Maximum Entropy Modeling with Feature Selection for Text Categorization

Jihong Cai and Fei Song

Department of Computing and Information Science
University of Guelph, Guelph, Ontario, Canada N1G 2W1
{jcai,fsong}@uoguelph.ca

Abstract. Maximum entropy provides a reasonable way of estimating probability distributions and has been widely used for a number of language processing tasks. In this paper, we explore the use of different feature selection methods for text categorization using maximum entropy modeling. We also propose a new feature selection method based on the difference between the relative document frequencies of a feature for both relevant and irrelevant classes. Our experiments on the Reuters RCV1 data set show that our own feature selection performs better than the other feature selection methods and maximum entropy modeling is a competitive method for text categorization.

Keywords: Text Categorization, Feature Selection, Maximum Entropy Modeling.

1 Introduction

Text categorization is the process of assigning predefined categories to textual documents. One of its many useful applications is for web content filtering, since only when we know the categories of the web pages can we decide whether they are offensive/inappropriate in order to block them from user access. Text categorization has been conducted with different machine learning techniques, including Naïve Bayes [4], k-Nearest Neighbors [3], Linear Least Squares Fit [8], Support Vector Machines [1], and Maximum Entropy Modeling [5].

This paper explores the use of different feature selection methods for text categorization in the context of maximum entropy modeling. In addition to some commonly-used feature selection methods, we propose a new feature selection method, called Count Difference, which is based on the difference between the relative document frequencies of a feature for both relevant and irrelevant classes. As Nigam et al. [5] points out, maximum entropy modeling may be sensitive to poor feature selection, and we want to see how these feature selection methods can affect the performance of a text classifier based on maximum entropy modeling and how is maximum entropy modeling is compared with other popular text categorization methods.

2 Feature Selection Methods

Most text categorization systems use word types and their frequency counts for document representation. We refer to these word types as features. Feature selection

H. Li et al. (Eds.): AIRS 2008, LNCS 4993, pp. 549–554, 2008.

not only reduces computational cost in space and time, but also improves performance by carefully selecting good features for classification [9].

2.1 Existing Feature Selection Methods

Document frequency stands for the number of documents in which a feature occurs in a document collection. This method favors features whose document frequencies fall into a mid-range, since low-frequency features do not contribute much to the distinction of most documents and high-frequency features are so common that they reduce the distinction between documents.

χ^2 ranking [3] favors features that are strongly dependent on relevant or irrelevant classes. One problem with this method is that it may give a high score to a rare feature. For example, a feature may only appear in 5 documents in a collection of 100,000 documents, but if all these 5 documents belong to the relevant class, the feature may still get a high score, which is counter-intuitive.

Likelihood ratio attempts to address the issue of assigning high scores to rare features in χ^2 ranking [3]. For a large sample size, it tends to behave similarly to χ^2 ranking, but it also works well for a small sample size.

Mutual Information only measures the dependency between a feature and its relevant class, and as a result, it tends to favor rare terms if they are mostly used for relevant documents [9].

Information Gain is a measure based on entropy, which has been successfully applied to the construction of an optimal decision tree [4]. Features that reduce the entropy the most are favored for this method.

Orthogonal Centroid chooses features by an objective function based on transformations on centroids. To overcome the time and space complexity of the original orthogonal centroid algorithm, an optimal orthogonal centroid algorithm was proposed to provide a simple solution for feature selection [7].

Term Discrimination tries to measure the ability of a feature for distinguishing one document from the others in a collection [6]. A very popular feature often has a negative discrimination value, since it tends to reduce the differences between documents, while a rare feature usually has a close-to-zero value, since it is not significant enough to affect the space density.

2.2 Count Difference

A feature whose document frequency for one class is higher than that for the other class is desirable since it helps distinguishing between the two classes. However, if the feature is rare in the training documents, its use will be limited since it only affects a small number of documents. This leads us to propose a new feature selection method called Count Difference (CD), which tries to reflect the above two factors in ranking features.

Given a feature, we can partition the set of training documents into four regions in the following contingency table.

Table 1. Feature-Class Contingency Table

	Relevant	Irrelevant
Feature Used	a	b
Feature Not Used	c	d

We first introduce the notation of relative document frequency, which is the ratio of the document frequency of a feature for one class over the average document frequency for the same class:

$$relativeDF(t, y) = a_t / \overline{a} \quad \text{and} \quad relativeDF(t, \tilde{y}) = b_t / \overline{b}$$

Here, $\overline{a}$ and $\overline{b}$ denote the average document frequencies for the relevant and irrelevant classes, which are computed as follows:

$$\overline{a} = \frac{1}{M}\sum_{t=1}^{M} a_t \quad \text{and} \quad \overline{b} = \frac{1}{M}\sum_{t=1}^{M} b_t$$

where M is the number of original features before the selection process.

