an associated cost of violating each constraint) between points in a dataset, in addition to having distances between the points \[13\]. Supervision in the form of constraints is generally more practical than providing class labels in the clustering framework, since true labels may be unknown a priori, while a human expert can easily specify whether pairs of points belong to the same cluster or different clusters.

Since K-Means clustering cannot handle pairwise constraints explicitly, we formulate the goal of clustering in the pairwise constrained clustering framework as minimizing a combined objective function, which is defined as the sum of the total distances between the points and their cluster centroids and the cost of violating the pairwise constraints. The mathematical formulation of this framework is motivated by the metric labeling problem and the generalized Potts model \[11, 5\].

In the pairwise constrained clustering framework, let \( \mathcal{M} \) be the set of unordered must-link pairs such that \( (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M} \) implies \( \mathbf{x}_i \) and \( \mathbf{x}_j \) should be assigned to the same cluster, and \( \mathcal{C} \) be the set of unordered cannot-link pairs such that \( (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C} \) implies \( \mathbf{x}_i \) and \( \mathbf{x}_j \) should be assigned to different clusters. Let \( W = \{ w_{ij} \} \) and \( \overline{W} = \{ \overline{w}_{ij} \} \) be two sets that give the weights corresponding to the must-link constraints in \( \mathcal{M} \) and the cannot-link constraints in \( \mathcal{C} \) respectively. Using this model, the problem of pairwise constrained
clustering is formulated as minimizing the following objective function, where point $\mathbf{x}_i$ is assigned to the partition $X_i$ with centroid $\mu_i$, and $I$ is the indicator function ($I[\text{true}]=1$, $I[\text{false}]=0$):

$$J_{\text{pckm}} = \sum_{\mathbf{x}_i \in X} ||\mathbf{x}_i - \mu_i||^2 + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in M} w_{ij} I[l_i \neq l_j] + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in C} w_{ij} I[|I]$$

(1)

We will refer to this model as the pairwise constrained K-Means (PC-KMEANS) model.

### 2.2 Semi-supervised Clustering via Metric Learning

Another avenue for utilizing labeled data involves adapting the distance metric employed by the clustering algorithm. Intuitively, this allows capturing the user’s view of which objects should be considered similar and which dissimilar. Since the original data representation may not be embedded in a space where clusters are sufficiently separated, modifying the distance metric transforms the representation so that distances between same-cluster objects are minimized, while distances between different-cluster objects are maximized. As a result, clusters discovered using the learned distance metric adhere more closely to the notion of similarity expressed by the labeled data than the clusters obtained using an untrained distance metric.

Following previous work [14], we can parameterize Euclidean distance with a symmetric positive-definite weight matrix $\mathbf{A}$ as follows: $||\mathbf{x}_i - \mathbf{x}_j||_A = \sqrt{(\mathbf{x}_i - \mu_i)^T \mathbf{A} (\mathbf{x}_i - \mu_i)}$. If $\mathbf{A}$ is restricted to be a diagonal matrix, then it scales each axis by a different weight and corresponds to feature weighting. In our clustering formulation, using the matrix $\mathbf{A}$ is equivalent to considering a generalized version of the K-Means model, where all the Gaussians have a covariance matrix $\mathbf{A}^{-1}$ [3]. The generalized K-Means objective function is then:

$$J_{\text{mkmeans}} = \sum_{\mathbf{x}_i \in X} ||\mathbf{x}_i - \mu_i||^2 - \log(\det(\mathbf{A}))$$

(2)

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

### 2.3 Unifying Constraints and Metric Learning in Clustering

Previous work on semi-supervised clustering [6, 14] that used labeled data for learning a metric only utilized the pairwise constraint information to learn weights that minimize constraint violations. We propose to incorporate metric learning directly into the clustering algorithm in a way that allows unlabeled data to influence the metric learning process along with pairwise constraints.

