Applying the Conjugate Gradient Method for Text Document Categorization

Vincent Tam[1], *Rudy Setiono*[2] *and A. Santoso*[2]

Abstract

In this paper, we investigate the effectiveness of two different methods to solve the linear least squares fit (LLSF) problem for document categorization. The first method is the Singular Value Decomposition (SVD) method that has been previously used to solve the document categorization problem. The second method is the Conjugate Gradient (CG) method that is one of the most effective algorithms for solving a linear equation problem. However, up to our knowledge, the CG method has never been applied to handle the document classification problem. Therefore, we compare the effectiveness of these two LLSF methods to categorize text documents. In addition, we examine the effect of using different term weighting schemes on their performance for document classification. Lastly, we compare the performance of the LLSF classifiers against the neighborhood-based Dt-kNN classifier, our best variant of the kNN classifier integrated with a dynamic threshold scheme, on the Reuters 21578 dataset. Besides being the first proposal to use the CG method for document classification, our work opens up many exciting directions for future investigation.

Keywords

Document Classification, Linear Least Squares Fit, Conjugate Gradient Method, Performance Measures.

1. Introduction

The linear least squares fit (LLSF) method was first proposed by Yang and Chute [3] for categorizing document collection from the MEDLINE database and Mayo patient records. Basically, this method works by learning the association between document terms and categories from manually categorized documents through adjusting a weight for each term in the document collection for each category, and using these weights to categorize new documents. The weight of each term for each category, as represented by a matrix of term-category regression coefficients, could be obtained by solving the least squares fit equation on the training set vectors.

Many algorithms such as Singular Value Decomposition (SVD) [5], Jacobi algorithm [6], Gauss-Seidel algorithm [6], Successive Over Relaxation (SOR) algorithm [6], and Conjugate Gradient (CG) [7] can be used to solve the LLSF problem. In [3], Yang and Chute used the SVD method to solve the LLSF problem, while Zhang and Oles [4] used the relaxation method to solve the least-squares problem. However, we are not aware of any work that compares the effectiveness of these different methods in categorizing documents. Therefore, we will study in this paper about the effectiveness of two different methods to solve the linear least-squares fit problem for document categorization. The first method is the SVD method, which has been used to solve the least-squares problem in some previous studies [1, 2, 3]. The second method is the Conjugate Gradient (CG) method, which is one of the most effective algorithms for solving a linear equation problem [4]. In addition, we examine the effect of using different term weighting methods on the performance of different methods when applied for solving the least-squares problem for categorizing documents. In the following sections, we will explain the SVD and CG methods individually on solving the LLSF mapping problem for document categorization.

This paper is organized as follows. Section 2 describes the vector-space model, the similarity measure, the performance measure for document classifiers and the statistical tests for performance analysis. Section 3 describes the text categorization methods including the singular value decomposition (SVD) method [5], firstly used by Yang and Chute [3] for classifying text documents, and the conjugate gradient (CG) method [7] that is an attractive approach for solving large sparse LLSF problems but never applied to categorize text documents yet. In particular, we will detail how to adapt the CG method for categorizing text documents. Section 4 evaluate and analyse the performance of the two LLSF methods against the heuristic-based Dt-*k*NN [8, 9] document classifier on handling the Reuters 21578 dataset [8]. Lastly, we conclude our work and shed light on several directions for future exploration in Section 5.

2. Preliminaries

Vector-space model: the model has been widely used in the area of Information Retrieval [9] and in particular document categorization [4]. Basically, it creates *m*-dimensional vectors $\vec{w} = (w_1, w_2, w_3, ..., w_m)$ to represent

[1] Department of Electrical and Electronic Engineering. The University of Hong Kong, Pokfulam, Hong Kong. Email: vtam@eee.hku.hk

