Proceedings of the 19th International Conference on Machine Learning (ICML-2002), pp. 19-26, Sydney, Australia, July 2002

Semi-supervised Clustering by Seeding

Sugato Basu SUGATO@CS.UTEXAS.EDU
Department of Computer Sciences, University of Texas, Austin, TX 78712

Arindam Banerjee ABANERJE@ECE.UTEXAS.EDU
Department of Electrical and Computer Engineering, University of Texas, Austin, TX 78712

Raymond Mooney MOONEY@CS.UTEXAS.EDU
Department of Computer Sciences, University of Texas, Austin, TX 78712

Abstract

Semi-supervised clustering uses a small amount of labeled data to aid and bias the clustering of unlabeled data. This paper explores the use of labeled data to generate initial seed clusters, as well as the use of constraints generated from labeled data to guide the clustering process. It introduces two semi-supervised variants of KMeans clustering that can be viewed as instances of the EM algorithm, where labeled data provides prior information about the conditional distributions of hidden category labels. Experimental results demonstrate the advantages of these methods over standard random seeding and COP-KMeans, a previously developed semi-supervised clustering algorithm.

1. Introduction

In many learning tasks, there is a large supply of unlabeled data but insufficient labeled data since it can be expensive to generate. Semi-supervised learning combines labeled and unlabeled data during training to improve performance. Semi-supervised learning is applicable to both classification and clustering. In supervised classification, there is a known, fixed set of categories and category-labeled training data is used to induce a classification function. In *semi-supervised classification*, training also exploits additional unlabeled data, frequently resulting in a more accurate classification function (Blum & Mitchell, 1998; Ghahramani & Jordan, 1994). In unsupervised clustering, an unlabeled dataset is partitioned into groups of similar examples, typically by optimizing an objective function that characterizes good partitions. In *semi-supervised clustering*, some labeled data is used along with the unlabeled data to obtain a better clustering. This paper explores the use of labeled data to generate seed clusters that initialize a clustering algorithm, as well as the use of constraints generated from the labeled data to guide the clustering process. Proper seeding biases clustering towards a good region of the search space, thereby reducing the chances of it getting stuck in poor local optima, while simultaneously producing a clustering similar to the user-specified labels.

If the initial labeled data represent all the relevant categories, then both semi-supervised clustering and semi-supervised classification algorithms can be used for categorization. However in many domains, knowledge of the relevant categories is incomplete. Unlike semi-supervised classification, semi-supervised clustering can group data using the categories in the initial labeled data, as well as extend and modify the existing set of categories as needed to reflect other regularities in the data.

This paper introduces two semi-supervised variants of KMeans clustering (MacQueen, 1967) that use initial labeled data for seeding. We motivate the algorithms using the Expectation Maximization (EM) framework (Dempster et al., 1977), showing that seeding can be explained using the conditional distribution of hidden category labels. We present results of experiments demonstrating the advantages of our methods over standard random seeding and COP-KMeans (Wagstaff et al., 2001), an alternative semi-supervised KMeans algorithm.

2. Background

KMeans is a clustering algorithm based on iterative relocation that partitions a dataset into K clusters, locally minimizing the average squared distance between the data points and the cluster centers. For a set of data points $\mathcal{X} = \{x_1, \cdots, x_N\}, x_i \in \mathbb{R}^d$, the KMeans algorithm creates a K-partitioning [1] $\{\mathcal{X}_l\}_{l=1}^K$ of $\mathcal{X}$ so that if $\{\mu_1, \cdots, \mu_K\}$ represent the K partition centers, then the following objective function

$$\mathcal{J}_{\text{kmeans}} = \sum_{l=1}^{K} \sum_{x_i \in \mathcal{X}_l} \|x_i - \mu_l\|^2 \tag{1}$$

is locally minimized.

2.1 COP-KMeans algorithm

COP-KMeans (Wagstaff et al., 2001) is a semi-supervised variant of KMeans, where initial background knowledge, provided in the form of constraints between instances in the dataset, is used in the clustering process. There are two types of constraints, *must-link* (two instances have to be together in the same cluster) and *cannot-link* (two instances have to be in different clusters), which are used in the clustering process to generate a partition that satisfies all the given constraints. In this paper, we have developed two semi-supervised variants of KMeans and compared them to COP-KMeans.

2.2 SPKMeans algorithm

In the Spherical KMeans (SPKMeans) algorithm, standard KMeans is applied to data vectors that have been normalized to have unit L_2 norm, i.e., the data points lie on a unit sphere (Dhillon et al., 2001). Assuming $\|x_i\| = \|\mu_l\| = 1$, $\forall i, l$ in Eqn. 1, we get $\|x_i - \mu_l\|^2 = 2 - 2x_i^T \mu_l$. Then, the clustering problem can be equivalently formulated as that of maximizing the objective function:

$$\mathcal{J}_{\text{spkmeans}} = \sum_{l=1}^{K} \sum_{x_i \in \mathcal{X}_l} x_i^T \mu_l \tag{2}$$

The SPKMeans algorithm gives a local maximum of this objective function. The SPKMeans algorithm has computational advantages for sparse high dimensional data vectors, which are very common in domains like text clustering. For this reason, we have used SPKMeans in our experiments.