Combining objective functions (1) and (2) leads to the following objective function that attempts to minimize cluster dispersion under a learned metric along with minimizing the number of constraint violations:

$$J_{\text{combined}} = \sum_{\mathbf{x}_i \in X} ||\mathbf{x}_i - \mu_i||^2 + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in M} w_{ij} I[l_i \neq l_j] + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in C} w_{ij} I[|I]$$

(3)

If we assume uniform weights $w_{ij}$ and $w_{ij}$, as traditionally done in the generalized Potts model [5], one problem with this objective function would be that all constraint violations are treated equally. However, the cost of violating a must-link constraint between two **close** points should be higher than the cost of violating a must-link constraint between two points that are **far apart**. Such cost assignment reflects the intuition that it is a “worse error” to violate a must-link constraint between similar points, and such an error should have more impact on the metric learning framework. Multiplying the weights $w_{ij}$ with $f_M(\mathbf{x}_i, \mathbf{x}_j) = \max(\alpha_{\min}, \alpha_{\max} - ||\mathbf{x}_i - \mathbf{x}_j||_A^2)$ gives us the overall cost of violating a
must-link constraint between two points \( x_i \) and \( x_j \), where \( \alpha_{\text{min}} \) and \( \alpha_{\text{max}} \) are non-negative constants that correspond to minimum and maximum penalties respectively. They can be set as fractions of the square of the maximum must-link distance \( \max_{(x_i, x_j) \in \mathcal{M}} \|x_i - x_j\|^2 \), thus guaranteeing that the penalty for violating a constraint is always positive. Overall, this formulation enables the penalty for violating a must-link constraint to be proportional to the “seriousness” of the violation.

Analogously, the cost of violating a cannot-link constraint between two distant points should be higher than the cost of violating a cannot-link constraint between points that are close, since the former is a “worse error”. Multiplying weights \( w_{ij} \) with \( f_C(x_i, x_j) = \min(\alpha_{\text{min}} + \|x_i - x_j\|^2, \alpha_{\text{max}}) \) allows us to take the “seriousness” of the constraint violation into account. The combined objective function then becomes:

\[
J_{\text{mpckm}} = \sum_{x_i \in \mathcal{X}} \|x_i - \mu_h\|^2 - \log(\det(A)) + \sum_{(x_i, x_j) \in \mathcal{M}} w_{ij} f_M(x_i, x_j)\mathbb{I}[i \neq j] + \sum_{(x_i, x_j) \in \mathcal{C}} w_{ij} f_C(x_i, x_j)\mathbb{I}[i = j]
\]

This objective function \( J_{\text{mpckm}} \) is greedily optimized by our proposed metric pairwise constrained K-Means (MPC-KMEANS) algorithm that uses a K-Means-type iteration.

3 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 weight sets \( \mathcal{W} \) and \( \overline{\mathcal{W}} \), and the number of clusters to form \( K \), metric pairwise constrained K-Means (MPC-KMEANS) finds a disjoint \( K \) partitioning \( \{\mathcal{X}_h\}_{h=1}^K \) of \( \mathcal{X} \) (with each partition having a centroid \( \mu_h \)) such that \( J_{\text{mpckm}} \) is (locally) minimized.

The algorithm MPC-KMEANS has two components. Utilizing constraints during cluster initialization and satisfaction of the constraints during every cluster assignment step constitutes the search-based component of the algorithm. Learning the distance metric by re-estimating the weight matrix \( A \) during each algorithm iteration based on current constraint violations is the similarity-based component.

The search-based component uses the constraints to bias the clustering towards a good region of the search space, thereby possibly reducing the chances of it getting stuck in poor local optima. The similarity-based technique transforms the space where datapoints are embedded to respect the user-provided constraints, thus capturing the notion of similarity appropriate for the dataset from the user’s perspective.

3.1 Initialization

To generate the seed clusters during the initialization step of MPC-KMEANS, we take the transitive closure of the must-link constraints [13] and augment the set \( \mathcal{M} \) by adding these entailed constraints, assuming consistency of the constraints. Let the number of connected components in the augmented set \( \mathcal{M} \) be \( \lambda \). These \( \lambda \) connected components are used to create \( \lambda \) neighborhood sets \( \{\mathcal{N}_p\}_{p=1}^\lambda \), where each neighborhood set consists of points connected by must-links from the augmented set \( \mathcal{M} \). For every pair of neighborhoods \( \mathcal{N}_p \) and \( \mathcal{N}_{p'} \) that have at least one cannot-link between them, we add cannot-link constraints between every pair of points in \( \mathcal{N}_p \) and \( \mathcal{N}_{p'} \) and augment the cannot-link set \( \mathcal{C} \) by these entailed constraints, assuming consistency of the 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 \( \lambda \) neighborhood sets \( \{\mathcal{N}_p\}_{p=1}^\lambda \). The cluster centroids are initialized from these neighborhood sets (for details, see Figure 1).
3.2 E and M steps

MPC-KMEANS alternates between cluster assignment in the E-step, and centroid estimation and metric learning in the M-step (see Figure 1).

