Experiments on Supervised Learning Algorithms for Text Categorization

Setu Madhavi Namburu, Haiying Tu, Jianhui Luo and Krishna R. Pattipati
Dept. of ECE, University of Connecticut
Storrs, CT-06269, USA
860-486-2890
krishna@engr.uconn.edu

Abstract—Modern[1,2] information society is facing the challenge of handling massive volume of online documents, news, intelligence reports, and so on. How to use the information accurately and in a timely manner becomes a major concern in many areas. While the general information may also include images and voice, we focus on the categorization of text data in this paper. We provide a brief overview of the information processing flow for text categorization, and discuss two supervised learning algorithms, viz., support vector machines (SVM) and partial least squares (PLS), which have been successfully applied in other domains, e.g., fault diagnosis **Error! Reference source not found.**. While SVM has been well explored for binary classification and was reported as an efficient algorithm for text categorization, PLS has not yet been applied to text categorization. Our experiments are conducted on three data sets: Reuter's-21578 dataset about corporate mergers and data acquisitions (ACQ), WebKB and the 20-Newsgroups. Results show that the performance of PLS is comparable to SVM in text categorization. A major drawback of SVM for multi-class categorization is that it requires a voting scheme based on the results of pair-wise classification. PLS does not have this drawback and could be a better candidate for multi-class text categorization.

TABLE OF CONTENTS

1. INTRODUCTION

Text categorization or classification, which assigns text documents to pre-specified classes or topics, plays a key role in organizing the massive sources of unstructured text information into an organized format. In the modern world, there is a growing need for text categorization, because the sources of information to be managed, such as webpages, news sources, and database management, are increasing rapidly. Filtering of spam mails, quick search of interesting topics from large databases, and retrieving the information based on user's preferences from information sources, are some other examples where text categorization can play an important role. These sources can contain images, voice, and multimedia data besides text. Most research efforts have assumed that the text components are sufficient for classification tasks. If an automatic classification engine is developed, categorization task can be achieved with less cost and in less time, while improving analyst's productivity.

Many popular algorithms have been successfully applied to text categorization. Nigam et al. [1] investigated expectation maximization (EM) algorithm for text classification from labeled and unlabeled documents. Zelikovitz et al. [3] applied latent semantic indexing (LSI) using background text. Joachims [8] applied transductive support vector machines (TSVM) for text categorization. Craven et al. [28] explored learning from symbolic knowledge. Among supervised learning algorithms, Naive Bayes and SVM have been well explored, and have proven to be promising techniques for text categorization [2] [7] [15] [17] [18].

A text categorization task using supervised learning algorithms involves a training phase and a test phase. The training phase of text categorization involves feature extraction and indexing processes. The vector space model has been used as the conventional method for representing text [5] [23] [24]. In these models, each document contains a feature vector extracted from the initial text of the

[1] 0-7803-8870-4/05/$20.00© 2005 IEEE

[2] IEEEAC paper #1260, Version 8, Updated December 10, 2004

document. Feature indexing will result in a set of consistent features for all the documents. Most text classification problems are linearly separable in very high dimensional spaces. Inherently, text classification problems have high dimensionality with very sparse representation [17], because each unique word in a corpus represents a feature and each document contains only a small percentage of the overall feature space. Once the data is prepared, learning algorithms are applied in the training phase to create a model. The model is then used by the classifier to test on the new data during the test phase (see Figure 1).

In this paper, we focus on two supervised learning algorithms, support vector machines (SVM) and partial least squares (PLS), which have been successfully applied in fault diagnosis **Error! Reference source not found.**. Even though SVM has been well explored for text classification, PLS has not yet been investigated. Our preliminary experimental results on three data sets show that PLS is comparable to SVM in performance, and could be a better candidate for multi-class categorization. This is because SVM requires a voting scheme based on results of pair-wise classification for the multi-class problem, while PLS does not have this limitation. Another advantage of PLS is that it represents the data in a lower dimensional space.

