International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:1, 2008

A Content Vector Model for Text Classification

Eric Jiang

***Abstract*—**As a popular rank-reduced vector space approach, Latent Semantic Indexing (LSI) has been used in information retrieval and other applications. In this paper, an LSI-based content vector model for text classification is presented, which constructs multiple augmented category LSI spaces and classifies text by their content. The model integrates the class discriminative information from the training data and is equipped with several pertinent feature selection and text classification algorithms. The proposed classifier has been applied to email classification and its experiments on a benchmark spam testing corpus (PU1) have shown that the approach represents a competitive alternative to other email classifiers based on the well-known SVM and naïve Bayes algorithms.

***Keywords*—**Feature Selection, Latent Semantic Indexing, Text Classification, Vector Space Model.

I. Introduction

TEXT classification is a problem applied to natural language texts that assigns a document to one or more predefined categories, based on its content. With the growth of the Internet and advances of computer technologies, more textual documents have been digitized and stored electronically, and text classification has become an increasingly important task. The applications of text classification range from Web page indexing, to document content filtering, information security, customer survey coding and help desk automation.

Over the years, a number of machine learning algorithms have been successfully used in text classification problems [11]. Among them, naïve Bayes [7], decision tree [8] and boosting [10], Racchio [9], Support Vector Machines [2] are the most popular. In this paper, a content vector model for text classification is proposed. It builds multiple augmented category Latent Semantic Indexing (LSI) spaces and classifies documents into categories by using their content vectors projected in these spaces. The approach has been applied to a special yet important and challenging text classification problem – email spam filtering – and the experiments of the classifier on a benchmark email testing corpus PU1 are presented. The rest of the paper is organized as follows. In Section II, LSI is briefly introduced and its original reference is provided. In Section III, the new content vector model for text classification is described, and its experiments on PU1 and a performance comparison with the SVM and naïve Bayes approaches are presented in Section IV. Some concluding remarks are provided in Section V.

Manuscript received April 17, 2006. This work was supported in part by an FRG grant from the University of San Diego, 2005-2006.

E. Jiang is with the University of San Diego, San Diego, CA 92110 USA (phone: 619-260-5956; fax: 619-260-4293; e-mail: jiang@sandiego.edu).

II. Latent Semantic Indexing

As a vector space model for information retrieval (IR), LSI [3] employs a rank-reduced term-document space through the singular vector decomposition (SVD) [5], and effectively, it transforms individual documents into their content vectors in the space to estimate the major associative patterns of terms and documents and to diminish the obscuring noise in term usage. Since the search in the space is based on the semantic content of documents, this approach is capable of retrieving relevant documents even when the query and such documents do not share any common terms.

Several variations of the LSI model have been proposed recently. For instance, an enhanced LSI implementation, which updates LSI by performing nonlinear perturbations to the LSI space, has been developed in [6] and it represents a more accurate semantic model for effective information retrieval.

III. Content Vector Model

LSI can be used as a learning algorithm for text classification by replacing the notion of query-relevance with the notion of category-membership. An experiment of this approach on an email corpus, Ling-Spam, was reported in [4]. However, there is a need to fully justify the validity of this approach as a competitive text classifier and to investigate the practicability of the approach in incorporating further space dimensionality reduction and category discriminative information in the training data. First, as pointed out recently by the author of Ling-Spam [1], the performance of a learning based spam filter on Ling-Spam can be over-optimistic because all legitimate messages in the corpus are topic-specific, and hence it may not reflect the performance that can be achieved on the incoming messages of a real email user. Secondly, the SVD computation can be computationally expensive for large data sets, and the exploration of additional dimensionality reduction with LSI is particularly valuable in order to make it a viable text classifier. This can be accomplished by reducing both sizes of the feature set and the training data set. Thirdly, LSI itself is a completely unsupervised learning algorithm and when it is applied to the (supervised) text classification problem, the valuable existing class discrimination information in the training data should be utilized and integrated in the model learning. Lastly, certain text classification problems are cost sensitive in the sense that the misclassifications of some categories carry a higher cost than others. Email classification is such an example. The potential utilization of category semantic spaces in this aspect is worth investigating. In this section, the proposed content

vector model is described in terms of its structure and major components.

A. Feature Selection

Two separate levels of feature selection have been used in our model for dimensionality reduction. In this paper, a term, or a feature is referred to as a word, a number, a symbol, or simply a punctuation mark. Dimensionality reduction aims to trim down the number of terms to be modeled while the content of individual documents is preserved.

First, features are selected in an unsupervised setting. The process is carried out by removing the stop or common terms and applying a term stemming procedure. Then, the terms with low document frequencies or low global frequencies are eliminated from the training data, as these terms may not help much in differentiating documents for categories and instead they can add some obscuring noises in documents classification. The selection process also removes the terms with very high global frequencies in the training data. The high frequent terms can mislead the classification process in our model due to the tf portion of the weighting scheme (see the next subsection) and might not be valuable in characterizing documents in different categories.

