

Hierarchical Document Classification Using Automatically Generated Hierarchy

Tao Li*

Shenghuo Zhu †

Abstract

Automated text categorization has witnessed a booming interest with the exponential growth of information and the ever-increasing needs for organizations. The underlying hierarchical structure identifies the relationships of dependence between different categories and provides valuable sources of information for categorization. Although considerable research has been conducted in the field of hierarchical document categorization, little has been done on automatic generation of topic hierarchies. In this paper, we propose the method of using linear discriminant projection to generate more meaningful intermediate levels of hierarchies in large flat sets of classes. The linear discriminant projection approach first transforms all documents onto a low-dimensional space and then clusters the categories into hierarchies accordingly. The paper also investigates the effect of using generated hierarchical structure for text classification. Our experiments show that generated hierarchies improve classification performance in most cases. A preliminary short version of the paper has appeared in [8].

1 Introduction

Many studies in document classification focus on *flat classification*, in which the predefined categories are treated individually and equally so that no structures exist to define relationships among them [10, 1]. Limitations to the flat classification approach exists in the fact that, as the Internet grows, the number of possible categories increases and the borderlines between document classes are blurred. To resolve this issue, recently several researchers have studied the use of hierarchies for text classification and obtained promising results [6, 1, 9]. However, the previous studies were mostly conducted on corpora with predefined hierarchical structures and little has been done on automatic generation of topic hierarchies.

This motivates us to address the issue of automatically building hierarchies of documents. Such studies are meaningful for the following reasons: First, manual building of hierarchies is an expensive task since the process requires domain experts to evaluate the relevance of documents to the topics. Second, there may exist document domains in which there are no natural hierarchies and even domain experts have difficulties in evaluating the semantics. Third, automatic hierarchy generation based upon document statistics may generate hierarchies that provide better statistical correlations among categories. Once such a statistically more significant hierarchy has been build, the hierarchy can be incor-

porated into various classification methods to help achieve better performance.

2 Linear Discriminant Projection Approach

The first step in hierarchy generation is to define the similarity measure between categories upon which hierarchies are built. In this section, we present the linear discriminant projection approach for inferring class relationships. Its core idea is to compare the class representatives in a low-dimensional space so that the comparison is more “meaningful”. More specifically, after finding the transformation, the similarity between classes is defined to be the distance between their centroids in the transformed spaces. The notations used through the discussion of this paper are listed in the Table 1.

Notations	Descriptions
A	document-term matrix
n	number of data points, i.e., documents
N	number of the dimensions, i.e, terms
k	number of class
S_i	covariance matrix of the i -th class
S_b	between-class scatter matrix
S_w	within-class scatter matrix
G	reduction transformation
m_i	centroid of the i -th class
m	global centroid of the training set

Table 1: Notations

2.1 Finding the Transformation Given a document-term matrix $A = (a_{ij}) \in \mathfrak{R}^{n \times N}$, where each row corresponds to a document and each column corresponds to a particular term, we consider finding a linear transformation $G \in \mathfrak{R}^{N \times \ell}$ ($\ell < N$) that maps each row a_i ($1 \leq i \leq n$) of A in the N -dimensional space to a row y_i in the ℓ -dimensional space. The resulting data matrix $A^L = AG \in \mathfrak{R}^{n \times \ell}$ contains ℓ columns, i.e. there are ℓ features for each document in the reduced (transformed) space. It is also clear that the features in the reduced space are linear combinations of the features in the original high dimensional space, where the coefficients of the linear combinations depend on the transformation G . Linear discriminant projection tries to compute the optimal transformation matrix G such that the

*School of Computer Science, Florida International University, taoli@cs.fiu.edu .

†NEC Labs America, Inc., zsh@sv.nec-labs.com. Major work was completed when the author was in University of Rochester.

class structure is preserved. More details are given below.

