Feature Reduction for Document Clustering and Classification

submitted to SIGIR 2000

Abstract

Often users receive search results which contain a wide range of documents, only some of which are relevant to their information needs. To address this problem, ever more systems not only locate information for users, but also organize that information on their behalf. We look at two main automatic approaches to information organization: interactive clustering of search results and pre-categorizing documents to provide hierarchical browsing structures. To be feasible in real world applications, both of these approaches require accurate yet efficient algorithms. Yet, both suffer from the curse of dimensionality — documents are typically represented by hundreds or thousands of words (features) which must be analyzed and processed during clustering or classification. In this paper, we discuss feature reduction techniques and their application to document clustering and classification, showing that feature reduction improves efficiency as well as accuracy. We validate these algorithms using human relevance assignments and categorization.

Keywords:

text clustering, text categorization, navigation vs ad hoc search, automated presentation of information

1 Introduction

Search Engines are a common gateway to huge document collections, be it the World Wide Web (WWW) or a collection of abstracts in an electronic book shop. It has been recognized that one limiting factor of search engine technology is the low precision of the results returned. It is not uncommon to get thousands or even millions of matches for a query such as "computer games". Even sophisticated ranking algorithms cannot know whether the user wants to browse documents on the latest advance in technology in this area or rather on entertaining products.

There are three basic approaches to this problem: 1) interactively clustering documents to allow users to preview a large collection more efficiently; 2) organizing document collections into a set of categories to allow users to navigate in a structured way; 3) learning more about the user's interests and/or task to allow the system to better identify the relevant documents for this user and/or task. For the first approach, we believe that cluster analysis of the documents returned by a query system provides a way to confront a user with different clusters/types of documents. Each cluster would either contain a high or a very low concentration of documents relevant to the user in accordance with the commonly accepted *cluster hypothesis* (van Rijsbergen 1979) for query results. This would allow users to quickly weed out whole clusters of irrelevant documents. In Section 3 we discuss three technical questions that arise in this context: 1) how to represent documents to avoid the curse of dimensionality; 2) how to efficiently (in roughly one second of CPU time) cluster 1000s of documents; and 3) how to validate the cluster hypothesis with the chosen representation and clustering algorithm. A visual-navigation search engine based on this work has been successfully implemented and is described elsewhere (Sewraz 1999).

As evidenced by the success of Yahoo and many other Internet index sites, browsing hierarchies (the second approach) are extremely popular methods to navigate the WWW. However, many of these indices are created manually, a time-consuming and expensive procedure. In Section 4 we will discuss the effectiveness of an automatic text categorization algorithm and the effects of dimensionality reduction on its accuracy.

Finally, the system could learn about the user's interests and attempt to use this information to expand the query and/or post-process the search results. In addition to technical barriers, this approach requires users to allow their Web surfing behavior to be monitored and that information to be shared with a search engine, which will be a social barrier for many users. As it is not directly related to dimensionality reduction, this approach will not be further discussed in this paper, but a discussion can be found in (Pretschner and Gauch 1999).

2 The Curse of Dimensionality and Feature Reduction

The natural features of text documents are words or phrases, and a document collection can contain millions of different features. Even after applying standard feature reduction techniques, the number of features remains large: in our clustering experiments with 528,155 US-American newspaper articles, we only kept nouns based on Brill's tagger (Brill 1994) with a medium document frequency: the noun had to appear in least three documents and in no more that 33% of all documents. Additionally, a list of stop-words was used to eliminate obvious function words of the language. This resulted in a vocabulary of around 280,000 so-called *potentially interesting words*. In our system we store a set of around 100 potentially interesting words per document along with the metadata of the document at index time. A set H of documents returned by a query may still have a potentially-interesting-words vocabulary of 10,000s of different words. Consequently, the often-used word histogram representation of documents leads to high-dimensional vectors.

