Text Categorization Based On Concept Indexing and Principal Component Analysis

Huang Ke, Ma Shaoping
State Key Lab of Intelligent Technology and Systems, Department of Computer Science and Technology, Tsinghua University, 100084, Beijing, China
hk@s1000e.cs.tsinghua.edu.cn

Abstract

A major problem in text categorization is the high dimensionality of feature vector space, which is about ten thousands in common. To reduce the dimensionality of the space while keeping the categorization accuracy is useful for improving categorization effectiveness and applying new categorization algorithms. Current feature selection methods for text categorization are partially effective in reducing dimensionality. We put forward a new algorithm, which combines algorithm of concept indexing and principal component analysis, for reducing dimensionality. From the experiments, we find that this algorithm can effectively reduce dimensionality without sacrificing categorization accuracy.

Keyword

Text Categorization Concept Indexing Principal Component Analysis Machine Learning

Introduction

With rapid development of Internet, more and more online texts are appearing on the network. While the users of Internet can access more information, they are being swamped by the voluminous information at the same time. Texts on the network are not managed orderly; they are just an unordered set of various data. To search specific information from Internet usually costs much time and energy. If the online texts are well indexed and summarized, people can access these texts more efficiently and effectively. Text categorization is such a solution: it assigns predefined categories to free text documents. Recently much research attention has been paid to text categorization. Most of the researchers based their work on the vector space model (VSM) of text document.

Vector Space Model and Feature Selection

Vector space model(VSM) of text document is put forward by Salton【1】and used in SMART system. In VSM, a text document is represented by a vector and all subsequent calculations are based on the vectors. Usually each dimension (or, feature) in vector space represents one word appearing in the documents, while the weight of the dimension reflects relative significance of the word to a document. Many machine learning technologies have been successfully applied to text categorization,

such as k-nearest-neighbor classifier(KNN), naïve Bayesian classifier, decision tree classifier, artificial neural network classifier, support vector machine classifier and so on 【2,3,4,5】.

A major characteristic, or difficulty of text categorization problems is the high dimensionality of the feature space. The native feature space consists of the unique terms (words or phrases) that occur in documents, which can be tens of thousands of terms even for a moderate-sized text collection. This is prohibitively high for many learning algorithms. Few neural networks, for example, can handle such a large number of input nodes. Bayes belief network model, as another example, will be computationally intractable unless an independent assumption (often not true) among features is imposed. From another view of point, in such a high-dimension space, not every feature is helpful for promoting classification accuracy. Current research results show that different terms play different roles in reflecting documents' contents. With some "noisy" terms kept, categorization accuracy may even be hurted. What's more, the calculation complexity of high-dimension obviously hampers online classification of text documents on the network. It's highly desirable to reduce the native space without sacrificing the categorization accuracy. It's also highly desirable to achieve such a goal automatically, i.e. no manual definition or construction of features is required.

Current research on feature selection concentrates on two fields. One type of such technology focus on feature selection methods: the calculation of Document Frequency (DF), Information Gain (IG), χ^2 statistic, Mutual Information (MI), Term Strength (TS) and so on【2,6】. These values can be easily calculated from training samples. Although these values can reflect the relative significance of terms to documents' contents and thus can be used to reduce dimensionality, the reduced dimensionality usually remains several thousands. If the dimensionality is reduced to several hundreds, the classification accuracy will be sacrificed greatly. Another type of technology tends to apply natural language processing (NLP) to text categorization. Such technology includes extracting nouns from documents and classifying documents based on the nouns extracted; n-gram language model; exploring semantic relations between terms; the application of WORDNET and so on【7,8,9】. But the efforts in this direction are not encouraging. It's believed that the introduction of NLP in text categorization may even cause negative effects on the categorization accuracy while NLP technology has not been fully developed.

In this paper we'll combine concept indexing (CI) and principal component analysis (PCA) to reduce the dimensionality of native vector space. PCA has been proved an effective method to aggressively reduce dimensionality. For text categorization, it's not viable to apply PCA directly to the high-dimension native feature space for the complexity of PCA. We first apply CI to native feature space and then apply PCA to the processed vectors. We expect to achieve the goal of dimensionality reduction through combination of the two technologies.