3. Algorithms

In this section, we explain how semi-supervision can be incorporated into the KMeans algorithm by *seeding* and propose two variants of the KMeans algorithm that use the seeds; then we give the mathematical motivation behind the two proposed algorithms.

Algorithm: `Seeded-KMeans`
Input: Set of data points $\mathcal{X} = \{x_1, \cdots, x_N\}, x_i \in \mathbb{R}^d$, number of clusters K, set $\mathcal{S} = \cup_{l=1}^K \mathcal{S}_l$ of initial seeds
Output: Disjoint K partitioning $\{\mathcal{X}_l\}_{l=1}^K$ of $\mathcal{X}$ such that KMeans objective function is optimized
Method:
1. `intialize`: $\mu_h^{(0)} \leftarrow \frac{1}{|\mathcal{S}_h|}\sum_{x \in \mathcal{S}_h} x$, for $h = 1, \ldots, K; t \leftarrow 0$
2. Repeat until *convergence*
2a. `assign_cluster`: Assign each data point x to the cluster h^* (i.e. set $\mathcal{X}_{h^*}^{(t+1)}$), for $h^* = \arg\min_h \|x - \mu_h^{(t)}\|^2$
2b. `estimate_means`: $\mu_h^{(t+1)} \leftarrow \frac{1}{|\mathcal{X}_h^{(t+1)}|}\sum_{x \in \mathcal{X}_h^{(t+1)}} x$
2c. $t \leftarrow (t+1)$

Figure 1. Seeded-KMeans algorithm

Algorithm: `Constrained-KMeans`
Input: Set of data points $\mathcal{X} = \{x_1, \cdots, x_N\}, x_i \in \mathbb{R}^d$, number of clusters K, set $\mathcal{S} = \cup_{l=1}^K \mathcal{S}_l$ of initial seeds
Output: Disjoint K partitioning $\{\mathcal{X}_l\}_{l=1}^K$ of $\mathcal{X}$ such that the KMeans objective function is optimized
Method:
1. `intialize`: $\mu_h^{(0)} \leftarrow \frac{1}{|\mathcal{S}_h|}\sum_{x \in \mathcal{S}_h} x$, for $h = 1, \ldots, K; t \leftarrow 0$
2. Repeat until *convergence*
2a. `assign_cluster`: For $x \in \mathcal{S}$, if $x \in \mathcal{S}_h$ assign x to the cluster h (i.e., set $\mathcal{X}_h^{(t+1)}$). For $x \notin \mathcal{S}$, assign x to the cluster h^* (i.e. set $\mathcal{X}_{h^*}^{(t+1)}$), for $h^* = \arg\min_h \|x - \mu_h^{(t)}\|^2$
2b. `estimate_means`: $\mu_h^{(t+1)} \leftarrow \frac{1}{|\mathcal{X}_h^{(t+1)}|}\sum_{x \in \mathcal{X}_h^{(t+1)}} x$
2c. $t \leftarrow (t+1)$

Figure 2. Constrained-KMeans algorithm

3.1 Seeding

Given a dataset $\mathcal{X}$, as previously mentioned, KMeans clustering of the dataset generates a K-partitioning $\{\mathcal{X}_l\}_{l=1}^K$ of $\mathcal{X}$ so that the KMeans objective is locally minimized. Let $\mathcal{S} \subseteq \mathcal{X}$, called the *seed set*, be the subset of data-points on which supervision is provided as follows: for each $x_i \in \mathcal{S}$, the user provides the cluster $\mathcal{X}_l$ of the partition to which it belongs. We assume that corresponding to each partition $\mathcal{X}_l$ of $\mathcal{X}$, there is typically atleast one seedpoint $x_i \in \mathcal{S}$. Note that we get a disjoint K-partitioning $\{\mathcal{S}_l\}_{l=1}^K$ of the seed set $\mathcal{S}$, so that all $x_i \in \mathcal{S}_l$ belongs to $\mathcal{X}_l$ according to the supervision. This partitioning of the seed set $\mathcal{S}$ forms the *seed clustering* and is used to guide the KMeans algorithm.

3.2 Two Semi-supervised KMeans Algorithms

In *Seeded-KMeans*, the seed clustering is used to initialize the KMeans algorithm. Thus, rather than initializing KMeans from K random means, the mean of the lth cluster is initialized with the mean of the lth partition $\mathcal{S}_l$ of the seed set. The seed clustering is only

[1] K disjoint subsets of $\mathcal{X}$, whose union is $\mathcal{X}$

used for initialization, and the seeds are not used in the following steps of the algorithm. The algorithm is presented in detail in Fig. 1.

