Semi-Supervised Clustering with Partial Background Information

Jing Gao* Pang-Ning Tan† Haibin Cheng ‡

Abstract

Incorporating background knowledge into unsupervised clustering algorithms has been the subject of extensive research in recent years. Nevertheless, existing algorithms implicitly assume that the background information, typically specified in the form of labeled examples or pairwise constraints, has the same feature space as the unlabeled data to be clustered. In this paper, we are concerned with a new problem of incorporating partial background knowledge into clustering, where the labeled examples have moderate overlapping features with the unlabeled data. We formulate this as a constrained optimization problem, and propose two learning algorithms to solve the problem, based on hard and fuzzy clustering methods. An empirical study performed on a variety of real data sets shows that our proposed algorithms improve the quality of clustering results with limited labeled examples.

1 Introduction

The topic of semi-supervised clustering has attracted considerable interests among researchers in the data mining and machine learning community [2, 3, 4, 8]. The goal of semi-supervised clustering is to obtain a better partitioning of the data by incorporating background knowledge. Although current semi-supervised algorithms have shown significant improvements over their unsupervised counterparts, they assume that the background knowledge are specified in the same feature space as the unlabeled data. These algorithms are therefore inapplicable when the background knowledge is provided by another source with a different feature space.

There are many examples in which the background knowledge does not share the same feature space as the data to be mined. For example, in computational biology, analysts are often interested in the clustering of genes or proteins. Extensive background knowledge is available in the biomedical literature that might be useful to aid the clustering task. In these examples, the background knowledge and the data to be mined may share some common features, but their overall feature sets are not exactly identical. We termed this task as semi-supervised clustering with partial background information.

One possible approach is to consider only the common features between the labeled and unlabeled data, and then apply existing semi-supervised learning techniques to guide the clustering process on the reduced feature sets. However, the clustering results for such an approach tend to be poor because the approach does not fully utilize all the information available in the labeled and unlabeled data. As will be shown in Section 4, when the number of shared features is too few, the clustering results might be worse than applying unsupervised clustering on the unlabeled data.

The key challenge of semi-supervised clustering with partial background knowledge is to determine how to utilize both the shared and non-shared features while performing clustering on the full feature set of the unlabeled data. The semi-supervised algorithm also has to recognize the possibility that the shared features might be useful for identifying certain clusters but not for others.

To overcome these challenges, we propose a novel approach for incorporating partial background knowledge into the clustering algorithm. The idea is to first assign a confidence-rated label to each unlabeled example by using a classifier built from the shared feature set of the data. A constrained clustering algorithm is then applied to the unlabeled data, where the constraints are given by the unlabeled examples whose classes are predicted with confidence greater than a user specified threshold, δ. The algorithm also assigns a weight factor to each class, in proportion to the ability of the shared feature set to discriminate between examples that belong to the class from other classes in the labeled data. We develop hard and soft (fuzzy) solutions to the constrained clustering problem and perform a variety of experiments using real data sets to demonstrate the effectiveness of our algorithms.

2 Incorporating Background Information

Let $\mathbf{U}$ be a set of n unlabeled examples drawn from a d-dimensional feature space, while $\mathbf{L}$ be a set of s labeled examples drawn from a d'-dimensional feature space. Without loss of generality, we assume that the first p features for the labeled and unlabeled data are identical—together, they form the shared feature set between $\mathbf{U}$ and $\mathbf{L}$. Each labeled example $l_j \in \mathbf{L}$ has a corresponding label $y_j \in \{1, 2, \cdots, q\}$, where q is the number of classes. The objective of clustering is to assign the unlabeled examples into their corresponding

*Michigan State University.
†Michigan State University.
‡Michigan State University.

clusters. We use the notation $C = \{c_1, c_2, \cdots, c_k\}$ to represent the k cluster centroids, where $\forall i : c_i \in \Re^d$.

Our proposed framework for incorporating partial background knowledge into the clustering task consists of the following steps:

Step 1: Build a classifier $\mathcal{C}$ using the shared feature set of the labeled examples, **L**.

Step 2: Apply the classifier $\mathcal{C}$ to predict the labels of the examples in **U**.

Step 3: Cluster the examples in **U** subjected to the constraints given by the predicted labels in Step 2.