Concept Indexing (CI) and its application in text categorization

Concept Indexing (CI) is a simple and effective way to reduce dimensionality【10, 11】. In the case of supervised learning, CI constructs the subspace (CI subspace) consisting of category centers (or, in another word, prototype vectors) as base vectors and projects native document vectors to this subspace. All subsequent processing is based on the vectors in the subspace. So the dimension of the subspace will be the number of categories of training samples, which is usually less than one hundred and thus is much less than the dimensionality of native vector space. The steps of CI can be formulated as following:

(1) Calculation of categories' prototype vectors. For i th prototype vector $Center_i$, it can be calculated as

$$Center_i = \frac{1}{N_i}\sum_{j=1}^{N_i} Doc_{ij}$$

Where N_i is the number of documents in category C_i, and Doc_{ij} is j th document vector of C_i.

(2) Projection of original document vectors to subspace. Assume that M categories appear in the training samples, any original document vector Doc_x will be represented by a M-dimension vector. The value of j th dimension of the projected vector is the dot product between the original document vector and the j th category prototype vector, which can be represented as

$$\cos(Doc_x, Center_j) = \frac{Doc_x \bullet Center_j}{\left\|Doc_x\right\|_2 \times \left\|Center_j\right\|_2}$$

From step (1) and (2), the native document vectors can be projected to a low-dimension subspace. Now we analyze the characteristic of such algorithm. Category prototype vector summarizes contents of documents belonging to this category. The more important one term to the content of a category, the higher the weight of corresponding dimension of the prototype vector will be. It's also found that most of weights concentrates on relatively a small part of dimensions. Synonyms usually co-occur in the same category and the weights of corresponding dimension in the prototype vectors are close. In the case of polysemy, their corresponding weights in different categories tend to be not similar. So the prototype vectors of categories partially embody the latent semantic relations of documents and thus weaken the negative effects of synonyms and polysemy to categorization. As stated before, not every term is useful for categorization; some terms are "noise" for categorization. The process of projection of each document vectors to the prototype vectors can partially filter these "noise". From the analysis above, it's plausible to project document vectors to the subspace consisting of category prototype vectors as base vectors. In the subspace, each prototype vector can be viewed as "concept" and the projection of document vectors can be viewed as calculating the "indexing" of "concept". This is what the term "concept indexing" means.

Principal Component Analysis and its application in text categorization

Principal Component Analysis (PCA) is a widely adopted method in pattern recognition and signal processing. PCA is effective in data compression and feature extraction【12,13,14】. It's natural for us to apply PCA in text categorization to get the low-dimension representation of document vectors. Because the complexity of PCA is prohibitively high in high-dimension space, we apply PCA to the CI subspace, i.e. we first apply concept indexing to native document vectors and then apply PCA to the low-dimension CI subspace. We define the subspace got after CI and PCA processing CI_PCA subspace. We expect to get low-dimension representation of document vectors in a relatively low cost.

The steps of PCA can be formulized as following.

(1) Assume that M categories appear in the training samples, calculate each inner category co-variance matrix respectively:

$$\Sigma_i = E\left[(x-\mu_i)(x-\mu_i)^T\right]$$

Where Σ_i is the inner category co-variance matrix of i th category; x is the document vector in i th category; μ_i is the prototype vector of i th category.

（2）Calculate weighted average inner-category co-variance matrix S_w :

$$S_w = \sum_{i=1}^{M} P_i \Sigma_i$$

Where P_i is the pre-possibility of i th category and can be estimated from training samples.

（3）Calculate the eigenvalue d_j and eigenvector V_j of S_w :

$$S_w V_j = d_j V_j$$

Because S_w is real symmetric matrix, M linear-independent eigenvectors exist and the dimension of each eigenvector is M.

(4) Transform vectors through the eigenvectors of S_w . Sort the eigenvalues of S_w in descending order: $d_1 \geq d_2 \geq ... \geq d_M$ and arrange the corresponding eigenvectors in the transformation matrix $W = [V_1, V_2, ..., V_M]$. For a vector x, it can be transformed to the new subspace as:

$$y = W^T x$$

Where W is the matrix consisting M linear-independent eigenvectors of S_w 。

(5) Compress the vectors in the transformed space (PCA subspace): according to the predefined dimensionality D, simply keeping the top D dimensions and omitting other dimensions.

