Clustering Large and Sparse Co-occurrence Data

Inderjit S. Dhillon and Yuqiang Guan
Department of Computer Sciences
University of Texas, Austin TX 78712

February 14, 2003

Abstract

A novel approach to clustering co-occurrence data poses it as an optimization problem in information theory — in this framework, an optimal clustering is one which minimizes the loss in mutual information. Recently a divisive clustering algorithm was proposed that monotonically reduces this loss function. In this paper we show that sparse high-dimensional data presents special challenges which can result in the algorithm getting stuck at poor local minima. We propose two solutions to this problem: (a) a prior to overcome infinite relative entropy values as in the supervised Naive Bayes algorithm, and (b) local search to escape local minima. Finally, we combine these solutions to get a powerful algorithm that is computationally efficient. We present detailed experimental results to show that the proposed method is highly effective in clustering document collections and outperforms previous approaches.

1 Introduction

Clustering is a central problem in unsupervised learning [7]. Presented with a set of data points, clustering algorithms group the data into clusters according to some notion of similarity between data points. However, the choice of similarity measure is a challenge and often an *ad hoc* measure is chosen. Information Theory comes to the rescue in the important situations where non-negative co-occurrence data is available. A novel formulation poses the clustering problem as one in information theory: find the clustering that minimizes the loss in (mutual) information [16, 6]. This information-theoretic formulation leads to a "natural" divisive clustering algorithm that uses relative entropy as the measure of similarity and monotonically reduces the loss in mutual information [6].

However, sparse and high-dimensional data presents special challenges and can lead to qualitatively poor local minima. In this paper, we demonstrate these failures and then propose two solutions to overcome these problems. First, we use a prior as in the supervised Naive Bayes algorithm to overcome infinite relative entropy values caused by sparsity. Second, we propose a local search strategy that is highly effective for high-dimensional data. We combine these solutions to get an effective, computationally efficient algorithm. A prime example of high-dimensional co-occurrence data is word-document data; we show that our algorithm returns clusterings that are better than those returned by previously proposed information-theoretic approaches.

The following is a brief outline of the paper. Section 2 discusses related work while Section 3 presents the information-theoretic framework and divisive clustering algorithm of [6]. The problems due to sparsity and high-dimensionality are illustrated in Section 4. We present our two-pronged solution to the problem in Section 5 after drawing an analogy to the supervised Naive Bayes algorithm in Section 5.1. Detailed experimental results are given in Section 6. Finally we present our conclusions and ideas for future work in Section 7.

2 Related work

Clustering is a widely studied problem in unsupervised learning, and a good survey of existing methods can be found in [7]. For the case of co-occurrence data, our information-theoretic framework is similar to the one used in the Information Bottleneck method [18], which yields a "soft" clustering of the data. An agglomerative hard clustering version of the method is given in [17] while methods based on sequential optimization are used in [2, 16]. As we demonstrate in Section 6, our proposed algorithm yields better clusterings than the above approaches, while being more computationally efficient.

Information-theoretic methods have been used for a variety of tasks in machine learning [14] including text classification [15]. Distributional clustering of words was first proposed in [13] and subsequently used by [1] for reducing the feature size for text classifiers. A general statistical framework for analyzing co-occurrence data based on probabilistic clustering by mixture models was given in [8].

3 Divisive Information-Theoretic Clustering

Let X and Y be two discrete random variables that take values in the sets $\{x_1, \ldots, x_m\}$ and $\{y_1, \ldots, y_n\}$ respectively. Suppose that we know their joint probability distribution $p(X, Y)$; often this can be estimated using co-occurrence data. A useful measure of the information that X contains about Y (and vice versa) is given by their mutual information, $I(X; Y)$ [4].

Suppose that we want to cluster Y. Let $\hat{Y}$ denote the random variable that ranges over the disjoint clusters $\hat{y}_1, \ldots, \hat{y}_k$, i.e.,

$$\cup_{i=1}^{k} \hat{y}_i = \{y_1, \ldots, y_n\}, \text{ and } \hat{y}_i \cap \hat{y}_j = \phi, \quad i \neq j.$$

A novel information-theoretic approach to clustering is to seek that clustering which results in the smallest loss in mutual information [16, 6]. This loss can be quantified by the following theorem.

Theorem 1 *[6] The loss in mutual information due to clustering Y is given by*

$$I(X; Y) - I(X; \hat{Y}) = \sum_{j=1}^{k} p(\hat{y}_j) JS_{p'}(\{p(X|y_i) : y_i \in \hat{y}_j\})$$

where $p(\hat{y}_j) = \sum_{y_i \in \hat{y}_j} p(y_i)$, $p'(y_i) = p(y_i)/p(\hat{y}_j)$ for $y_i \in \hat{y}_j$, and JS denotes the generalized Jensen-Shannon divergence as defined in [11].