The problem with this kind of representations is that any two randomly picked vectors in a high-dimensional hypercube tend to have a constant distance from each other, no matter what the measure is! As an example, let $x, y \in [0,1]^n$ be drawn independently from a uniform distribution. The expectation value of their sum-norm distance is $n/3$ with a variance of $n/18$. For $n = 1,800$ (corresponding to a joint vocabulary of just 1,800 words for a word histogram representation) this means a typical distance of $|x - y|_1 = 600 \pm 10$. With increasing n the ratio between standard deviation and vector size gets ever smaller, as it scales with $1/\sqrt{n}$. This is a generic statistical property of high-dimensional spaces with any standard distance measure and can be traced down to the law of large numbers.

Although word histogram document representations are by no means random vectors, each additional dimension tends to not only spread the size of a cluster but also dilute the distance of two previously well-separated clusters. Hence, it seems prohibitive involving all semantic features (e.g., the words) of a

document collection for document clustering and classification.

3 Clustering

Document clustering has attracted much interest in the recent decades, eg (Salton 1968; Croft 1978; Voorhees 1985; Rasmussen 1992), and much is known about the importance of feature reduction in general, eg (Krishnaiah and Kanal 1982) and, in particular, clustering (van Rijsbergen 1979), but little has been done so far to facilitate feature reduction for document clustering of query results.

3.1 Feature Reduction for Query Result Clustering

Let D be a set of documents and H the subset that is matched by a particular query. We suggest ranking the importance of each such word j with a weight

$$w_j = \frac{h_j}{d_j} \cdot h_j \log(|H|/h_j),$$

where h_j is the number of documents in H containing the word j, and d_j is the number of documents in the whole document collection D containing j. The second factor prefers medium matched-document frequency h_j, while the first factor prefers words that specifically occur in the matched documents. The highest-ranked words are meant to be related to the query. Indeed, we have "hardware", "software", "IBM" etc as the top-ranked words when querying for "computer". This seems to be a powerful approach to restrict the features of the matched documents to the top k ranked words, which we will call the *related words*. One important aspect is that the features are computed at query time. Hence, when the query "computer" is refined to "computer hardware", a completely new set of features would emerge automatically.

We represent each matched document i as a k-dimensional vector v_i, where the j-th component v_{ij} is a function of the number of occurrences t_{ij} of the j-th ranked related word in the document i:

$$v_{ij} = \log(1 + t_{ij}) \cdot \log(|D|/d_j)$$

This is a variation of the tf-idf weight that stresses the term frequency less. We project the vector v_i onto the k-dimensional unit sphere obtaining a normalized vector u_i that represents the document i. We deem the Euclidean distance between u_a and u_b a sensible *semantic distance* between two documents a and b in the document subset H returned by a query with respect to the complete document collection D. Note that the Euclidean distance is related to the cosine similarity measure owing to the process of normalization.

3.2 Clustering Algorithm

Post-retrieval document clustering has been well studied in the recent years, eg (Cutting, Karger, Pedersen and Tukey 1992; Allen, Obry and Littman 1993; Leouski and Croft 1996; Zamir and Etzioni 1998).

We deploy a variant of the Buckshot algorithm (Cutting, Karger, Pedersen and Tukey 1992), a two-phase process to cluster the document representations u_i in an Euclidean k-dim space. Each cluster contains a certain number of document vectors and is represented by their normalized arithmetic mean, the so-called centroid vector. In the first phase, hierarchical clustering with complete, single or average linkage operates on the best-ranked 150 documents (in contrast to the original Buckshot method, which would use a random document sample). This can be done in a fraction of one second CPU time. Hierarchical clustering has the advantage that one can either prescribe the number N of clusters or let the number of clusters determine by demanding a certain minimum similarity within a cluster. Either way, once clusters within the top-ranked documents are identified, their N centroids can be computed and used as a seed for standard iterative clustering of the remaining documents. This algorithm consumes an amount of time linear in the number $|D|$ of documents and in the number N of clusters. In our experience, one cycle of iterative clustering is not only adequate but also preserves the cluster structure given by the top-ranked documents, which are thought to be the most important ones. 1,000s of documents can thus be clustered in less than a second.

linkage	dim 2	6	10	14	20	28	36	64	96	128	256
complete	0.15	**0.29**	**0.35**	**0.29**	0.27	0.27	0.28	0.28	0.24	0.26	0.28
average	0.15	0.28	0.32	0.31	0.25	0.22	0.24	0.20	0.22	0.18	0.17
single	0.15	0.27	0.32	0.28	0.27	0.28	0.24	0.21	0.13	0.12	0.09

