Fast Supervised Dimensionality Reduction Algorithm with Applications to Document Categorization & Retrieval*

George Karypis
Department of Computer Science
Army HPC Research Center
University of Minnesota
Minneapolis, MN 55455
karypis@cs.umn.edu

Eui-Hong (Sam) Han
Department of Computer Science
Army HPC Research Center
University of Minnesota
Minneapolis, MN 55455
han@cs.umn.edu

ABSTRACT

Retrieval techniques based on dimensionality reduction, such as Latent Semantic Indexing (LSI), have been shown to improve the quality of the information being retrieved by capturing the latent meaning of the words present in the documents. Unfortunately, the high computational and memory requirements of LSI and its inability to compute an effective dimensionality reduction in a supervised setting limits its applicability. In this paper we present a fast supervised dimensionality reduction algorithm that is derived from the recently developed cluster-based unsupervised dimensionality reduction algorithms. We experimentally evaluate the quality of the lower dimensional spaces both in the context of document categorization and improvements in retrieval performance on a variety of different document collections. Our experiments show that the lower dimensional spaces computed by our algorithm consistently improve the performance of traditional algorithms such as C4.5, k-nearest-neighbor, and Support Vector Machines (SVM), by an average of 2% to 7%. Furthermore, the supervised lower dimensional space greatly improves the retrieval performance when compared to LSI.

1. INTRODUCTION

The emergence of the World-Wide-Web has led to an exponential increase in the amount of documents available electronically. At the same time, various digital libraries, news sources, and company-wide intranets provide huge collections of online documents. These developments have led to an increased interest in methods that allow users to quickly and accurately retrieve and organize these types of information.

Traditionally, information has been retrieved by literally matching terms in documents with those present in a user's query. Unfortunately, methods that are based only on lexical matching can lead to poor retrieval performance due to two effects. First, because most terms have multiple meanings, many unrelated documents may be included in the answer set just because they matched some of the query terms. Second, because the same concept can be described by multiple terms, relevant documents that do not contain any of the query terms will not be retrieved. These problems arise from the fact that the ideas in a document are more related to the concepts described in them than the words used in their description. Thus, effective retrieval methods should match the concept present in the query to the concepts present in the documents. This will allow retrieval of documents that are part of the desired concept even when they do not contain any of the query terms, and will prevent documents belonging to unrelated concepts from being retrieved even if they contain some of the query terms. This concept-centric nature of documents is also one of the reasons why the problem of document categorization (i.e., assigning a document into a pre-determined class or topic) is particularly challenging.

To address these problems, techniques based on *dimensionality reduction* have been explored for capturing the concepts present in a collection. The main idea behind these techniques is to map each document (and a query or a test document) into a lower dimensional space that can potentially take into account the dependencies between the terms. The associations present in the lower dimensional representation can then be used to improve the retrieval or categorization performance. The various dimensionality reduction techniques can be classified as either *supervised* or *unsupervised*. Supervised dimensionality reduction refers to the set of techniques that take advantage of class-membership information while computing the lower dimensional space. These techniques are primarily used for document classification and for improving the retrieval performance of pre-categorized document collections. Examples of such techniques include a variety of feature selection schemes [18, 32, 22] that reduce the dimensionality by selecting a subset of the original features, and techniques that create new fea-

*This work was supported by NSF CCR-9972519, by Army Research Office contract DA/DAAG55-98-1-0441, by the DOE ASCI program, and by Army High Performance Computing Research Center contract number DAAH04-95-C-0008. Access to computing facilities was provided by AHPCRC, Minnesota Supercomputer Institute. Related papers are available via WWW at URL: http://www.cs.umn.edu/~karypis

tures by clustering the terms [2]. On the other hand, unsupervised dimensionality reduction refers to the set of techniques that compute a lower dimensional space without using any class-membership information. These techniques are primarily used for improving the retrieval performance, and to a lesser extent for document categorization. Examples of such techniques include Principal Component Analysis (PCA) [13], Latent Semantic Indexing (LSI) [3, 8], Kohonen Self-Organizing Map (SOFM) [19] and Multi-Dimensional Scaling (MDS) [14]. In the context of document data sets, LSI is probably the most widely used of these techniques, and experiments have shown that it significantly improves the retrieval performance [3, 8] for a wide variety of document collections.

Recently, a new class of dimensionality reduction algorithms for document data sets have been developed that derive the axes of the lower dimensional space using document clustering [7, 17]. In these algorithms, the original set of documents is first clustered into k similar groups, and then for each group, the centroid vector (i.e., the vector obtained by averaging the documents in the group) is used as one of the k axes of the lower dimensional space. The key motivation behind this dimensionality reduction approach is the view that each centroid vector represents a *concept* present in the collection, and the lower dimensional representation expresses each document as a function of these concepts. This interpretation of the lower dimensional representation of each document is the reason that this dimensionality scheme is call *concept indexing* (CI) [17], and each one of the centroid vectors are called *concept vectors* [7]. Extensive theoretical analysis presented in [7] and experimental evaluation presented in [17] show that concept indexing leads to lower dimensional spaces that are similar to those obtained by LSI and lead to similar information retrieval performance. However, unlike the high computational and memory requirements of LSI, CI can compute the lower dimensional space very fast by using near linear time document clustering algorithms [7, 17, 6, 20, 1]. Experiments presented in [17] show that CI is an order of magnitude when compared to LSI and has a linear memory complexity.

The focus of this paper is to extend and experimentally evaluate concept indexing in the context of supervised dimensionality reduction. Since CI derives the axes of the lower dimensional space by using the centroid vectors of groups of similar documents, it can potentially be modified to take into account class-membership information. In this paper we explore one way of achieving this, by simply forcing the groups used to derive the axes of the lower dimensional space to only contain document of a single class. We experimentally evaluate the quality of this lower dimensional space both in the context of document categorization and improvements in retrieval performance on a variety of different document collections. Our experiments show that the lower dimensional spaces computed by CI consistently improve the performance of traditional algorithms such as C4.5 [25], k-nearest-neighbor [31, 28], and support vector machines (SVM) [15], by an average of 3% to 7%. Furthermore, the lower dimensional space greatly improves the retrieval performance when compared to LSI that cannot take class-membership information when performing the dimensionality reduction.