Assume there are k classes in the data set. Suppose m_i , S_i , P_i are the mean vector, covariance matrix, and a prior probability of the i -th class, respectively, and m is the total mean. For the covariance matrix S_i for the i th class, we can decompose it as $S_i = X_i X_i^T$, where X_i has the same number of columns as the number of data points in the i -th class. Define the matrices

$$H_b = [\sqrt{P_1}(m_1 - m), \dots, \sqrt{P_k}(m_k - m)] \in \mathfrak{R}^{N \times k},$$

$$H_w = [\sqrt{P_1}X_1, \dots, \sqrt{P_k}X_k] \in \mathfrak{R}^{N \times n}.$$

Then the between-class scatter matrix S_b , the within-class scatter matrix S_w , and the total scatter matrix S_t are defined as follows [3]:

$$S_b = \sum_{i=1}^k P_i (m_i - m)(m_i - m)^T = H_b H_b^T,$$

$$S_w = \sum_{i=1}^k P_i S_i = H_w H_w^T.$$

In the lower-dimensional space resulting from the linear transformation G , the within-cluster and between-cluster matrices become

$$S_w^L = (G^T H_w)(G^T H_w)^T = G^T S_w G,$$

$$S_b^L = (G^T H_b)(G^T H_b)^T = G^T S_b G.$$

An optimal transformation G would maximize $\text{Trace}(S_b^L)$ and minimize $\text{Trace}(S_w^L)$. A common optimization for computing optimal G is

$$G^* = \arg \max_G \text{Trace} \left((G^T S_w G)^{-1} G^T S_b G \right).$$

The solution can be readily obtained by solving a eigenvalue decomposition problem on $S_w^{-1} S_b$, provided that the within-class scatter matrix S_w is nonsingular. Since the rank of the between-class scatter matrix is bounded above by $k-1$, there are at most $k-1$ discriminant vectors.

2.2 Extension on General Cases In general, the within-class scatter matrix S_w may be singular especially for document-term matrix where the dimension is very high. A common way to deal with it is to use generalized eigenvalue decomposition [4, 7]

Let $K = [H_b \ H_w]^T$, which is a $k+n$ by N matrix. By the generalized singular value decomposition, there exist orthogonal matrices $U \in \mathfrak{R}^{k \times k}$, $V \in \mathfrak{R}^{n \times n}$, and a nonsingular matrix $X \in \mathfrak{R}^{N \times N}$, such that

$$(2.1) \quad \begin{bmatrix} U^T & 0 \\ 0 & V^T \end{bmatrix} K X = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \\ \Sigma_2 & 0 \\ 0 & 0 \end{bmatrix},$$

where

$$\Sigma_1 = \text{diag}(\overbrace{1, \dots, 1}^r, \alpha_1, \dots, \alpha_s, \overbrace{0, \dots, 0}^{t-r-s}),$$

$$\Sigma_2 = \text{diag}(\overbrace{0, \dots, 0}^r, \beta_1, \dots, \beta_s, \overbrace{1, \dots, 1}^{t-r-s}),$$

$$t = \text{rank}(K), \quad r = t - \text{rank}(H_w^T),$$

$$s = \text{rank}(H_b) + \text{rank}(H_w) - t,$$

satisfying

$$1 > \alpha_1 \geq \dots \geq \alpha_s > 0,$$

$$0 < \beta_1 \leq \dots \leq \beta_s < 1,$$

and $\alpha_i^2 + \beta_i^2 = 1$ for $i = 1, \dots, s$.

From Eq. (2.1), we have

$$(X^T H_b)(X^T H_b)^T = \begin{bmatrix} \Sigma_1^T \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix},$$

$$(X^T H_w)(X^T H_w)^T = \begin{bmatrix} \Sigma_2^T \Sigma_2 & 0 \\ 0 & 0 \end{bmatrix}.$$

Hence a natural extension of the proposed linear discriminant projection in Section 2.1 is to choose the first $q = r + s$ columns of the matrix X in Eq. (2.1) as the transformation matrix G^* .

