Document Clustering using Word Clusters via the Information Bottleneck Method

Noam Slonim and Naftali Tishby

School of Computer Science and Engineering and
The Interdisciplinary Center for Neural Computation
The Hebrew University, Jerusalem 91904, Israel
email: {noamm,tishby}@cs.huji.ac.il

Abstract

We present a novel implementation of the recently introduced *information bottleneck method* for unsupervised document clustering. Given a joint empirical distribution of words and documents, $p(x, y)$, we first cluster the words, Y, so that the obtained word clusters, $\tilde{Y}$, maximally preserve the information on the documents. The resulting joint distribution, $p(X, \tilde{Y})$, contains most of the original information about the documents, $I(X;\tilde{Y}) \approx I(X;Y)$, but it is much less sparse and noisy. Using the same procedure we then cluster the documents, X, so that the information about the word-clusters is preserved. Thus, we first find *word-clusters* that capture most of the mutual information about the set of documents, and then find *document clusters*, that preserve the information about the word clusters. We tested this procedure over several document collections based on subsets taken from the standard $20Newsgroups$ corpus. The results were assessed by calculating the correlation between the document clusters and the correct labels for these documents. Finding from our experiments show that this *double clustering* procedure, which uses the information bottleneck method, yields significantly superior performance compared to other common document distributional clustering algorithms. Moreover, the double clustering procedure improves all the distributional clustering methods examined here.

1 Introduction

Document clustering has long been an important problem in information retrieval. Early works suggested improving the efficiency and increasing the effectiveness of document retrieval systems by first grouping the documents into clusters (cf. [27] and the references therein). Recently, document clustering has been put forward as an important tool for Web search engines [15] [16] [18] [30], navigating and browsing document collections [5] [6] [8] [9] [23] and distributed retrieval [29]. Two types of clustering have been studied in the context of information retrieval systems: clustering the documents on the basis of the distributions of words that co-occur in the documents, and clustering the words using the distributions of the documents in which they occur (see [28] for in-depth review). In this paper we propose a new method for document clustering, which combines these two approaches under a single information theoretic framework. A recently introduced principle, termed the *information bottleneck method* [26] is based on the following simple idea. Given the empirical joint distribution of two variables, one variable is compressed so that the mutual information about the other is preserved as much as possible. In our case these two variables correspond to the set of documents and the set of words. Thus, we may find *word-clusters* that capture most of the information about the document corpus, or we may extract *document clusters* that capture most of the information about the words that occur. In this work we combine the two alternatives. We approach this problem using a two stage algorithm. First, we extract word-clusters that capture most of the information about the documents. In the second stage we replace the original representation of the documents, the co-occurrence matrix of documents versus words, by a much more compact representation based on the co-occurrences of the word-clusters in the documents. Using this new document representation, we reapply the same clustering procedure to obtain the desired document clusters. The main advantage of this *double-clustering* procedure lies in a significant reduction of the inevitable noise of the original co-occurrence matrix, due to its very high dimension. The reduced matrix, based on the word-clusters, is denser and more robust, providing a better reflection of the inherent structure of the document corpus.

Our main concern is how well this method actually discovers this inherent structure. Therefore, instead of evaluating our procedure by its effectiveness for an IR system (e.g. [30]), we evaluate the method on a standard *labeled* corpus, commonly used to evaluate *supervised* text classification algorithms. In this way we circumvent the bias caused by the use of a specific IR system. In addition, we view the 'correct' labels of the documents as objective knowledge on the inherent structure of the dataset. Specifically we used the $20Newsgroups$ dataset, collected by Lang [12], which contains about $20,000$ articles evenly distributed over 20 UseNet discussion groups. From this corpus we generated several subsets and measured clustering performance via the correlation between the obtained document clusters and the original newsgroups. We compared several clustering algorithms including the single-stage information bottleneck algorithm [24], Ward's method [1] and complete-linkage [28] using the standard *tf-idf*

term weights [20]. We found that double-clustering, using the information bottleneck method, was significantly superior to all the other examined algorithms. In addition, the double-clustering procedure improved performance over other algorithms in all our experiments. In other words, clustering the documents by their words was always inferior to clustering by word-clusters.

2 The Information Bottleneck Method

Most clustering algorithms start either from pairwise 'distances' between points (pairwise clustering) or with a distortion measure between a data point and a class centroid (vector quantization). Given the distance matrix or the distortion measure, the clustering task can be adapted in various ways into an optimization problem consisting of finding a small number of classes with low intraclass distortion or with high intraclass connectivity. The main problem with this approach is in the choice of the distance or distortion measures. Too often this is an arbitrary choice, sensitive to the specific representation, which may not accurately reflect the structure of the various components in the high dimensional data.

In the context of document clustering, a natural measure of similarity of two documents is the similarity between their word conditional distributions. Specifically, let X be the set of documents and let Y be the set of words, then for every document we can define

$$p(y|x) = \frac{n(y|x)}{\sum_{y \in Y} n(y|x)} \ , \tag{1}$$

where $n(y|x)$ is the number of occurrences of the word y in the document x. [1] Roughly speaking, we would like documents with similar conditional word distributions to belong to the same cluster. This formulation of finding a cluster hierarchy of the members of one set (e.g. documents), based on the similarity of their conditional distributions w.r.t the members of another set (e.g. words), was first introduced in [17] and was called "distributional clustering".