The reminder of this paper is organized as follows. Section 2 describes the vector-space document model used in our algorithm. Section 3 describes the concept indexing dimensionality reduction algorithm. Section 4 provides the experimental evaluation of the quality of the supervised lower dimensional space computed by concept indexing. Finally, Section 5 offers some concluding remarks and directions for future research.

2. VECTOR-SPACE MODELING OF DOCUMENTS

The documents in concept indexing are represented using the popular vector-space model [26]. In this model, each document d is considered to be a vector in the term-space. In its simplest form, each document is represented by the *term-frequency* (TF) vector $\vec{d}_{tf} = (tf_1, tf_2, \ldots, tf_n)$, where tf_i is the frequency of the ith term in the document. A widely used refinement to this model is to weight each term based on its *inverse document frequency* (IDF) in the document collection. The motivation behind this weighting is that terms appearing frequently in many documents have limited discrimination power, and for this reason they need to be de-emphasized. This is commonly done [26] by multiplying the frequency of each term i by $\log(N/df_i)$, where N is the total number of documents in the collection, and df_i is the number of documents that contain the ith term (i.e., document frequency). This leads to the *tf-idf* representation of the document, i.e., $\vec{d}_{tfidf} = (tf_1 \log(N/df_1), tf_2 \log(N/df_2), \ldots, tf_n \log(N/df_n))$. Finally, in order to account for documents of different lengths, the length of each document vector is normalized so that it is of unit length, i.e., $\|\vec{d}_{tfidf}\|_2 = 1$. In the rest of the paper, we will assume that the vector representation $\vec{d}$ of each document d has been weighted using *tf-idf* and it has been normalized so that it is of unit length.

In the vector-space model, the similarity between two documents d_i and d_j is commonly measured using the cosine function [26], given by

$$\cos(\vec{d_i}, \vec{d_j}) = \frac{\vec{d_i} \cdot \vec{d_j}}{\|\vec{d_i}\|_2 * \|\vec{d_j}\|_2}, \tag{1}$$

where "$\cdot$" denotes the dot-product of the two vectors. Since the document vectors are of unit length, the above formula is simplified to $\cos(\vec{d_i}, \vec{d_j}) = \vec{d_i} \cdot \vec{d_j}$.

Given a set S of documents and their corresponding vector representations, we define the *centroid* vector $\vec{C}$ to be

$$\vec{C} = \frac{1}{|S|} \sum_{d \in S} \vec{d}, \tag{2}$$

which is the vector obtained by averaging the weights of the various terms in the document set S. We will refer to S as the *supporting set* for the centroid $\vec{C}$. Analogously to individual documents, the similarity between a document d and a centroid vector $\vec{C}$ is computed using the cosine measure, as follows:

$$\cos(\vec{d}, \vec{C}) = \frac{\vec{d} \cdot \vec{C}}{\|\vec{d}\|_2 * \|\vec{C}\|_2} = \frac{\vec{d} \cdot \vec{C}}{\|\vec{C}\|_2}. \tag{3}$$

Note that even though the document vectors are of length one, the centroid vectors will not necessarily be of unit length.

Intuitively, this document-to-centroid similarity function tries to measure the similarity between a document and the documents belonging to the supporting set of the centroid.

A careful analysis of Equation 3 reveals that this similarity captures a number of interesting characteristics. In particular, the similarity between $\vec{d}$ and $\vec{C}$ is the ratio of the dot-product between $\vec{d}$ and $\vec{C}$, divided by the length of $\vec{C}$. If S is the supporting set for $\vec{C}$, then it can be easily shown [6, 10] that

$$\vec{d} \cdot \vec{C} = \frac{1}{|S|} \sum_{x \in S} \cos(\vec{d}, \vec{x}),$$

and that

$$\|\vec{C}\|_2 = \sqrt{\frac{1}{|S|^2} \sum_{d_i \in S} \sum_{d_j \in S} \cos(\vec{d_i}, \vec{d_j})}. \tag{4}$$

Thus, the dot-product is the average similarity between d and all other documents in S, and the length of the centroid vector is the square-root of the average pairwise similarity between the documents in S, including self-similarity. Note that because all the documents have been scaled to be of unit length, $\|\vec{C}\|_2 \leq 1$. Hence, Equation 3 measures the similarity between a document and the centroid of a set S, as the average similarity between the document and all the documents in S, amplified by a function that depends on the average pairwise similarity between the documents in S. If the average pairwise similarity is small, then the amplification is high, whereas if the average pairwise similarity is high, then the amplification is small. One of the important features of this amplification parameter is that it captures the degree of dependency between the terms in S [10]. In general, if S contains documents whose terms are positively dependent (e.g., terms frequently co-occurring together), then the average similarity between the documents in S will tend to be high, leading to a small amplification. On the other hand, as the positive term dependency between documents in S decreases, the average similarity between documents in S tends to also decrease, leading to a larger amplification. Thus, Equation 3 computes the similarity between a document and a centroid, by both taking into account the similarity between the document and the supporting set, as well as the dependencies between the terms in the supporting set.

3. CONCEPT INDEXING

As discussed in Section 1, concept indexing computes a lower dimensional space by finding groups of similar documents and using them to derive the axes of the lower dimensional space. In the rest of this section we describe the details of the CI dimensionality reduction algorithm for both an unsupervised and a supervising setting, and analyze the nature of its lower dimensional representation.