The generalized Jensen-Shannon(JS) divergence used above is a measure of the "distance" between probability distributions. The smaller the JS-divergence the more "cohesive" or "compact" is the cluster. The above theorem implies that the loss in mutual information equals a weighted sum of within-cluster JS-divergences. A proof of Theorem 1 is given in [6]. By properties of the JS-divergence, the loss in mutual information may be re-written as $I(X; Y) - I(X; \hat{Y}) = \sum_{j=1}^{k} \sum_{y_i \in \hat{y}_j} p(y_i) KL(p(X|y_i), p(X|\hat{y}_j))$ and so KL, which denotes the relative entropy or Kullback-Leibler divergence [4], emerges as the "natural" distortion measure.

The divisive clustering algorithm of Figure 1 provides a way to obtain a sequence of clusterings that monotonically reduces the loss in mutual information given by Theorem 1. The algorithm iteratively (i) re-partitions the distributions $p(X|y_i)$ by their closeness in KL-divergence to the cluster distributions $p(X|\hat{y}_j)$, and (ii) subsequently, given the new clusters, re-computes the optimal cluster distributions. This algorithm is general in that it can be applied to any co-occurrence table. In [6] the algorithm was shown to converge. It was applied to feature clustering to reduce the model size of the resulting classifiers and was found to outperform previously proposed agglomerative strategies [1]. The word-class co-occurrence data tackled here was not sparse.

4 Challenges due to Sparsity and High-Dimensionality

We now demonstrate how the divisive information-theoretic clustering algorithm can falter due to sparsity and high-dimensionality.

Algorithm: DITC $(p(X,Y), k, \{\hat{y}_j\}_{j=1}^k)$
Input: $p(X,Y)$ is the empirical joint probability distribution, k is the number of desired clusters.
Output: The set of clusters of $\hat{y}$, $\{\hat{y}_j^{(\tau)}\}_{j=1}^k$.
Method:
1. **Initialization:** $\tau \leftarrow 0$. Choose $\{p(X|\hat{y}_j^{(\tau)})\}_{j=1}^k$ to be the conditional distributions $p(X|y)$'s that are "maximally" far apart.
2. Repeat until convergence
   2a.**assign_cluster:** for each y_i, find its new cluster index as $j^*(y_i) = \arg\min_j KL(p(X|y_i), p(X|\hat{y}_j^{(\tau)}))$,
   2b.**compute cluster distributions:** $p(\hat{y}_j^{(\tau)}) = \sum_{y_i \in \hat{y}_j^{(\tau)}} p(y_i)$, $p(X|\hat{y}_j^{(\tau)}) = \frac{1}{p(\hat{y}_j^{(\tau)})} \sum_{y_i \in \hat{y}_j^{(\tau)}} p(y_i) p(X|y_i)$.
   2c.$\tau \leftarrow \tau + 1$.

Figure 1: Divisive information-theoretic clustering

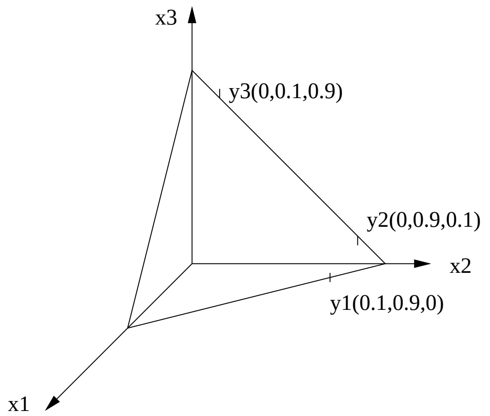

Figure 2: Probability simplex showing 3 distributions

Example 1 *(Sparsity) Consider the three conditional distributions:*

y_1	y_2	y_3
0.1	0	0
0.9	0.9	0.1
0	0.1	0.9

. These distributions may be thought of as points on the three-dimensional probability simplex as shown in Figure 2. Suppose we want to cluster y_1, y_2 and y_3 into two clusters; clearly the optimal clustering puts $\{y_1, y_2\}$ in one cluster and $\{y_3\}$ in the other. However suppose the initial clusters are $\hat{y}_1 = \{y_1\}$ and $\hat{y}_2 = \{y_2, y_3\}$. Then the cluster distributions will be $\hat{y}_1 = (0.1, 0.9, 0)$ and $\hat{y}_2 = (0, 0.5, 0.5)$, respectively (note that we are using $\hat{y}_i$ to denote a cluster as well as its distribution $p(X|\hat{y}_i)$—the particular usage should be clear from context). As shown in Table 1, the Kullback-Leibler divergences $KL(y_1, \hat{y}_2)$, $KL(y_2, \hat{y}_1)$ and $KL(y_3, \hat{y}_1)$ are infinite. Therefore Algorithm DITC of Figure 1 gets stuck in this initial clustering and the resulting percentage loss in mutual information equals 55.3% assuming $p(y_i) = \frac{1}{3}, 1 \le i \le 3$ (see Theorem 1). On the other hand, the percentage mutual information lost for the optimal partition, $\hat{y}_1^* = \{y_1, y_2\}$ and $\hat{y}_2^* = \{y_3\}$, equals 10.4%, Clearly, the DITC algorithm misses the optimal partition due to the presence of zeros in the cluster distributions that result in infinite KL-divergences.