We will compare the performance of PLS with SVM. We also list the results associated with the same data sets from some previous research papers. Our initial experiments are conducted on Rueters-21578 data set using ModApte split about corporate mergers and data acquisitions (ACQ). Later we utilized two other data sets, WebKB and the 20-News-groups, to compare the classification performance of our schemes.

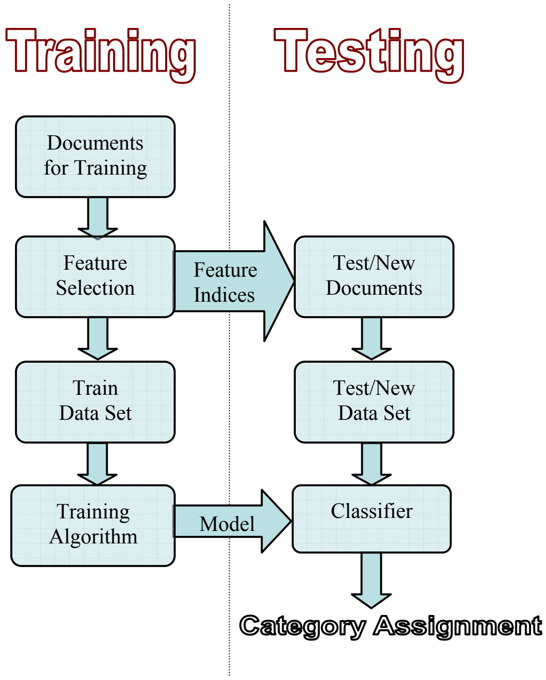
The rest of the paper is organized as follows. Section 2 presents an overview of text categorization task and our proposed scheme. Section 3 provides a brief summary of the SVM and PLS algorithms, and how they are applied to text classification. Section 4 describes the data sets we used. Section 5 reports on the results with a discussion. Finally, we conclude the paper in section 6.

2. TEXT CATEGORIZATION OVERVIEW

The goal of text categorization is to classify a set of documents into a fixed number of predefined categories. Each document may belong to more than one class. Using supervised learning algorithms, the objective is to learn classifiers from known examples (labeled documents) and perform the classification automatically on unknown examples (unlabeled documents).

Figure.1 shows the overall flow diagram of the text categorization task. Consider a set of labeled documents from a source $D=[d_1,d_2,\dots d_n]$ belonging to a set of classes $C=[c_1,c_2,\dots,c_p]$. The text categorization task is to train the classifier using these documents, and assign categories to new documents. In the training phase, the n documents are arranged in p separate folders, where each folder corresponds to one class. In the next step, the training data set is prepared via a feature selection process.

Figure 1. Flow Diagram of Text Categorization



Text data typically consists of strings of characters, which are transformed into a representation suitable for learning. It is observed from previous research that words work well as features for many text categorization tasks. In the feature space representation, the sequences of characters of text documents are represented as sequence of words. Feature selection involves tokenizing the text, indexing and feature space reduction. Text can be tokenized using term frequency (TF), inverse document frequency (IDF), term frequency inverse document frequency (TFIDF) [27], or using binary representation. Using these representations the global feature space is determined from entire training document collection.

Feature selection plays a major role in achieving better classification performance. A significant amount of research has been done on feature selection for better representation of data for text categorization. Nigam et al. [1] compared the performance of naive bayes (NB) and expectation maximization (EM) using labeled and unlabeled documents. Madsen and Hansen [4] used part-of-speech tagger for feature representation. Yang and Pedersen [26] have performed a comparative study of feature selection in text categorization. Ko et al. [22] performed text categorization based on importance of sentences.

In the case of very large text collections, storage space and efficiency of handling the data become a major concern. Consequently, feature reduction is a must. Feature space reduction in a feature selection process improves the accuracy of the learning algorithm performance, decreases the data size, controls the classification time, and avoids overfitting of the data. There are many ways of feature space reduction such as stemming, stop-word removal, and using information gain or mutual information criteria [27]. An alternative way of forming the dataset is to consider all the labeled documents and split them into training and test sets after feature selection.