3.1 Unsupervised Dimensionality Reduction

CI computes the reduced dimensional space in the unsupervised setting as follows. If k is the number of desired dimensions, CI first computes a k-way clustering of the documents (using variants of k-means document clustering [6, 20, 1, 7, 17]), and then uses the centroid vectors of the clusters as the axes of the reduced k-dimensional space. In particular, let D be an $n \times m$ document-term matrix, (where n is the number of documents, and m is the number of distinct terms in the collection) such that the ith row of D stores the vector-space representation of the ith document (i.e., $D[i, *] = \vec{d_i}$). CI uses a clustering algorithm to partition the documents into k disjoint sets, $S_1, S_2, \ldots, S_k$. Then, for each set S_i, it computes the corresponding centroid vector $\vec{C_i}$ (as defined by Equation 2). These centroid vectors are then scaled so that they have unit length. Let $\{\vec{C'}_1, \vec{C'}_2, \ldots, \vec{C'}_k\}$ be these unit length centroid vectors. Each of these vectors form one of the axis of the reduced k-dimensional space, and the k-dimensional representation of each document is obtained by projecting it onto this space. This projection can be written in matrix notation as follows. Let C be the $m \times k$ matrix such that the ith column of C corresponds to $\vec{C'}_i$. Then, the k-dimensional representation of each document $\vec{d}$ is given by $\vec{d}C$, and the k-dimensional representation of the entire collection is given by the matrix $D_k = DC$. Similarly, the k-dimensional representation of a query $\vec{q}$ for a retrieval is given by $\vec{q}C$. Finally, the similarity between two documents in the reduced dimensional space is computed by calculating the cosine between the reduced dimensional vectors.

3.2 Supervised Dimensionality Reduction

In the case of supervised dimensionality reduction, CI can be modified to use the pre-existing clusters of documents (i.e., the classes or topics in which the documents belong to) in finding the groups of similar documents. In the simplest case, each one of these groups can correspond to one of the classes in the data set. In this case, the rank of the lower dimensional space will be identical to the number of classes in the collection. A lower dimensional space with a rank k that is greater than the number of classes, l, can be computed by using the l classes to obtain an initial l-way clustering of the documents (a cluster for each class) and then using a clustering algorithm to obtain a k-way clustering by repeatedly partitioning some of these clusters. Note that in the final k-way clustering, each one of these finer clusters will contain documents from only one class. The reverse of this approach can be used to compute a lower dimensional space that has a rank that is smaller than the number of distinct classes, by repeatedly combining some of the initial clusters using an agglomerative clustering algorithm. However, this lower dimensional space tend to lead to poor classification performance as it combines together potentially different concepts, and is not recommended. Note that once these clusters have been identified, then the algorithm proceeds to compute the lower dimensional space in the same fashion as in the unsupervised setting (Section 3.1).

3.3 Analysis & Discussion

In order to understand this dimensionality reduction scheme, it is necessary to understand two things. First, we need to understand what is encapsulated within the centroid vectors, and second, we need to understand the meaning of the reduced dimensional representation of each document. For the rest of this discussion we focus on the supervised dimensionality reduction computed by concept indexing, but similar observations can be made for the unsupervised setting.

Given a set of documents, each belonging to a different class, the centroid vector provides a mechanism to summarize their content. In particular, the prominent dimensions of the vector (i.e., terms with the highest weights), correspond to the terms that are most important within the set.

	re1									
cocoa	0.62 cocoa	0.40 buffer	0.29 icco	0.25 deleg	0.23 stock	0.18 rule	0.12 consum	0.11 council	0.10 ghana	0.09 compromis
grain	0.37 wheate	0.27 corn	0.27 tonne	0.24 grain	0.16 export	0.16 mln	0.14 soviet	0.14 usda	0.13 maize	0.13 crop
veg	0.44 palm	0.35 oil	0.25 tax	0.24 veget	0.21 ec	0.18 tonne	0.15 fate	0.13 indonesia	0.13 olein	0.12 rbd
wheat	0.52 wheate	0.28 tonne	0.17 stg	0.16 intervent	0.16 bonu	0.16 home	0.15 market	0.15 flour	0.13 barlei	0.13 fe
copper	0.72 copper	0.21 mine	0.17 ct	0.17 cent	0.17 magma	0.16 cathod	0.14 ton	0.14 lb	0.12 noranda	0.11 miner
coffee	0.67 coffee	0.26 ico	0.26 quota	0.17 bag	0.16 export	0.15 brazil	0.14 colombia	0.14 meet	0.12 ibc	0.12 produc
sugar	0.72 sugar	0.22 tonne	0.22 white	0.15 trader	0.14 intervent	0.14 ec	0.13 tender	0.12 ecu	0.12 rebat	0.11 cargoe
ship	0.33 ship	0.27 port	0.23 strike	0.20 vessel	0.20 seamen	0.14 union	0.13 cargo	0.13 tanker	0.12 gulf	0.12 worker
cotton	0.77 cotton	0.35 bale	0.14 plant	0.12 upland	0.11 weather	0.11 crop	0.10 certif	0.08 china	0.08 exchang	0.08 pct
carcass	0.46 beef	0.34 meate	0.19 iowa	0.16 slaughter	0.15 dakota	0.15 plant	0.15 pork	0.15 citi	0.15 lockout	0.14 ufcwu
crude	0.41 oil	0.24 crude	0.24 barrel	0.21 opec	0.17 bpd	0.17 dlr	0.17 mln	0.14 price	0.13 bble	0.12 energi
nat	0.59 ga	0.27 natur	0.25 feet	0.21 pipelin	0.18 cubic	0.17 butan	0.12 lt	0.11 flow	0.11 energi	0.11 co
meal	0.33 meal	0.30 fe	0.24 tonne	0.22 pellet	0.18 cake	0.18 compound	0.16 mln	0.16 guarante	0.15 credit	0.14 fish
alum	0.59 aluminium	0.28 alcan	0.27 aluminum	0.26 smelter	0.15 alumina	0.14 lme	0.12 tonne	0.12 metal	0.12 suralco	0.11 capac
oilseed	0.49 soybean	0.24 tonne	0.21 crusher	0.21 rapese	0.21 oilsee	0.16 loan	0.16 shipment	0.15 cargill	0.14 japanes	0.12 bought
gold	0.64 gold	0.38 ounce	0.25 mine	0.22 ton	0.14 coin	0.13 feet	0.12 silver	0.12 ore	0.11 assai	0.10 reserv
tin	0.63 tin	0.28 miner	0.18 atpc	0.17 itc	0.17 strike	0.16 bolivia	0.11 comibol	0.11 bolivian	0.10 paz	0.10 hunger
livestock	0.40 beef	0.35 cattle	0.29 pork	0.23 meate	0.17 dairi	0.16 lb	0.14 head	0.13 japan	0.12 bonu	0.11 nppc
iron	0.69 steel	0.19 iron	0.13 mln	0.13 industri	0.12 ore	0.12 product	0.12 coal	0.11 steelmak	0.10 tonne	0.10 plate
rubber	0.65 rubber	0.25 pact	0.24 inra	0.16 confer	0.15 consum	0.15 price	0.14 natur	0.14 xuto	0.12 agreem	0.11 adopt
zinc	0.71 zinc	0.16 pound	0.16 grade	0.15 metal	0.14 februari	0.14 januari	0.13 smelter	0.12 mint	0.12 smelt	0.11 ct
orange	0.46 orang	0.41 juice	0.31 fcoj	0.27 duti	0.26 gallon	0.21 frozen	0.17 florida	0.16 citru	0.13 brazil	0.12 depart
pet	0.31 resin	0.28 ethylen	0.25 pound	0.19 dow	0.19 chemic	0.19 plant	0.18 polypropylen	0.17 ventur	0.16 ct	0.15 petrochem
dlr	0.65 dollar	0.32 yen	0.28 bank	0.17 dealer	0.16 japan	0.16 baker	0.15 rate	0.15 currenc	0.14 interven	0.14 pari
gas	0.59 gasolin	0.23 unlead	0.17 mln	0.16 distill	0.16 tax	0.14 fuel	0.13 refin	0.13 eia	0.13 octan	0.11 compon