The issue of selecting the 'right' distance measure between distributions remains, however, unresolved in that earlier work. Recently, Tishby, Pereira, and Bialek [26] proposed a principled approach to this problem, which avoids the arbitrary choice of a distortion or a distance measures. In this new approach, given the empirical joint distribution of two random variables $p(x, y)$, one looks for a compact representation of X, which preserves as much information as possible about the relevant variable Y. This simple intuitive idea has a natural information theoretic formulation: *find clusters of the members of the set X, denoted here by $\tilde{X}$, such that the mutual information $I(\tilde{X}; Y)$ is maximized, under a constraint on the information extracted from X, $I(\tilde{X}; X)$.*

The mutual information, $I(X; Y)$, between the random variables X and Y is given by (e.g. [4])

$$I(X;Y) = \sum_{x \in X, y \in Y} p(x)p(y|x) \log \frac{p(y|x)}{p(y)} \ , \tag{2}$$

and is the only consistent statistical measure of the information that variable X contains about variable Y. The compactness of the representation is determined by $I(\tilde{X}; X)$, while the quality of the clusters, $\tilde{X}$, is measured by the fraction of the information they capture about Y, namely, $I(\tilde{X}; Y)/I(X; Y)$. Perhaps surprisingly, this general problem has an exact optimal formal solution without any assumption about the origin of the joint distribution $p(x, y)$ [26]. This solution is given in terms of the three distributions that characterize every cluster $\tilde{x} \in \tilde{X}$: the prior probability for this cluster, $p(\tilde{x})$, its membership probabilities $p(\tilde{x}|x)$, and its distribution over the relevance variable, $p(y|\tilde{x})$. In general, the membership probabilities, $p(\tilde{x}|x)$, are 'soft', i.e. every $x \in X$ can be assigned to every $\tilde{x} \in \tilde{X}$ in some (normalized) probability. The information bottleneck principle determines the distortion measure between the points x and $\tilde{x}$ to be the $D_{KL}[p(y|x) \| p(y|\tilde{x})] = \sum_y p(y|x) \log \frac{p(y|x)}{p(y|\tilde{x})}$, the Kulback-Libeler divergence [4] between the conditional distributions $p(y|x)$ and $p(y|\tilde{x})$. Specifically, the formal solution is given by the following equations which must be solved together,

$$\begin{cases} p(\tilde{x}|x) = \frac{p(\tilde{x})}{Z(\beta,x)} \exp\left(-\beta D_{KL}[p(y|x) \| p(y|\tilde{x})]\right) \\ p(y|\tilde{x}) = \frac{1}{p(\tilde{x})} \sum_x p(\tilde{x}|x)p(x)p(y|x) \\ p(\tilde{x}) = \sum_x p(\tilde{x}|x)p(x) \ , \end{cases} \tag{3}$$

where $Z(\beta, x)$ is a normalization factor, and the single positive (Lagrange) parameter β determines the "softness" of the classification. Intuitively, in this procedure the information contained in X about Y is 'squeezed' through a compact 'bottleneck' of clusters $\tilde{X}$, that is forced to represent the 'relevant' part in X w.r.t. to Y.

2.1 Relation to previous work

An important information theoretic based approach to word clustering was carried out by Brown et al [3] who used n-gram models, and about the same time by Pereira, Tishby and Lee [17] who introduced an early version of the bottleneck method, using verb-object tagging for word sense disambiguation. Hofmann [10] has recently proposed another procedure, called probabilistic latent semantic indexing (PLSI) for automated document indexing, motivated and based upon our earlier work. Using this procedure one can represent documents (and words) in a low-dimensional 'latent semantic space'. The latent variables defined in this scheme are somewhat analogous to our $\tilde{X}$ variable. However, there are important differences between these approaches. First, while PLSI assumes a generative hidden variable model for the data and uses maximum likelihood for estimating the latent variables, the information bottleneck method makes no assumption about the structure of the data distribution (it is *not* a hidden variable model) and uses a variational principle to optimize directly the *relevant information* in the co-occurrence data to extract the new representation. Second, the PLSI model is based on a conditional-independence assumption, i.e. given the latent variables the words and documents are independent, which is not needed in our approach. Another important advantage of our method is that it has a complete (formal) analytic solution, enabling better understanding of the resulting classification.

3 The Agglomerative Information Bottleneck Algorithm

As has been shown in [24] [25], there is a simple implementation of the information bottleneck method, restricted to the case of 'hard' clusters. In this case every $x \in X$ belongs to precisely one cluster $\tilde{x} \in \tilde{X}$. This restriction, which corresponds to the limit $\beta \to \infty$ in Eqs. (3), yields a natural distance measure between distributions which can be easily implemented in an agglomerative hierarchical clustering procedure.

[1] Note that under this definition the priors of all documents are uniformly normalized to $p(x) = \frac{1}{|X|}$, thus we avoid an undesirable bias due to different document lengths.

