Text Categorization, Supervised Learning, and Domain Knowledge Integration

Ah-Hwee Tan
Kent Ridge Digital Labs
21 Heng Mui Keng Terrace
Singapore 119613
ahhwee@krdl.org.sg

Fon-Lin Lai
Kent Ridge Digital Labs
21 Heng Mui Keng Terrace
Singapore 119613
fllai@krdl.org.sg

ABSTRACT

This paper introduces a predictive self-organizing neural network known as Adaptive Resonance Associative Map for classification of free-text documents. Whereas most statistical approaches to text categorization derive classification knowledge based on training examples alone, ARAM performs supervised learning and integrates user-defined classification knowledge in the form of IF-THEN rules. Through our experiments on the Reuters-21578 database, we show that ARAM performs reasonably well in mining categorization knowledge from sparse and high dimensional document feature space. In addition, ARAM predictive accuracy and learning efficiency can be improved by incorporating rules derived from the Reuters category description. The impact of rule insertion is most significant for categories with a small number of positive training documents.

Keywords

Categorization, text mining, learning, knowledge integration

1. INTRODUCTION

Text categorization refers to the task of automatically assigning documents into one or more predefined classes or categories. In recent years, there has been an increasing number of statistical and machine learning techniques that automatically generate text categorization knowledge based on training examples. Such techniques including decision trees, K-nearest-neighbor system (KNN), rule induction, gradient descent neural networks, regression models, Linear Least Square Fit (LLSF), and support vector machines (SVM) assume the availability of a large pre-labeled or tagged training corpus. In specific domains, such corpora may not be readily available. In a personalized information filtering application, for example, few users would have the patience to provide feedback to a large number of documents for training the classifier. On the other hand, most users are willing to specify what they want explicitly. In these cases, it is desirable to have the flexibility of building a text classifier from examples as well as obtaining categorization knowledge directly from the users.

This paper reports our evaluation of Adaptive Resonance Associative Map (ARAM) [4] for text classification based on a popular public domain document database, namely Reuters-21578. The objective of our experiments is two-folded. First, we study ARAM's capability in mining categorization rules from sparse and high dimensional document feature vectors. Second, we investigate if ARAM's predictive accuracy and learning efficiency can be enhanced by incorporating a set of rules derived from the Reuters category description.

2. ARAM

ARAM belongs to a family of predictive self-organizing neural networks known as predictive Adaptive Resonance Theory (predictive ART) [3] that performs incremental supervised learning of recognition categories (pattern classes) and multidimensional maps of patterns. An ARAM system can be visualized as two overlapping Adaptive Resonance Theory (ART) [2] modules consisting of two input fields F_1^a and F_1^b with an F_2 category field. For classification problems, the F_1^a field serves as the input field containing the input activity vector and the F_1^b field servers as the output field containing the output class vector. The F_2 field contains the activities of the recognition categories that are used to encode the patterns. During learning, given an input pattern presented at the F_1^a input layer and an output pattern presented at the F_1^b output field, a F_2 category node is selected to encode the pattern pair.

When performing classification tasks, ARAM creates recognition categories of input patterns, and associates each category with its respective prediction. The knowledge that ARAM discovers during learning, is compatible with IF-THEN rule-based representation. Specifically, each node in the F_2 field represents a recognition category associating a set of F_1^a input patterns with a set of F_1^b output patterns. Learned weight template vectors, one for each F_2 node, constitute a set of rules that link antecedents to consequents. At any point during the incremental learning process, the system architecture can be translated into a compact set of rules. Similarly, domain knowledge in the form of IF-THEN rules can be inserted into ARAM architecture.

3. EXPERIMENTS

To facilitate comparison, we used the recommended *ModApte* split of Reuters-21578 [1, 5] to partition the database into training and testing data. By selecting the 90 (out of a total of 135) categories that contain at least one training and one testing documents, there were 7770 training documents and 3019 testing documents.

3.1 Performance Measures

ARAM experiments adopted the most commonly used performance measures, namely *recall*, *precision*, and the F_1 measure. These scores were calculated for a series of binary classification experiments, one for each category, and then averaged across the experiments. Two types of averaging methods were used: (1) *micro-averaging* technique that gave equal weights to each document; and (2) *macro-averaging* technique that gave equal weight to each category.

3.2 Paradigm

ARAM experiments used the following parameter values: choice parameters $\alpha_a = 0.01$, $\alpha_b = 0.01$; learning rates $\beta_a = \beta_b = 1.0$ for fast learning; contribution parameter $\gamma = 1.0$, vigilance parameters $\rho_a = 0.7$ and $\rho_b = 1.0$. The number of keyword features was fixed at 100 through empirical experiments. Using a voting strategy, 10 voting ARAM produced a probabilistic score between 0 and 1. The score was then thresholded at a specific cut off point to produce a binary class prediction.