In our experiments, we used the Rainbow toolkit [12] for feature selection. We used the term-frequency (TF) representation of features, and also compared the performance of SVM and PLS with term-frequency-inverse-document-frequency (TFIDF) representation for the WebKB and the 20-Newsgroups data sets. Each document, d_j is represented by the TF vector $\underline{f}_j = (f_{1j}, f_{2j}, \dots f_{sj})$, where f_{ij} is the frequency of the i^{th} term in document d_j, or the TFIDF vector $\underline{g}_j = (g_{1j}, g_{2j}, \dots, g_{sj})$, where g_{ij} represents TFIDF value of the i^{th} term given by

$$g_{ij} = f_{ij} \log_2 \left(\frac{n}{n_i} + 1 \right) \tag{1}$$

Here n_i is the number of documents containing term i.

The above feature space represents the entire term space of dimension s. We used Mutual Information (MI) [25] measure for feature space reduction. The mutual information between a feature, F and a category, Y is defined as:

$$MI(F,Y) = \sum_{F \in \{f, \overline{f}\}} \sum_{Y \in \{C\}} P(F,Y) \log \frac{P(F,Y)}{P(F)P(Y)} \tag{2}$$

We selected the top m features that have the highest mutual information within each category, thereby reducing the feature space into m dimensions. The result is a feature-document matrix $A = [a_{ij}]$ of dimension m by n, where the elements represent the TF or the TFIDF of the corpus of documents. In the next step, the training algorithm (SVM, PLS) is applied on the data set. The classifier uses the trained model to categorize new documents during the test phase. In this paper, we assumed that each document belongs to only one class.

3. Description Of Algorithms

Support Vector Machines (SVM): Support vector machine (SVM), as a statistical learning theory, has gained popularity in recent years because of its two distinct features. First, SVM is often associated with the physical meaning of the data, so that it is easy to interpret. Second, it requires only a small number of training samples. The SVM has been successfully used in many applications, such as pattern recognition, multiple regression, nonlinear model fitting, and fault diagnosis.

The essential idea of SVM classification is to transform the input data to a higher dimensional feature space, and find an optimal hyperplane that maximizes the margin between the classes. The group of examples that lie closest to the separating hyperplane is referred to as support vectors. We implemented a generalized form of SVM for overlapped and nonlinearly separable data. Consider a feature-document matrix, A with m features (as rows) and n documents (as columns). Each column, denoting a document, belongs to one of two classes: a target class or non-target class. Briefly, the training data for the two classes is arranged as

$$\begin{gathered} F = \left[\left(\underline{\mathrm{a}}_1, y_1 \right), \left(\underline{\mathrm{a}}_2, y_2 \right), \cdots, \left(\underline{\mathrm{a}}_n, y_n \right) \right], \\ \underline{\mathrm{a}}_i \in R^m, \;\; y_i \in \{-1, \; 1\} \end{gathered} \tag{3}$$

where R^m is the m-dimensional feature space.

y_i is the class label with

$$y_i = \begin{cases} 1 & \text{if } \underline{a}_i \text{ belongs to the target class} \\ -1 & \text{otherwise} \end{cases}$$

For non-separable case, a separating hyperplane must satisfy the following constraints

$$y_i [\langle \underline{\mathrm{w}}, \underline{a}_i \rangle + b] \;\geq\; 1 - \xi_i, \; i = 1, 2, ..., n \tag{4}$$

where $\xi_i \geq 0$ is the slack variable and $\langle . \, , \, . \rangle$ denotes the dot product. To determine the vector w and scalar b, the following function is minimized:

$$\Phi \left(\underline{\mathrm{w}}, \xi \right) = \frac{1}{2} \left\| \mathrm{w} \right\|^2 + G \sum_{i=1}^{n} \xi_i \tag{5}$$