Next, features are selected by their frequency distributions among documents in the training data. This supervised feature selection step intends to, through those classified documents in the training data, further identify the features that distribute most differently among categories. Our model uses Information Gain (IG) [13] in this selection process. The measure IG quantifies the amount of information gained for category prediction by the knowledge of the presence or absence of a term in a document. More specifically, the Information Gain of a term T about a category C can be defined as

$$IG(T,C) = \sum_{c\in\{C,\bar{C}\}} \sum_{t\in(T,\bar{T})} P(t,c)\log\frac{P(t,c)}{P(t)P(c)}$$

where $P(c)$ and $P(t)$ denotes the probability that a document belongs to the category c and the probability of t in a document, respectively, and $P(t, c)$ is the joint probability of t and c.

B. Document Vector and Term Weighting

After feature selection, each document is encoded as a numeric vector whose elements are the values of the retained feature set. Each term value is associated with a local and global term weight, representing the relative importance of the term in the document and the overall importance of the term in the corpus, respectively. It is our belief that term frequencies can be more informative than the simple binary coding for document classification.

There are several choices to weight a term locally and globally based on its frequencies. Some preliminary experiments we performed on several weighting combinations have indicated that the traditional log(tf)-idf weighting scheme [6] produces very satisfactory performance and was used in our experiments.

C. Augmented Category LSI Spaces

For a given document set, LSI builds a rank-reduced vector space that characterizes the principal correlations of terms and documents in the set. In regard to document classification, multiple LSI spaces can be constructed, one for each category, and each of the spaces is constructed from the only documents of one category. It is assumed that a pure category-based LSI space offers a more accurate content profile, and more representative term and document correlations for the category. In practice, however, this approach may not work very well because documents from different categories can be quite similar and difficult to be distinguished from each other. It is especially true in email filtering. Many spam messages are purposely written in a way to have legitimate looks and to mislead spam filters. In our model, assuming that the training data have been separated into individual category sets, this problem is ameliorated by augmenting each of the category sets to include a small number of training samples that are most close to the category set but belong to other categories. Because of their closeness to a category set, the new documents that are similar to those augmented samples are prone to be misclassified in the LSI spaces built from pure category sets but can be correctly classified in the LSI space built from the augmented category sets. The similar strategy has been used in data compression [12]. Each of the augmented category sets is then used to build the corresponding semantic space for document classification.

In this work, the expansion of the category training sets is carried out by clustering the sets and finding their corresponding cluster centroids for sample comparison and selection. Given a set D of documents and their vector representations, the centroid c of D is a vector computed by averaging the term weights in D as:

$$c = \frac{1}{size(D)}\sum_{d\in D} d \qquad (1)$$

For a category set, once its centroid is formed, all samples from other category sets are compared against to the centroid, and the most similar samples are selected and added to the category set. The similarity between a sample document d and a centroid c is measured by their cosine value as:

$$\cos(d,c) = \frac{d\cdot c}{\|d\|\times\|c\|} \qquad (2)$$

Assume that a category set contain documents of the same topic, the centroid c in (1) provides an effective mechanism to summarize the topic of documents in the set. Since the clustering is done after feature selection, the centroid itself is an encoded content vector representing the most important retained features within the set. The cosine similarity in (2) offers a comparison of a sample with all documents in the set where the centroid is constructed. Mathematically, it measures an average similarity between the sample and these documents.

A variant of the well-known k-means clustering algorithm is used in our model. The sizes of augmented samples to

category sets can vary depending on the corpus and the space dimensionality, and in our experiments presented in Section IV, a unified augmented sample size 18 is used. A preliminary analysis on our model has indicated that the size of augmented samples to a category set has certain impact on the classification accuracy for the category. This characteristic can potentially be used as a powerful control in some cost sensitive classification problems such as spam filtering (see Section IV).

It should be pointed out that the proposed model of using augmented category-based semantic spaces is effective in integrating the most valuable class discrimination information into the LSI learning and characterizing the principal semantic content structures of each category. Each training sample document is represented as a content vector in one or more category spaces and collectively, these content vectors profile themes of categories. The multi-space configuration is embedded in our classification algorithms, which are described in the next subsection, also helps improve the classification accuracy of incoming documents. This model requires the construction of multiple LSI spaces. However, the dimensionality of each space in this model is substantially reduced and this can be especially useful for dealing with a large training set.