Example 2 *(High dimensionality) For the second example, we took a collection of 30 documents consisting of 10 documents each from the three distinct* classes *MEDLINE, CISI and CRAN (see Section 6.1 for more details). These 30 documents contain a total of 1073 words and so the data is very high-dimensional. However, when we run DITC using the word-document co-occurrence data, there is hardly any movement of*

Table 1: The KL-divergences from y_1, y_2, y_3 to $\hat{y}_1, \hat{y}_2$

	$\hat{y}_1$	$\hat{y}_2$
y_1	0	∞
y_2	∞	0.531
y_3	∞	0.531

Table 2: Typical confusion matrix returned by *DITC* on a small document collection.

	MED	CRAN	CISI
$\hat{y}_1$	3	0	4
$\hat{y}_2$	7	1	4
$\hat{y}_3$	0	9	2

documents between clusters irrespective of the starting partition. The left confusion matrix (see Section 6.2 for definition) in Table 2 shows that a typical clustering returned by Algorithm DITC is quite poor. Note that this set of documents is easy to cluster since the documents come from three very different classes.

5 Proposed Algorithm

In this section, we propose a computationally efficient algorithm that avoids the above problems. To motivate our first solution, we draw a parallel between our unsupervised method and the supervised Naive Bayes algorithm.

5.1 The Naive Bayes Connection

Consider a supervised setting where $\hat{y}_1, ..., \hat{y}_k$ correspond to k given document classes and $x_1, ..., x_m$ correspond to the m words/features to be used for classification. The Naive Bayes classifier assumes class conditional word independence and computes the most probable class for test document d as

$$\text{argmax}_{\hat{y}_i} p(\hat{y}_j) \prod_{t=1}^{m} p(x_t|\hat{y}_j)^{N(x_t,d)} \tag{1}$$

where $N(x_t, d)$ is the number of occurrences of word x_t in document d [12]. Taking logarithms in (1), dividing throughout by the length of the document $|d|$ and adding the entropy $-\sum_{t=1}^{m} p(x_t|d) \log p(x_t|d)$ (where $p(x_t|d) = N(x_t, d)/|d|$), the Naive Bayes rule (1) is transformed to

$$\text{argmin}_{\hat{y}_j} KL(p(X|d), p(X|\hat{y}_j)) - \frac{p(\hat{y}_j)}{|d|}.$$

Note the similarity of this rule to Step 2*a* of Algorithm *DITC* (Figure 1). The *MLE* estimate of $p(x_t|\hat{y}_j)$ equals the fraction of occurrences of word x_t in class $\hat{y}_j$; however this implies that if x_t does not occur in $\hat{y}_j$ then $p(x_t|\hat{y}_j) = 0$. This is a well known problem with *MLE* estimates and can lead to infinite KL-divergence values resulting in incorrect classifications. To overcome this problem, it is common to modify the *MLE* estimate by introducing a "prior", for example, Laplace's rule of succession:

$$p'(x_t|\hat{y}_j) = \frac{1 + N(x_t, \hat{y}_j)}{m + \sum_{t=1}^{m} N(x_t, \hat{y}_j)}. \tag{2}$$

Algorithm: DITC_prior $(p(X,Y), k, \{\hat{y}_j\}_{j=1}^k, \alpha)$
Input: $p(X,Y)$ is the empirical joint probability distribution, k is the number of desired clusters, α is the prior.
Output: The set of clusters of $\hat{y}$, $\{\hat{y}_j^{(\tau)}\}_{j=1}^k$.
Method:
1. Initialization: $\tau \leftarrow 0$. Choose $\{p(X|\hat{y}_j^{(\tau)})\}_{j=1}^k$ to be the conditional distributions $p(X|y)$'s that are "maximally" far apart.
2. Repeat until convergence
   2a.assign_cluster: for each y_i, find its new cluster index as $j^*(y_i) = \arg\min_j KL(p(X|y_i), p(X|\hat{y}_j^{(\tau)}))$,
   2b.compute cluster distributions: $p(\hat{y}_j^{(\tau)}) = \sum_{y_i \in \hat{y}_j^{(\tau)}} p(y_i)$, $p(X|\hat{y}_j^{(\tau)}) = \frac{1}{1+\alpha}\left(\frac{1}{p(\hat{y}_j^{(\tau)})}\sum_{y_i \in \hat{y}_j^{(\tau)}} p(y_i)p(X|y_i) + \alpha u(X)\right)$
   2c.$\tau \leftarrow \tau + 1$, $\alpha \leftarrow \alpha/2$.