In *Constrained-KMeans*, the seed clustering is used to initialize the KMeans algorithm as described for the Seeded-KMeans algorithm. However, in the subsequent steps, the cluster memberships of the data points in the seed set are not re-computed in the `assign_cluster` steps of the algorithm – the cluster labels of the seed data are kept unchanged, and only the labels of the non-seed data are re-estimated. The algorithm is given in detail in Fig. 2.

Constrained-KMeans seeds the KMeans algorithm with the user-specified labeled data and keeps that labeling unchanged throughout the algorithm. In Seeded-KMeans, the user-specified labeling of the seed data may be changed in the course of the algorithm. Constrained-KMeans is appropriate when the initial seed labeling is noise-free, or if the user does not want the labels on the seed data to change, whereas Seeded-KMeans is appropriate in the presence of noisy seeds. This and other aspects of these two algorithms are studied in detail through experiments in Sec. 4.

3.3 Semi-supervised KMeans as EM

The EM algorithm is a very general method of finding the maximum-likelihood estimate of the parameters of an underlying distribution, or, more generally, a probabilistic data generation process, from a set of observed data that has incomplete or missing values. If $\mathcal{X}$ denotes the observed data, Θ denotes the current estimate of the parameter values and $\mathcal{Z}$ denotes the missing data, then, in the E-step, the EM algorithm computes the expected value of the complete-data log-likelihood $\log p(\mathcal{X}, \mathcal{Z}|\Theta)$ over the distribution $p(\mathcal{Z}|\mathcal{X}, \Theta)$ (Bilmes, 1997). As we shall demonstrate, the semi-supervision provided to the KMeans algorithm essentially determines this conditional distribution over which the expectation is computed. We shall take a closer look at the assumptions one makes on this distribution in the EM framework for solving the KMeans problem so that the effect of semi-supervision will become evident.

The KMeans clustering algorithm is essentially an EM algorithm on a mixture of K Gaussians under certain assumptions. The data-generation process in KMeans is assumed to be as follows – first, one Gaussian is chosen out of the K following their prior probability distribution; then, a data-point is sampled following the distribution of the chosen Gaussian. Let $\mathcal{X} = \{x_1, \cdots, x_N\}$ be the set of data-points we want to cluster with each $x_i \in \mathbb{R}^d$. The missing data $\mathcal{Z}$ is the cluster assignment of the data-points. It takes values in $\{1, \cdots, K\}$ and is always conditioned on the data-point under consideration. We denote $(\mathcal{Z} = l)$ by z_l. For deriving KMeans, we assume that the prior distribution π of the Gaussians is uniform, i.e., $\pi_l = 1/K, \forall l$, and that each Gaussian has identity covariance. Then, the parameter set Θ consists of just the K means $\mu_1, \cdots, \mu_K$. With these assumptions, one can show that (Bilmes, 1997):

$$\begin{aligned}
&\mathbf{E}_{\mathcal{Z}|\mathcal{X},\Theta}[\log p(\mathcal{X}, \mathcal{Z}|\Theta)] \\
&= \sum_{l=1}^{K}\sum_{i=1}^{N} \log(\pi_l \cdot \frac{1}{(2\pi)^{d/2}} e^{-\|x_i-\mu_l\|^2})\, p(z_l|x_i, \Theta) \\
&= -\sum_{l=1}^{K}\sum_{i=1}^{N} \|x_i - \mu_l\|^2\, p(z_l|x_i, \Theta) + c \qquad (3)
\end{aligned}$$

where c is a constant. Further assuming that

$$p(z_l|x_i, \Theta) = \begin{cases} 1 & \text{if } l = \arg\min_h \|x_i - \mu_h\|^2, \\ 0 & \text{otherwise,} \end{cases} \qquad (4)$$

and replacing it in Eqn. 3, we note that the expectation term comes out to be the negative of the well-known KMeans objective function with an additive constant.[2] Thus, the problem of maximizing the expectation of the complete-data log-likelihood under these assumptions is same as that of minimizing the KMeans objective function. Keeping in mind the assumption in Eqn. 4, the KMeans objective can be written as

$$\mathcal{J}_{\text{kmeans}} = \sum_{l=1}^{K}\sum_{i=1}^{N} \|x_i - \mu_l\|^2\, p(z_l|x_i, \mu_l) \qquad (5)$$

The only "missing data" for the KMeans problem are the conditional distributions $p(z_l|x_i, \mu_l)$. Knowledge of these distributions solves the problem, but normally there is no way to compute it. In the semi-supervised clustering framework, the user provides information about some of the data points that specifies the corresponding conditional distributions.

Example: If x_i and x_j are two data-points with a *must-link* constraint between them (Sec. 2.1), then $p(z_l|x_i, \mu_l)$ and $p(z_l|x_j, \mu_l)$ are identically distributed. In fact, all data-points in the transitive closure of a connected set of *must-link* constraints will be identically distributed.