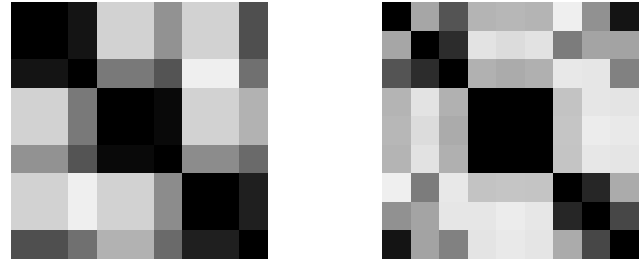


Figure 1: Document similarity. Each block represents the similarity between the corresponding row and column documents. The darker the contrast, the more similar the documents. For perfect class structure preserving, we expect three consecutive dark squares along the main diagonal.

2.3 Defining the Similarity After finding the transformation G , we define the similarity between classes to be the distance between their centroids in the transformed spaces. In other words, two categories are similar if they are “close” to each other in the transformed space. The linear discriminant projection finds the transformation that preserves the class structure by minimizing the sum of squared within-class scatter while maximizing the sum of squared between-class scatter and hence the distances in the transformed space should be able to reflect the inherent structure of the dataset.

To make it more clear on the linear discriminant projection approach, we compare the method with the well-known Latent Semantic Indexing (LSI) [2] and give a concrete example. LSI projects a document onto the latent semantic space. Although LSI has been proven to be extremely useful in various information retrieval tasks, it is not an optimal transformation for text categorization since LSI is completely unsupervised. In other words, LSI aims at optimal transformation of the original data into the lower dimensional space in terms of mean squared error but it pays no attention to the underlying class structure. Linear discriminant projection explicitly utilizes the intra-class and inter-class covariance matrices and tends to preserve the class structure.

We consider a dataset consisting of nine sentences from three different topics: user interaction, graph theory and distributed systems:

- 1(1) Human interface for user response
- 2(1) A survey of user opinion of computer system response time
- 3(1) Relation of user-perceived response time to error measurement
- 4(2) The generation of random, binary, unordered trees
- 5(2) The intersection graph of paths in trees
- 6(2) Graph Minors IV: Widths of trees and well-quasi-ordering
- 7(3) A survey of distributed shared memory system
- 8(3) RADAR: A multi-user distributed system
- 9(3) Management interface tools for distributed computer system

By removing words/terms that occur only once, we obtain the document-term matrix. Suppose that the first and second sentences in each class are used for training data. Then the transformation shown in Figure 1(a) is obtained and the plot of the LSI algorithm in Figure 1(b). The example shows that linear discriminant projection has discrimination power and is able to reflect the inherent similarity structure of the classes. Hence the distance between the centroids is a good measure for the similarity between categories.

3 Hierarchy Generation

After obtaining the similarities/distances between classes, we use the Hierarchical Agglomerative Clustering (HAC) algorithm of [5] to generate automatic topic hierarchies from a given set of flat classes. The result of hierarchical clustering is a dendrogram where similar classes are organized into hierarchies. We choose UPGMA (Unweighted Pair-Groups Method Average method), which is known to be simple, efficient and stable [5]. In UPGMA, the average distance between clusters is calculated from the distance between each point in a cluster and all other points in another cluster. The two clusters with the lowest average distance are joined together to form the new cluster.