Table 1: The ten highest weight terms in the centroids of the classes for a subset of the Reuters-21578 text collection.

One example of such centroid vectors for a subset of the topics in the Reuters-21578 [21] text collection is shown in Table 1. For each of these vectors, Table 1 shows the ten highest weight terms[1]. The number that precedes each term in this table is the weight of that term in the centroid vector. Also note that the terms shown in this table are not the actual words, but their stems.

A number of observations can be made by looking at the terms present in the various centroids. First, looking at the weight of the various terms, we can see that for each centroid, there are relatively few terms that account for a large fraction of its length; that is, each centroid can be described by a relative small number of *keyword* terms. This is a direct consequence of the fact that the supporting sets for each centroid correspond to groups of documents belonging to the same topic, and not just random subsets of documents. Second, these terms are quite effective in providing a summary of the topics that the documents belong to, and their weights provide an indication of how central they are in these topics. This feature of centroid vectors has been used successfully in the past to build very accurate summaries [6, 20], and to improve the performance of clustering algorithms [1]. Third, the prevalent terms of the various centroids often contain terms that act as synonyms within the context of the topic they describe. This can easily be seen in most of the centroids in Table 1. For example, the terms *ship*, *vessel*, and *tanker* are all present in the centroid corresponding to the topic "ship"; similarly, the terms *aluminium*, *aluminum*, and *alumina* are all present in the centroid corresponding to the topic "alum". Note that these terms may not necessarily be present in a single document; however, such terms will easily appear in the centroid vectors if they are used interchangeably to describe the underlying topic. Fourth, looking at the various terms of the centroid vectors, we can see that the same term often appears in multiple centroids. This is because many terms have multiple meanings (*polysemy*). For example, this happens in the case of the term *oil* that appears in the centroids for the topics "veg" and "crude". The meaning of *oil* in the "veg" centroid is is that of cooking oil, whereas the meaning of *oil* in the "crude" centroid is that of fuel. To summarize, the centroid vectors provide a very effective mechanism to represent the *concepts* present in the supporting set of documents, and these vectors capture actual as well as latent associations between the terms that describe the concept.

Given a set of k centroid vectors and a document d, the ith coordinate of the reduced dimensional representation of this document is the similarity between document d and the ith centroid vector as measured by the cosine function (Equation 3). Note that this is consistent with the earlier definition (Section 3.1), in which the ith coordinate was defined as the dot-product between $\vec{d}$, and the unit-length normalized centroid vector $\vec{C'}_i$. Thus, the different dimensions of the document in the reduced space correspond to the degree at which each document matches the concepts that are encapsulated within the centroid vectors. Note that documents that are close in the original space will also tend to be close in the reduced space, as they will match the different concepts to the same degree. Moreover, because the centroids capture latent associations between the terms describing a concept, documents that are similar but are using somewhat different terms will be close in the reduced space even though they may not be close in the original space, thus improving the retrieval of relevant information. Similarly, documents that are close in the original space due to polysemous words, will be further apart in the reduced dimensional space; thus, eliminating incorrect retrievals.

4. EXPERIMENTAL RESULTS

In this section we experimentally evaluate the quality of the supervised dimensionality reduction performed by CI. Three different sets of experiments are presented. The first two sets focus on evaluating the document categorization performance achieved by traditional categorization algorithms when operating on the reduced dimensional space. We present two sets of experiments, one evaluating the multi-class categorization (where each document can belong to multiple classes) performance, and the second evaluating the single-class k-way categorization (where each document belongs to only one of k classes) performance. The third set of experiments focuses on evaluating the retrieval performance achieved in the supervised lower dimensional space. Note

[1] The centroid vectors were scaled so that they are of length one.

that in all the experiments using LSI, we used the same unit length *tf-idf* document representation used by CI.