subject to the constraints in (4). In the above equation the first and second terms represent model complexity and model accuracy respectively. G is a regularization parameter to control the trade-off between these two terms. The solution of (5) is given by the following dual optimization problem [30][31]:

$$\underset{\underline{\alpha}}{\text{maximize}}\ W\left(\underline{\alpha}\right)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\underline{a}_i^T,\underline{a}_j)\alpha_i\alpha_j y_i y_j,$$

$$\text{subjet to: }\sum_{i=1}^{n}\alpha_i y_i=0,\alpha_i\in\left[0,G\right],\ i=1,2,...,n. \tag{6}$$

If a nonlinear mapping $K\left(\underline{a}_i,\underline{a}_j\right)$ is chosen *a priori*, the optimization problem of (6) becomes,

$$\underset{\underline{\alpha}}{\text{maximize}}\ W\left(\underline{\alpha}\right)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K\left(\underline{a}_i,\underline{a}_j\right) \tag{7}$$

subject to the same set of constraints. The nonlinear mapping (or kernel function) K is used to transform the original input $\underline{a}$ to a feature space Ω via $K\left(\underline{a}_i,\underline{a}_j\right)=\left\langle\varphi\left(\underline{a}_i\right),\varphi\left(\underline{a}_j\right)\right\rangle$. The decision function becomes

$$f\left(\underline{\mathrm{a}}\right)=\operatorname{sgn}\left(\sum_{i\in SVs}\alpha_i y_i K\left(\underline{a}_i,\underline{a}\right)+b\right). \tag{8}$$

where SVs is an index set, which contains the indices of the support vectors. In practice, various kernel functions are used, such as linear, polynomial, radial basis functions (RBF), and sigmoid functions. In this paper, we use the RBF kernel functions, $K\left(\underline{a}_i,\underline{a}_j\right)=\exp\left(-\left\|\underline{a}_i-\underline{a}_j\right\|^2/2\gamma^2\right)$. Here, ξ,G,γ are the design parameters for the SVM. In this paper the parameters used for the experiments are G and γ and these are determined through a simple grid-search using cross-validation. For multi-class categorization, classifiers are trained pair wise and voting is used to select the class in the test phase.

Partial Least Squares (PLS): Partial least squares, also known as projection to latent structures, is a dimensionality reduction technique for maximizing the covariance between the $n\times m$ independent training data matrix X=A^T (the transpose of feature-document matrix) and the $n\times p$ dependent matrix Y (corresponding to p text categories) for each component of the reduced space. It builds a regression model between X and Y. While forming the data sets, in each row of Y the class label corresponding to the document in X is represented by 1 in the corresponding column of Y. PLS is applicable when the terms are large-sized and redundant, when the data set has missing values, and when there is no well-understood relationship between the terms (features) and document class variables but categorization is the main goal [10].

PLS generates uncorrelated latent variables ("concepts"), which are linear combinations of the original terms. The basic idea is to select the weights of the linear combination to be proportional to the covariance between the terms and document classes. Once the concepts are extracted, a least squares regression is performed to estimate the document class. Both matrices X and Y are decomposed into a number of concepts (called components in the PLS parlance), which is known as the model reduction order, plus residuals. The reduction order (i.e., number of concepts) is determined by cross-validation. The decompositions are given by

$$X=\sum_{i=1}^{k}\underline{t}_i\underline{p}_i^T+E=TP^T+E \tag{9}$$

$$T\in R^{n\times k},P\in R^{m\times k},E\in R^{n\times m}$$

$$Y=\sum_{i=i}^{k}b_i\underline{t}_i\underline{q}_i^T+F=UQ^T+F \tag{10}$$

$$U=TDiag(b_i)\in R^{n\times k},Q\in R^{p\times k},F\in R^{n\times p}$$

Here, T, U are score matrices (latent vectors); P, Q are loading matrices; E, F are residual matrices, and k is the model reduction order (no of concepts).