![Figure 1: MPC-KMEANS algorithm](image)

In the E-step of MPC-KMEANS, every point \( \mathbf{x} \) is assigned to a cluster so that the sum of the distance of \( \mathbf{x} \) to the cluster centroid and the cost of constraint violations possibly incurred by this cluster assignment is minimized. Note that this assignment step is order-dependent, since the subsets of \( \mathcal{M} \) and \( \mathcal{C} \) associated with each cluster may change with the assignment of a point. In the cluster assignment step, each point moves to a new cluster only if the component of \( J_{\text{mpckm}} \) contributed by this point decreases. So when all points are given their new assignment, \( J_{\text{mpckm}} \) will decrease or remain the same.

In the M-step, the cluster centroids \( \mu_k \) are first re-estimated using the points in \( \mathcal{X}_t \). As a result, the contribution of each cluster to \( J_{\text{mpckm}} \) is minimized. The pairwise constraints do not take part in this centroid re-estimation step because the constraints are not an explicit function of the centroid. Thus, only the first term (the distance component) of \( J_{\text{mpckm}} \) is minimized in this step. The centroid re-estimation step effectively remains the same as K-Means.

The second part of the M-step is metric learning, where the matrix \( \mathbf{A} \) is re-estimated to decrease the objective function \( J_{\text{mpckm}} \). The updated matrix \( \mathbf{A} \) is obtained by taking the partial derivative \( \frac{\partial J_{\text{mpckm}}}{\partial \mathbf{A}} \) and setting it to zero, resulting in:

\[
\mathbf{A} = \left( \sum_{i \in \mathcal{X}} (\mathbf{x}_i - \mu_{i_1})(\mathbf{x}_i - \mu_{i_2})^T - \sum_{(i,j) \in \mathcal{M}} w_{ij}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \mathbb{I}[i \neq j] \right)^{-1} - \sum_{(i,j) \in \mathcal{C}} w_{ij}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \mathbb{I}[i = j]
\]

Since estimating a full matrix \( \mathbf{A} \) from limited training data is difficult, we limit ourselves to diagonal \( \mathbf{A} \), which is equivalent to learning a metric via feature weighting. In that case, the \( d \)-th diagonal element of \( \mathbf{A} \), \( \alpha_{dd} \), corresponds to the weight of the \( d \)-th feature.
\begin{align*}
    \alpha_{dd} &= \left( \sum_{x_i \in \mathcal{X}} (x_{id} - \mu_{kd})^2 - \sum_{(x_i, x_j) \in \mathcal{M}} w_{ij} (x_{id} - x_{jd})^2 \mathbb{1}[l_i \neq l_j] + \sum_{(x_i, x_j) \in \mathcal{C}} \omega_{ij} (x_{id} - x_{jd})^2 \mathbb{1}[l_i = l_j] \right)^{-1} 
\end{align*}

The first term in the sum, \( \sum_{x_i \in \mathcal{X}} (x_{id} - \mu_{kd})^2 \), scales the weight of each feature proportionately to the feature’s contribution to the overall cluster dispersion, analogously to scaling performed when computing Mahalanobis distance. The second two terms that depend on constraint violations, \(- \sum_{(x_i, x_j) \in \mathcal{M}} w_{ij} (x_{id} - x_{jd})^2 \mathbb{1}[l_i \neq l_j] \) and \( \sum_{(x_i, x_j) \in \mathcal{C}} \omega_{ij} (x_{id} - x_{jd})^2 \mathbb{1}[l_i = l_j] \), respectively contract and 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 equalized, while constraint violations are minimized.

The objective function decreases after every cluster assignment, centroid re-estimation and metric learning step till convergence, implying that the MPC-KMEANS algorithm will converge to a local minima of \( J_{mpckm} \).