Thus, semi-supervision essentially provides information about the conditional distributions $p(z_l|x_i, \mu_l)$.

[2] The assumption in Eqn. 4 can also be derived by assuming the covariance of the Gaussians to be $\epsilon\mathbb{I}$ and letting $\epsilon \to 0^+$ (Kearns et al., 1997).

In standard KMeans without any initial supervision, the K means are chosen randomly in the initial M-step and the data-points are assigned to the nearest means in the subsequent E-step. As explained above, every point x_i in the dataset has K possible conditional distributions associated with it (each satisfying Eqn. 4) corresponding to the K means to which it can belong. This assignment of data point x_i to a random cluster in the first E-step is similar to picking one conditional distribution at random from the K possible conditional distributions.

In Seeded-KMeans, the initial supervision is equivalent to specifying the conditional distributions $p(z_l|x_i, \mu_l)$ for the seed points $x_i \in \mathcal{S}$. The specified conditional distributions of the seed data are just used in the initial M-step of the algorithm, and $p(z_l|x_i, \mu_l)$ is re-estimated for all $x_i \in \mathcal{X}$ in the following E-steps of the algorithm.

In Constrained-KMeans, the initial M-step is same as Seeded-KMeans. The difference is that for the seed data points, the initial labels, i.e., the conditional distributions $p(z_l|x_i, \mu_l)$, are kept unchanged throughout the algorithm, whereas the conditional distribution for the non-seed points are re-estimated at every E-step.

In our experiments, we will be using the SPKMeans framework (Sec. 2.2). In this framework, since every point lies on the unit sphere so that $\|x_i\| = \|\mu_l\| = 1$, the expectation term in Eqn. 3 becomes equivalent to

$$\mathbf{E}_{\mathcal{Z}|\mathcal{X},\Theta}[\log p(\mathcal{X}, \mathcal{Z}|\Theta)] = \sum_{l=1}^{K} \sum_{i=1}^{N} x_i^T \mu_l \, p(z_l|x_i, \Theta) + c$$

So, maximizing the SPKMeans objective function is equivalent to maximizing the expectation of the complete-data log-likelihood in the E-step of the EM algorithm.

4. Experiments

In our experiments, we used 2 data sets – CMU 20 Newsgroups data and Yahoo! News data. For each dataset, we ran 4 algorithms – Seeded-KMeans, Constrained-KMeans, COP-KMeans, and Random-KMeans. In Random-KMeans, the K means were initialized by taking the mean of the entire data and randomly perturbing it K times (Fayyad et al., 1998). This technique of initialization has given good results in unsupervised KMeans in previous work (Dhillon et al., 2001). We compared the performance of these methods on the 2 datasets with varying seeding and noise levels, using 10-fold cross validation. For each dataset, SPKMeans was used as the underlying KMeans algorithm for all the 4 KMeans variants.

4.1 Datasets

The 20 Newsgroups dataset (`20 Newsgroups`) is a collection of 20,000 messages, collected from 20 different Usenet newsgroups – 1000 messages from each of the 20 newsgroups were chosen, and the dataset was partitioned by newsgroup name.[3] In our experiments, we used the MC toolkit[4] for creating the vector space model for text documents. For the `20 Newsgroups` dataset, MC generated a vocabulary of 21,631 words – each message is represented as a (sparse) vector in a 21,631 dimensional space, with TFIDF weighting. The Yahoo! News K-series (`Yahoo! News`) dataset[5] is a collection of 2340 Yahoo! news articles belonging to one of 20 different Yahoo! categories. The vector space model of the K1 set from the Yahoo! K-series has 12,229 words – the data-points reside in a 12,229 dimensional space and are TFIDF weighted. For the text datasets, "non-content-bearing" stop-words, high-frequency words and low-frequency words were removed, following the methodology of Dhillon et al. (2001).

From the original `20 Newsgroups` dataset, some other datasets were generated: (1) `Small-20 Newsgroups` – contains a random subsample of 100 documents from each of the 20 newsgroups (2) `Different-3 Newsgroups` – selects 3 very different newsgroups from the original `20 Newsgroups` dataset (alt.atheism, rec.sport.baseball, sci.space) (3) `Same-3 Newsgroups` – selects 3 very similar newsgroups from the original `20 Newsgroups` dataset (comp.graphics, comp.os.ms-windows, comp.windows.x). The dataset `Small-20 Newsgroups` was created to study the effect of dataset size on the performance of the algorithms. `Different-3 Newsgroups` and `Same-3 Newsgroups` were generated to study the effect of data separability on the algorithms.

4.2 Evaluation Measures

We have used two evaluation measures in our experiments. One is the objective function of KMeans – for SPKMeans, the higher the objective function, the better is the performance. This measure does not take into account the user-labeling of the data. The other measure is *mutual information* (MI), which determines the amount of statistical information shared by the random variables representing the cluster and the (user-labeled) class assignments of the data points. In this work, MI is computed following the methodology of Strehl et al. (2000).