Let $\tilde{x} \in \tilde{X}$ denote a specific (hard) cluster, then following [25] we define,

$$\begin{cases} p(\tilde{x}|x) = \begin{cases} 1 & \text{if } x \in \tilde{x} \\ 0 & \text{otherwise} \end{cases} \\ p(y|\tilde{x}) = \frac{1}{p(\tilde{x})} \sum_{x \in \tilde{x}} p(x) p(y|x) \\ p(\tilde{x}) = \sum_{x \in \tilde{x}} p(x). \end{cases} \quad (4)$$

Using these distributions one can easily evaluate the mutual information between the set of clusters $\tilde{X}$ and Y using Eq.(2). As stated earlier, the objective of the information bottleneck method is to extract partitions of X, defined by the mapping $p(\tilde{x}|x)$, that maximize the mutual information functional. Note that the roles of X and Y in this scenario can be switched. We may extract clusters of words which capture maximum mutual information about the documents, or find clusters of documents that capture the mutual information about the words. We utilize this symmetry in the *double-clustering* procedure (see section 5). The general framework applied here is an agglomerative greedy hierarchical clustering algorithm. The algorithm starts with trivial partitioning into $|X|$ singleton clusters, where each cluster contains exactly one element of X. At each step we *merge* two components of the current partition into a single new component in a way that *locally* minimizes the loss of mutual information $I(\tilde{X};Y)$. Every merger, $(\tilde{x}_i\ \tilde{x}_j) \Rightarrow \tilde{x}_*$, is formally defined by the following equations

$$\begin{cases} p(\tilde{x}_*|x) = \begin{cases} 1 & \text{if } x \in \tilde{x}_i \text{ or } x \in \tilde{x}_j \\ 0 & \text{otherwise} \end{cases} \\ p(y|\tilde{x}_*) = \frac{p(\tilde{x}_i)}{p(\tilde{x}_*)} p(y|\tilde{x}_i) + \frac{p(\tilde{x}_j)}{p(\tilde{x}_*)} p(y|\tilde{x}_j) \\ p(\tilde{x}_*) = p(\tilde{x}_i) + p(\tilde{x}_j). \end{cases} \quad (5)$$

The decrease in the mutual information $I(\tilde{X};Y)$ due to this merger is defined by $\delta I(\tilde{x}_i, \tilde{x}_j) \equiv I(\tilde{X}_{before};Y) - I(\tilde{X}_{after};Y)$, where $I(\tilde{X}_{before};Y)$ and $I(\tilde{X}_{after};Y)$ are the information values before and after the merger, respectively. After a little algebra [25] one can see that

$$\delta I(\tilde{x}_i, \tilde{x}_j) = (p(\tilde{x}_i) + p(\tilde{x}_j)) \cdot D_{JS}[p(y|\tilde{x}_i), p(y|\tilde{x}_j)] \quad (6)$$

where the functional D_{JS} is the *Jensen-Shannon* (JS) divergence (see [13] [7]) defined as

$$D_{JS}[p_i, p_j] = \pi_i D_{KL}[p_i \| \bar{p}\,] + \pi_j D_{KL}[p_j \| \bar{p}\,], \quad (7)$$

where in our case

$$\begin{cases} \{p_i, p_j\} \equiv \{p(y|\tilde{x}_i), p(y|\tilde{x}_j)\} \\ \{\pi_i, \pi_j\} \equiv \{\frac{p(\tilde{x}_i)}{p(\tilde{x}_*)}, \frac{p(\tilde{x}_j)}{p(\tilde{x}_*)}\} \\ \bar{p} = \pi_i p(y|\tilde{x}_i) + \pi_j p(y|\tilde{x}_j)\ . \end{cases} \quad (8)$$

The JS-divergence is non-negative and equals zero if and only if both arguments are identical. It is upper bounded (by 1) and symmetric though it is not a metric. Note that the "merger cost", $\delta I(\tilde{x}_i, \tilde{x}_j)$, can now be interpreted as the multiplication of the 'weight' of the merged elements, $p(\tilde{x}_i) + p(\tilde{x}_j)$, by their 'distance', $D_{JS}[p(y|\tilde{x}_i), p(y|\tilde{x}_j)]$.

By introducing the information optimization criterion the resulting similarity measure directly *emerges* from the analysis. The algorithm is now very simple, where at each step we perform "the best possible merge", i.e. merge the clusters $\{\tilde{x}_i, \tilde{x}_j\}$ which minimize $\delta I(\tilde{x}_i, \tilde{x}_j)$. In figure 1 we provide the pseudo code of this agglomerative procedure.

Input: Joint probability distribution $p(x, y)$

Output: A partition of X into m clusters, $\forall m \in \{1...|X|\}$

Initialization:

- Construct $\tilde{X} \equiv X$
- $\forall i, j = 1...|X|,\ i < j$, calculate $d_{i,j} = (p(\tilde{x}_i) + p(\tilde{x}_j)) D_{JS}[p(y|\tilde{x}_i), p(y|\tilde{x}_j)]$

Loop:

- For $m = |X| - 1 \ ... \ 1$
  - Find the indices $\{i, j\}$ for which $d_{i,j}$ is minimized
  - Merge $\{\tilde{x}_i, \tilde{x}_j\} \Rightarrow \tilde{x}_*$
  - Update $\tilde{X} = \{\tilde{X} - \{\tilde{x}_i, \tilde{x}_j\}\} \bigcup \{\tilde{x}_*\}$
  - Update $d_{i,j}$ costs w.r.t. $\tilde{x}_*$
- **End For**

Figure 1: Pseudo-code of the agglomerative information bottleneck algorithm.