The details of these steps are explained in the next three subsections. To better understand the overall process, we shall explain the constrained clustering step first before discussing the classification steps.

2.1 Constrained Clustering with Partial Background Knowledge The objective function for the clustering task has the following form:

$$Q = \sum_{i=1}^{n}\sum_{j=1}^{k}(t_{ij})^m d_{ij}^2 + R\sum_{i=1}^{n'}\sum_{j=1}^{k} w_j |t_{ij} - f_{ij}|^m \tag{2.1}$$

The first term is equivalent to the objective function for standard k-means algorithm. The second term penalizes any violations of constraints on the predicted labels of examples in **U**. d_{ij} is the square distance between data point u_i and the cluster centroid c_j. R is a regularization parameter that determines the tradeoff between the distance to centroids and the amount of constraint violations. w_j is a weight factor that represents how informative is the shared feature set in terms of discriminating cluster j from other clusters. Finally, t_{ij} is the cluster membership function, which indicates whether the unlabeled example u_i should be assigned to cluster j.

For hard clustering, $m = 1$ and f_{ij} is a discrete variable in $\{0, 1\}$ that satisfies the constraint $\sum_{j=1}^{k} f_{ij} = 1$. For fuzzy clustering, $m = 2$ and f_{ij} is a continuous variable in the range $[0, 1]$ that satisfies $\sum_{j=1}^{k} f_{ij} = 1$. The next subsection explains how the confidence-rated labels (f_{ij}) are obtained.

2.2 Classification of Unlabeled Examples We employ the nearest-neighbor classifier for labeling the examples in **U**. For each example $u_i \in$ **U**, we compute $NN(i, j)$, which is the distance between u_i to its nearest neighbor in **L** from class j. Note that the distance is computed based on the shared feature set between **U** and **L**. If $\min_j NN(i, j) \geq \delta$, then the example is ignored from the second term of the objective function. δ is the threshold for filtering unnecessary constraints due to examples that have low confidence values.

Once we have computed the nearest-neighbor distances, the class membership function f_{ij} is determined in the following way. For hard clustering, each example $u_i \in$ **U** is labeled as follows:

$$f_{ij} = \begin{cases} 1 & \text{if } j = \min_{1 \leq l \leq k} NN(i, l) \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

For fuzzy clustering, f_{ij} measures the confidence that u_i belongs to the class c_j. In [5], several confidence measures are introduced for nearest neighbor classification algorithms. We adopt the following confidence measure for classification output:

$$M(i, j) = \frac{1}{\alpha + NN(i, j)},$$

where α is a smoothing parameter. The value of f_{ij} is then computed as follows:

$$f_{ij} = \frac{M(i, j)}{Z} \tag{2.3}$$

where Z is a normalization factor so that $\sum_{j=1}^{k} f_{ij} = 1$.

To incorporate f_{ij} into the objective function Q, we need to identify the class label associated with each cluster j. If $k = q$, the one-to-one correspondence between the classes and clusters is established during cluster initialization. The cluster centroids are initialized in the following way: after classifying the unlabeled examples, we choose examples that have confidence greater than δ and use them to compute the centroid for each class. Because there is a high penalty for assigning an example to a cluster that is different than its predicted class, we expect the class labels for the clusters to remain unchanged even though the centroid locations may change and some of their examples are re-assigned to other clusters in later rounds.

If the number of clusters exceeds the number of classes, the initial centroids for the remaining clusters that do not have a corresponding class label are chosen randomly. Furthermore, the unlabeled examples that are assigned to these clusters do not contribute to the second term of the objective function. Finally, if the number of clusters is less than the number of classes, we may either ask the domain experts to determine the k labels in q that are contained in the unlabeled data set or to assign each cluster to the majority class of examples within the cluster.

2.3 Cluster Weighting The proposed objective function should also take into account how informative is the shared feature set. We use the weight w_j to reflect the importance of the shared feature set in terms of discriminating cluster j from other clusters. The concept of feature weighting has been used in clustering by Frigui et. al. [7].