[3] http://www.ai.mit.edu/people/jrennie/20_newsgroups
[4] http://www.cs.utexas.edu/users/jfan/dm
[5] ftp://ftp.cs.umn.edu/users/boley/PDDPdata

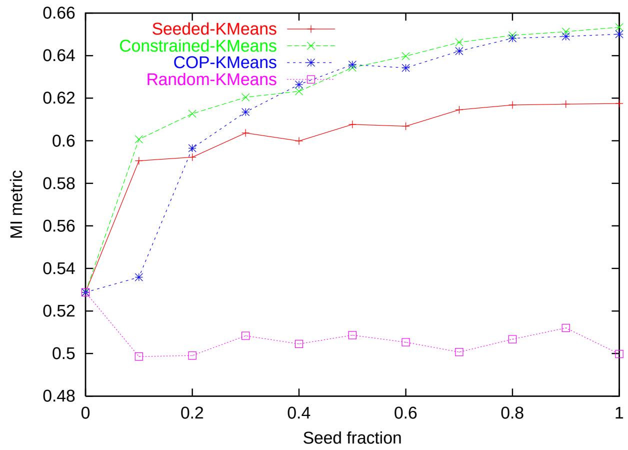

Figure 3. Comparison of MI of algorithms on `20 Newsgroups` data, noise fraction = 0

4.3 Learning curves with cross validation

For all the algorithms, on each dataset, we have generated learning curves with 10-fold cross-validation. For studying the effect of seeding, 10% of the dataset is set aside as the test set at any particular fold. The training sets at different points of the learning curve are obtained from the remaining 90% of the data by varying the seed fraction from 0.0 to 1.0 in steps of 0.1, and the results at each point on the learning curve are obtained by averaging over 10 folds. The clustering algorithm is run on the whole dataset, but the MI measure is calculated only on the test set. For studying the effects of noise in the seeding, similar learning curves are generated by keeping a fixed fraction of seeding and varying the noise fraction.

4.4 Seed and Noise generation

In Seeded-KMeans and Constrained-KMeans, the seeds at any point on the learning curve were selected from the dataset according to the corresponding seed fraction. In COP-KMeans, the *must-link* and the *cannot-link* constraints are generated from the specified seeds. The K cluster centers are chosen randomly, but as each one is chosen, any *must-link* constraints that it participates in are enforced, i.e., all items that the chosen instance must link to are assigned to the new cluster, so that they cannot later be chosen as the center of another cluster (Wagstaff et al., 2001).

In a real-life application, since the semi-supervision will be provided by a human user, there is a chance that the supervision may be erroneous in some cases. We simulate such labeling noise in our experiments by changing the labels of a fraction of the seed examples to a random incorrect value.

5. Analysis of Results

MI with respect to seeding: For the zero-noise case, the semi-supervised algorithms perform better than the unsupervised algorithm in terms of the MI measure (Figs. [3,4,5]), irrespective of the size of the dataset. Constrained-KMeans performs at least as good as the Seeded-KMeans, since the former uses the correct user bias introduced by the user-labeled seeds throughout the execution of the algorithm in the zero-noise case. Though both Constrained-KMeans and COP-KMeans treat the seeds as constraints, the fact that Constrained-KMeans uses all the seeds to initialize clusters, as opposed to COP-KMeans which does not necessarily do that, results in the former having better performance in most cases, with zero-noise. In fact, the effect of seeding seems to be so important that in some cases (Fig. 4), Seeded-KMeans performs significantly better than COP-KMeans.

Objective function with respect to seeding: Though the MI measure increases with an increase in seed fraction for the semi-supervised algorithms, the behavior of the objective function will depend on whether the user bias provided by the user-labeled seeds is consistent with the assumptions of KMeans. If the category structure created by the user-labeling of the dataset satisfies the KMeans assumptions, then the data partition induced by seeding will be close to the optimal partition, and KMeans is known to converge to a good local optimum in this case (Fig. 6) (Devroye et al., 1996). On the other hand, if the user bias is inconsistent with the KMeans assumptions, then constrained seeding will result in convergence to a sub-optimal solution (Fig. 7). Note that since Seeded-KMeans does not necessarily maintain the same assignments for the seed points in subsequent iterations, its objective function does not decrease due to conflict in bias; however, since Constrained-KMeans and COP-KMeans keep the seeds as constraints, their objective function decreases with increase in seeding. Since Random-KMeans never uses the seeds, its behavior is independent of this conflict.