PLS may be viewed as a two-phase optimization problem. For simplicity, assume that the response matrix Y is a single column, $\underline{y}\in R^n$. In the first phase, for a given latent vector $\underline{t}\in R^n$, PLS seeks to find a rank 1-matrix $\underline{t}\,\underline{p}^T$ that is closest to X in Forbenius norm, i.e., $\min_{\underline{p}}\left\|X-\underline{t}\,\underline{p}^T\right\|_F$. The solution is given by: $\underline{p}=\frac{X^T\underline{t}}{\underline{t}^T\underline{t}}$. In the second phase, PLS seeks to find $\underline{t}$ and a weight vector $\underline{w}$ such that the covariance between $\underline{t}$ and $\underline{y}$ is maximized:

$$\begin{aligned}&\max_{\underline{t},\underline{w}}\ \underline{t}^T\underline{y}\\&\text{s.t. }\underline{t}=X\underline{w}\\&\quad\underline{w}^T\underline{w}=1\end{aligned} \tag{11}$$

The result is:

$$\underline{w} = \frac{X^T \underline{y}}{\left\| X^T \underline{y} \right\|_2} \propto \underline{p} \quad \text{and} \quad \underline{t} = \frac{X X^T \underline{y}}{\left\| X^T \underline{y} \right\|_2} \qquad (12)$$

Since $\underline{t}$ and $\underline{w}$ (or $\underline{p}$) depend on each other, iteration is required. The score and loading vectors are determined using the nonlinear iterative partial least squares (NIPALS) algorithm [9].

4. Datasets

Reuter's-21578 Dataset: We have conducted our initial experiments on the Reuter's-21578 dataset about corporate mergers and data acquisitions (ACQ) with ModApte split [21]. After feature selection, the data set consists of 9947 features and 2600 documents belonging to two classes: ACQ, and not ACQ. We used 2000 documents for training and tested on the remaining 598 documents after removing documents with empty feature vectors.

WebKB Dataset: WebKB collection [20] consists of WWW pages provided by the carnegie mellon university (CMU) text learning group. The web pages were manually classified into 7 classes: course, department, faculty, other, project, staff, and student. We performed our experiments on the most popular categories used by previous researchers: course, faculty, project, and student. The resulting data set consists of 4199 documents. The headers and html tags are removed. We did stop-word removal for feature space reduction. To be consistent with the work done in [1], [3] and [7], we used only 300 features with the highest mutual information for each category, and removed the documents with empty feature vectors.

20 News Groups Dataset: The 20-Newsgroups data set [12] consists of Usenet articles from 20 different news groups. Each group contains approximately 1000 documents. In this paper, we considered four sub groups of the 20-Newsgroups corpus related to computers, recreation, science and talk as separate data sets. It is observed from the previous research that these are the most confusable clusters for the categorization task. Computers subgroup consists of documents from five classes, while the other three subgroups consist of documents from four classes each. The headers and html tags are removed while tokenizing the features from documents. The top 5000 features with highest mutual information for each category are used as feature vectors for each subgroup.

5. Experimental Results And Discussion

Different methods are adapted for each data set to show the performance measures of the classifiers. In the following, we discuss the results for each dataset considered in this paper.

Reuter's-21578 Dataset: Classification results for the Reuter's-21578, 1.0 distribution dataset about ACQ are shown in Table 1 along with the tuning parameters used. The accuracy measure shown here is Precision/Recall-Break even point. Precision is the ratio of the number of correct positive predictions to the number of positive predictions. Recall is the ratio of the number of correct positive predictions to the total number of examples. The Break-even point is where precision and recall are equal. We used this accuracy measure for this particular data set in order to be consistent with the previous results presented on this data.