4.1 Multi-Class Categorization Performance

We used the Reuters-21578 [21] text collection to evaluate the multi-class categorization performance achieved by the k-nearest-neighbor and SVM [15] document categorization algorithms both on the original as well as the reduced dimensional space. In particular, we used the "ModApte" split to divide the text collection into a set of 9603 training documents and 3299 test documents. After eliminating stopwords and removing terms that occur less than two times, the training corpus contains 11,001 distinct terms. Then, for each one of the 115 non-empty topic categories of the training set, we computed its concept vector by averaging the vectors of the documents in the training set that belong to that topic. Note that documents that belongs to multiple topics contribute to multiple concept vectors. These concept vectors were used as the axis of the lower dimensional space, and were used to obtained the lower dimensional representation for both the training and the test set.

Table 2 shows the *Precision/Recall Breakeven Point* achieved by the k-nearest-neighbor (kNN) and the SVM classifiers for the ten largest classes, for both the original and the reduced dimensional space. The columns labeled "kNN" and "SVM" show the performance achieved by these algorithms on the original space, whereas the columns labeled "CI-kNN" and "CI-SVM" shows the performance on the lower dimensional space. The kNN results were obtained using a distance-weighted version of the algorithm [28, 31], similar to that used by Yang and Liu [31] and $k = 30$, whereas the SVM results were obtained using the linear model offered by SVM^{light} [15]. The last row in Table 2 shows the *microaveraged* [30] *Precision/Recall Breakeven Point* over all Reuters topics.

Topic	kNN	CI-kNN	SVM	CI-SVM
earn	97.10	97.40	98.46	98.45
acq	91.00	92.60	92.89	92.35
money-fx	77.40	82.10	76.26	82.32
grain	85.40	89.20	92.66	93.83
crude	85.50	88.60	87.83	88.76
trade	74.80	81.80	76.32	80.00
interest	72.10	78.40	68.80	76.07
ship	81.30	85.60	83.79	87.20
wheat	80.30	80.00	83.33	87.14
corn	78.40	78.90	85.15	84.87
microaverage	83.13	86.10	85.15	87.62

Table 2: Precision/Recall breakeven point on the ten most frequent Reuters topics and microaveraged performance over all Reuters topics.

From the results in that table we can see that CI's lower dimensional space improves the categorization performance of both algorithms. In particular, "CI-kNN" achieves a microaverage of 86.10 that is higher than the 83.13 achieved by kNN, whereas "CI-SVM" achieves a microaverage of 87.62 which is higher than the 85.15 achieved by SVM on the original space. Note that the kNN and SVM results on the original space shown in Table 2 are comparable to those reported in [15] and are somewhat different from the results reported by [31]. We believe the difference is due to slight differences in pre-processing and the logarithmic *tf* model used in [31].

4.2 Single-Class k-way Categorization Performance

In our second set of experiments we evaluated the categorization performance of the lower dimensional representation computed by CI on a variety of documents data sets, each of which contained documents that belong to a single class. The characteristics of the various document collections used in this experiment are summarized in Table 3 [2].

The first three data sets are from the statutory collections of the legal document publishing division of West Group described in [5]. Data sets *tr11, tr12, tr21, tr31, tr41, tr45*, and *new3* are derived from TREC-5 [27], TREC-6 [27], and TREC-7 [27] collections. Data set *fbis* is from the Foreign Broadcast Information Service data of TREC-5 [27]. Data sets *la1*, and *la2* are from the Los Angeles Times data of TREC-5 [27]. The classes of the various *trXX*, *new3*, and *fbis* data sets were generated from the relevance judgment provided in these collections. The class labels of *la1* and *la2* were generated according to the name of the newspaper sections that these articles appeared, such as "Entertainment", "Financial", "Foreign", "Metro", "National", and "Sports". Data sets *re0* and *re1* are from Reuters-21578 text categorization test collection Distribution 1.0 [21]. We removed dominant classes such as "earn" and "acq" that have been shown to be relatively easy to classify. We then divided the remaining classes into 2 sets. Data sets *oh0, oh5, oh10, oh15*, and *ohscal* are from OHSUMED collection [12] subset of MEDLINE database. We took different subsets of categories to construct these data sets. Data set *wap* is from the WebACE project (WAP) [9, 4]. Each document corresponds to a web page listed in the subject hierarchy of Yahoo! [29].

To illustrate the performance improvement of supervised dimensionality reduction by CI on these data sets, we performed an experiment in which we used C4.5 [25] and k-nearest-neighbor, both on the original space, as well as on the reduced dimensional space. For each set of documents, the reduced dimensionality experiments were performed as follows. First, the entire set of documents was split into a training and test set. Next, the training set was used to find the axes of the reduced dimensional space by constructing an axis for each one of the classes[3]. Then, both the training and the test set were projected into this reduced dimensional space. Finally, in the case of C4.5, the projected training and test set were used to learn the decision tree and evaluate its accuracy, whereas in the case of kNN, the neighborhood computations were performed on the projected training and test. In our experiments, we used a value of $k = 20$ for kNN, both for the original as well as for the reduced dimensional space.

The classification accuracy of the various experiments are shown in Table 4. These results correspond to the average classification accuracies of 10 experiments, where in each experiment a randomly selected 80% fraction of the documents was used for training and the remaining 20% was used for testing. The first two columns of this table show the classification accuracy obtained by C4.5 and kNN on the original

[2] These data sets are available from *http://www.cs.umn.edu/~han/data/tmdata.tar.gz*.

[3] We also performed experiments in which the number of dimensions in the reduced space was two and three times greater than the number of classes. The overall performance of the algorithms did not change, and due to space limitations we did not include these results.