Dataset separability: Semi-supervision gives substantial improvement over unsupervised clustering for datasets that are difficult to cluster, in the sense that there is a lot of overlap between the clusters, e.g., `Same-3 Newsgroups`, (Fig. 8), whereas for datasets that are easily separable, e.g., `Different-3 Newsgroups` (Fig. 9), the improvement over Random-KMeans is marginal. If the dataset is easily separable, then there are not many bad local minima and even Random-KMeans can easily find the cluster structure. However, for datasets with overlapping cluster struc-

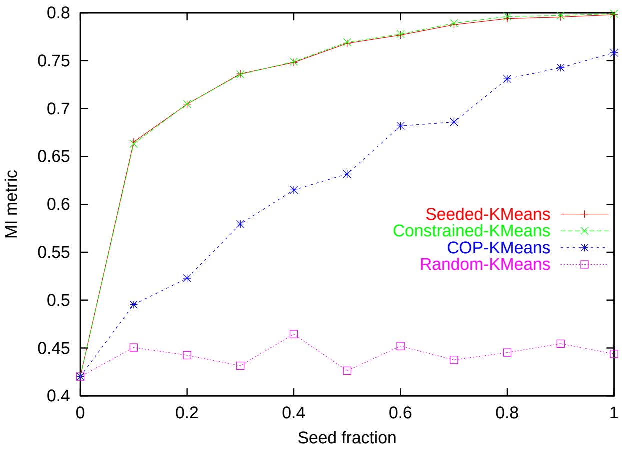

Figure 4. Comparison of MI of algorithms on `Small-20 Newsgroups` data, noise fraction = 0

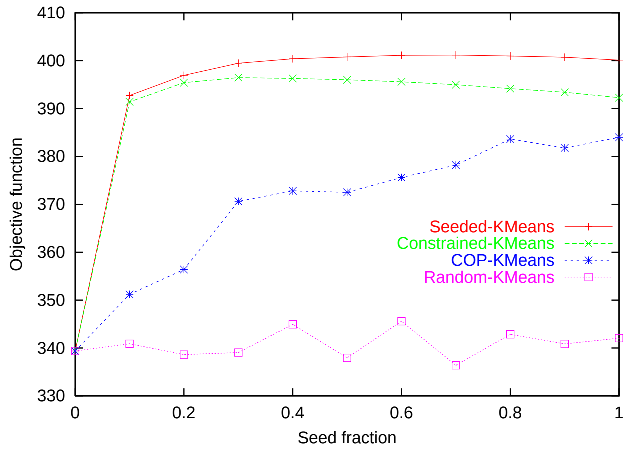

Figure 6. Comparison of objective functions of algorithms on `Small-20 Newsgroups` data, noise fraction = 0

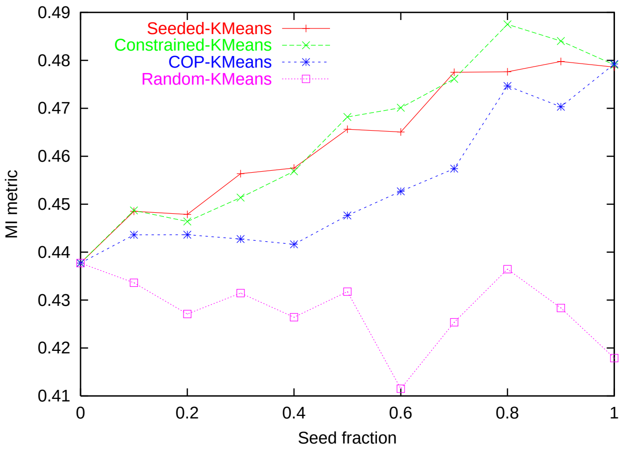

Figure 5. Comparison of MI of algorithms on `Yahoo!` data, noise fraction = 0

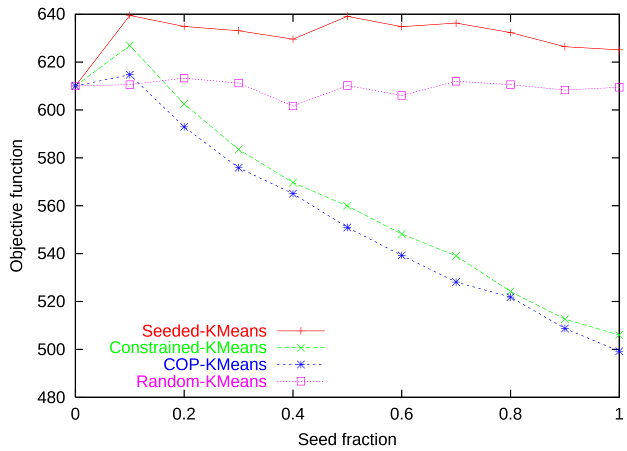

Figure 7. Comparison of objective functions of algorithms on `Yahoo!` data, noise fraction = 0

ture, seeding seems to be an important factor in helping the algorithm find a good clustering. The MI measure for the separable dataset is in general much higher than for the overlapping dataset even with high seeding, because the latter one is a harder problem to solve.

Performance with incomplete seeding: We also ran initial experiments with *incomplete* seeding, where seeds are not specified for every cluster – from Fig. 10, it can be seen that the MI metric did not decrease substantially with increase in the number of unseeded categories, showing that the semi-supervised clustering algorithms could extend the seed clusters and generate more clusters, in order to fit the regularity of the data.