Let $D(d, j)$ denote the discriminating ability of the

feature set $\Re^d$ for cluster j:

$$D(d,j) = \frac{\sum_{u_i \in c_j} d(x_i, c_l, \Re^d)}{\sum_{\substack{l \neq j \\ u_i \in c_j}} d(x_i, c_j, \Re^d)} \tag{2.4}$$

where $d(u_i, c_j, \Re^d)$ is the distance between u_i and centroid c_j based on the feature set $\Re^d$. Intuitively, $D(d,j)$ is the ratio of the between-cluster distance and the within-cluster distance. The higher the ratio, the more informative is the feature set in terms of discriminating cluster j from other clusters.

The weight of cluster j is then determined as follows:

$$w_j = D(p,j)/D(d,j) \tag{2.5}$$

where p is the number of shared features and d is the total number of features in **U**.

3 Algorithms

This section presents the SemiHard and SemiFuzzy clustering algorithms for solving the constrained optimization problem posed in the previous section. The EM algorithm [1] provides an iterative method for solving the optimization problem. The basic idea of the algorithm is as follows: first, an initial set of centroids is chosen. During the E-step, the cluster centroids are fixed while the configuration matrix $T = [t_{ij}]$ is varied until the objective function is minimized. During the M step, the configuration matrix is fixed while the centroids are recomputed to minimize the objective function. The procedure is repeated until the objective function converges.

3.1 SemiHard Clustering In SemiHard clustering, $[t_{ij}]$ takes the value from $\{0,1\}$. The objective function for hard clustering can be re-written as follows:

$$Q = \sum_{i=1}^{n} A_i^t T_i + b_i$$

where b_i is a constant, T_i is the i-th row of the configuration matrix, $A_i = \{a_{ij}\}_{j=1}^k$ is the coefficient matrix with

$$a_{ij} = \begin{cases} d_{ij}^2 - Rw_j & \text{if } f_{ij} = 1 \\ d_{ij}^2 + Rw_j & \text{otherwise} \end{cases} \tag{3.6}$$

During the E-step, we should minimize the contribution of each point to the objective function Q. Clearly, minimizing Q subject to the constraints is a linear programming problem. From [6], the minimum can be achieved by setting t_{ij} as follows:

$$t_{ij} = \begin{cases} 1 & \text{if } j = \min_l a_{il} \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

During the M-step, since the configuration matrix is fixed, the second term of the objective function is unchanged. Minimizing Q is therefore equivalent to minimizing the first term. The centroid update formula is :

$$c_j = \frac{\sum_{i=1}^n t_{ij} u_i}{\sum_{i=1}^n t_{ij}} \tag{3.8}$$

3.2 SemiFuzzy Clustering In fuzzy clustering, $[t_{ij}]$ are continuous variables. During the E-step, we should minimize Q with respect to T subject to the constraints. We may use the Lagrange multiplier method and obtain:

$$Q = \sum_{i=1}^{n}\sum_{j=1}^{k}(t_{ij})^2 d_{ij}^2 + \sum_{i=1}^{n}\sum_{j=1}^{k} w_j (t_{ij} - f_{ij})^2 - \sum_{i=1}^{n} \lambda_i (\sum_{j=1}^{k} t_{ij} - 1)$$

Differentiating Q yields the following:

$$t_{ij} = \frac{\lambda_i + 2Rw_j f_{ij}}{2(d_{ij}^2 + Rw_j)} \tag{3.9}$$

The solution for λ_i is:

$$\lambda_i = (1 - \sum_{j=1}^{k} \frac{Rw_j f_{ij}}{d_{ij}^2 + Rw_j}) / \sum_{j=1}^{k} \frac{1}{2(d_{ij}^2 + Rw_j)}$$

During the M-step, the objective function once again depends only on the first term. Therefore,

$$c_j = \frac{\sum_{i=1}^n t_{ij}^2 u_i}{\sum_{i=1}^n t_{ij}^2} \tag{3.10}$$

4 Experiments

The following data sets are used to evaluate the performances of our algorithms:

Table 1: Data set descriptions.

	# of instances	# of attributes	# of classes
Waveform	5000	21	3
Shuttle	4916	9	2
Web	8500	26	2
Physio-logical	13239	10	3

Among these data sets, Waveform and Shuttle are obtained from the UCI machine learning repository. Web is a data set used for categorizing Web sessions into accesses by human users and Web crawlers. The physiological data set was taken from the Physiological Data Modeling contest at ICML 2004, which corresponds to sensor measurements collected using an armband device.