4 Experiments

4.1 Methodology and Datasets

Experiments were conducted on several datasets from the UCI repository: Iris, Wine, and representative randomly sampled subsets from the Pen-Digits and Letter datasets. For Pen-Digits and Letter, we chose two sets of three classes: \{I, J, L\} from Letter and \{3, 8, 9\} from Pen-Digits, sampling 20% of the datapoints from the original datasets randomly. These classes were chosen from the handwriting recognition datasets since they intuitively represent difficult visual discrimination problems.

We have used pairwise F-measure to evaluate the clustering results. It is defined as the harmonic mean of precision and recall, the traditional information retrieval measures adapted for evaluating clustering by considering same-cluster pairs [2]. We generated learning curves with 10-fold cross-validation for each dataset to determine the effect of utilizing the pairwise constraints. Each point in the learning curve represents a particular number of pairwise constraints given as supervision to the algorithm. Unit constraint weights \( W \) and \( \tilde{W} \) were used, since the datasets did not provide individual weights for the constraints. The clustering algorithm is run on the whole dataset, but the pairwise F-measure is calculated only on the test set. Results were averaged over 50 runs of 10-fold cross-validation.

4.2 Results and Discussion

Figures 2-4 show learning curves for the four datasets. For each dataset, we compared four semi-supervised clustering schemes:

- MPC-KMEANS clustering, which involves both seeding and metric learning in the unified framework described in Section 2.3;
- M-KMEANS, which is K-Means clustering with the metric learning component described in Section 2.2 without utilizing the constraints for seeding;
- PC-KMEANS clustering, which utilizes constraints for seeding the initial clusters and forces the cluster assignments to respect the constraints;
- Unsupervised K-Means clustering.

On the presented datasets, the unified approach outperforms individual seeding and metric learning approaches. The learning curves illustrate that providing pairwise constraints is beneficial to clustering quality. For the Wine and Letter-IJL datasets, the difference between methods that utilize metric learning (MPC-KMEANS and M-KMEANS) and those that do not (PC-KMEANS and regular K-Means) with no pairwise constraints indicates that even
in the absence of constraints, weighting features by their variance (essentially using 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 outperforms the initial point of the learning curve. We conjecture that this phenomenon is due to the fact that feature weights learned from few constraints are unreliable, while increasing the number of constraints provides the metric learning mechanism enough data to estimate good metric parameters.

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., PC-KMEANS 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 MPC-KMEANS results.

Overall, our results indicate that the unified approach to utilizing pairwise constraints in clustering outperforms using seeding and metric learning individually and leads to improvements in cluster quality.

5 Future Work

Extending our approach to high-dimensional datasets like text, where Euclidean distance performs poorly, is the primary avenue for future research. We are currently working on a formulation that utilizes an objective function similar to $J_{mpckm}$ for spherical K-Means [1].
The weight matrix $A$ is likely to be singular for high-dimensional data; such a scenario could be handled by regularization. Alternatively, dimensionality reduction methods such as Latent Semantic Indexing [7] could be used to project high-dimensional data onto a lower-dimensional space.

Comparing our unified approach to other search-based and similarity-based techniques, e.g., those of [14] and [6], is another area for future work. We are also planning to incorporate active selection of pairwise constraints into our proposed approach.

In some situations, obtaining data labels directly instead of pairwise constraints may be possible. While it is possible to infer pairwise constraints in such scenarios from the labels, the number of pairwise constraints grows quadratically with the amount of labeled data, making training on the entire set of pairwise constraints intractable. Selection mechanisms for pairwise constraints are an interesting topic for future work.

6 Conclusions

This paper has presented a new approach to semi-supervised clustering that unifies the previous search-based and similarity-based methods. We have presented a general formalization of the semi-supervised learning problem that has allowed us to develop a variation of the standard K-Means clustering algorithm that uses supervised data both as seeds and constraints during search, as well as for adapting the underlying distance metric. By ablating the individual components of this unified approach, we have experimentally compared the individual approaches to each other and to their combination. When only small amounts of supervised data are used, the search-based approach produces more accurate clusters than the similarity-based approach. By combining the advantages of both techniques, the unified approach generally performs better than either of the approaches individually.

References