Figure 3: Adding priors to Algorithm *DITC*

5.2 Using a prior

As in the supervised Naive Bayes method, we wish to perturb $p(x_t|\hat{y}_j)$ to avoid zero probabilities. Recall that in our unsupervised case $p(X|\hat{y}_j)$ refers to a cluster distribution. The important question is: what should be the perturbation? For reasons that will become clearer later, we perturb the cluster distribution to:

$$p'(X|\hat{y}_j) = \frac{1}{1+\alpha}\left(p(X|\hat{y}_j) + \alpha \cdot u(X)\right), \tag{3}$$

where α is a constant and $u(X)$ is the uniform distribution $(\frac{1}{m}, ..., \frac{1}{m})$. The value of this prior has a pleasing property: the perturbed cluster distribution $p'(X|\hat{y}_j)$ can be interpreted as the mean distribution for $\hat{y}_j$ obtained after perturbing each element of the joint distribution $p(x,y)$ to $p'(x,y) = \frac{1}{1+\alpha}\left(p(x,y) + \frac{\alpha}{m}p(y)\right)$. This can be easily verified: a little algebra reveals that $p'(y) = \sum_x p'(x,y) = p(y)$ (thus the marginals $p(y)$ are unchanged) and $p'(X|\hat{y}_j) = \frac{1}{p'(\hat{y}_j)}\sum_{y_i \in \hat{y}_j} p'(y_i)p'(X|y_i) = \frac{1}{p(\hat{y}_j)}\sum_{y_i \in \hat{y}_j} \frac{p(y_i)}{1+\alpha}\left(p(X|y_i) + \alpha \cdot u(X)\right)$ is seen to give (3).

Another notable observation about our prior is that if α is taken to equal $\frac{m}{\sum_t N(w_t, \hat{y}_j)}$ in (3), then we get Laplace's rule of succession (2) that is used in the supervised Naive Bayes method. What should be the value of α in our clustering algorithm? Experimental results reveal that an "annealing" approach helps, i.e., start the algorithm with a large value of α and decrease α progressively as the number of iterations increase. Algorithm *DITC_prior* in Figure 3 summarizes our method. Note that our approach is different from the deterministic annealing approach of [14], which performs a "soft" clustering at every step and needs to identify cluster splits as a "temperature" parameter is decreased.

Algorithm *DITC* was shown to converge in [6] and so the convergence of Algorithm *DITC_prior* follows since $\alpha \to 0$ as the iteration count increases. This is another reason to halve α with every iteration. Back to our Example 1, with α set to 1, it can be shown that Algorithm *DITC_prior* is able to generate the optimal partition. Our use of priors also leads to better document clustering than Algorithm *DITC* (detailed results are in Section 6). However, problems due to high-dimensionality as in Example 2 are not completely cured.

5.3 Local Search

To further improve our algorithm, we now turn to a local search strategy that allows us to escape undesirable local minimum, especially in the case of high-dimensionality. Our local search principle, called *first variation* in [5], refines a given clustering by incrementally moving a distribution from one cluster to another in order to achieve a better objective function value. Precisely, a first variation of a partition $\{\hat{y}_j\}_{j=1}^k$ is a partition $\{\hat{y}'_j\}_{j=1}^k$ obtained by removing a distribution y from a cluster $\hat{y}_i$ of and assigning it to an existing cluster $\hat{y}_l$. Among all the kn possible first variations, corresponding to each combination of y and $\hat{y}_l$, we denote the one that gives the smallest loss in mutual information as `nextFV` $\left(\{\hat{y}_j\}_{j=1}^k\right)$. We now present the details