D. Document Classification

To classify incoming documents, three document classification algorithms are considered. Each of them treats incoming documents as individual queries and utilizes the embedded class discrimination information in the model. The first algorithm is simple, and uses the most semantically similar document in the training data, which is determined by the content vectors in all augmented LSI spaces, to classify incoming documents. The algorithm is referred to as *Single*. It is computationally efficient. However, it can be less accurate as some documents from different categories can have similar looks. The second algorithm classifies incoming documents by using a group of the top m most similar samples in the training data compared in the LSI spaces. The counts or sums of cosine similarity values, of category-labeled sample documents in the group make the classification decision. The algorithm uses the latter approach is referred to as *Multiple*. The third message classification algorithm is a hybrid approach that combines the ideas of *Single* and *Multiple* with the hope to mollify some problems associated with the algorithms. This algorithm is named as *Hybrid*. It has a few parameters that are set heuristically, and can be configured by the user, depending upon users' tolerance level to potential misclassification errors for certain categories.

IV. Experiments

The proposed content vector model has been applied to email filtering, a special two-category text classification problem. In this section, the experiments of the model on the benchmark spam testing corpus PU1 are presented. A comparison with the SVM and naïve Bayes classifiers on the same corpus is also provided.

A. Performance Evaluation

As in general text classification, the performance of a spam filter can be evaluated by both spam and legitimate precisions and recalls. In brief, the precision is gauged by the percentage of messages classified to a category which actually are, whereas the recall is quantified by the percentage of messages from a category that are categorized by the classifier. These measurements, however, do not take an unbalanced misclassification cost into consideration. Spam filtering can be a cost sensitive learning process in the sense that misclassifying a legitimate message to spam is typically a more severe error than misclassifying a spam message to legitimate. In this paper, a cost-sensitive and unified weighted accuracy [1] is used as a performance criterion and it can be defined as

$$WAcc(\lambda) = \frac{\lambda n_{L->L} + n_{S->S}}{\lambda(n_{L->L} + n_{L->S}) + (n_{S->S} + n_{S->L})}$$

where $n_{L->L}$, $n_{L->S}$, $n_{S->S}$ and $n_{S->L}$ denotes the count of the classification $L->L$ (legitimate classified as legitimate), $L->S$ (legitimate misclassified as spam), $S->S$ (spam classified as spam), and $S->L$ (spam misclassified as legitimate), respectively, and λ is a cost parameter. The *WAcc* formula assumes that the error of $L->S$ is λ times more costly than the error of $S->L$. In our experiments, $\lambda = 1$ ($L->S$ and $S->L$ have the same cost) and $\lambda = 9$ ($L->S$ has a higher cost than $S->L$) are used. The setting of $\lambda = 999$ has also been proposed in literature. However, this setting can be inaccurate when the training data are not large enough. For this reason, the setting is not used here.

B. Experiments on PU1

PU1 is a benchmark spam testing corpus that was released recently [1]. It contains a total of 1099 real email messages, with 618 legitimate and 481 spam. The experiments on PU1 are performed using stratified 10-fold cross validation. More specifically, the PU1 corpus is partitioned into ten equally-sized subsets. Each experiment takes one subset for testing and the remaining for training, and the process repeats ten times with each subset takes a turn for testing. The performance is then evaluated by averaging over the ten experiments.

For text classifiers, the size of feature set can have an effect on their classification performance. Most of spam experiments have been reported in literature use relatively small feature sets. In our experiments, various feature set sizes have been used, with small ones ranging from 50 to 650 incremented by 100, and with large ones ranging from 1650 to 7650 incremented by 1000.

As a comparison to some other popular classifiers, our content vector model is evaluated against the Support Vector Machines (SVM) and naïve Bayes (NB) approaches. Both SVM and NB classifiers have been applied for email filtering.

In this work, the Weka[i] implementation of SVM and NB is used and the input data to both classifiers are the same as to our proposed model, namely, the processed set of message vectors after feature selection and term weighting.

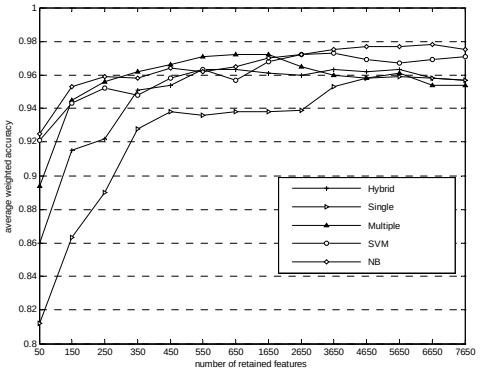

Fig. 1 Average weighted accuracy with $\lambda = 1$ (PU1)