4 Other Clustering Methods

In the same general framework of the agglomerative clustering algorithm, we applied two other similarity criteria to construct two other algorithms for purposes of comparison. First, a common natural distance measure between probability distributions is the $L1$ norm (or the *variational distance*), defined as,

$$L_1(p_i, p_j) \equiv \sum_{y \in Y} |p_i(y) - p_j(y)|. \quad (9)$$

Unlike the JS-divergence, the $L1$ norm is a *distance* measure satisfying all the metric properties, including triangle inequality. It also approximates the JS-divergence for close distributions [13]. Our second clustering algorithm therefore used the following distributional similarity measure

$$d_{i,j} = (p(\tilde{x}_i) + p(\tilde{x}_j)) \cdot L_1(p(y|\tilde{x}_i), p(y|\tilde{x}_j))\ . \quad (10)$$

Note that multiplication by the 'weight' of the clusters to be merged is crucial. Otherwise there is a strong bias for assigning all objects into one cluster. Besides these two algorithms, which are motivated by probability theory, our third comparison algorithm is the standard Ward's method which is based on the Euclidean distance [1]. The similarity measure for this algorithm is thus given by

$$d_{i,j} = \frac{p(\tilde{x}_i)p(\tilde{x}_j)}{p(\tilde{x}_i) + p(\tilde{x}_j)} \cdot \sum_{y \in Y} (p(y|\tilde{x}_i) - p(y|\tilde{x}_j))^2\ . \quad (11)$$

In addition we also implemented a *complete-linkage* (agglomerative) algorithm (see e.g. [28]) which uses the conventional *tf-idf* term-weights [20] to represent the documents in a vector space model. In this method the least similar pair of documents, one of

Input: Joint probability distribution $p(x, y)$

First Stage: Find clusters $\tilde{Y}$ using $\{p(x|y)\}$

Second Stage:

- For every $x \in X$, replace $p(y|x)$ by the more compact representation $p(\tilde{y}|x)$
- Find clusters $\tilde{X}$ using $\{p(\tilde{y}|x)\}$

Figure 2: The double-clustering procedure.

each cluster, determines the similarity between clusters. Specifically,

$$SIM(\tilde{x}_i, \tilde{x}_j) = min_{x \in \tilde{x}_i, x' \in \tilde{x}_j}(SIM(x, x')) \ , \qquad (12)$$

where for $SIM(x, x')$ we used the cosine of the angle between the two *tf-idf* vectors representing the documents. We also implemented a *single-linkage* algorithm for which the *most* similar pair of documents, one from each cluster, determines the similarity between clusters. However, the results for this method were significantly inferior to the complete-linkage algorithm (due to a strong tendency to cluster all documents into one huge cluster), thus we do not report these data here.

5 The Double Clustering Procedure

The three criteria described in Eqs.(6, 10, 11) are essentially symmetric with regard to the roles of X and Y. In other words, there are no prior requirements regarding which variable should be compressed. In this work we suggest a combination of these two options. In order to do that we introduce a two-stage clustering procedure. In the first stage we represent each word y by its conditional distribution over the set of documents, $p(x|y)$. We then use a distributional clustering algorithm to obtain *word-clusters*, denoted by $\tilde{Y}$, with $|\tilde{Y}| \ll |Y|$. In the second stage we use these word-clusters to replace the original representation of the documents. Instead of representing a document by its conditional distribution of words, $p(y|x)$, we represent it by its conditional distribution of **word-clusters**, $p(\tilde{y}|x)$, defined by

$$p(\tilde{y}|x) = \frac{n(\tilde{y}|x)}{\sum_{\tilde{y} \in \tilde{Y}} n(\tilde{y}|x)} = \frac{\sum_{y \in \tilde{y}} n(y|x)}{\sum_{\tilde{y} \in \tilde{Y}} \sum_{y \in \tilde{y}} n(y|x)} \ . \qquad (13)$$

Using this compact representation, we re-apply the distributional clustering algorithm to extract the desired document clusters, $\tilde{X}$. In figure 2 we outline the double clustering procedure.

Using the information bottleneck method in this double-clustering framework provides clear-cut information on the nature of the clusters obtained. In the first stage the algorithm extracts word-clusters which capture most of the relevant information about the given documents. More formally stated, in the first stage the algorithm finds a set of clusters $\tilde{Y}$ such that $I(X; \tilde{Y}) \lessapprox I(X; Y)$. In the second stage, the algorithm extracts document clusters, $\tilde{X}$, that capture most of the relevant information about the word-clusters. Therefore we obtain significant reduction in both dimensions of the original variables, without losing too much of their mutual information: $I(\tilde{X}; \tilde{Y}) \lessapprox I(X; \tilde{Y}) \lessapprox I(X; Y)$.

6 The Experimental Design

In this section we describe our experiments and present a new objective method for evaluating document clustering procedures. In addition we describe the datasets used in our experiments, which are all based on a standard IR corpus, the $20Newsgroups$ corpus.