Algorithm: DITC_LocalSearch $(p(X,Y), k, \{\hat{y}_j\}_{j=1}^k, f)$
Input: $p(X,Y)$ is the empirical joint probability distribution, k is the number of desired clusters, f is first variation chain length.
Output: The set of clusters of $\hat{y}$, $\{\hat{y}_j^{(\tau)}\}_{j=1}^k$.
Method:
1. $\tau \leftarrow 0$.
2. Repeat until convergence.
   $l \leftarrow 0, \mathcal{U} \leftarrow \hat{y}$, do the following f times:
   2a. $\{\hat{y}_j^{(\tau+l+1)}\}_{j=1}^k \leftarrow$ `nextFV` $\left(\{\hat{y}_j^{(\tau+l)}\}_{j=1}^k, \mathcal{U}\right)$. Mark the moved distribution and delete it from $\mathcal{U}$.
   2b. Record the change of loss in mutual information in `ObjChange[l]`. $l \leftarrow l+1$.
   2c. `MaxChange` $\leftarrow \max_i \sum_{l=0}^i$ `ObjChange[l]` `MaxI` $\leftarrow \arg\max_i \sum_{l=0}^i$ `ObjChange[l]`, $i = 0, 1..., f-1$.

Figure 4: Algorithm *DITC* enhanced by local search

of performing this move. Suppose we move y from $\hat{y}_j$ to $\hat{y}_l$, then by Theorem 1 the change in the loss in mutual information equals

$$\begin{aligned}\delta &= p(\hat{y}_j^-)JS_{p'}(\{p(X|y_i) : y_i \in \hat{y}_j^-) - p(\hat{y}_j)JS_{p'}(\{p(X|y_i) : y_i \in \hat{y}_j) \\ &\quad + p(\hat{y}_l^+)JS_{p'}(\{p(X|y_i) : y_i \in \hat{y}_l^+) - p(\hat{y}_l)JS_{p'}(\{p(X|y_i) : y_i \in \hat{y}_l),\end{aligned}$$

where $\hat{y}_j^- = \hat{y}_j - \{y\}$ and $\hat{y}_l^+ = \hat{y}_l \cup \{y\}$. From [11], $JS_\pi(\{p_i : 1 \le i \le n\}) = H\left(\sum_i \pi_i p_i\right) - \sum_i \pi_i H(p_i)$, where $H(p_i)$ is the entropy of p_i. Hence, we can expand the above JS-divergences to get

$$\begin{aligned}\delta &= p(\hat{y}_j^-)H(\hat{y}_j^-) - \sum_{y_i \in \hat{y}_j^-} p(y_i)H(y_i) - p(\hat{y}_j)H(\hat{y}_j) - \sum_{y_i \in \hat{y}_j} p(y_i)H(y_i) \\ &\quad + p(\hat{y}_l^+)H(\hat{y}_l^+) - \sum_{y_i \in \hat{y}_l^+} p(y_i)H(y_i) - p(\hat{y}_l)H(\hat{y}_l) - \sum_{y_i \in \hat{y}_l^+} p(y_i)H(y_i) \\ &= p(\hat{y}_j^-)H(\hat{y}_j^-) - p(\hat{y}_j)H(\hat{y}_j) + p(\hat{y}_l^+)H(\hat{y}_l^+) - p(\hat{y}_l)H(\hat{y}_l). \qquad (4)\end{aligned}$$

Note that in the above, for brevity we have used y and $\hat{y}$ to also denote the distributions $p(X|y)$ and $p(X|\hat{y})$ respectively. The above analysis shows that δ can be computed in time that is linear in the dimensionality m. Equation (4) suggests that δ is expected to be large when changing cluster affiliation leads to a substantial change in the cluster distributions $\hat{y}_j^-$ and $\hat{y}_l^+$; this happens exactly in the cases where *DITC_prior* is seen to fail, i.e., when data is high-dimensional and clusters are small in size.

So far we have only considered moves that reduce loss in mutual information. An enhancement is to allow a chain of first variation moves that may lead to temporary increases. The algorithm searches for a sequence of successive moves that result in the largest possible reduction in the loss of mutual information (this enhancement was inspired by the successful graph partitioning heuristic of [9]). Algorithm *DITC_LocalSearch* in Figure 4 gives an outline of our proposed strategy. Note that the algorithm maintains a set of unmarked y's that correspond to the distributions not moved.

Finally, Figure 5 presents our algorithm that incorporates both the ideas of priors and local search.

6 Experimental Results

We now present experimental results when we apply our information-theoretic algorithm to the task of clustering document collections using word-document co-occurrence data.

Algorithm: DITC_PLS $(p(X,Y), k, \{\hat{y}_j\}_{j=1}^k, \alpha, f)$
Input: $p(X,Y)$ is the empirical joint probability distribution, k is the number of desired clusters,
α is the prior, f is LS_chain length.
Output: The set of clusters of $\hat{y}$, $\{\hat{y}_j^{(\tau)}\}_{j=1}^k$.
Method:
1. Initialization: $\tau \leftarrow 0$. Choose $\{p(X|\hat{y}_j^{(\tau)})\}_{j=1}^k$ to be the conditional distributions $p(X|y)$'s that are "maximally" far apart.
2. Repeat until convergence
   2a.Run DITC_prior $(p(X,Y), k, \{\hat{y}_j^\tau\}_{j=1}^k, \alpha)$
   2c.Run DITC_LocalSearch $(p(X,Y), k, \{\hat{y}_j^\tau\}_{j=1}^k, f)$

Figure 5: Divisive information theoretic clustering with prior and local search

6.1 Data sets

For our test data, we use various subsets of the 20-newsgroup data (NG20) [10] and the SMART collection (ftp://ftp.cs.cornell.edu/pub/smart).