Table 1: Average clustering qualities over 46 queries

3.3 Validation

Any clustering method — even random assignment — leads to a partition of the documents. We propose a method to assess the quality of the clustering process based on human-expert data. We have used the 1997-1998 collection of the TREC data (Voorhees and Harman 1999) with 528,155 documents, mainly newspaper articles, 100 queries and corresponding relevance assessments. We ignored queries that contained fewer than 40 relevant documents and divided the remaining 61 queries randomly into test data (15) and training data (46). For each of the training queries we would run the query in a standard search engine and partition the set H of 1000 best-ranked documents into six clusters $H_1, \ldots, H_6$ according to above scheme. A certain proportion p_* of the documents in H would be relevant according to the human relevance assessments. A clustering, where the proportions $p_1, \ldots, p_6$ of relevant documents in each cluster, respectively, were either 0 or 1 would be ideal. Contrary, a clustering where $p_1 = \ldots = p_6 = p_*$, could not assist a human user at all to quickly weed out big sets of irrelevant documents (see Figure 1). We came up with a quality measure that assigns 1 in the best case and 0 in the worst case and linearly interpolates between these cases. Averaging over all available training queries gives insight into the typical behavior of a clustering routine in the context of a particular search engine.

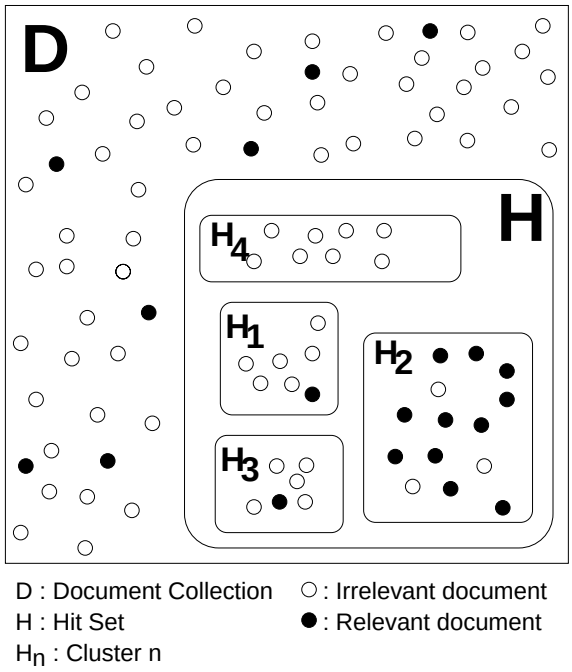

Figure 1: The document collection after searching and clustering

3.4 Experimental Results and Discussion

The experimental procedure was applied to queries with different dimensions for the document representation and three linkage methods each (single, average and complete) for the seeding of the iterative-clustering centers. For each of the combinations, we computed a cluster quality as described above and

averaged this quality over the training queries, see Table 1.

Our studies confirmed that the average cluster quality is significantly higher than random cluster assignments, which supports the cluster hypothesis. We were also able to tune the parameters of the clustering procedure, most notably the number k of features used in the document representation. Our preliminary findings indicate that $k \approx 10$ is sufficient for good clustering results, and that complete linkage for the hierarchical clustering seeds seems to outperform the other linkage methods. All our findings have been confirmed on the test queries.

4 Classification

Browsing is another method to find information. Several search engines provide subject hierarchies that can be browsed, but the associated Web pages are manually placed in the categories, e.g., Infoseek's Ultraseek (Infoseek 1998) which limits the amount of information available. In contrast, we build a browsing structure automatically by categorizing documents into concepts in a pre-defined ontology, or subject hierarchy. In one application, all the pages for a given Web site are categorized into concepts in a standard ontology to produce a site map for that site (Zhu, Gauch, Gerhard, Kral and Pretschner 1999). A site map represents the information space of a site by either relating a group of Web pages to a particular subject or depicting the links specified in the Web pages. To generate a subject-based site map, one must describe a subject space, which typically has a hierarchical structure, and then associate documents with the appropriate subjects. Other automatic approaches may employ neural networks (Lin 1995) document clustering (Cartia 1998) or the links among the Web pages (Maarek, Ben-Shaul, Jacovi, Shtalhaim, Ur and Zernik 1997)