6.1 The evaluation method

In general, measuring clustering effectiveness is not a trivial issue. Standard measures such as the average distance between data points and candidate class centroids are rather abstract for our needs, and furthermore, as already mentioned, it is not clear what distance measure should be used. In most of the previous work on document clustering the performance of the clustering has been measured in terms of its effectiveness over some information retrieval system. Specifically, the clustering results are used to reorder the list of documents returned by the IR system, under the assumption that the user is able to select the clusters with the highest relevant document density [9] [22] [30]. There are several problems in this evaluation method. First, as noted in [30], empirical tests have shown that users fail to choose the best cluster about 20% of the time [9]. Second, generating the document collections by the results obtained by an IR system w.r.t. some queries is sensitive to the specific IR system and queries being used, which may result in some unclear bias over the datasets. Third, this evaluation method does not measure directly how well the inherent structure of the document corpus is revealed by the clustering procedure, but rather provide *indirect estimates*, through the IR system performance. To overcome these problems we propose a simple solution, which is essentially estimating document clustering performance by tools used for *supervised* text classification tasks. In other words, since our interest is in measuring how well the clustering process can reveal the inherent structure of a given document collection, we use a standard *labeled* text classification corpus to construct our datasets, while using the labels as clear objective knowledge reflecting the dataset inherent structure. In addition, we adopt the accuracy measure used by supervised learning algorithms to our needs. Specifically, we measure clustering performance by the *accuracy* given by the contingency table of the obtained clusters and the 'real' document categories.

6.2 The datasets

We constructed 10 different document subsets of the $20Newsgroups$ corpus collected by Lang [12]. This corpus contains about $20,000$ articles evenly distributed among 20 UseNet discussion groups, and is usually employed for evaluating supervised text classification techniques (e.g. [2] [21]). Many of these groups have similar topics (e.g. five groups discuss different issues concerning computers). In addition, as pointed out by Schapire and Singer [21] about 4.5% of the documents in this corpus are present in more than one group (since people tend to post articles to multiple newsgroups). Therefore, the 'real' clusters are inherently fuzzy. For our tests we used 10 different randomly chosen subsets from this corpus. The details of these subsets are given in table 1. Our pre-processing included ignoring all file headers, lowering the upper case characters and ignoring all words that contained digits or non alpha-numeric characters. We did not use a stop-list or any stemming procedure. However, we included a standard feature selection mechanism, where for each dataset we selected the 2000 words with the highest contribution to the mutual information between the words and the documents. More formally stated, for each dataset, we sorted all words by $I(y) \equiv p(y) \sum_{x \in X} p(x|y) \log \frac{p(x|y)}{p(x)}$ and selected the top 2000.

Dataset	Newsgroups included	#documents per group	Total #documents
$Science$	sci.crypt, sci.electronics, sci.med, sci.space.	500	2000
$Binary_{1,2,3}$	talk.politics.mideast, talk.politics.misc.	250	500
$Multi5_{1,2,3}$	comp.graphics, rec.motorcycles, rec.sport.baseball, sci.space talk.politics.mideast.	100	500
$Multi10_{1,2,3}$	alt.atheism, comp.sys.mac.hardware, misc.forsale, rec.autos, rec.sport.hockey sci.crypt, sci.electronics, sci.med, sci.space, talk.politics.gun.	50	500

Table 1: Datasets details. For example, for each of the three $Binary$ datasets we randomly chose 500 documents, evenly distributed between the news groups talk.politics.mideast and talk.politics.misc. This resulted in three document collections, $Binary_1$, $Binary_2$ and $Binary_3$, each of which consisted of 500 documents.

7 Experimental Results

For each of our 10 document collections we tested the following clustering algorithms:

- IB_{double}: The double-clustering procedure using the distance measure derived from the information bottleneck method (Eq. 6).
- $L1_{double}$: The double-clustering procedure using the $L1$-norm distance measure (Eq. 10).
- $Ward_{double}$: The double-clustering procedure using Ward's distance measure, i.e. the Euclid norm (Eq. 11).
- IB_{single}: Clustering the documents based on the original co-occurrence matrix of documents versus words, using the distance measure derived from the information bottleneck method (Eq. 6).
- $L1_{single}$: Clustering the documents based on the original co-occurrence matrix of documents versus words, using the $L1$-norm distance measure (Eq. 10).
- $Ward_{single}$: Clustering the documents based on the original co-occurrence matrix of documents versus words, using Ward's distance measure (Eq. 11).
- $Complete_{tf-idf}$: Clustering the documents based on the original co-occurrence matrix of documents versus words, using a complete-linkage algorithm and the *tf-idf* term weights. The similarity between documents was estimated by the cosine of the angle between the two *tf-idf* vectors representing the documents.

To avoid bias due to the number of word-clusters used by the double-clustering procedures, we tested the performance of these algorithms for various numbers of word-clusters. Specifically we tested performance using $10, 20, 30, 40$ and 50 word-clusters. Performance was estimated as the *accuracy* given by the contingency table of the obtained clusters and the real document categories. [2] For example, in table 2 we present the contingency table and the accuracy for the IB_{double} algorithm over the $Multi5_2$ dataset using 10 word-clusters. Note that this accuracy was obtained in an *unsupervised* manner, without using any of the document labels. The number of document clusters used for evaluating the contingency table was generally set to be identical to the number of 'real' categories (except for the *Binary* datasets for which we used 4 document clusters instead of 2). This is equivalent to a simplifying assumption that a user is approximately aware of the number of 'real' categories in the document collection. Choosing the appropriate number of document clusters without any prior knowledge about the data is a question of model selection which is beyond the scope of this work. We note, however, that this problem could be addressed using standard techniques such as cross-validation.

	graphics	*motorcycles*	*baseball*	*space*	*mideast*
$\tilde{x}_1$	**78**	3	11	6	10
$\tilde{x}_2$	3	**68**	7	5	5
$\tilde{x}_3$	4	5	**59**	8	9
$\tilde{x}_4$	6	14	13	**68**	13
$\tilde{x}_5$	9	10	10	13	**63**

Table 2: Contingency table for the IB_{double} algorithm over the $Multi5_2$ dataset using 10 word-clusters. The accuracy is 0.67.