We compared the performance of SemiHard and SemiFuzzy algorithms against the unsupervised clustering algorithm **K-means** as well as the supervised learning algorithm

Nearest Neighbor (NN). The evaluation metrics we use is an external cluster validation measure, i.e., the error rate of each algorithm.

4.1 Comparison with Baseline Methods Table 2 summarizes the results of our experiments when comparing SemiHard and SemiFuzzy to K-means and NN algorithms. In all of these experiments, 1% of the data set is labeled while the remaining 99% is unlabeled. Furthermore, 10% of the features are randomly selected as shared features. Clearly, both SemiHard and SemiFuzzy algorithms outperform their unsupervised and supervised learning counterparts even at 1% labeled examples.

Table 2: Performance comparison for various algorithms.

	SemiHard	SemiFuzzy	K-means	NN
Waveform	0.4534	**0.4186**	0.4700	0.4706
Shuttle	0.2054	**0.1617**	0.2168	0.2316
Web	0.4766	**0.4317**	0.4994	0.4597
Physio-logical	0.4930	**0.4719**	0.5828	0.5009

4.2 Variation in the Amount of Labeled Examples In this experiment, we vary the amount of labeled examples from 1% up to 20% and compare the performances of the four algorithms. We use the Waveform data set for this experiment. The percentage of shared features is set to be 10%. Figure 1 shows the results of our experiment. The error rate for SemiFuzzy is lowest, followed by the SemiHard algorithm.

For K-means, the error rate is almost unchanged (the slight variability is due to different choices of initial centroids). For nearest neighbor classification, adding more labeled data helps to decrease the error rate. But when the amount of labeled examples continues to increase, the error rate reaches a steady state. This result suggests that having 10% of the labeled examples is sufficient to discriminate records from different classes in the Waveform data set; adding more labeled data does not always help.

4.3 Variation in the Amount of Common Features Figure 2 shows the effect of varying the percentage of shared features from 10% to 60%. We use the Waveform data set for our experiment. We fix the percentage of labeled examples at 1%. It can be seen that adding more shared features improves the error rates of SemiHard, SemiFuzzy, and NN algorithms. This is because the more features shared by the labeled and unlabeled data, the more informative it is to aid the clustering task. In all of these experiments, SemiFuzzy gives the lowest error rates, followed by SemiHard and the NN algorithm.

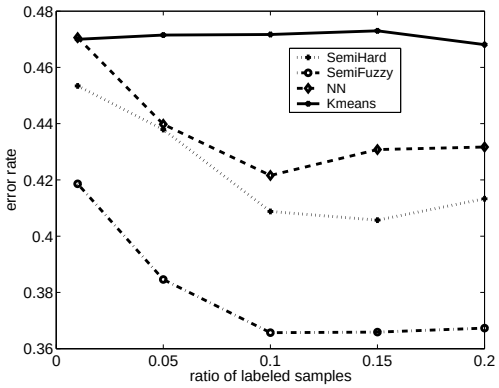

Figure 1: Error rate for various percentage of labeled examples.

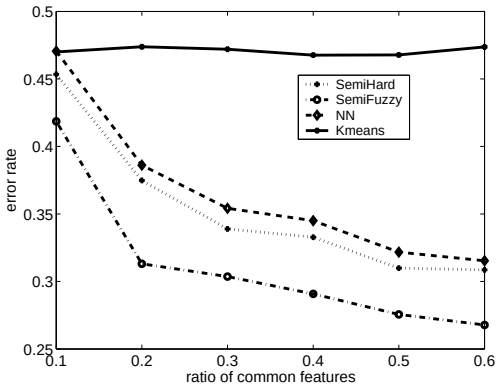

Figure 2: Error rate for various percentage of shared features.

4.4 Limited Labeled Information This section analyzes the performances of our algorithms when not all classes are present in the labeled data. As in the previous two experiments, we report the results for the Waveform data set and assume 1% of the examples are labeled. Furthermore, the labeled data set contains only examples from two of the three classes. We then add examples from the third class to the unlabeled data set incrementally from 10% to 40%. Figure 3 shows how the error rate varies as the percentage of examples from the third class is increased. Notice that SemiHard and SemiFuzzy outperform K-means when there are few examples from the omitted class. However, when more examples from the omitted class are added to the unlabeled data set, the performances of our algorithms may become worse than K-means.