Our classification approach uses a pre-existing ontology (in our case, a browsing hierarchy used by a prominent Web site) as a reference ontology. Each Web page from the local site to be characterized is automatically classified into the best matching concept in the reference ontology. The documents that have been manually attached to each concept in the reference ontology are used as training data for categorizing the new documents. We use a vector space approach, calculating the cosine similarity measure to identify the closest match between a vector representing the Web page and the vectors representing the concepts in the reference ontology. All the training documents associated with a given concept are concatenated to form a superdocument. These superdocuments are then indexed using a vector space retrieval engine. Each Web page to be categorized is then treated as a query and the retrieval engine returns the top matching superdocuments for that "query" (i.e., the top matching categories for that Web page). Each document is attached to only the top matching concept in the ontology. The weights of all the document matches to a particular concept are then accumulated to determine the weight of that concept in the site being studied.

To validate the quality of our categorization, the next section discusses series of experiments that compare the performance of the vector space categorization approach with one of the top classifiers, Naïve-Bayes (Friedman, Geiger and Goldszmidt 1997).

4.1 Validation

We compared the performance of our approach with a Naïve-Bayes implementation from Carnegie-Mellon University (McCallum 1998), which has been found to be among the most effective classifiers (Friedman and Goldszmidt 1996). In the Naïve-Bayes approach (Langley, Iba and Thompson 1992), the conditional probability of each attribute value given a particular category is determined as well as the probability of the category appearing. This probabilistic information is typically established through a learning procedure, which takes in a new training instance each time and adjusts the probabilistic information of the corresponding category.

For the experiments, we selected an ontology containing 1,274 concepts and spidered 10 associated Web pages for each concept. We evaluated the accuracy of classifiers with a variable numbers of Web pages (one through eight per concept) in the training set, using the remaining two Web pages per concept as the test set.

4.2 Experimental Results and Discussion

Table 2 shows the mean distance between the concept chosen by the classifiers and the concept with which the test document was originally associated. We defined the distance between two concepts as the length of the path between them. For example, the distance between a parent and its child in the ontology

	Vector Space		Naïve Bayes	
training pages	mean dist	# of matches	mean dist	# of matches
1	6.07	127.0	7.07	32.0
2	5.69	202.5	6.87	61.0
3	5.11	289.0	6.61	95.5
4	4.69	355.0	6.43	127.0
5	4.42	407.5	6.29	149.0
6	4.27	434.5	6.11	184.0
7	4.32	423.5	6.09	188.0
8	4.39	415.0	5.96	184.0
Average	4.87	331.75	6.43	127.56

Table 2: Mean distance and number of matches per test page for the vector space and Naïve-Bayes approaches

would be one and the distance between siblings would be two. Since the ontology had a maximum depth of 11, the maximum possible distance between the assigned and target concepts would be 22. Table 2 also shows the number of exact matches, where the classifier chose the same category with which the Web page was originally associated. The results in Table 2 are based on classifying two test documents per concept (ie, 2,548 test documents) when variable numbers of pages per concept were used for training. The statistical t-test analysis shows that the mean distance produced by the vector space approach is significantly smaller than that for the Naïve-Bayes approach (t-statistics $= 5.56 > t(14)$ for $p = 0.001$). The number of matches for the vector space is also significantly larger than that for the Naïve-Bayes approach (t-statistics $= 4.45 > t(14)$ for $p = 0.001$).

Examining Table 2 in more detail, we can see the effect of the amount of training data on the classification accuracy, which shows that, for the Vector Space method, the highest accuracy occurs with six training pages per concept. Adding more pages per concept degrades it slightly. Figure 2 illustrates the classifiers' performance in more detail, showing the histogram of distances when the number of training documents was six.