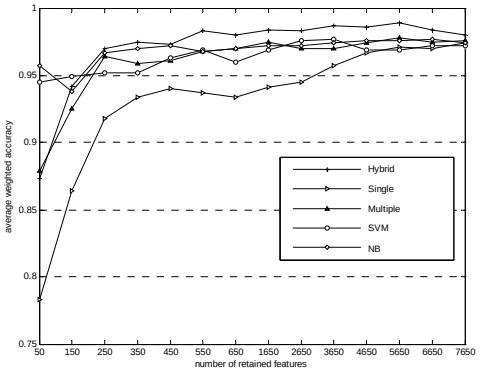

Fig. 2 Average weighted Accuracy with $\lambda = 9$ (PU1)

Fig. 1 and Fig. 2 show the average weighted accuracies *WAcc* with *Hybrid*, *Single*, *Multiple* of the content vector model, SVM and NB for the cost parameter $\lambda = 1$ and $\lambda = 9$, respectively. The results are obtained over all the feature sets that have been considered. For $\lambda = 1$, *Single* is clearly inferior to all other four algorithms that have most of their average accuracy values above 95%. Among these top four, *Multiple* is the best performer over small feature sets whereas NB and SVM are top-rated when large feature sets are considered. For $\lambda = 9$, all algorithms except *Single* deliver quite good performance. And for most of the feature sets, *Hybrid* consistently achieves the top weighted accuracy that peaks at 5650 with 98.9%. It is interesting to note that NB does extremely well on this corpus and likely, its performance is boosted by our feature selection process.

It can be observed that, for all classifiers except *Single*, there should be a minimum required size of feature set (say around 350) on this corpus to achieve an acceptable classification performance. A further performance analysis on the content vector model has also revealed that their precisions and recalls for both categories are generally improved as the feature set gets larger. However, when the size of a feature set reaches to a certain point (250 or 350), further enlarging feature set would produce higher spam precision and legitimate recall but lower or same spam recall and legitimate precision. This is especially noticeable for the *Hybrid* algorithm.

The experiments on PU1 have demonstrated that the proposed content vector model is very effective for spam detection and filtering, and represents a very competitive alternative to other well-known classifiers such as SVM and naïve Bayes…

V. Conclusion

As a rank-reduced vector space model, LSI has been successfully used in information retrieval and other applications. In this paper, an LSI-based content vector classification model is proposed that classifies documents by their semantic content. The model utilizes the valuable email discriminative information in the training data and incorporates several pertinent feature selection and document classification algorithms. The experiments of the model on an email testing corpus have shown that it is very effective in learning to classify spam email messages. The competitive performance of the proposed classifier is also demonstrated by comparing it with two popular classifiers: SVM and naïve Bayes. As future work, we plan to experiment the proposed classifier with general text classification corpora such as the Reuters-21578, and improve the accuracy and efficiency of the model by further exploring feature-document associations and investigating the optimal size setting of augmented training samples to be added to a category training set.

[1] Weka: www.cs.waikato.ac.nz/ml/weka/

References

[1] Androutsopoulos, G. Paliouras, and E. Michelakis (2004). "Learning to filter unsolicited commercial e-mail".Technical Report 2004/2, NCSR Demokritos.

[2] N. Christianini and J. Shawe-Taylor (2000). *An introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press.

[3] S. Deerwester, S. Dumais, G. Furnas, T. Landauer and R. Harshman (1990) "Indexing by Latent Semantic Analysis". *Journal of the American Society for Information Science*. 41, 391-409.

[4] K. Gee (2003). "Using Latent Semantic Indexing to Filter Spam". *Proceedings of the 2003 ACM Symposium on Applied Computing,* 460-464.

[5] G. Golub and C. Van Loan (1996). *Matrix Computations*. John-Hopkins, Baltimore, 3rd edition.

[6] E. Jiang and M. Berry (2000). "Solving Total Least-Squares Problems in Information Retrieval. *Linear Algebra and its Applications*, 316, 137-156.

[7] T. Mitchell (1997). *Machine Learning.* McGraw-Hill.

[8] J. Quinlan (1993). *C 4.5*: *Programs for Machine Learning.* Morgan Kaufmann.

[9] J, Rocchio (1971). "Relevance feedback information retrieval". *The Smart retrieval system-Experiments in automatic document processing,* (G. Salton ed.). Prentice-hall, 313-323.

[10] R. Schapier and Y. Singer (2000). "BoosTexter: a boosting-based system for text categorization". *Machine Learning,* 39, 2/3, 135-168.
[11] F. Sebastiani (2002). "Machine learning in automated text categorization". *ACM Computing Surveys* 334, 1, 1-47.
[12] H. Schutze, D.A. Hall and J.O. Pedersen (1995). "A Comparison of Classifiers and Document Representations for the Routing Problem". Proceedings of SIGIR, 1995, 229-237.
[13] Y. Yang and J. Pedersen (1997). "A comparative study on feature selection in text categorization". *Proceedings of the 14th International conference on Machine Learning,* 412-420.