Data	Source	# of doc	# of class	min class size	max class size	avg class size	# of words
west1	West Group	500	10	39	73	50.0	977
west2	West Group	300	10	18	45	30.0	1078
west3	West Group	245	10	17	34	24.5	1035
oh0	OHSUMED-233445	1003	10	51	194	100.3	3182
oh5	OHSUMED-233445	918	10	59	149	91.8	3012
oh10	OHSUMED-233445	1050	10	52	165	105.0	3238
oh15	OHSUMED-233445	913	10	53	157	91.3	3100
ohscal	OHSUMED-233445	11162	10	709	1621	1116.2	11465
re0	Reuters-21578	1504	13	11	608	115.7	2886
re1	Reuters-21578	1657	25	10	371	66.3	3758
tr11	TREC	414	9	6	132	46.0	6429
tr12	TREC	313	8	9	93	39.1	5804
tr21	TREC	336	6	4	231	56.0	7902
tr31	TREC	927	7	2	352	132.4	10128
tr41	TREC	878	10	9	243	87.8	7454
tr45	TREC	690	10	14	160	69.0	8261
la1	TREC	3204	6	273	943	534.0	31472
la2	TREC	3075	6	248	905	512.5	31472
fbis	TREC	2463	17	38	506	144.9	2000
new3	TREC	9558	44	104	696	217.2	83487
wap	WebACE	1560	20	5	341	78.0	8460

Table 3: Summary of data sets used.

data sets. The next two columns show the classification accuracy results obtained by the same algorithms on the reduced dimensional space computed by CI. The next four columns show the classification accuracy obtained by these algorithms when used on the reduce dimensional space computed by LSI. For each algorithm, we present two sets of results, obtained on a 25- and on a 50-dimensional space. Note that these lower dimensional spaces were computed without taking into account any class information, as LSI cannot perform dimensionality reduction in a supervised setting. Finally, the last column shows the results obtained by the naive Bayesian (NB) classification algorithm in the original space. In our experiments, we used the NB implementation with the multinomial event model provided by the Rainbow [24] software library. The NB results are presented here to provide a reference point for the classification accuracies. Note that we did not use the NB algorithm in the reduced dimensional space, as NB cannot effectively handle continuous attributes [16]. Also, for each of these data sets, we highlighted the scheme that achieved the highest classification accuracy, by using a boldface font.

Looking at the results, we can see that both C4.5 and kNN, benefit greatly by the supervised dimensionality reduction computed by CI. For both schemes, the classification accuracy achieved in the reduced dimensional space is greater than the corresponding accuracy in the original space for all 21 data sets. In particular, over the entire 21 data sets, CI improves the average accuracy of C4.5 and kNN by 7%, and 6%, respectively. Comparing these results against those obtained by naive Bayesian, we can see that kNN, when applied on the reduced dimensional space, substantially outperforms naive Bayesian, which was not the case when comparing the performance of kNN in the original space. In particular, over the entire 21 data sets, the accuracy of kNN in the reduced space is 5% greater than that of naive Bayesian. Looking at the various classification results obtained by C4.5 and kNN on the lower dimensional spaces computed by LSI, we can see that the performance is mixed. In particular, comparing the best performance achieved in either one of the lower dimensional spaces over that achieved in the original space, we can see that LSI improves the results obtained by C4.5 in only four data sets, and by kNN in only ten data sets. Note also that none of the best performance achieved in either one of the lower dimensional spaces by LSI is better than the best performance achieved in the lower dimensional space by CI.

We have not included the results of C4.5 and kNN using feature selection techniques due to the inconsistent performance of such schemes in these data sets. In particular, the right number of dimensions for different data sets varies considerably. For detailed experiments showing the characteristics of feature selection schemes in text categorization, readers are advised to see [32, 11].

4.3 Query Retrieval Performance

One of the goals of dimensionality reduction techniques such as CI and LSI is to project the documents of a collection onto a low dimensional space so that similar documents (i.e., documents that are part of the same topic) come closer together, relative to documents belonging to different topics. This transformation, if successful, can lead to substantial improvements in the accuracy achieved by regular queries. The query performance is often measured by looking at the number of relevant documents present in the top-ranked returned documents. Ideally, a query should return most of the relevant documents (*recall*), and the majority of the documents returned should be relevant (*precision*). Unfortunately, a number of the larger collections in our experimental testbed did not have pre-defined queries associated with them, so we were not able to perform this type of evaluation. For this reason our evaluation was performed in terms of how effective the reduced dimensional space was in bringing closer together documents that belong to the same class.

To evaluate the extent to which a dimensionality reduction scheme is able to bring closer together similar documents, we performed the following experiment for each one of the data sets shown in Table 3. Let D be one of these datasets. For each document $d \in D$, we computed the k-nearest-neighbor sets both in the original as well as in the reduced dimensional space. Let K_d^o and K_d^r be these sets in the original and reduced space, respectively. Then, for each of these sets, we counted the number of documents that belong to the same class as d, and let n_d^o and n_d^r be