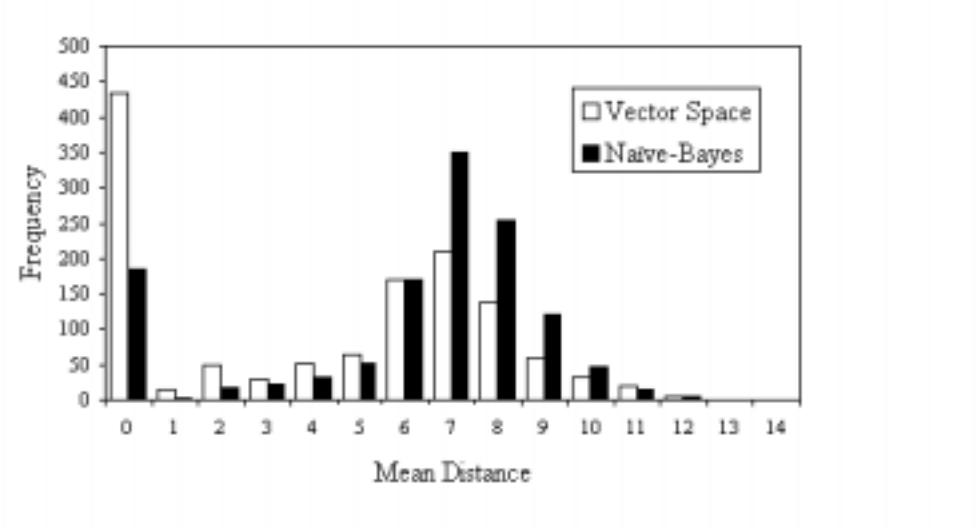

Figure 2: The histogram of distances between assigned concept and the correct concept (number of training pages - 6) for the vector space approach and the naïve-Bayes approach

In addition to varying the number of training pages, we also investigated the effect of using only a subset of words rather than the complete test documents as the input to the vector space classifier. The words were weighted using $\mathrm{tf} \cdot \mathrm{idf}$ (where tf was the frequency of the term in the test document and idf was the inverse document frequency calculated over all available training documents) and we selected the top-weighted words. Figure 3 shows the results when we varied the number of words selected. The data

show that the mean distance decreased as the number of selected words increased, but little improvement is seen beyond 40 words selected. These results confirm the power of dimensionality reduction in two ways. First, our best result was found with only six training pages, adding more pages (and more words per concept) did not improve the classification. Similarly, representing pages to be classified by more than the 40 most important words (features) did not appreciably improve the classification.

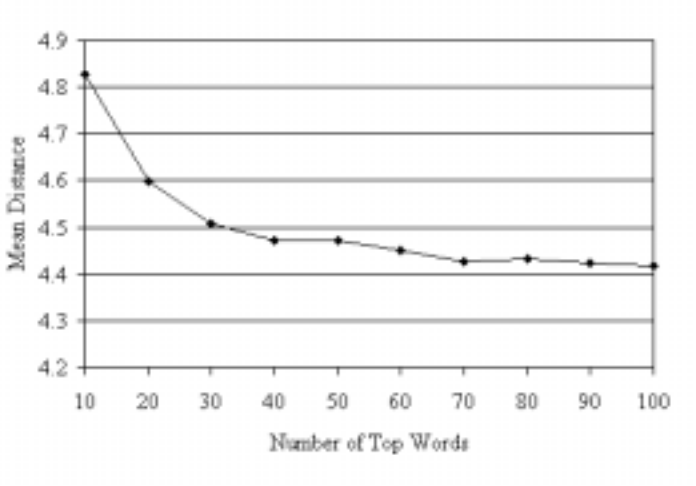

Figure 3: Mean distance as a function of the number of top words in test documents

In summary, the data from the experiments show that the vector space approach was more accurate than the Naïve-Bayes approach, and that a modest number of training documents (approximately six per concept) are sufficient for good performance. The results also show that a small number of the most important words for a document convey enough information for classification. Overall, the site map generated by the our Local Classification Agent (LCA) seems to be quite accurate for an entirely automatic approach.

5 Conclusions and Further Work

We have presented a method of assessing the quality of a document clustering procedure using human relevance assessments. We have demonstrated that this measure and the respective measure for the quality of a document classification procedure show the effects of the curse of dimensionality in accordance to a well-known statistical effect. We used these data to reduce the number of features used to represent concepts (for classification) or documents in a collection subset (for clustering), respectively. Not only do the reduced feature representations reduce computing time but they also show a better discriminatory behavior. Owing to the generic nature of the curse of dimensionality it has to be assumed that our feature reduction techniques are likely to improve the relevance and speed of any document clustering or classification algorithm which is based on feature vector representations.