A set of IF-THEN rules was crafted by the authors based on a description of the Reuters categories in the Reuters-21578 documentation (cat-descriptions_120396.txt). The rules simply linked the keywords mentioned in the description to their respective category labels. Creation of such rules was rather straight-forward. A total of 146 rules was created without the help from any domain expert in less than half an hour.

3.3 Results

Table 1 compares the results obtained by ARAM with and without rule insertion on the 10 most populated Reuters categories. The micro-averaged F_1 and the macro-averaged F_1 scores across the top 10 and all the 90 categories are also given. Five out of the top 10 categories, namely *acq*, *money-fx*, *grain*, *interest*, and *corn*, showed noticeable improvement in F_1 measures by incorporating rules. Interestingly, one category, namely *crude*, produced worse results. The differences in F_1 for the other four were not significant. Overall improvement on the micro-averaged F_1 scores across the top 10 and all the 90 categories were 0.005 and 0.006 respectively. The improvement obtained on the macro-averaged F_1 scores, 0.013 for the top 10 and 0.038 for the 90 categories, were much more significant. This showed that rule insertion was most effective for categories with a smaller number of positive training documents. The results were encouraging as even a simple set of rules was able to produce a noticeable improvement in accuracy. In addition, eight of the 10 categories produced a smaller number of categories through rule insertion. The other two had no or insignificant difference. The impact on the number of learning iterations was much weaker. Nevertheless, six out of 10 converged in a smaller number of iterations. Among the other four categories, one had no change and three had slight increases ranging from 0.1 to 0.2 iterations.

Table 1: Predictive performance of ARAM on the top 10 categories. N - number of learning iterations. C - number of recognition categories created.

	ARAM			ARAM w/rules		
Category	N	C	F_1	N	C	F_1
earn	5.6	565.3	0.983	5.6	**535.3**	0.982
acq	6.8	486.0	0.913	**6.5**	**457.5**	**0.926**
money-fx	6.5	200.3	0.739	**6.2**	**190.8**	**0.776**
grain	5.4	88.8	0.875	5.6	**87.3**	**0.901**
crude	6.4	68.5	0.831	6.5	68.5	0.819
trade	7.6	182.0	0.634	**7.0**	**174.5**	**0.636**
interest	6.7	157.9	0.704	**6.1**	**150.1**	**0.714**
ship	5.3	46.4	0.822	**5.0**	47.1	0.816
wheat	5.3	72.2	0.761	**4.8**	**70.2**	0.758
corn	4.3	71.4	0.701	4.5	**70.0**	**0.762**
Top 10 (miF_1,maF_1)		(0.892,0.796)		(**0.897**,**0.809**)		
All 90 (miF_1,maF_1)		(0.831,0.550)		(**0.837**,**0.588**)		

Table 2: Performance of ARAM compared with other top performing text classification systems.

Classifiers	*miR*	*miP*	miF_1	maF_1
ARAM w/rules	0.8380	0.8369	0.8374	0.5876
ARAM	0.8251	0.8376	0.8313	0.5497
SVM	0.8120	0.9147	0.8599	0.5251
KNN	0.8339	0.8807	0.8567	0.5242
LLSF	0.8507	0.8489	0.8498	0.5008
Gradient descent NNet	0.7842	0.8785	0.8287	0.3765
Native Bayes	0.7688	0.8245	0.7956	0.3886

Table 2 compares ARAM results with top performing classification systems on Reuter-21578 [5]. ARAM performed noticeably better than the gradient descent neural networks and the Native Bayes classifiers. Its miF_1 scores were about one to two percents lower than those of SVM, KNN, and LLSF, but the maF_1 scores were significantly higher. This indicates that ARAM tends to outperform the others in small categories and thus might be suitable for on-line text classification applications such as document filtering and personalization.

4. REFERENCES

[1] C. Apte, F. Damerau, and S.M. Weiss. Automated learning of decision rules for text categorization. *ACM Trans. Info. Syst.*, 12(3):233–251, 1994.

[2] G. A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37:54–115, 1987.

[3] G. A. Carpenter, *et. al.* Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Trans. Neural Networks*, 3:698–713, 1992.

[4] A.-H. Tan. Adaptive Resonance Associative Map. *Neural Networks*, 8(3):437–446, 1995.

[5] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings, SIGIR'99*, 42–49, 1999.