Detailed results for all three double-clustering algorithms in all 10 datasets are given in figure 3. In table 3 we list the results for the three single-clustering procedures, as well as for the $Complete_{tf-idf}$ algorithm. For purposes of comparison we also include the average results of the double-clustering procedures. Several results should be noted specifically:

- Using the information bottleneck algorithm with the double clustering procedure (i.e. algorithm IB_{double}) resulted in superior performance compared to the other algorithms. Specifically the average performance over all datasets attained 0.55 accuracy, while the second best result was 0.47 accuracy (for the $Complete_{tf-idf}$ algorithms). [3]

[2] Another possible measure for the quality of the obtained clusters is the *mutual information* given in the contingency table. This measure is highly correlated with the accuracy measure we used in this work. For purposes of comparison, we also present the averaged results using the mutual information as the quality measure in figure 3.

[3] To gain some perspective we also tested the performance of *Rainbow* software package [14] using a naive Bayes *supervised* classifier. The training set for each

Data/algorithm	IB_{double}	IB_{single}	$L1_{double}$	$L1_{single}$	$Ward_{double}$	$Ward_{single}$	$Complete_{tf-idf}$
$Science$	0.59	0.49	0.41	0.34	0.33	0.29	0.47
$Binary_1$	0.70	0.71	0.61	0.62	0.59	0.56	0.67
$Binary_2$	0.68	0.60	0.60	0.57	0.56	0.51	0.61
$Binary_3$	0.75	0.70	0.66	0.65	0.60	0.60	0.52
$Multi5_1$	0.59	0.42	0.43	0.38	0.34	0.27	0.51
$Multi5_2$	0.58	0.40	0.43	0.36	0.29	0.28	0.34
$Multi5_3$	0.53	0.50	0.46	0.34	0.34	0.29	0.63
$Multi10_1$	0.35	0.24	0.31	0.24	0.20	0.19	0.27
$Multi10_2$	0.35	0.26	0.28	0.27	0.21	0.20	0.33
$Multi10_3$	0.35	0.29	0.27	0.28	0.20	0.17	0.34
Average	**0.55**	**0.46**	**0.45**	**0.40**	**0.37**	**0.34**	**0.47**

Table 3: Averaged results for all clustering procedures in all datasets. For the double-clustering algorithms the results for every dataset are averaged over the different numbers of word-clusters used in the process.

- The double-clustering algorithms were tested using $10, 20, 30, 40$ and 50 word-clusters for every dataset, i.e. 50 runs (5 for each dataset). In almost all runs (48 out of 50) the IB_{double} performance was superior to the other double-clustering algorithms. Of these, the $L1_{double}$ was usually better than the $Ward_{double}$. In addition, in 46 out of these 50 runs, double-clustering improved the performance for all the distance measures used in the clustering process. In other words, IB_{double}, $L1_{double}$ and $Ward_{double}$ were almost always superior to IB_{single}, $L1_{single}$ and $Ward_{single}$ respectively. The most significant improvement was for the information bottleneck algorithms. (IB_{double} versus IB_{single}).
- The single-stage procedures were tested once for each dataset, i.e. 10 runs. Averaging over these runs, the $Complete_{tf-idf}$ algorithm was slightly better than the IB_{single} algorithm, which on its own was significantly better than the $L1_{single}$ algorithm. The $Ward_{single}$ algorithm exhibited the weakest performance for almost all datasets.
- The best performance of all algorithms was over the three *Binary* datasets. For the *Multi*5 and *Science* datasets performance was usually similar. The weakest performance was obtained consistently for the *Multi*10 datasets. In other words, as expected, increasing the number of categories resulted in poorer performance, regardless of the algorithm used.

8 Discussion and Further Work

In this paper we presented a novel principled approach to the clustering of documents, which outperformed several other commonly used algorithms. The combination of two novel ingredients contributed to this work. The first is the *information bottleneck method*, which is a principled information theoretic approach to distributional clustering. It provides an objective measure of the quality of the obtained clusters - the extracted relevant information - as well as a well justified distributional similarity measure. This measure, which emerges directly from first principles, is the *KL-divergence* to the mean, or the *Jensen-Shannon* divergence, and is in this information theoretic sense the optimal similarity measure. The second ingredient is the *double clustering* procedure. This mechanism, which can be used with *any* distributional clustering algorithm, amounts to clustering in both dimensions - first, words based on their document distribution, and second - documents based on their *word-clusters* distribution. We demonstrated that the double-clustering procedure is useful for all the similarity measures we examined. When combined with the information bottleneck, the results were clearly better. The method is shown to provide good document classification accuracy for the $20Newsgroups$ dataset, in a fully *unsupervised* manner, without using any training labels. We argue that these results demonstrate both the validity of the information bottleneck method and the power of the double-clustering procedure for this problem.