NG20 consists of approximately 20,000 newsgroup postings collected from 20 different usenet newsgroups. We report results on NG20 and various subsets of this data set of size 500 each: *Binary* (talk.politics.mideast, talk.politics.misc; 250 documents from each category), *Multi5* (comp.graphics, rec.motorcycles, rec.sport.baseball, sci.space, talk.politics.mideast; 100 from each) and *Multi10* (alt.atheism, comp.sys.mac.hardware, misc.forsale, rec.autos, rec.sport.hocky, sci.crypt, sci.electronics, sci.med, sci.space, talk.politics.gun; 50 from each). By merging the five "comp", the three "religion", the three "politics", the two "sport" and the two "transportation" categories, we generated ten meta-categories in this corpus, which we call *NG10*. In order for our results to be comparable, we applied the same preprocessing as in [16] to all the news group data sets, i.e. we removed stopwords and selected the 2000 words with the highest contribution to the mutual information, removed documents with less than 10 word occurrences and removed all the headers except the subject line.

From SMART, we used MEDLINE, CISI, and CRANFIELD subcollections. MEDLINE consists of 1033 medical journals abstracts, CISI consists of 1460 abstracts from information retrieval papers, while CRANFIELD consists of 1400 aerodynamical systems abstracts. We also created 3 subsets of 30, 150, and 300 documents respectively; each data set was created by equal sampling of the three collections. After removing stopwords, the number of words for the 30, 150 and 300 document data sets is 1073, 3658 and 5577 respectively. We will refer to the entire data set as CLASSIC3 and the subsets as C30, C150 and C300 respectively.

6.2 Evaluation measures

Since we know the underlying class labels for our data sets, we can evaluate clustering results by forming a confusion matrix where entry(i,j) gives the number of documents in cluster i that belong to the true class j. For an objective evaluation measure, we use micro-averaged precision which was used in [16]. The idea is to first associate each cluster with the most dominant class label in that cluster. Then for each class $c \in C$, we define $\alpha(c,\hat{y})$ to be the number of points correctly assigned to c, $\beta(c,\hat{y})$ to be the number of documents incorrectly assigned to c and $\gamma(c,\hat{y})$ to be the number of documents incorrectly not assigned to c. The micro-averaged precision is

$$P(\hat{y}) = \frac{\sum_c \alpha(c,\hat{y})}{\sum_c \alpha(c,\hat{y}) + \beta(c,\hat{y})}$$

and the micro-averaged recall is

$$R(\hat{y}) = \frac{\sum_c \alpha(c,\hat{y})}{\sum_c \alpha(c,\hat{y}) + \gamma(c,\hat{y})}$$

Note that for uni-labeled data, $P(\hat{y}) = R(\hat{y})$.

Table 3: Confusion matrices for 3893 documents.

$\hat{y}_1$	**847**	41	275		**1016**	1	2
$\hat{y}_2$	142	**954**	86		1	**1389**	1
$\hat{y}_3$	44	405	**1099**		16	9	**1457**
	DITC partition				*DITC_prior* partition		

6.3 Results and Analysis

We first show that Algorithm *DITC_PLS* (with prior and local search) is superior to Algorithms *DITC_prior* and *DITC_LocalSearch*. Note that all our algorithms are deterministic since we choose initial cluster distributions that are "maximally" far apart from each other [3].

Algorithm *DITC_prior* cures the problem of sparsity to some extent and its results are superior to *DITC*, for example, Table 3 shows the confusion matrices resulting from the two algorithms. An interesting option in *DITC_prior* is the starting value of α. Indeed, as Figure 6 shows, the starting values of α can result in quite different values of mutual information preserved and micro-averaged precision. The trend seems to be that larger starting values of α lead to better results (we observe this trend over other data sets too). This is interesting in itself since larger α values correspond to starting with "smeared" cluster distributions, or in other words, with high joint entropy values $H(X, \hat{Y})$. This phenomena needs to be further studied and we feel it might have some relationship to the temperature parameter in deterministic annealing [14].