Our work focused on finding a typical optimal number of features for clustering and classification purposes, yet it can be expected that individual document subsets or concepts each require a different number of features. It is left for further studies to fine-tune the average number of features (generated with the above methods) using properties of the particular concept or document set without reference to training data or human assessments.

References

Allen, R. B., P. Obry and M. Littman (1993). An interface for navigating clustered document sets returned by queries. In *Proceedings of the ACM Conference on Organizational Computing Systems*, pp. 166–171.

Brill, E. (1994). Some advances in rule-based part of speech tagging. In *AAAI*.

Cartia (1998). *ThemeScape Technology Overview*. http://www.cartia.com/products/

techoverview.html.

Croft, W. B. (1978). *Organizing and searching large files of documents*. Ph. D. thesis, University of Cambridge.

Cutting, D. R., D. R. Karger, J. O. Pedersen and J. W. Tukey (1992). Scatter/gather: a cluster-based approach to browsing large document collections. In *Proceedings of the 15th International ACM SIGIR Conference*, pp. 318–329.

Friedman, N., D. Geiger and M. Goldszmidt (1997). Bayesian network classifiers. *Machine Learning 29*, 131–163.

Friedman, N. and M. Goldszmidt (1996). Building classifiers using Bayesian networks. In *Thirteenth National Conference on Artificial Intelligence (AAAI)*.

Infoseek (1998). *Home Page*. http://software.infoseek.com/products/products.htm.

Krishnaiah, P. R. and L. N. Kanal (Eds) (1982). *Handbook of Statistics: Classification, Pattern Recognition and Reduction of Dimensionality*, Volume 2. North-Holland Publishing Company.

Langley, P., W. Iba and K. Thompson (1992). An analysis of Bayesian classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, San Jose. AAAI Press.

Leouski, A. V. and W. B. Croft (1996). An evaluation of techniques for clustering search results. Technical Report IR-76, Department of Computer Science, University of Massachusetts, Amherst.

Lin, X. (1995). Searching and browsing on map displays. In *Proceedings of the 58th ASIS Annual Meeting (ASIS '95)*.

Maarek, Y., I. Ben-Shaul, M. Jacovi, M. Shtalhaim, S. Ur and D. Zernik (1997). Webcutter: A system for dynamic and tailorable site mapping. In *The 6th International World Wide Web Conference*.

McCallum, A. (1998). *A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering*. http://www.cs.cmu.edu/~mccallum/bow/.

Pretschner, A. and S. Gauch (1999). Ontology based personalized search. In *Proc of the 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '99)*, Chicago, IL, pp. 391–398.

Rasmussen, E. (1992). Clustering algorithms. In W. B. Frakes and R. Baeza-Yates (Eds), *Information Retrieval: Data Structures and Algorithms*, pp. 419–442. Prentice Hall.

van Rijsbergen, C. J. (1979). *Information Retrieval* (2nd ed). London: Butterworth.

Salton, G. (1968). *Automatic information organization and retrieval*. New York: McGraw-Hill.

Sewraz, S. (1999). *A Visual Information-Retrieval Navigator*. MSc Thesis, Imperial College.

Voorhees, E. (1985). The cluster hypothesis revisited. In *Proceedings of ACM SIGIR*, pp. 188–196.

Voorhees, E. M. and D. K. Harman (1999). *Information Technology: The Seventh Text REtrieval Conference (TREC-7)*. NIST. http://trec.nist.gov.

Zamir, O. and O. Etzioni (1998). Web document clustering: A feasibility demonstration. In *Proceedings of the 21th International ACM SIGIR Conference*, pp. 46–54.

Zhu, X., S. Gauch, L. Gerhard, N. Kral and A. Pretschner (November 1999). Ontology-based web site mapping for information exploration. In *Proc of the 8th International Conference on Information and Knowledge Management (CIKM '99)*, Kansas City, MO, pp. 188–194.