The agglomerative procedure used in this work has time complexity of $O(|X|^3)$, which is not suitable for very large datasets. Several techniques have been proposed for dealing with this issue, which can also be employed here. For example, a somewhat similar clustering procedure was recently used by Baker and McCallum [2] for finding word clusters in *supervised* text categorization. To avoid high complexity they suggested using a fixed small number of clusters, where in each step the two most similar clusters are joined and another new word is added as a new cluster. Their work pointed out that distributional clustering can be useful for significant reduction of the feature dimensionality with minor decrease in classification accuracy. The present work, in contrast, shows that for *unsupervised* document classification, using word clusters is not only more efficient, but also leads to significant improvement in performance.

The double-clustering procedure used here is a two stage process. First we find word-clusters and then use them to obtain document clusters. A natural generalization is to try and compress both dimensions of the original co-occurrence matrix *simultaneously*, and we are working in this direction. In addition the agglomerative information bottleneck algorithm used here is a special case of a more general algorithm which yields even better performance on the same data. This more general approach is beyond the scope of this work and will be presented elsewhere [25].

Acknowledgments

Useful discussions with Yoram Singer and Fernando Pereira are greatly appreciated. This research was supported by grants from the Israeli Ministry of Science, and by the US-Israel Bi-national Science Foundation (BSF).

dataset was set to 200 documents (evenly distributed). The test sets were identical to those listed in table 1. Averaging over 5 runs, the averaged performance over all data sets attained 0.71 accuracy. Specifically for the three $Multi10$ datasets, *Rainbow* averaged performance attained 0.38 accuracy while the *unsupervised* IB_{double} average performance attained 0.35 accuracy, which is definitely comparable.

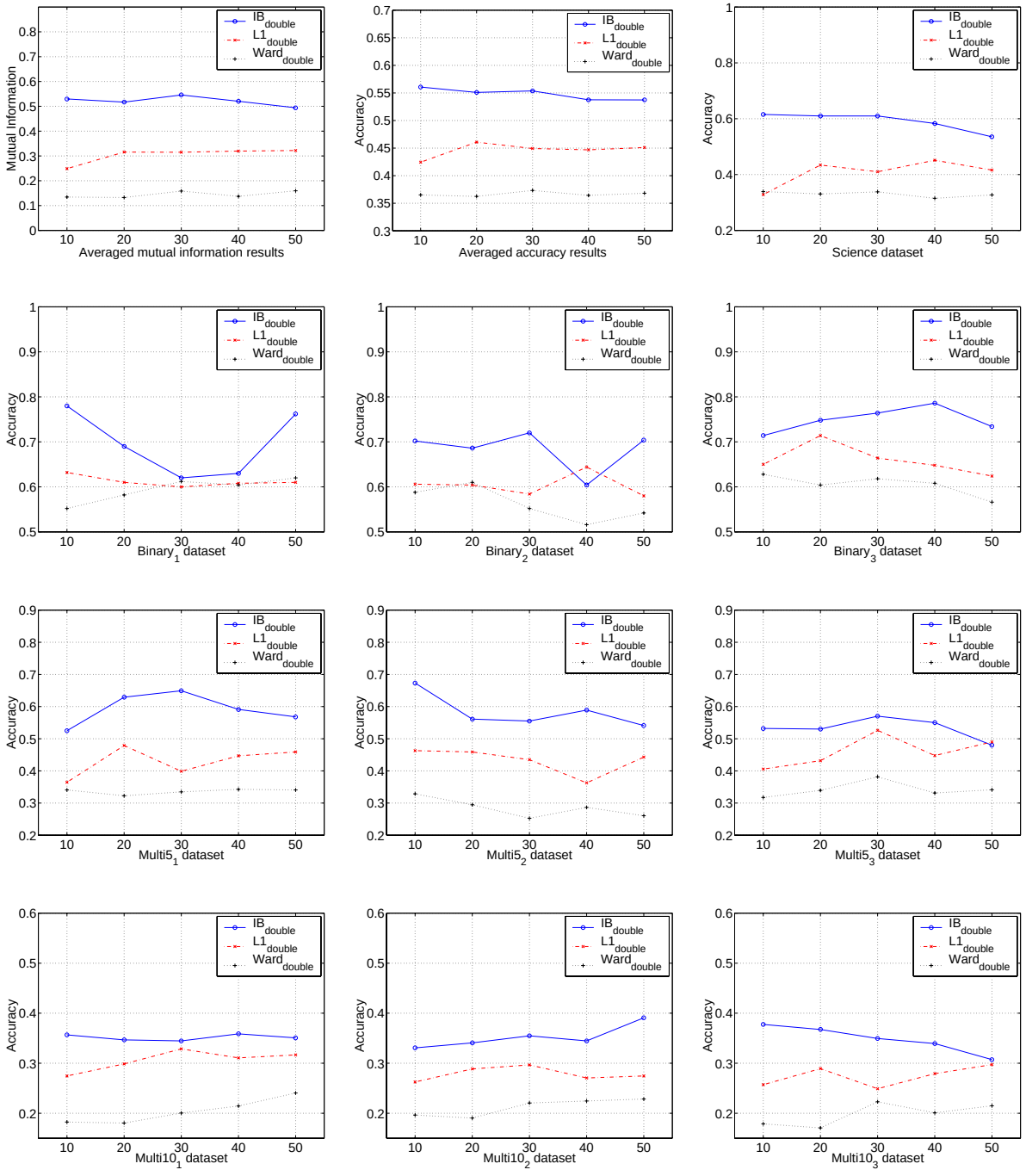

Figure 3: Results for all three double-clustering procedures over all data sets. The horizontal axis corresponds to the number of word-clusters used to cluster the documents. The top left figure presents averaged results in terms of the mutual information given in the contingency table of the document clusters and the real categories. Note the similarity with the accuracy averaged results.