However, the starting values of α cease to be an issue when we use *DITC_PLS*, which is seen to be "immune" to different starting values in Figure 6. Note that these figures also validate our optimization criterion: there is a definite correlation between the mutual information preserved and micro-averaged precision, which was also observed in [16]. Thus *DITC_PLS* is seen to be more stable than *DITC_prior* in addition to yielding higher quality results. Tables 4 and 5 further show that *DITC_LocalSearch* also yields better clustering than *DITC_prior*. However, *DITC_PLS* is much more computationally efficient than *DITC_LocalSearch* since it yields better starting partitions before invoking the slower local search procedure; hence *DITC_PLS* is our method of choice.

We now compare our Algorithm *DITC_PLS* with previously proposed information-theoretic algorithms. [17] proposed the use of an agglomerative algorithm that first clusters words, and then uses this clustered feature space to cluster documents using the same agglomerative information bottleneck method. More recently [16] improved their clustering results by using sequential information bottleneck (sIB). We implemented the sIB algorithm for purpose of comparison; since the sIB method starts with a random partition we ran 10 trials and report the performance numbers with error bars (standard deviation) in Figures 7 and 8, which also contain performance results for our algorithms (recall that our algorithm are deterministic). Figures 7 and 8 again reveal the correlation between the preserved mutual information and micro-averaged precision. *DITC_PLS* is seen to be the best performing algorithm, and beats sIB on at least 3 of the data sets; for example, the average micro-averaged precision of sIB on *Multi5* is .8 while *DITC_PLS* yields .95. Note that numbers for our sIB implementation are averages of 10 runs while the published numbers in [16] are the best among 15 restarts (they did not give error bars). Also, the *Binary*, *Multi10* and *Multi5* datasets are formed by a sampling of the newsgroups, so our data sets are a bit different from theirs. Note that the NG10 and NG20 data sets used by us and [16] are identical, and so are the micro-averaged precision values (in this case they do give averaged results over 10 runs, see [16, Table 2].

For large data sets, *DITC_prior* gives results that are comparable to those with prior and local search; this leads to big savings in time since *DITC_prior* is much faster than sIB as shown in Table 6.

7 Conclusions and future work

In this paper, we have proposed the use of priors and local search to yield a computationally efficient clustering algorithm that directly optimizes the preservation of mutual information. We draw an analogy between our unsupervised method and the supervised Naive Bayes algorithm. In future work we would like to investigate the relationship between our choice of α and the temperature schedule in deterministic annealing [14]. At one

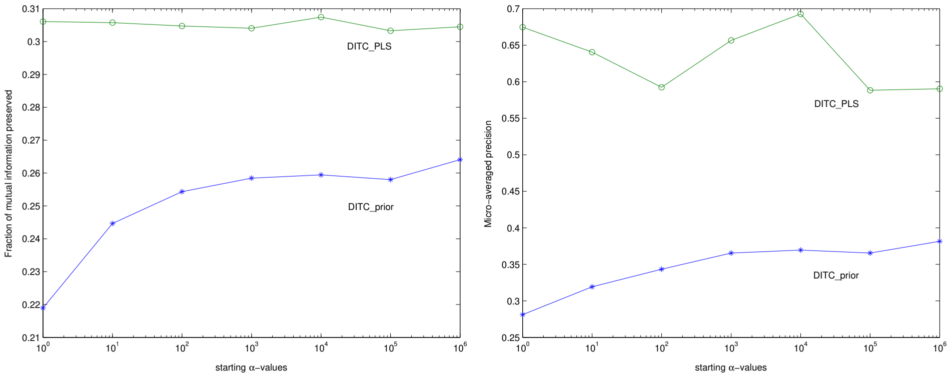

Figure 6: Mutual information preserved and Micro-averaged precision returned by *DITC_prior* with various starting α-values on *Multi10*

Table 4: Algorithm *DITC_LocalSearch* yields better confusion matrix than *DITC_prior* (150 documents, 3652 words, $f = 20$)

DITC_prior partition

$\hat{y}_1$	1	15	29
$\hat{y}_2$	13	11	8
$\hat{y}_3$	36	24	13

DITC_LocalSearch partition

50	0	1
0	50	0
0	0	49

Table 5: Algorithm *DITC_LocalSearch* yields better confusion matrix than *DITC_prior* (300 documents, 5577 words, $f = 20$)