Through steps 1 to 5, one can obtain the low-dimension representation of vector in PCA subspace. The key property of PCA is that it attains the best linear map from M-dimension space to D-dimension space in the senses of:

- Least squared sum of errors of the reconstructed data;
- Maximum mutual information (assuming the data vector x distributed normally) between the original vectors x and their projection y: $I(x,y) = \frac{1}{2}\ln((2\pi e)^D \lambda_1 ... \lambda_D)$, where $\lambda_1 ... \lambda_D$ are the first D eigenvalues of the covariance matrix.

Experimental Results

In this section, we'll compare the classification accuracy based on the document vectors in CI_PCA subspace and the classification accuracy based on native vector representations. The experiments are based on English text collection of Reuters-21578【15】. The text collection consists of economic news of Reuters in 1987. Every document of the collection is manually assigned one or more categories and the partition of training samples and testing samples is predefined. The size of this collection is about 13,000 documents, of which contains about 8,000 training samples and 5,000 testing samples. All the documents in the collection are partitioned into 135 categories, of which 115 categories appear in training samples. The collection is organized following SGML grammar and widely used in the field of text categorization.

First, the documents in the collection are preprocessed, eliminating the stop words in the documents and stemming. Native document vectors are constructed based on the terms extracted, the frequency of single term as the corresponding weight in the document vectors. These vectors are then projected to CI subspace and then projected to PCA subspace. After the two projections, we can obtain the 115-dimension representations of native document vectors.

Effect in Dimensionality Reduction

Current experiments indicate that keeping those terms with high document frequency is helpful to promote classification accuracy. In this following experiment we'll compare the effect of dimensionality reduction by DF value and the effect of CI and PCA. We compare the classification accuracy based on the vectors processed by CI_PCA and DF feature respectively. We select KNN classifier with K = 10 for classification. The result is showed as figure1:

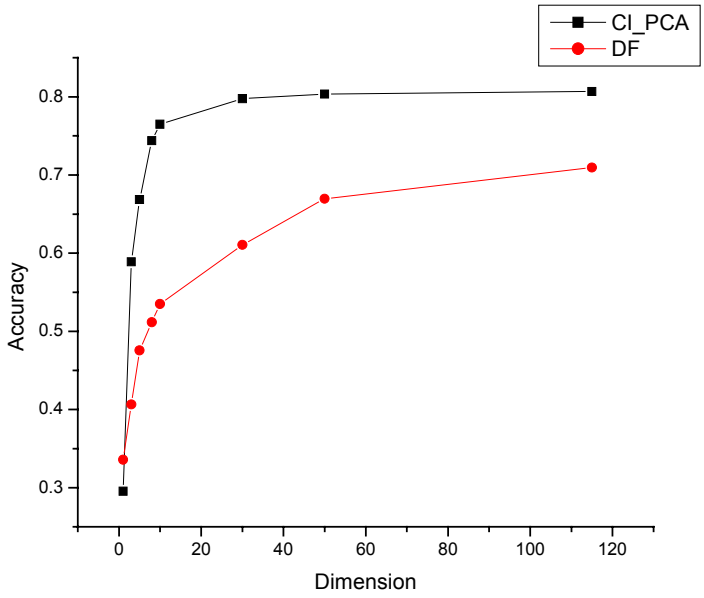

Figure 1 Comparison of effects in dimensionality reduction

In figure1, the horizontal coordinate represents the dimensionality of the vector space and the vertical coordinate represents the classification accuracy. The curve labeled "CI_PCA" is the

experiment result got by the CI and PCA algorithm described above, and the curve labeled “DF” is the experiment result of dimensionality reduction by document frequency values. From the comparison of the classification accuracy, it’s very obvious that the algorithm of combing CI and PCA is much more effective in dimensionality reduction.