WebKB Dataset: The documents for each class consist of four different universities documents and miscellaneous documents. Leave-one-university-out cross-validation, i.e., testing on one university documents and training on the miscellaneous and the other three university documents, is performed. Depending on the university, the test documents varied from 226 to 308. A simple grid search is done for different values of parameters of SVM and PLS for each run of the cross-validation. Best parameters are selected based on the maximum average accuracy. Table 2 shows the performance of SVM and PLS classifiers for the WebKB data with TF and TFIDF representations, along with the parameters used. The results shown for both classifiers are average accuracies over the four cross-validation runs. It also lists some of the results from previous research papers based on naive bayes (NB) [7], expectation maximization (EM) [1], and latent semantic indexing with back ground text (LSI-bg) [3].

Table 1. Precision/Recall-Breakeven point for the Reuter's ACQ Data

Classifier	Parameters	ACQ
SVM	γ =1.2 $G = 5$	**96.1**
PLS	$k = 11$	**96.32**

A note of caution is pertinent on the results. The three researchers used different feature selection methods and also different numbers of training examples. The result shown here for the EM from [1] was with 240 labeled documents, 2500 unlabeled documents and the top 300 words as features. The result shown for LSI-bg [3] was with 200 training examples with background text of 2500 documents and the top 300 words as features. For NB [7], a multi-variate Bernoulli model was used with top 100 words and 70% as training data. With TFIDF feature representation, PLS performance improved by 2.6%. As opposed to PLS, SVM's performance decreased with TFIDF representation of features in this case.

Table 2. Classification Results for WebKB data along with the results from previous research.

Classifier	**Accuracy (%)**	
SVM	TF $\gamma = 0.00001$ G =1000	TFIDF $\gamma = 0.000001$ G = 1000
	90.21	89.92
PLS	TF $k = 24$	TFIDF $k = 24$
	87.00	**89.6**
NB	87.00	
EM	82.00	
LSI-bg	75.56	

Table 3. SVM and PLS classification results on four sub-groups of 20-Newsgroups data

Group Name	**SVM (%)**		**PLS (%)**	
	TF	TFIDF	TF	TFIDF
Computers	**74.85** (γ = 0.00002, G = 100)	**77.52** (γ = 0.00001, G = 100)	**71.63** (k = 38)	**75.69** (k = 30)
Recreation	**92.22** (γ = 0.0005, G = 100)	**93.43** (γ = 0.00001, G = 50)	**91.74** (k = 20)	**94.15** (k = 19)
Science	**87.94** (γ = 0.00005, G = 1000)	**89.12** (γ = 0.00001, G = 100)	**86.79** (k = 20)	**91.17** (k = 17)
Talk	**76.87** (γ = 0.0001, G = 100)	**79.47** (γ = 0.00001, G = 50)	**77.87** (k =26)	**81.85** (k = 19)

20 News Groups Dataset: Table 3 shows the classification performance of SVM and PLS classifiers on four subgroups of the 20-Newsgroups data along with the parameters used. The results shown are average accuracies over two-fold cross-validation. Their performance is compared using both the TF and TFIDF representations of features. It is observed that the TFIDF representation consistently improved the performance of both classifiers.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a brief overview of the text categorization task. We applied two supervised learning algorithms, SVM and PLS, for text categorization. We also compared their performance with two representations of the data (TF, TFIDF).

While SVM has been well explored for binary classification and was reported as an efficient algorithm for text categorization, PLS has not yet been applied to text categorization. A major drawback of SVM for multi-class categorization is that it requires a voting scheme based on pair-wise classification results. PLS does not have this drawback and could be a better candidate for multi-class categorization. Another advantage of PLS is that it represents the data in a lower dimensional space. Our experimental results on three data sets: Reuter's-21578 data about ACQ, WebKB and 20-Newsgroups, showed that the performance of PLS is competitive with SVM in text categorization.

In future, we plan to conduct more experiments on SVM and PLS with multi-class documents and a large number of single class documents with different forms of feature representation.

7. ACKNOWLEDGEMENTS

We thank Dr. A. McCallum for making the Rainbow toolkit available online. We also thank the CMU text learning group and Dr. David. D. Lewis for collections of the datasets. We thank Sumit Narayan for helping with Rainbow software in retrieving the data from text documents. This work is supported by DARPA.