					LSI Reduced Space				
	Original Space		CI Reduced Space		C4.5		*k*NN		
	C4.5	*k*NN	C4.5	*k*NN	25 Dims	50 Dims	25 Dims	50 Dims	NB
west1	85.5%	82.9%	86.2%	**86.7%**	73.7%	74.5%	83.0%	81.4%	**86.7%**
west2	75.3%	77.2%	75.3%	**78.7%**	63.8%	59.2%	75.5%	73.8%	76.5%
west3	73.5%	76.1%	74.5%	**80.6%**	57.8%	55.3%	75.5%	77.3%	75.1%
oh0	82.8%	84.4%	87.3%	**89.8%**	74.5%	72.8%	83.9%	81.9%	89.1%
oh5	79.6%	85.6%	88.4%	**92.0%**	76.5%	76.7%	87.0%	86.8%	87.1%
oh10	73.1%	77.5%	79.6%	**82.6%**	70.9%	65.5%	79.4%	77.7%	81.2%
oh15	75.2%	81.7%	84.6%	**86.4%**	67.5%	64.9%	81.3%	80.7%	84.0%
re0	75.8%	77.9%	82.3%	**85.0%**	69.1%	64.4%	79.5%	76.3%	81.1%
re1	77.9%	78.9%	80.0%	**81.6%**	59.8%	60.6%	71.2%	75.4%	80.5%
tr11	78.2%	85.3%	87.0%	**88.9%**	79.3%	80.5%	81.3%	83.0%	85.3%
tr12	79.2%	85.7%	88.4%	**89.0%**	76.2%	72.5%	80.8%	82.7%	79.8%
tr21	81.3%	89.1%	**90.3%**	90.0%	74.6%	73.1%	87.6%	88.5%	59.6%
tr31	93.3%	93.9%	94.7%	**96.9%**	90.2%	87.5%	93.0%	92.3%	94.1%
tr41	89.6%	93.5%	95.3%	**95.9%**	89.9%	87.3%	93.4%	92.4%	94.5%
tr45	91.3%	91.1%	92.9%	**93.6%**	80.3%	80.9%	91.1%	92.1%	84.7%
la1	75.2%	82.7%	85.7%	**87.6%**	76.1%	74.2%	83.4%	82.1%	**87.6%**
la2	77.3%	84.1%	87.2%	88.6%	78.2%	76.1%	85.9%	84.7%	**89.9%**
fbis	73.6%	78.0%	81.3%	**84.1%**	59.7%	56.0%	76.4%	76.3%	77.9%
wap	68.1%	75.1%	77.5%	**82.9%**	62.3%	60.2%	74.3%	76.1%	80.6%
ohscal	71.5%	62.5%	73.5%	**77.8%**	59.4%	57.5%	70.9%	69.6%	74.6%
new3	72.7%	67.9%	73.1%	**77.2%**	41.1%	43.5%	53.9%	63.1%	74.4%

Table 4: The classification accuracy of the original and reduced dimensional data sets.

these counts. Let $N_o = \sum_{d \in D} n_d^o$, and $N_r = \sum_{d \in D} n_d^r$, be the cumulative counts over all the documents in the data set, for the original and reduced space, respectively. Given these two counts, then the performance of a dimensionality reduction scheme was evaluated by comparing N_r against N_o. In particular, if the ratio N_r/N_o is greater than one, then the reduced space was successful in bringing a larger number of similar documents closer together than they were in the original space, whereas if the ratio is less than one, then the reduced space is worse. We will refer to this ratio as the *retrieval improvement* (RI) achieved by the dimensionality reduction scheme.

The RI measures for the different classes in each one of these data sets are shown in Table 5. Note that the number of dimension in the CI-reduced space for each data set is different, and is equal to the number of classes in the data set. For the LSI results, the dimension for *new3* is 125 and the dimension for the rest of data sets is 50. The LSI results had the best performance with these dimensions [17].

As we can see from this table, the supervised dimensionality reduction computed by CI dramatically improves the retrieval performance for all the different classes in each data set and outperforms LSI in all classes. Note also that the retrieval performance of CI on the smaller classes tends to improve the most. This is because in supervised dimensionality reduction by CI, each class is equally represented as one dimension, regardless of its size. In contrast, LSI results show that the performance of the smaller classes tend to be worse than that of larger classes. This is because smaller classes contribute less to the error of the reduced ranked approximation, and the resulting lower dimensional representation fails to capture the characteristics of smaller classes.

5. CONCLUSIONS AND DIRECTIONS OF FUTURE WORK

In this paper we presented a fast supervised dimensionality reduction technique based on the recently developed cluster-based unsupervised dimensionality reduction algorithms. Our experimental evaluation shows that this lower dimensional space greatly improves the categorization performance of traditional algorithms, and leads to substantial gains in information retrieval performance in pre-categorized document collections.

The quality of the lower dimensional spaces can be further improved by using techniques that adjust the importance of the different features in a supervised setting. A variety of such techniques have been developed in the context of k-nearest-neighbor classification [28, 23, 11], all of which can be used to scale the various dimensions prior to the dimensionality reduction for computing centroid vectors and to scale the reduced dimensions for the final classification.

6. REFERENCES

[1] Charu C. Aggarwal, Stephen C. Gates, and Philip S. Yu. On the merits of building categorization systems by supervised clustering. In *Proc. of the Fifth ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, pages 352–356, 1999.

[2] L. Baker and A. McCallum. Distributional clustering of words for text classification. In *SIGIR-98*, 1998.

[3] M.W. Berry, S.T. Dumais, and G.W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37:573–595, 1995.

[4] D. Boley, M. Gini, R. Gross, E.H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Document categorization and query generation on the world wide web using WebACE. *AI Review*, 13(5-6), 1999.

[5] T. Curran and P. Thompson. Automatic categorization of statute documents. In *Proc. of the 8th ASIS SIG/CR Classification Research Workshop*, Tucson, Arizona, 1997.

[6] D.R. Cutting, J.O. Pedersen, D.R. Karger, and J.W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the ACM SIGIR*, pages pages 318–329, Copenhagen, 1992.

[7] I.S. Dhillon and D.S. Modha. Concept decomposition for large sparse text data using clustering. Technical Report Research Report RJ 10147, IBM Almadan Research Center, 1999.

[8] S.T. Dumais. Using LSI for information filtering: TREC-3 experiments. In *Proc. of the Third Text REtrieval Conference (TREC-3), National Institute of Standards and Technology*, 1995.

[9] E.H. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. WebACE: A web agent for document categorization and exploartion. In *Proc. of the 2nd International Conference on Autonomous Agents*, May 1998.