Comparison With Original Vector Space

From Figure1 one can find that in the case of dimensionality reduction by DF values, the classification accuracy will rise with larger dimensionality. The following experiment will compare the classification accuracy based on CI_PCA subspace and native vector space without dimensionality reduction. In CI_PCA subspace, we keep dimensionality as 50 and compare the classification accuracy in the subspace with the classification accuracy in the native vector space. We adjust the parameter of K in KNN classifier to get different classification result. The comparison is showed in figure2:

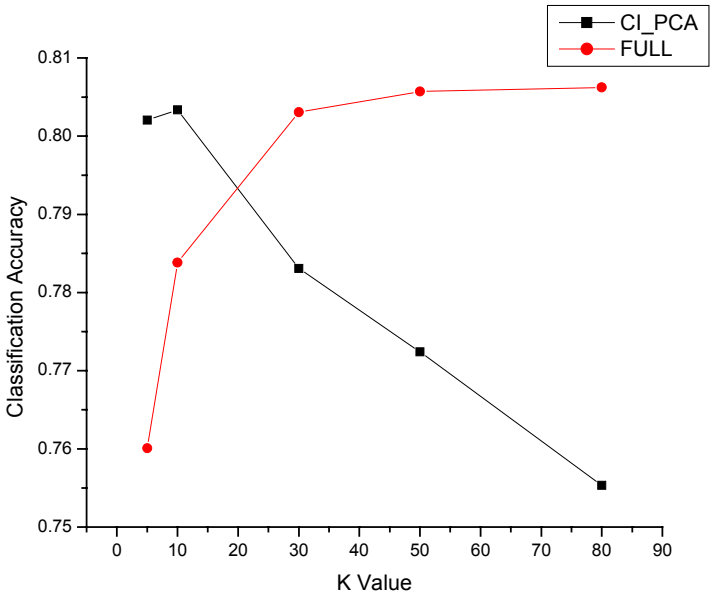

Figure 2 Comparison with native vector space

In figure2, the horizontal coordinate represents the parameter K of the KNN classifier and the vertical coordinate represents the classification accuracy. The curve labeled “CI_PCA” is the experiment result got by the CI and PCA algorithm described above, and the curve labeled “FULL” is the experiment result based on native vector space. The result shows that even in the low-dimension CI_PCA subspace (dimensionality = 50), the highest classification accuracy is very close to the highest classification accuracy got in the original vector space. Such result means that the algorithm of CI_PCA achieves aggressive dimensionality reduction without sacrificing classification accuracy.

The Influence of Parameter of KNN Classifier

While KNN classifier is used, one important issue is how to select parameter K. Unsuitable K value may cause less similar vectors be counted in (in the case of small K), or much noisy vectors affect the classification result (in the case of large K). The following experiments show the influence of K of KNN classifier to the classification result. The experiments show how the classification accuracy varies with the dimensionality of subspace under different K values. The result is showed in figure3:

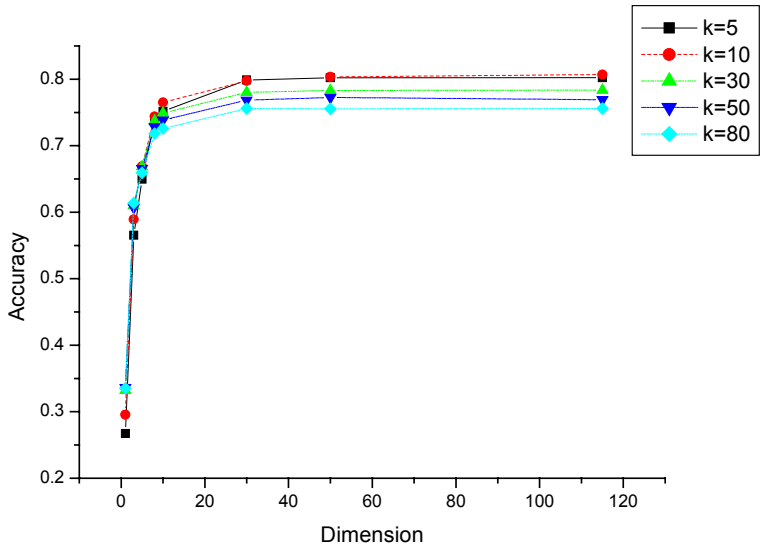

Figure 3 The influence of parameter K of KNN classifier

In figure3, the horizontal coordinate represents the dimensionality of the CI_PCA subspace and the vertical coordinate represents the classification accuracy. The curve labeled different K value is the experiment results with different K value of KNN classifier. The experiment results show that to achieve optimal classification accuracy, the dimensionality can be reduced as low as 30. Compared with the classification accuracy based on native vector space (the dimensionality is about 20 thousands), the classification accuracy is not sacrificed. The results also show that a smaller K value is better than a larger K value. Such result means after CI_PCA processing, similar document vectors are closer in the subspace, which verifies our analysis of prototype vector above: prototype vectors weaken the negative effect of synonyms and polysemy, and the projection of original document vectors can partially filter "noisy" dimensions of document vectors.

Conclusion

A major characteristic, or difficulty of text categorization problems is the high dimensionality of the feature space. Current methods cannot aggressively reduce dimensionality without sacrificing classification accuracy. Concept indexing and principal component analysis are effective algorithms in dimensionality reduction. Because of the complexity of PCA in high-dimension vector space, we combine concept indexing and principal component analysis to aggressively reduce dimensionality of document vector space. From the experiment results, we find that the method of combing CI and PCA can compress the vector space dimensionality from tens of thousands to less than 50 without sacrificing classification accuracy. The experiment results are encouraging. The method put forwarded in the paper is meaningful to online text categorization, application of more machine learning algorithms.

Reference

【1】 Salton, Gerard: *Introduction to modern information retrieval.*，Auckland : McGraw-Hill , 1983.

【2】 Kjersti Aas, Line Eikvil: *Text Categorisation: A Survey*, Rapport Nr. 941, June, 1999. ISBN 82-539-0425-8

【3】 Erik Wiener, Jan O. Pedersen, Andreas S. Weigend: *A Neural Network Approach to Topic Spotting*, Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval

【4】 Yiming Yang: *An Evaluation of Statistical Approaches to Text Categorization,* Journal of Information Retrieval, Vol 1, No. 1/2, pp 67--88, 1999.

【5】 Leah S. Larkey and W. Bruce Croft: *Combing Classifiers in Text Categorization,* Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval

【6】 Yiming Yang, Jan O. Pedersen, *A Comparative Study on Feature Selection in Text Categorization*，Proc. of the 14th International Conference on Machine Learning ICML97, pp. 412---420, 1997

【7】 Sam Scott, Stan Matwin, *Feature Engineering for Text Classification*, Proc. 16th International Conf. on Machine Learning,379--388,Morgan Kaufmann, San Francisco, CA,1999

【8】 William W. Choen, Yoram Singer, *Context-Sensitive Learning Methods for Text Categorization*, Proceedings of the 19th Annual International ACM Conference on Research and Development in Information Retrieval, pages 307--315, Zurich, Switzerland, 1996. ACM Press

【9】 Robert Basili, Alessandro Moschitti, Maria Teresa Pazienza, *Language Sensitive Text Classification*, Proceeding of {RIAO}-00, 6th International Conference ``Recherche d'Information Assistee par Ordinateur", Paris, FR,331--343,2000

【10】 George Karypis, Eui-Hong Han, *Concept Indexing: A Fast Dimensionality Reduction* Algorithm with *Applications to Document Retrieval & Categorization*, Technical Report TR-00-0016, University of Minnesota, 2000

【11】 George Karypis, Eui-Hong Han, *Fast Supervised Dimensionality Reduction Algorithm with Applications to Document Categorization & Retrieval*, Proceedings of the 2000 ACM CIKM International Conference on Information and Knowledge Management, McLean, VA, USA, 2000

【12】 Bian Zhaoqi, Zhang Xuegong, *Pattern Recognition (Chinese)*, Tsinghua University Press, Beijing, 2000.1

【13】 Matthew Partridge, Rafael Calvo, *Fast Dimensionality and Simple PCA*, Intelligent Data Analysis, 2(3), 1998

【14】 Miguel A. Carreira-Perpinan, *A Review of Dimension Reduction Techniques*, Techical Report CS-96-09, Department of Computer Science, U. of Sheffield, 1997

【15】 http://www.research.att.com/~lewis/reuters21578.html