Belief Learning in Certainty Inference for Text Categorization

Weidong Qu and Katsuhiko Shirai
Department of Information and Computer Science, Waseda University

1. Introduction

Text categorization is one of the active research topics in the area of information retrieval. The objective of text categorization is the automated assigning documents into predefined categories based on their content. In recent years, many statistical and machine learning methods are proposed to address text categorization task. Such methods like Support vector machines (SVM), K-nearest-neighbor system (KNN), Sleeping-experts, and BoosTexter assume that a large pre-labeled or tagged training corpus is available [1,2]. In some specific domains, to collect this kind of large corpus seems somewhat difficult. For example, in a personalized information-filtering or personalized recommendation systems, few people have patient to collect a large number documents to train their system. Moreover, at the beginning, users usually have only some basic consciousness about what they want. In these cases, most users prefer specifying what they want explicitly in some convenient manner. Combine machine learning method with user's description is a feasible approach.

This paper describes a method of belief learning in certainty inference for text categorization. Unlike most machine learning methods derive classification knowledge based on training samples alone, this method can easily integrates user defined IF-THEN rules due to being compatible with expert system's rules combination frame. Our method uses a multiplicative update algorithm to perform the belief learning of rules and classifies a document by a certainty (plausible) inference mechanism. The initial experiments show that the performance of this method is comparable to Sleeping-experts method. Moreover, it has better time and space efficiency than Sleeping-experts method.

2. Method

Certainty method is a plausible inference mechanism used in MYCIN Expert System [3]. The rules in certainty method are as following forms:

$$\text{Rule: hypothesis} \rightarrow \text{consequence with } \{CF\}$$
$$CF = MB - MD$$
$$0 <= MB(h|e),\ MD(h|e) <= 1.0$$
$$p(h|e) + p(\neg h|e) <= 1.0$$

Where, the CF is a certainty factor. It is defined by two measures MB (measure of belief) and MD (measure of disbelief). Given two rules, MYCIN uses the following combination formulas to combine them.

$$CF(h|e1 \wedge e2) = \begin{cases} X + Y(1-X) & \text{if } X,Y >= 0 \\ X + Y(1+X) & \text{if } X,Y <= 0 \\ (X+Y)/(1-\min(|X|,|Y|)) & \text{if } X>0 \text{ and } Y<0 \text{ or } X<0 \text{ and } Y>0 \end{cases}$$

Where: $X = CF(h|e1)$, $Y = CF(h|e2)$

Certainty method is used in our system to infer if a document belongs to a category. The rules can be defined by users according to their description or learn from a pre-labeled corpus. Given a document, we can extract the text features (hypothesis) such as meaningful words or phrases and convert them into rules. Two measures MB and MD are assigned to each rule. In our system, MB would mean features (words or phrases) are somewhat relevant to this category and MD can be seen that these features are irrelevant to this category. Having this rule set, we can use rules to infer if a document belongs to this category or not. In expert system, these rule's certainty factors are usually assigned by domain experts. Since all the CF values are assigned by users is infeasible, we use machine learning method to estimate the values from a training corpus. Note that our system can works in an online manner, users can first define their rules and use them to categorize some documents. Then the feedback about system's output can be used to re-estimate the rule's CF values. We have found that the multiplicative update rules are a good compromise between speed and ease of implementation for solving parameter estimate problems. The certainty factor learning algorithm is shown in following Fig.1. Here θ is a threshold and α is the learning rate.

Parameters: $\theta \in (0,1)$, $\alpha \in (0,1)$

a) Using combination rule to combine all rules CF(CFi,CFj), after combining all evidences we got the final $CF(\sum CFi)$.
b) If $CF(\sum CFi) > \theta$ and the result is of class A, do nothing.
c) If $CF(\sum CFi) <= \theta$ and the result is of $\neg A$, do nothing.
d) If $CF(\sum CFi) <= \theta$ and the result is of class A, for each rule's MDi, set MDi to αMDi.
e) If $CF(\sum CFi) > \theta$ and the result is of class $\neg A$,

for each rule's MBi, set MBi to αMBi.

f) Renormalize certainty factor.