Performance with respect to noise: As noise is increased, the performance of Constrained-KMeans and COP-KMeans starts to degrade compared to Seeded-KMeans. COP-KMeans and Constrained-KMeans keep using the same noisy seeds in every subsequent iteration of the algorithm, whereas Seeded-KMeans can abandon noisy seed labels in subsequent iterations (Fig. 11). So Seeded-KMeans is quite robust against noisy seeding, and can take full advantage of the seeding if it gives the algorithm a good initialization.

The statistical significance of the conclusions in this section have been tested across various datasets. For example, on the `Small-20 Newsgroup` dataset, the conclusions are significant for seed fraction $>= 0.2$ ($p < 0.001$) for the first three aspects discussed above, using two-tailed paired t-test. For the noise experiments, the conclusion is significant for noise fraction < 0.5 ($p < 0.001$).

6. Related Work

Several semi-supervised classification algorithms have shown improvements in classification accuracy over

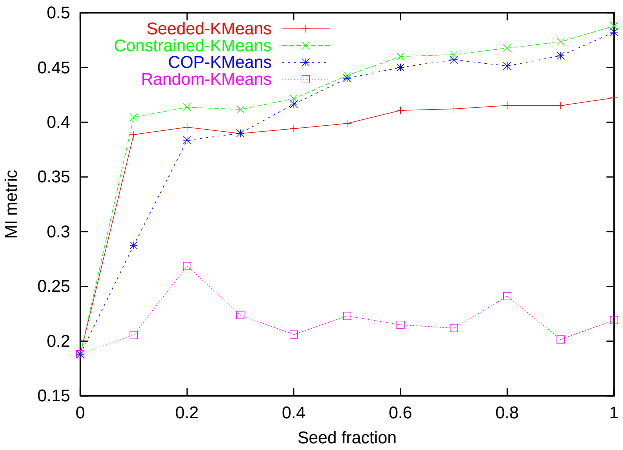

Figure 8. Comparison of MI of algorithms on `Same-3 Newsgroups` data, noise fraction = 0

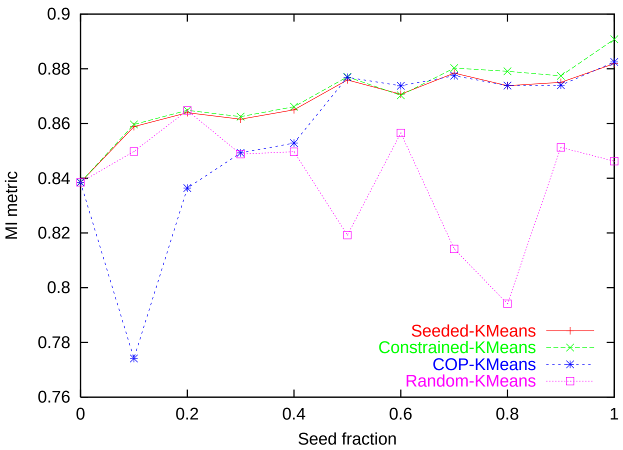

Figure 9. Comparison of MI of algorithms on `Different-3 Newsgroups` data, noise fraction = 0

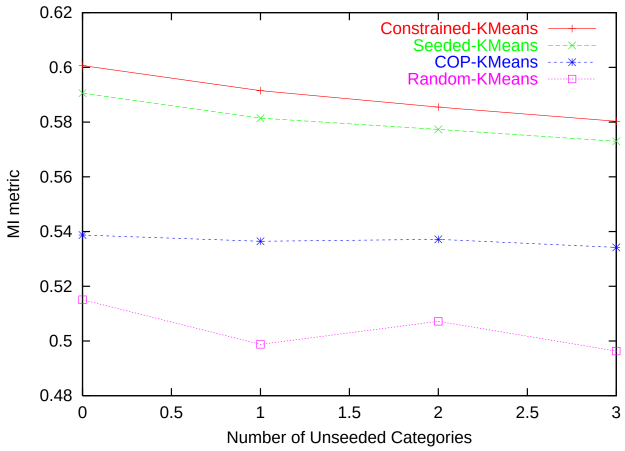

Figure 10. Comparison of MI of algorithms on `20 Newsgroups` data, seed fraction = 0.1

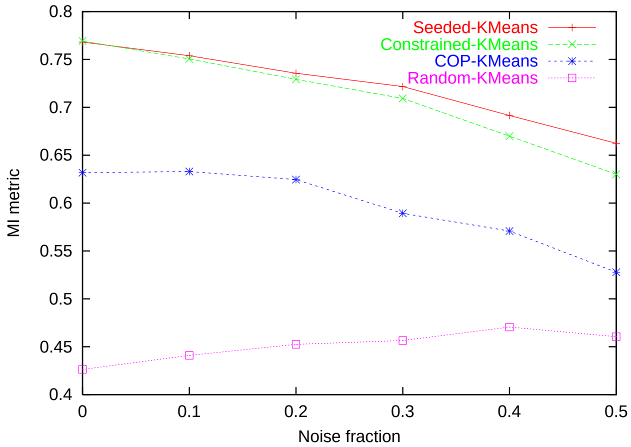

Figure 11. Comparison of MI of algorithms on `Small-20 Newsgroups` data, seed fraction = 0.5

purely supervised algorithms, e.g. co-training (Blum & Mitchell, 1998), transductive Support Vector Machines (SVMs) (Joachims, 1999), and semi-supervised EM (Ghahramani & Jordan, 1994; Nigam et al., 2000).