4 Experiments on hierarchy Generation

We use a wide range of datasets in our experiments and anticipate that these data sets would provide us enough insights on automatic hierarchy generation. The datasets and their characteristics are summarized in Table 2. **20Newsgroups**

Datasets	# documents	# classes
20Newsgroups	20,000	20
WebKB	8,280	7
Industry Sector	9,637	105
Reuters-top 10	2,900	10
Reuters-2	8,000	42
K-dataset	2,340	20

Table 2: Data Sets Descriptions

dataset¹ contains about 20,000 articles evenly divided among 20 Usenet newsgroups. **WebKB** dataset contains web-pages gathered from university computer science departments. There are about 8300 documents and they are divided into seven categories: student, faculty, staff, course, project, department and other. **Industry Sector** dataset consists of company homepages classified in a hierarchy of industry sectors². **Reuters:** The Reuters-21578 Text Categorization Test collection contains documents collected from the Reuters newswire in 1987. It is a standard text categorization benchmark and contains 135 categories. In our experiments, we used two subsets of the data collection. The first one includes the ten most frequent categories among the 135 topics, which we call Reuters-top10. The second one contains the documents that have unique topics (documents that have multiple class assignments are ignored), which we call Reuters-2. **K-dataset**³ contains 2340 documents consisting news articles from Reuters news service via the Web in October 1997.

4.1 Data Preprocessing To preprocess the datasets, we remove the stop words using a standard stop list and perform the stemming operations with a Porter stemmer. All HTML tags and all header fields except subject and organization are ignored. In all our experiments, we first randomly choose 70% for hierarchy building (and later training in categorization), the remaining 30% is then used for testing. The 70% training set is further preprocessed by selecting the top 1000 words by information gain. The feature selection is done with the rainbow package⁴. All of our experiments are performed on a P4 2GHz machine with 512M memory running Linux 2.4.9-31.

¹<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.html>.

²<http://www.cs.cmu.edu/~TextLearning/datasets.html>.

³<ftp://ftp.cs.umn.edu/dept/users/boley/PDDPdata/>.

⁴<http://www.cs.cmu.edu/~mccalum/bow>.

4.2 Experimental Results Figure 2 shows the hierarchies of WebKB, 20Newsgroups and Reuters-top10 built via linear discriminant projection⁵ The block in the graphs represents the similarity between the corresponding row and column document categories and the darker the more similar.

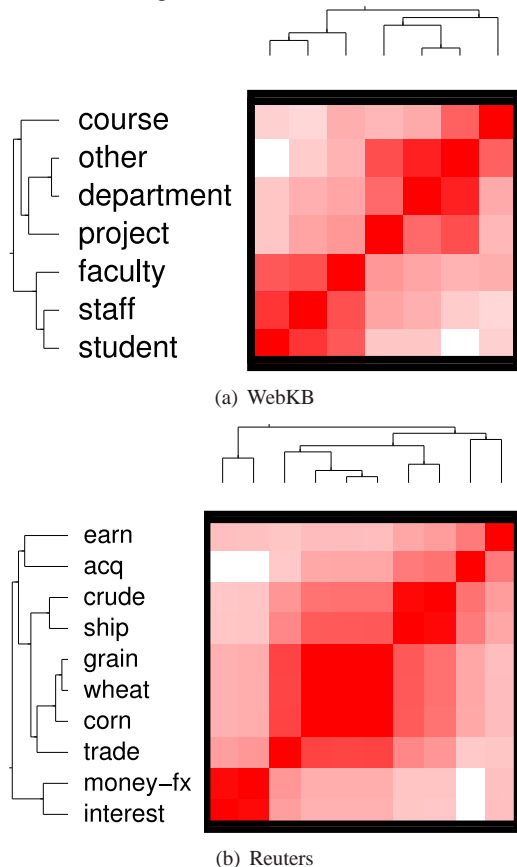


Figure 2: Hierarchies of WebKB, 20Newsgroups and Reuters-top 10 using linear discriminant projection.

We can observe from the dendrogram the semantic similarity of classes. For example, on WebKB, *faculty*, *staff* and *student* are close to each other. Note that *faculty*, *staff* and *student* are people and they are different from other datasets. Hence the linear discriminant projection approach tends to group them together. We can observe similar phenomena on 20Newsgroup and Reuters-top10. On 20Newsgroups, *talk.politics.guns* and *talk.politics.misc* are grouped together. On Reuters-top10, for example, the close pairs of classes using linear discriminant projection are: (*ship*, *crude*), (*money-fx*, *interest*), (*grain*, *wheat*) and (*earn*, *acq*).