References

[1] M. R. Anderberg. *Cluster Analysis for Applications*. Academic Press, 1973.

[2] L. D. Baker and A. K. McCallum. Distributional Clustering of Words for Text Classification In *ACM SIGIR 98*, 1998.

[3] P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai and R. L. Mercer. Class-based n-gram models of natual language. Computational Linguistics, 18(4), pages 467–477, 1992

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

[5] D. R. Cutting, D. R. Karger, J. O. Pedersen and J. W. Tukey. Scatter/Gother: A Cluster Based Approach to Browsing Large Document Collections. In *ACM SIGIR 92*, pages 318–329, 1992.

[6] D. R. Cutting, D. R. Karger and J. O. Pedersen. Constant Interaction-Time Scatter/Gother Browsing of Very Large Document Collections. In *ACM SIGIR 93*, pages 126–134, 1993.

[7] R. El-Yaniv, S. Fine, and N. Tishby. Agnostic classification of Markovian sequences. In *Advances in Neural Information Processing (NIPS-97)*, pages 465–471, 1997.

[8] K. Eguchi. Adaptive Cluster-based Browsing Using Incrementally Expanded Queries and Its Effects. In *ACM SIGIR 99*, pages 265–266, 1999.

[9] M. A. Hearst and J. O. Pedersen. Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. In *ACM SIGIR 96*, pages 76–84, 1996.

[10] T. Hofmann. Probabilistic Latent Semantic Indexing. In *ACM SIGIR 99*, pages 50–57, 1999.

[11] M. Iwayama and T. Tokunaga. Cluster-Based Text Categorization: A Comparison of Category Search Strategies. In *ACM SIGIR 95*, pages 273–280, 1995.

[12] K. Lang. Learning to filter netnews. In *Proc. of the 12th Int. Conf. on Machine Learning*, pages 331–339, 1995.

[13] J. Lin. Divergence Measures Based on the Shannon Entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

[14] McCallum, Andrew Kachites. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow, 1996.

[15] M. Mechkour, D. J. Harper and G. Muresan. The WebCluster Project: Using Clustering for Mediating Access to the WWW In *ACM SIGIR 98*, pages 357–358, 1998.

[16] G. Muresan, D. J. Harper and M. Mechkour. WebCluster, a Tool for Mediated Information Access. In *ACM SIGIR 99*, page 337, 1999.

[17] F. C. Pereira, N. Tishby, and L. Lee. Distributional clustering of English words. In *30th Annual Meeting of the Association for Computational Linguistics, Columbus, Ohio*, pages 183–190, 1993.

[18] D. Roussinov, K. Tolle, M. Ramsey and H. Chen. INteractive Internet Search through Automatic Clustering: an Empirical Study. In *ACM SIGIR 99*, pages 289–290, 1999.

[19] G. Salton. The SMART retrieval system. Englewood Cliffs, NJ:Prentice-Hall; 1971..

[20] G. Salton. Developments in Automatic Text Retrieval. Science, Vol. 253, pages 974–980, 1990.

[21] R. E. Schapire and Y. E. Singer. BoosTexter: A System for Multiclass Multi-label Text Categorization, 1998.

[22] H. Schutze and C. Silverstein. Projections for Efficient Documents Clustering In *ACM SIGIR 97*, pages 74–81, 1997.

[23] C. Silverstein and J. O. Pedersen. Almost-Constant-Time Clustering for Arbitrary Corpus Subsets. In *ACM SIGIR 97*, pages 60–66, 1997.

[24] N. Slonim and N. Tishby. Agglomerative Information Bottleneck. In Proc. of Neural Information Processing Systems (NIPS-99), pages 617–623, 1999.

[25] N. Slonim and N. Tishby. The Hard Clustering Limit of the Information Bottleneck Method. In preperation.

[26] N. Tishby, F.C. Pereira and W. Bialek. The Information Bottlencek Method In Proc. of the 37-th Allerton Conference on Communication and Computation, 1999.

[27] C. J. van Rijsbergen. Information Retrieval. London: Butterworths; 1979.

[28] P. Willett. Recent Trends in Hierarchic Document Clustering: A Crtical Review. Information Processing & Management, Vol. 24(5), pp. 577-597, 1988.

[29] J. Xu and W. B. Croft. Cluster-based Language Models for Distributed Retrieval. In *ACM SIGIR 99*, pages 254–261, 1999.

[30] O. Zamir and O. Etzioni. Web Document Clustering: A Feasibility Demonstration. In *ACM SIGIR 98*, pages 46–54, 1998.