[2] School of Computing,The National University of Singapore, Singapore. Email: rudys@comp.nus.edu.sg

the text document with respect to a set of *m* unique terms. The weight w_i associated with the term t_i depends on the term frequency f_i (the number of occurrence of term t_i in any document *x*), and the inverse document frequency idf_i, which is equal to $log\ (N/D_i)$, where *N* is the total number of documents in the document collection and D_i is the number of documents in the document collection that contain the term t_i. The weight of term t_i in the document *x* can be calculated as follows:

$$w_i(t_i,x) = f_i * log(N/D_i) \qquad (1)$$

Similarity measure: the similarity between two documents $\vec{d}_j$ and $\vec{d}_k$ is normally measured using the cosine function between two vectors. This function is written as:

$$\cos(\vec{d}_j, \vec{d}_k) = \frac{\vec{d}_j \cdot \vec{d}_k}{|\vec{d}_j| * |\vec{d}_k|} \qquad (2)$$

where "." denotes the dot product of the two vectors and $|\vec{d}_j|$ denotes the length of vector $\vec{d}_j$. We use this function to measure the similarity between two documents in all experiments reported in this paper.

Performance measure: the performance measures for document classification algorithms [2,4] vary from the simple precision and recall measures [9] to the more complicated micro and macro F_1 values [3]. Precision is the percentage of retrieved documents that are relevant while recall is the percentage of relevant documents retrieved. F_1 measure was introduced by Rijsbergen [3] to combine recall (*r*) and precision (*p*) with an equal weight in the following formula:

$$F_1(r,p) = \frac{2rp}{r+p} \qquad (3)$$

The F_1 score can be computed on each individual category first and then averaged over categories, or computed globally over all the test documents. The former is known as the macro averaging, while the latter is called micro averaging. Micro-averaged F_1 value is widely used in the cross-method comparison, while macro-averaged F_1 is often used to measure the performance of a classifier in rare categories.

Significance test: to compare the performance of two different classifiers, we follow Yang and Liu [1] to perform the micro and macro sign tests. The micro sign test is a sign test designed for comparing two systems based on their binary decision on all test document/category pairs. By counting *n*, the number of times where systems A and system B differ and *k*, the number of times where system A is better than system B, we can measure the performance of systems A compared to system B using the binomial distribution method (if $n \leq 12$) or the normal distribution method (if $n > 12$). The macro sign test is similar to the micro sign test. However in the macro sign test, we compare the two systems based on their decision solely on each individual category.

3. The Text Categorization Methods

3.1. Singular Value Decomposition (SVD)

SVD is a direct method to solve the linear least-squares problem. This method is a very powerful technique for dealing with sets of equations where the coefficients matrices are either singular or very close to singular. In many cases, where the other direct methods for solving linear least-squares problem such as Gaussian elimination and LU decomposition failed to give a satisfactory result, the SVD technique can still give a good result [5].

Using the SVD method, any *M x N* matrix *A* whose number of rows *M* is greater than or equal to its number of columns *N*, can be written as the product of an *M x N* column-orthogonal matrix *U*, an *N x N* diagonal matrix *S* with positive or zero elements (the *singular values*), and the transpose of an *N x N* orthogonal matrix *V*. In other words, it can be described using the following figure:

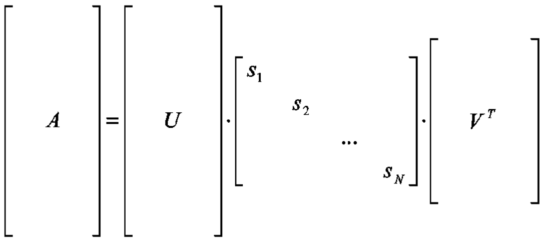

Figure 1: SV Decomposition of Matrix A

The matrices *U* and *V* are each orthogonal with their columns as orthonormal:

$$U^T \cdot U = V^T \cdot V = I \quad \text{and} \quad V \cdot V^T = I$$

To obtain the term-category association matrix *F*, we need to form matrices *A* and *B* where a row in matrix *A* represents document i^{th} in the training set, a column in the matrix *A* represents a term *j*th in the document collection, a row in the matrix *B* represents assignment of i^{th} document to the category set, and a column in the matrix *B* represents a category in the category set. For example, if b_{ij} value is equal to *1*, it means that the i^{th} document in the training document collection belong to the jth category and if the value is equal to *0*, it means that the *ith* document does not belong to the *j*th category.