5 Exploiting the Generated Hierarchy for Classification

In this section, we investigate the effects of exploiting the automatically generated hierarchy for classification and use

⁵Due to the space limit, we do not include the hierarchies on K-dataset, Reuters-2 and Industry sector.

classification accuracy as the evaluation measure.

An obvious approach to utilization of the hierarchy is a top-down level-based approach that arranges the clusters in a two-level tree hierarchy and trains a classifier at each internal node. We analyze the generated dendrogram to determine the clusters that provide maximum inter-class separation and find the best grouping of classes at the top level. The dendrogram is scanned in a bottom-up fashion to find the distances at which successive clusters get merged. We clip the dendrogram at the point where the cluster merge distances begin increasing sharply. In our experiments, we clip the dendrogram when the current merge distance is at least two times larger than previous one. For example, on 20Newsgroups dataset with linear discriminant projection approach, we have 8 top-level groups as shown in Table 3. The experimental results reported here are obtained via two-level classification.

groups	members
1	<i>alt.atheism</i> , <i>talk.region.misc</i> , <i>talk.politics.guns</i> , <i>talk.politics.misc</i> , <i>talk.politics.mideas</i>
2	<i>sci.space</i> , <i>sci.med</i> <i>sci.electronic</i>
3	<i>comp.os.mswindows.misc</i> , <i>comp.sys.ibm.pc.hardware</i> , <i>comp.graphs</i> , <i>comp.sys.mac.hardware</i> , <i>comp.windows.x</i>
4	<i>rec.sport.baseball</i> , <i>rec.sport.hockey</i> , <i>rec.motorcycles</i>
5	<i>misc.forsale</i>
6	<i>soc.religion.christian</i>
7	<i>rec.autos</i>
8	<i>sci.crypt</i>

Table 3: Top level groups for 20Newsgroups via linear projection.

LIBSVM⁶ is used as our classifier. LIBSVM is a library for support vector classification and regression and supports multi-class classification. In addition, we use linear kernel in all our experiments as it gives best results on our experiments.

We first build a top-level classifier (L1 classifier) to discriminate among the top-level clusters of labels. At the second level (L2) we build classifiers within each cluster of classes. Each L2 classifier can concentrate on a smaller set of classes that confuse with each other. In practice, each classifier has to deal with a more easily separable problem, and can use an independently optimized feature set; this should lead to slight improvements in accuracy apart from the gain in training and testing speed. Table 4 gives the performance comparisons of flat classification with hierarchical classification. We observe the improved performance on all datasets.

From Table 4, we observe that Reuters-top10 has the most significant gain in accuracy using hierarchy. Figure 3 presents the accuracy comparison for each class of Reuters-top10. Shown in Figure 3, each class' accuracy is improved by using the generated hierarchy except the accuracy of

⁶<http://www.csie.ntu.edu.tw/~cjlin/libsvm>

Datasets	Flat	Linear Projection	
		Level One	Overall
20Newsgroups	0.952	0.985	0.963
WebKB	0.791	0.860	0.804
Industry Sector	0.691	0.739	0.727
Reuters-top 10	0.826	0.963	0.921
Reuters-2	0.923	0.938	0.927
K-dataset	0.915	0.961	0.921

Table 4: Accuracy Table. The flat column gives the accuracy of flat classification, the “Level One” column shows the level one accuracy while the “Overall” column represent the overall accuracy.

trade stays unchanged. The accuracy of *corn* is improved significantly from about 7% to 60%. In flat classification, almost all the documents in *corn* class are misclassified to *grain* and *wheat* classes. Using hierarchical classification, by grouping *corn*, *grain* and *wheat* together, A second-level classifier using an independently optimized feature set can then be designed to focus on separation of the three similar classes. The performance improvement is thus obtained.