REFERENCES

[1] Nigam, K., McCallum, A., Thrun, S., and Mitchell, T., “Text Classification from Labeled and Unlabeled Documents using EM”, Machine Learning, 39(2/3), 2000, pp. 103-134.

[2] Sun, A., Lim, E.-P., and Ng, W.-K., "Web Classification using Support Vector Machine", In Proceedings of the fourth international workshop on Web information and data management, ACM Press, 2002, pp. 96-99.

[3] Zelikovitz, S. and Hirsh, H., “Using LSI for Text Classification in the Presence of Background Text”, Proceedings for the Conference on Information and Knowledge Management (CIKM), 2001.

[4] Madsen, R. E. and Hansen, L. K., “Part-of-Speech Enhanced Context Recognition”, MLSP, 2004.

[5] Dumais, S. T., Platt, J., Heckerman, D., and Sahami, M., “Inductive Learning Algorithms and Representations for Text Categorization”, In Proceedings of ACM-CIKM98, Nov. 1998, pp. 148-155.

[6] Bingham, E., and Mannila, H., “Random Projection in Dimensionality Reduction: Applications to Image and Text Data”, Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001), August 26-29, 2001, San Francisco, CA, USA, pp. 245-250.

[7] McCallum, A., and Nigam, K., “A Comparison of Event Models for Naive Bayes Text Classification”, In AAAI/ICML-98 Workshop on Learning for Text Categorization, Technical Report WS-98-05, AAAI Press. 1998, pp. 41-48.

[8] Joachims, T., “Transductive Inference for Text Classification Using Support Vector Machines”, Proceedings of the 16th International Conference on Machine Learning, 1999, pages 200-209.

[9] Geladi, P., and Kowalski, B. R., “Partial Least-Squares Regression: A Totorial”, Anal. Chim. Acta, 185, 1985, pp-1-17.

[10] Tobias, R. D., “An Introduction to Partial Least Squares Regression”, TS-509, SAS Institute Inc., Cary, N.C., April 1997.

[11] Rasmus, B., “Multiway Calibration. Multilinear PLS”, Vol 10, 1996, pp 47-61.

[12] McCallum, A. K., “Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering”, http://www.cs.cmu.edu/#mccallum/bow, 1996.

[13] Joachims, T., “A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization”, Proceedings of International Conference on Machine Learning (ICML), 1997.

[14] Fraggoudis, D., Meretakis, D., and Likothanassis, S., “Integrating Feature and Instance Selection for Text Classification”, SIGKDD 2002, pp. 501-506.

[15] Rennie, J. D. M., Shi, L., Teevan, J., and Karger, D. R., “Tackling the Poor Assumptions of Naive Bayes Text Classifiers”, ICML 2003.

[16] Lewis, D. D., and Ringuette, M., “A Comparision of Two Learning Algorithms for Text Categorization”, Proceedings of SDAIR, 3rd Annual Symposium on Document Analysis and Information Retrieval, 1994.

[17] Joachims, T., “Text Categorization with Support Vector Machines: Learning with Many Relevant Features”, In Proceedings of the Tenth European Conference on Machine Learning, 1998, pages 137-142.

[18] Cooley, R., “Classification of News Stories using Support Vector Machines”, In IJCAI'99 Workshop on Text Mining, Stockholm, Sweden, August 1999.

[19] Liao, C., Alpha, S., and Dixon, P., “Feature Preparation in Text Categorization”, ADM03 workshop (The Australian Data Mining Workshop), Canberra, Australia, 9 Dec., 2003.

[20] WebKB Dataset: http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/.

[21] Reuter’s Dataset : http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.

[22] Ko, Y., Park, J., and Seo, J., "Automatic Text Categorization using the Importance of Sentences", COLING 2002.