The solution for the linear least squares fit problem can

$$F = B^T (A^+)^T = B^T U S^{-1} V^T \qquad (4)$$

be obtained using the following formula:
After the solution matrix F was computed, we need to run a cross validation test on the training documents to compute the category specific thresholds as in the *k*NN classifier. Besides, the categories for each test document is determined based on F, the original test document vector $\vec{d}$, and the cosine function for similarity measures. For detail, refer to [10].

3.2. Conjugate Gradient (CG) Method

Conjugate Gradient is the most prominent iterative method for solving a large sparse linear least squares problem:

$$Ax = b \qquad (5)$$

where matrix A is a known large sparse positive definite matrix, x is an unknown matrix, and b is a known vector [7]. In the text document classification problem, A is the matrix representation of all documents in the training set, x is the term-category association vector and b is a vector representing the category assignment for each training document. This method is especially good for document classification problem because text document classification normally creates a large sparse representation matrix.

The CG method proceeds by generating vector sequences of iterates (i.e., successive approximations to the solution), residuals corresponding to the iterates, and directions used in updating the iterates and residuals:

$$p^{(0)} = r^{(0)} = b - Ax^{(0)} \qquad (6)$$

In every iteration of the method, two inner products are performed in order to compute update scalars that are defined to make the sequences satisfy certain orthogonality conditions. For a symmetric positive definite linear system, these conditions imply that the distance to the true solution is minimized in some norm.
Unlike the SVD method which could compute the solution matrix F (matrix F represents the weight of each term in the collection for all the categories used), the CG method could only compute the solution vector x (vector x represents the weight of each term in the document collection for a single category). Therefore, after the solution vector x for each of the categories had been computed, we should combine all the solution vectors to form a solution matrix F which is similar to the one used for the SVD method. The solution matrix F could then be used to compute the category specific threshold by running the cross validation test on the training documents (see [1, 9] for detail on cross validation test). The categories assignment for each test document can be determined by using similar steps as explained previously for the SVD method.

4. Experiments

For this experiment, we did not study the effect of using our full dataset since we did not have enough resources to handle the full dataset. Both the SVD and the CG method were implemented using Matlab Version *6.1* and were run under UNIX environment. The SVD method took approximately *2* hours to finish while the CG method needed roughly *10.6* hours. The CG methods needed much longer time because this method could only process one category at each time, while the SVD method could process all the categories at once. However, our experimental results show that the CG method could return a more accurate result compared to the SVD algorithm, especially for the rare categories. In addition, we include the result of the Support Vector Machine (SVM) classifier [1], a machine-learning algorithm adapted for document classification, for comparison purpose. Furthermore, we compare these results against those of a heuristic-based *k*-Nearest Neighborhood improved with a dynamic threshold scheme (Dt-*k*NN)[10].

Classifier	Recall	Precision	Micro F_1	Macro F_1
SVD (*tf*)	0.8524	0.8272	**0.8396**	0.5572
SVD (*tf*idf*)	0.8538	0.8224	0.8378	**0.5622**
CG (*tf*)	0.8609	0.8114	0.8354	0.6116
CG (*tf*idf*)	0.8711	0.8084	**0.8386**	**0.6280**
Dt-*k*NN	0.8734	0.8605	0.8669	0.5790
SVM	0.8120	0.9137	0.8599	0.5251

Table 1: Results of the SVD, SVM and CG Classifiers.

Table 1 details our experimental results using the SVD and CG algorithm for classifying text document in the Reuters 21578 dataset [9]. These results clearly show that in terms of macro averaged *F1* score, the CG method outperforms the SVD method. On the other hand, in terms of micro-averaged *F1* both the SVD and CG methods perform almost similarly. The best result obtained by the SVD method is only slightly higher than the result obtained by the CG method. The micro *F1* score of CG method is lower compared to the SVM (Singular Value Decomposition) method as in [1] (*0.8599*). However, the macro *F1* score of CG method is higher than the SVM (*0.5251*). To study the performance of each classifier further, we also conducted an investigation on the classifier performance on the top 10 categories. For more detail, refer to [9].