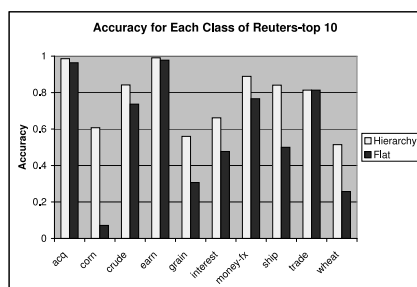


Figure 3: Accuracy comparison on each class of Reuters-top10

6 Manually-Built Hierarchy vs Generated Hierarchy

Once we have the automatic approach for hierarchy generation, it is natural to compare the generated hierarchy with the manually-built one. In this section, we illustrate their difference via three experiments on 20Newsgroups, WebKB and Reuters-top10.

Due to space limit, we only present the human-built (two-level) hierarchies for the 20Newsgroups as shown in Table 5. The manual hierarchy is generated by the authors to group the categories with strong confidence. To further understand the differences between two kinds of hierarchies, we also compare their hierarchical categorization performances, as listed in Table 6. For comparison purposes, the experimental results are based on two-level classifiers.

As you can observe from the comparisons, human-built hierarchy is purely based on “human semantics”, but not necessarily optimized for classification purpose. In all the three datasets, using the automatic generated hierarchy, the classification accuracies are slightly higher than those using human-built hierarchy. Hence, an important research

direction is to combine the automatic and manual approaches for generating both statistically significant and intuitively meaningful hierarchies.

groups	members
1	<i>talk.region.misc, talk.politics.guns, talk.politics.misc, talk.politics.mideas</i>
2	<i>sci.electronic, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware</i>
3	<i>comp.os.mswindows.misc, sci.crypt, comp.graphs, comp.windows.x</i>
4	<i>rec.sport.baseball, rec.sport.hockey</i>
5	<i>misc.forsale</i>
6	<i>alt.atheism, soc.religion.christian</i>
7	<i>rec.autos, rec.motorcycles</i>
8	<i>sci.space, sci.med</i>

Table 5: Human-generated 8 top-level groups for 20Newsgroups.

Datasets	Human-generated	Automatic Generated
20Newsgroups	(0.956, 0.954)	(0.985, 0.963)
WebKB	(0.811, 0.795)	(0.860, 0.804)
Reuters-top 10	(0.928, 0.9475)	(0.963, 0.901)

Table 6: Performance comparisons of human-generated hierarchy with automatic generated hierarchy. The entries are in the format of (level one, flat). The accuracy of automatic generated hierarchy was taken from the linear projection approach.

References

- [1] D’Alessio, S., Murray, K., Schiaffino, R., & Kershenbaum, A. (2000). The effect of using hierarchical classifiers in text categorization. *RIAO-00*.
- [2] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41, 391–407.
- [3] Fukunaga, K. (1990). *Introduction to statistical pattern recognition*. New York: Academic Press. 2nd edition.
- [4] Howland, P., & Park, H. (2004). Generalizing discriminant analysis using the generalized singular value decomposition. *IEEE PAMI*, 26, 995–1006.
- [5] Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice Hall.
- [6] Koller, D., & Sahami, M. (1997). Hierarchically classifying documents using very few words. *ICML*.
- [7] Li, T., Zhu, S., & Ogihara, M. (2003a). Efficient multi-way text categorization via generalized discriminant analysis. *ACM CIKM* (pp. 317–324).
- [8] Li, T., Zhu, S., & Ogihara, M. (2003b). Topic hierarchy generation via linear discriminant projection. *ACM SIGIR* (pp. 421–422).
- [9] Sun, A., & Lim, E.-P. (2001). Hierarchical text classification and evaluation. *ICDM* (pp. 521–528).
- [10] Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. *SIGIR*.