[10] E.H. Han and G. Karypis. Centroid-based document

re0		
Size	CI	LSI
608	1.12	1.01
319	1.31	1.11
219	1.28	1.12
80	1.89	1.30
60	1.26	0.99
42	2.17	1.14
39	1.30	1.14
38	1.38	0.82
37	1.66	1.16
20	1.54	1.06
16	1.60	1.00
15	1.32	0.76
11	1.64	0.73

re1		
Size	CI	LSI
371	1.25	1.05
330	1.18	1.06
137	1.51	1.24
106	1.23	1.13
99	1.11	1.04
87	1.11	1.04
60	1.44	1.14
50	1.94	0.90
48	1.05	0.99
42	2.13	1.01
37	1.59	1.22
32	1.33	1.19
31	1.67	1.23
31	1.72	1.26
27	1.84	1.30
20	2.01	1.06
20	1.41	1.27
19	1.81	0.93
19	2.18	0.80
18	1.69	0.97
18	3.67	1.09
17	1.49	0.83
15	3.75	0.98
13	1.40	0.80
10	2.27	0.43

fbis		
Size	CI	LSI
506	1.07	1.03
387	1.02	0.99
358	1.31	1.14
190	1.07	0.99
139	1.17	1.04
125	1.32	1.15
121	1.17	1.09
119	1.03	0.99
94	1.33	1.20
92	1.44	1.09
65	1.40	1.04
48	1.80	1.29
46	1.80	1.14
46	1.09	1.06
46	1.73	0.97
43	2.26	0.91
38	2.68	0.94

wap		
Size	CI	LSI
341	1.05	1.04
196	1.72	1.32
168	1.31	0.94
130	1.42	1.03
97	1.17	1.09
91	1.75	1.29
91	1.94	1.74
76	1.37	1.14
65	1.22	0.99
54	1.71	1.09
44	3.81	1.34
40	1.14	0.88
37	2.36	1.27
35	2.98	1.52
33	2.83	1.10
18	3.63	0.52
15	3.49	0.76
13	2.57	0.87
11	2.66	1.02
5	2.78	0.78

new3		
Size	CI	LSI
696	1.13	1.05
568	1.03	0.98
493	1.87	1.24
369	1.31	1.11
330	1.09	1.03
328	1.49	1.08
326	1.24	1.09
306	1.08	1.05
281	1.18	1.05
278	1.16	1.06
276	1.07	1.03
270	1.23	1.14
253	1.63	1.29
243	1.07	1.04
238	1.35	1.08
218	1.24	1.11
211	1.17	1.02
198	1.85	1.38
196	1.20	1.14
187	1.34	1.16
181	1.39	1.23
179	1.14	1.02
174	1.84	0.99
171	1.92	1.35
171	1.09	1.00
161	1.19	1.11
159	1.41	1.19
153	1.25	1.02
141	1.69	1.16
139	1.25	1.10
139	1.27	1.11
136	1.19	1.08
130	1.29	1.22
126	1.66	1.08
124	1.06	1.03
123	1.23	1.16
120	1.03	0.97
116	1.53	0.92
115	1.18	1.03
110	1.18	1.08
110	1.11	1.07
106	1.04	1.02
105	1.28	1.16
104	2.54	1.17

la1		
Size	CI	LSI
943	1.33	1.12
738	1.11	1.07
555	1.21	1.11
354	1.34	1.25
341	1.41	1.14
273	2.22	1.08

la2		
Size	CI	LSI
905	1.31	1.13
759	1.10	1.06
487	1.25	1.13
375	1.20	1.15
301	1.48	1.14
248	1.75	1.09

ohscal		
Size	CI	LSI
1621	1.38	1.24
1450	1.56	1.37
1297	1.37	1.19
1260	1.46	1.29
1159	1.63	1.41
1037	1.81	1.39
1001	1.85	1.53
864	1.47	1.33
764	1.78	1.35
709	1.51	1.28

Table 5: The per-class RI measures for various data sets for supervised dimensionality reduction. The first column shows the number of documents in each class.

classification algorithms: Analysis & experimental results. Technical Report TR-00-017, Department of Computer Science, University of Minnesota, Minneapolis, 2000. Available on the WWW at URL *http://www.cs.umn.edu/~karypis*.

[11] Eui-Hong Han. *Text Categorization Using Weight Adjusted* k*-Nearest Neighbor Classification*. PhD thesis, University of Minnesota, October 1999.

[12] W. Hersh, C. Buckley, T.J. Leone, and D. Hickam. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In *SIGIR-94*, pages 192–201, 1994.

[13] J. E. Jackson. *A User's Guide To Principal Components*. John Wiley & Sons, 1991.

[14] A.K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[15] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proc. of the European Conference on Machine Learning*, 1998.

[16] G.H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proc. of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence*, pages 338–345, 1995.

[17] G. Karypis and E.H. Han. Concept indexing: A fast dimensionality reduction algorithm with applications to document retrieval & categorization. Technical Report TR-00-016, Department of Computer Science, University of Minnesota, Minneapolis, 2000. Available on the WWW at URL *http://www.cs.umn.edu/~karypis*.

[18] R. Kohavi and D. Sommerfield. Feature subset selection using the wrapper method: Overfitting and dynamic search space topology. In *Proc. of the First Int'l Conference on Knowledge Discovery and Data Mining*, pages 192–197, Montreal, Quebec, 1995.

[19] T. Kohonen. *Self-Organization and Associated Memory*. Springer-Verlag, 1988.

[20] Bjornar Larsen and Chinatsu Aone. Fast and effective text mining using linear-time document clustering. In *Proc. of the Fifth ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, pages 16–22, 1999.

[21] D. D. Lewis. Reuters-21578 text categorization test collection distribution 1.0. *http://www.research.att.com/∼lewis*, 1999.

[22] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, 1998.

[23] D.G. Lowe. Similarity metric learning for a variable-kernel classifier. *Neural Computation*, pages 72–85, January 1995.

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

[25] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[26] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[27] TREC. Text REtrieval conference. *http://trec.nist.gov*.

[28] D. Wettschereck, D.W. Aha, and T. Mohri. A review and empirical evaluation of feature-weighting methods for a class of lazy learning algorithms. *AI Review*, 11, 1997.

[29] Yahoo! Yahoo! http://www.yahoo.com.

[30] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval Journal*, May 1999.

[31] Y. Yang and X. Liu. A re-examination of text categorization methods. In *SIGIR-99*, 1999.

[32] Y. Yang and J. Pederson. A comparative study on feature selection in text categorization. In *Proc. of the Fourteenth International Conference on Machine Learning*, 1997.