Tables 2 and 3 below show the results of our significance test results on different classifiers. Table 3 clearly shows that based on the micro significance test, the SVD algorithm using the *tf* weighting scheme and the Dt-*k*NN performs the best. However, the CG method with the *tf*idf* weighting scheme outperforms all the other LLSF algorithms tested and performs as good as the Dt-*k*NN classifier based on the macro significance test (Table 4). This is probably because the CG method could find a more accurate term-category association matrix, especially for the category with low number of training documents compared to the direct methods for solving the LLSF problem such as the SVD method.

Classifier	SVD (tf)	SVD (tf*idf)	CG (tf)	CG (tf*idf)
Dt-kNN	≈	>	>>	>>
SVD (tf)		>>	>>	>>
SVD (tf*idf)			>>	>>
CG (tf)				≈

Table 2: Micro Significance Test Result for different classifiers[2]

Classifier	SVD (tf)	SVD (tf*idf)	CG (tf)	CG (tf*idf)
Dt-kNN	≈	≈	≈	≈
SVD (tf)		≈	≈	<<
SVD (tf*idf)			≈	<<
CG (tf)				<<

Table 3: Macro Significance Test Result for different classifiers[2]

Our experiments also found that the SVD algorithm performs better in conjunction with *tf* weighting scheme, while the CG method could perform better in conjunction with the *tf*idf* weighting scheme. These results reinforced the observations we obtained in some previous work [9] which stated that a classifier could perform optimally only in conjunction with a certain term weighting scheme.

5. Concluding remarks

In this paper, we experimented with different document classifiers such as the neighborhood-based Dt-*k*NN [8, 9] (a variant of the *k* Nearest Neighbor [8]) classifier against the LLSF (Linear Least Squares Fit) classifiers, including the Singular Value Decomposition (SVD) and Conjugate Gradient (CG) methods, for categorizing documents in the Reuters 21578 dataset. Besides being the first attempts to use the CG method as a document classifier, our work also investigated the effect of using different term weighting schemes towards the performance of each classifier. Agreed with our previous study [9], these experimental results showed that each classifier could perform optimally only in conjunction with a certain term weighting scheme. For example, the LLSF classifier performed better in conjunction with the *tf*idf* weighting scheme. Furthermore, our work opens numerous possibilities for future exploration. An example is the integration of the CG method into the modern digital libraries to categorize the vast amount of stored documents. Another potential application is the possible use of the SVD or CG method implemented on the e-mail or Short Message Service (SMS) server side to quickly categorize and possibly remove any unwanted emails or short messages in mobile phones due to spamming.

[2] The signs ">>" and ">" indicate that the classifier in the row is significantly better than the classifier in the column at level of 1% or 5%, respectively. The signs "<<" and "<" indicates that the classifier in the column is significantly better than the classifier in the row at the level of 1% or 5% respectively. The sign "≈" indicates the differences between both classifier performances are not statistically significant.

6. References

[1] Y. Yang, and X. Liu, *A re-examination of Text Categorization Methods*, In Proc. of ACM SIGIR 1999.

[2] W. Wang, W. Meng and C. Yu, *Concept Hierarchy Based Text Database Cate. in a Metasearch Engine Environ't,* In Proceedings of WISE, p.273-280, 2000.

[3] Y. Yang and C. Chute, *An Example-based Mapping Method for Text Cat. & Retrieval,* ACM TOIS,1994.

[4] T. Zhang & F. Oles, *Text Cate. Based Regularied Linear Class. Methods*, Information Retrieval, vol. 4, p. 5-31, Kluwer Academic Publisher, 2001.

[5] W. Press *et. al. Numerical Recipe in C: The Art of Sci Comp.,* Cambridge University Press, 1993.

[6] R. L. Burden, J. D. Faires, and A. C. Reynolds, *Numerical Analysis*, PWS Publisher, Boston, Massachusetts, 1981.

[7] R. Barret, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Donggara, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods,* Society for Industrial and Applied Mathematics, 1994.

[8] V. Tam and A. Santoso, *A Comp. Study of Centroid-, Neighborhood-based and Statistical Approach for Effective Text Cate.* In Proc. of ICPR, p.286-289, 2002.

[9] A. Santoso, *Improving Web Search Engines with Docu. Classification Technique*, MSc. Thesis, School of Computing, NUS, December 2001.