In semi-supervised clustering, previous work has been done on the use of labeled data to aid clustering by modifying clustering objective functions to incorporate labeled data (Demiriz et al., 1999), iterative user feedback (Cohn et al., 2000), and conditional distributions in an auxiliary space (Sinkkonen & Kaski, 2000). Previous work on cluster initialization includes comparisons of data-dependent and data-independent initialization techniques (Meila & Heckerman, 1998), and estimation of the modes of the data distribution for good initialization (Fayyad et al., 1998).

7. Future Work

The connection with the general EM framework and the interpretation of semi-supervision in terms of conditional distributions widens the applicability of the proposed methods to a variety of clustering problems. The most important of these is the concept of probabilistic or soft seeding – where semi-supervision gives the algorithm the probabilities of the seeds belonging to the various cluster labels, rather than explicitly stating which cluster it belongs to. In terms of the conditional distribution, we do not need the assumption in Eqn. 4 anymore, since the conditional distributions can now be any multinomial distribution defined over the K cluster labels. Semi-supervision by probabilistic seeding could be applicable to many learning tasks, such as volcano detection in planet-surface images (Smyth et al., 1994).

8. Conclusion

Semi-supervised clustering uses some labeled data to aid search and bias the partitioning of unlabeled data into conceptual groups. Seeded-KMeans and Constrained-KMeans are semi-supervised clustering algorithms that use labeled data to form initial clusters and constrain subsequent cluster assignment. Both methods can be viewed as instances of the EM algorithm, where labeled data provides prior information about the conditional distributions of hidden category labels. Experimental results demonstrate the advantages of these methods over standard random seeding and COP-KMeans (Wagstaff et al., 2001), an alternative semi-supervised KMeans algorithm. In particular, seeding without constraints is a robust semi-supervised method that is less sensitive to noise and imperfections in the supervised data.

9. Acknowledgment

This research was supported by NSF grants IIS-0117308 and ECS-9900353, and by an MCD Fellowship awarded by the University of Texas at Austin.

References

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

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *Proc. of 11th Annual Conf. on Computational Learning Theory.*

Cohn, D., Caruana, R., & McCallum, A. (2000). Semi-supervised clustering with user feedback. Unpublished manuscript. Available at `http://www-2.cs.cmu.edu/~mccallum/`.

Demiriz, A., Bennett, K., & Embrechts, M. (1999). Semi-supervised clustering using genetic algorithms. *Artificial Neural Networks in Engineering (ANNIE).*

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, *39*, 1–38.

Devroye, L., Gyorfi, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition.* Springer Verlag.

Dhillon, I. S., Fan, J., & Guan, Y. (2001). Efficient clustering of very large document collections. In *Data mining for scientific and engineering applications.* Kluwer Academic Publishers.

Fayyad, U. M., Reina, C., & Bradley, P. S. (1998). Initialization of iterative refinement clustering algorithms. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)* (pp. 194–198).

Ghahramani, Z., & Jordan, M. I. (1994). Supervised learning from incomplete data via the EM approach. *Advances in Neural Information Processing Systems 6* (pp. 120–127).

Joachims, T. (1999). Transductive inference for text classification using support vector machines. *Proc. of 16th Intl. Conf. on Machine Learning (ICML-99).*

Kearns, M., Mansour, Y., & Ng, A. Y. (1997). An information-theoretic analysis of hard and soft assignment methods for clustering. *Proc. of 13th Conf. on Uncertainty in Artificial Intelligence (UAI-97)* (pp. 282–293).

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297).

Meila, M., & Heckerman, D. (1998). *An experimental comparison of several clustering and initialization methods* (Technical Report MSR-TR-98-06). Microsoft Research.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, *39*, 103–134.

Sinkkonen, J., & Kaski, S. (2000). *Semisupervised clustering based on conditional distributions in an auxiliary space* (Technical Report A60). Helsinki University of Technology.

Smyth, P., Fayyad, U., Burl, M., & Perona, P. (1994). Inferring ground truth from subjective labelling of Venus images. *Advances in Neural Information Processing Systems 6.*

Strehl, A., Ghosh, J., & Mooney, R. (2000). Impact of similarity measures on web-page clustering. *Workshop on Artificial Intelligence for Web Search (AAAI 2000)* (pp. 58–64).

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