Fig.1 Multiplicative weight updated algorithm

We should notice that this algorithm is similar to the Sleeping-experts algorithms [4]. The difference between our method and Sleeping-experts is that we use the MYCIN inference mechanism (rule combination) and Sleeping-experts uses a weighted combination of the predictions of the experts. Moreover, Sleeping-experts algorithms use n-grams as their experts and our method only use words or some phrases. N-grams can not work in our frame.

3. Experiments

To investigate our system's performance, we compare our method with Sleeping-experts method. Not only because two methods use similar multiplicative weights updated algorithm, but also use different context information. Sleeping-experts uses n-gram based context information and our method works in a two layers neural network manner. The context constrains are constructed in its parameters estimating stage.

3.1 Data Collection

The experiments were conducted on a dataset of about 2000 documents tagged with 10 different categories, which extracted from Mainichi newspaper corpus 1994. The documents were split into training and test sets. The categories and the number of documents in each category are showed in Table 1.

Category	*Training*	*Test*
Sports	161	147
Government	135	142
Crime	156	148
Education	110	124
Traffic	112	103
Military	110	118
International affairs	96	97
Communications	76	83
Drama	86	95
Agriculture	78	72

Table 1: The categories and the number of documents.

3.2 Performance Measure

To evaluate the performance of document categorization, we use F-measure, a weighted combination of recall and precision, as the principal evaluation metric [5]. Precision is the number of correctly assigned documents in the category over the total number of documents assigned in that category. Recall is the number of correctly assigned documents in the category over the total number of assigned documents by human in that category. The F-measure is defined as follows:

$$Fmeasure = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

3.3 Results

The experiments used the following parameter values: threshold $\theta = 0.10$ and learning rate $\alpha = 0.4$. Figure 1 compares our method with Sleeping-experts method on test set. It is shown that the performance of our method is comparable to Sleeping-experts method. To achieve optional performance, Sleeping-experts method usually uses all words, bi-grams, tri-grams and 4-grams as its experts. The number of its experts is very huge; on the contrary, we only use word or some phrases, so our method has better time and space efficiency than Sleeping-experts method. Here we just give initial experiment results. In our future work, we will investigate if the predictive accuracy of our method can be improved by incorporating a set of rules derived from user's description.

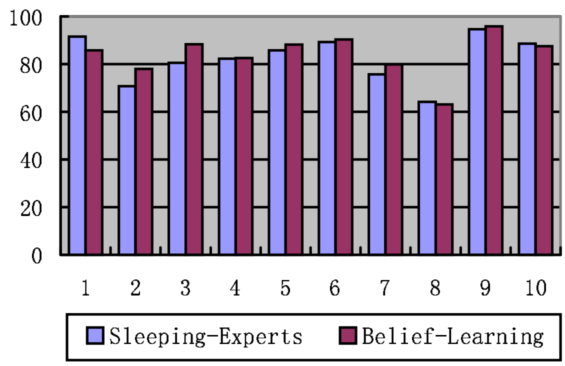

Figure 1: performance (in *F-measure*) of our method with Sleeping-experts method.

4. References

[1] Robert E. Schapire, Yoram Singer. BoosTexter: A Boosting-based System for Text Categorization. Machine Learning, 39(2/3), 135-168, 2000.

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

[3] R. Lu. Artificial intelligence. Science press (In Chinese), 1998.

[4] William W. Cohen. and Yoram Singer. Context-sensitive learning methods for text categorization. In Proceedings, SIGIR 96, 307–315, 1996.

[5] D. Lewis. Evaluating text categorization. Proceedings of speech and natural language workshop, pp. 312-318, 1991.