4.5 Real-Life Applications We demonstrate the application of semi-supervised clustering with partial background knowledge in the context of document clustering. For this

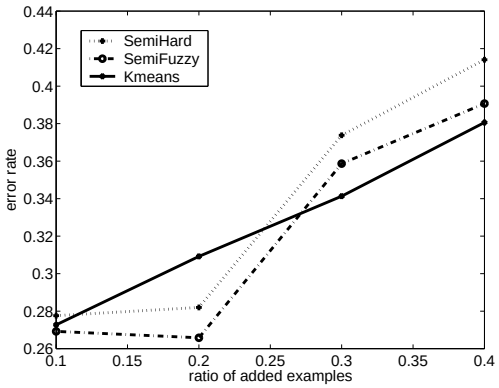

Figure 3: Error rate while varying the percentage of added examples

experiment, we use the Usenet Newsgroups data set, where we choose articles from 5 newsgroups, each of which contains 1000 articles. After preprocessing, we obtain a vocabulary of 19842 words. A document-term matrix is constructed, where the word count vector for each document is normalized to unit length.

We assume that the labels for some of the older articles are known and would like to use the information to cluster newer articles posted on the newsgroups. We first sort the documents according to their posting dates and create a labeled data set based on the first p% of the articles. The remaining documents form the unlabeled data set. Although the vocabulary of words used in the older and newer collections may not be exactly identical, the feature set constructed from both collections has some overlapping features. Unlike the previous experiments, we modify our algorithms slightly to account for the characteristics of the documents. First, we use a dissimilarity measure based on the cosine similarity. Second, we apply naïve Bayes (instead of nearest-neighbor) as the underlying classifier for Step 1 of our algorithm. Table 3 compares the error rate of SemiHard against K-means and the naïve Bayes classifier.

Table 3: Error rates on the Newsgroups Data.

Percentage of Labels	SemiHard	NB	K-means	Common Terms #
10%	**0.0233**	0.0455	0.2344	10452
5%	**0.0952**	0.1387	0.2244	8572
1%	0.2602	0.3812	**0.2040**	4915

The results suggest that the error rates for SemiHard and NB improve as the percentage of labeled examples increases. The performance of K-means degrades slightly because there are less unlabeled examples available for clustering. When there are sufficient labeled examples, both SemiHard and NB have better performances than K-means. However, when the amount of labeled examples is limited (1%), their error rates are worse than K-means since the information conveyed by the labeled examples might not be accurate enough. Finally, SemiHard always outperforms NB because it utilizes information from both the shared and non-shared features.

5 Conclusion

In this paper, we present a principled approach for incorporating partial background information into a clustering algorithm. The novelty of this approach is that the background knowledge can be specified in a different feature space than the unlabeled data. We illustrate how the objective function for K-means clustering can be modified to incorporate the constraints due to partially labeled information. In principle, our methodology is applicable to any base classifiers. We present both hard and fuzzy clustering solutions to the constrained optimization problem. Using a variety of real data sets, we demonstrate the effectiveness of our approach over standard K-means and the nearest-neighbor classification algorithms.

References

[1] N. L. A.P. Dempster and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, Series B(39):1–38, 1977.

[2] S. Basu. Semi-supervised clustering: Learning with limited user feedback. *Department of Computer Sciences, University of Texas at Austin*, 2003.

[3] S. Basu, A. Banerjee, and R. J. Mooney. Semi-supervised clustering by seeding. In *ICML '02: Proceedings of the International Conference on Machine Learning*, pages 27–34, 2002.

[4] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *KDD '04: Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 59–68. ACM Press, 2004.

[5] S. J. Delany, P. Cunningham, and D. Doyle. Generating estimates of classification confidence for a case-based spam filter. In *Proceedings of the 6th International Conference on Case-based Reasoning*, page To appear, 2005.

[6] R. Fletcher. *Practical methods of optimization*. John Wiley & Sons, Chichester, second edition, 1986.

[7] H. Frigui and O. Nasraoui. Unsupervised learning of prototypes and attribute weights. *Pattern Recognition*, 37(3):943–952, 2004.

[8] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained K-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 577–584, 2001.