[23] Salton, G., Fox, E. A., and Wu, W., "Extended Boolean information retrieval", Communications of the ACM 26 (12), 1983, pp.1022-1036.

[24] Salton G., "Automatic Text Processing: The Transformation, Analysis, and Retreival of Information by Computer", Addison-Wesley, 1989.

[25] Cover, T. M., and Thomas, J. A., "Elements of Information Theory", Wiley, 1991.

[26] Yang, Y., and Pedersen, J. O., "A Comparative Study on Feature Selection in Text Categorization", In Proc. of ICML-97, 14th International Conf. On machine Learning (Nashville, USA), 1997, pp. 412-420.

[27] Salton, G., and Buckley, C., "Term Weighting Approaches in Automatic Text Retrieval", Information Processing and Management, 24(5), 1988, pp.513-523.

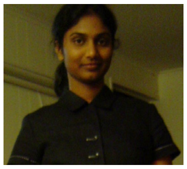
[28] Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., and Slattery, S., "Learning to Extract Symbolic Knowledge from the World Wide Web", In Proceedings of the Fifteenth National Conference on Artificial Intelligence, 1998.

[29] Ge, M., Du, R., Zhang, G., and Xu, Y., "Fault Diagnosis using Support Vector Machine with an Application in Sheet Metal Stamping Operations," Mechanical Systems and Signal Processing, Vol 18, 2004, pp.143-159.

[30] Christopher J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, Vol.2, 1998, pp.121-167.

[31] Smola, A. J, Barlett, P. L, Scholkopf, B and Schuurmans, D, "Advances in Large Margin Classifiers", Cambridge, Massachusetts: The MIT Press, 2000.

BIOGRAPHY



Setu Madhavi Namburu received her B.Tech degree in Electrical and Electronics Engineering from Jawaharlal Nehru Technological University, India, in 2002. She is presently pursuing her Master's degree in Electrical and Computer Engineering Department at the University of Connecticut. Her research interests include systems analysis, simulation, classification algorithms, model-based and data-driven fault diagnosis of engineering systems.

Haiying Tu received the BS degree in automatic control from Shanghai Institute of Railway Technology in 1993 and MS in transportation information engineering and control from Shanghai Tiedao University in 1996. She is currently a Ph.D. student of Electrical and Computer Engineering at the University of Connecticut (UCONN). Prior to joining UCONN, she was a lecturer of Tongji University in Shanghai, China and also worked as an employee of Computer Interlocking System Testing Center which belongs to the Ministry of Railway of China. Her current research interests include organizational design, Bayesian analysis, fault diagnosis and decision making.

Jianhui Luo is a graduate student in the Electrical and Computer Engineering Department at the University of Connecticut. Previously he was at Shanghai Institute of Railway Technology, China, from which he received his BASc in Automatic Control in 1993. He worked as a design engineer, and later deputy department head in CASCO Signal Inc. from 1993 to 2000 in Shanghai, China. His primary research interests are in the areas of system simulation, model-based system fault diagnosis, and safety critical system analysis.

Krishna R. Pattipati is a Professor of Electrical and Computer Engineering at the University of Connecticut, Storrs, CT, USA. He has published over 285 articles, primarily in the application of systems theory and optimization techniques to large-scale systems. Prof. Pattipati received the Centennial Key to the Future award in 1984 from the IEEE Systems, Man and Cybernetics (SMC) Society, and was elected a Fellow of the IEEE in 1995. He received the Andrew P. Sage award for the Best SMC Transactions Paper for 1999, Barry Carlton award for the Best AES Transactions Paper for 2000, the 2002 NASA Space Act Award, and the 2003 AAUP Research Excellence Award at the University of Connecticut. He also won the best technical paper awards at the 1985, 1990, 1994, 2002 and 2004 IEEE AUTOTEST Conferences, and at the 1997 and 2004 Command and Control Conferences. Prof. Pattipati served as Editor-in-Chief of the IEEE Transactions on SMC-Cybernetics (Part B) during 1998-2001.