DITC_prior partition

$\hat{y}_1$	45	38	35
$\hat{y}_2$	31	26	33
$\hat{y}_3$	24	36	32

DITC_LocalSearch partition

97	0	0
1	100	0
2	0	100

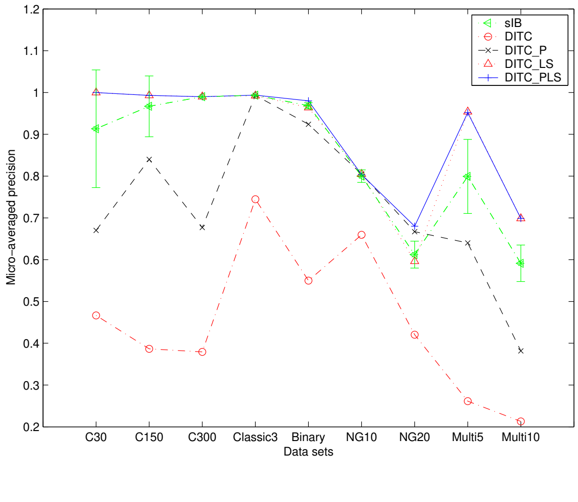

Figure 7: Micro-averaged precision results.

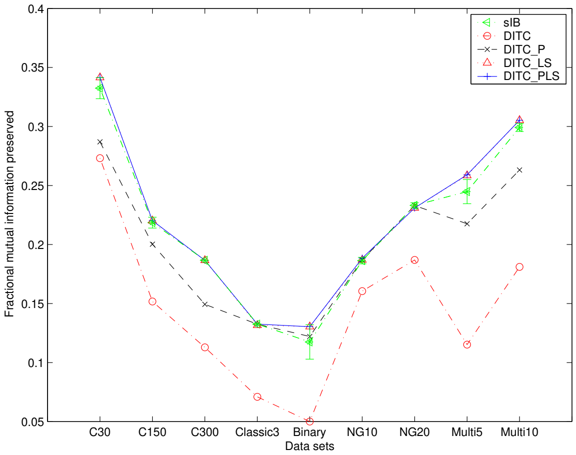

Figure 8: Fraction of mutual information preserved.

Table 6: Computation time (in seconds) on large data sets (≥ 3000).

DATA SET	sIB	DITC	DITC_PRIOR
CLASSIC3	94.92	1.35	1.67
NG20	6244	35.87	29.92
NG10	2459	16.71	14.75

level our algorithm differs from deterministic annealing since the latter does "soft clustering", however our large initial values of α at earlier iterations correspond to large entropy values $H(X;\hat{Y})$, so we suspect there might be a theoretical connection. The information theoretic framework is general and we intend to apply it to more interesting problems, such as clustering the states of a Markov chain by analyzing its probability transition matrix. We also believe that this information-theoretic framework is "natural" for applying MDL techniques for selecting the "right" number of clusters. Further, we intend to extend our algorithm to co-cluster or simultaneously cluster both X and Y so that the loss in mutual information $I(X;Y) - I(\hat{X};\hat{Y})$ is minimized.

Acknowledgments. This research was supported by NSF CAREER Award No. ACI-0093404.

References

[1] L. D. Baker and A. McCallum. Distributional clustering of words for text classification. In *ACM SIGIR*, pages 96–103. ACM, August 1998.

[2] P. Berkhin and J. D. Becher. Learning simple relations: Theory and applications. In *2nd SIAM Int'l Conference on Data Mining*, pages 420–436, 2002.

[3] P. Bradley, U. Fayyad, and C. Reina. Scaling clustering algorithms to large databases. In *KDD'03*. AAAI Press, 1998.

[4] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, USA, 1991.

[5] I. S. Dhillon, Y. Guan, and J. Kogan. Iterative clustering of high dimensional text data augmented by local search. In *Proceedings of The 2002 IEEE International Conference on Data Mining*, 2002.

[6] I. S. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research(JMLR): Special Issue on Variable and Feature Selection*, 3:1265–1287, March 2003.

[7] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* John Wiley & Sons, 2nd edition, 2000.

[8] T. Hofmann and J. Puzicha. Statistical models for co-occurrence and histogram data. In *ICPR98*, pages 192–194, 1998.

[9] B.W. Kernighan and S Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49(2):291–307, 1970.

[10] Ken Lang. News Weeder: Learning to filter netnews. In *Proc. 12th Int'l Conf. Machine Learning*, pages 331–339, San Francisco, 1995.

[11] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Trans. Inform. Theory*, 37(1):145–151, 1991.

[12] Tom M. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

[13] F. Pereira, N. Tishby, and L. Lee. Distributional clustering of English words. In *31st Annual Meeting of the ACL*, pages 183–190, 1993.

[14] K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proc. IEEE*, 86(11):2210–2239, 1998.

[15] Mehran Sahami. *Using Machine Learning to Improve Information Access.* PhD thesis, CS Department, Stanford University, 1998.

[16] N. Slonim, N. Friedman, and N. Tishby. Unsupervised document classification using sequential information maximization. In *ACM SIGIR*, 2002.

[17] N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In *ACM SIGIR*, pages 208–215, 2000.

[18] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.