With the relative document frequencies, we can then define the count difference score of a feature as the difference between its two relative document frequencies:

$$CD(t) = (a_t / \overline{a} - b_t / \overline{b})^2$$

Intuitively, the relative document frequency measures the importance of a feature against the average feature for one class. If a feature is rare, its relative document frequency will be low, whereas if a feature is popular, its relative document frequency will be high. The count difference tends to favor features whose relative document frequencies for one class are higher than those for the other class. If a feature is popular for both classes, its count difference score will be reduced.

3 Maximum Entropy Modeling

Maximum entropy provides a reasonable way of estimating probability distributions from training data. The key principle is that we should agree with everything that is known, but carefully avoid assuming anything that is unknown. In particular, when nothing is known about certain features, the distribution for them should be as uniform as possible (thus the maximum entropy).

3.1 Feature Functions

Following Nigam et al. [5], we represent features as feature functions. For text categorization, we can define a feature function for each word-class combination:

$$f_{w,c}(d, y) = \begin{cases} 0 & \text{if } y \neq c \\ N(d, w)/N(d) & \text{otherwise} \end{cases}$$

Here, N(d, c) is the number of times word w occurring in document d, and N(d) is the number of words in document d.

3.2 Log-Linear Models

A maximum entropy model generally takes the following parametric form:

$$p(x,c) = \frac{1}{Z}\prod_{i=1}^{K}\alpha_i^{f_i(x,c)}$$

where K is the number of feature functions, α_i is a weight for feature function f_i, and Z is normalization constant. The model is also called the log-linear model, since by taking the logarithm on both sides, we get a linear combination for the feature functions.

The log-linear model above allows overlapping/dependent features. Although we use individual words as features for text categorization, we could easily extend the set with word pairs, longer phrases, and even non-text features (such as links in web pages). Even for individual words, we do not require them to be independent as the case for Naïve Bayes method.

In addition, the log-linear model provides further differentiations among features by carefully assigning weights α_i to different feature functions. In particular, by setting $\alpha_i = 1$, we essentially eliminate that feature in the combination process.

4 Experimental Results

Reuters RCV1 data set is a benchmark collection for evaluating text categorization systems [2]. It has over 800,000 news articles collected over one year's time. We focus on the topic scheme, which is a hierarchy of 103 categories, with the top four categories being CCAT (Corporate/Industrial), ECAT (Economics), GCAT (Government/Social), and MCAT (Markets). We use $F_{1.0}$ (harmonic mean of recall and precision) to measure the classification performance. For multiple categories, we use both macro-average (per-class average) and micro-average (per-document average).

4.1 Classification Performance

There are several factors that can affect the classification results. For a hierarchical scheme, there is a choice of flat classification (testing all categories independently) and hierarchical classification (only the documents belonging to a parent category are tested further for its sub-categories). We choose the hierarchical classification, since it is naturally suited for a hierarchical scheme.

We use the Improved Iterative Scaling algorithm [5], which iteratively updates the weighting parameters until they are converged or a certain number of iterations are reached. To avoid over fitting to the training data, we can terminate the process after a certain number of iterations or monitor the performance with a validation data set and stop the training when the performance starts to decrease. For simplicity, we terminate the training after 50 iterations.

Table 2 shows the classification results based on the above setting, where the values are the micro-averages of $F_{1.0}$ measures. First, we see that the classification performance goes up for all selection methods as we increase the number of features but only to a certain degree. After that, the performance starts to decrease but very slowly. This indicates the need for feature selection, since it not only reduces the computational cost but also improves the performance.

Table 2. Effects of Feature Selection on Text Categorization

	DF	χ^2	CD	LR	MI	IG	OC	TD
100	.601	.267	.676	.292	.061	.448	.628	.599
500	.741	.553	.769	.703	.067	.736	.764	.728
1k	.774	.717	.795	.745	.094	.757	.791	.748
1.5k	.789	.755	.793	.752	.112	.762	.789	.748
2k	.790	.760	.791	.754	.135	.764	.788	.746
4k	.786	.763	.780	.756	.223	.763	.778	.740
8k	.776	.758	.768	.749	.514	.757	.766	.728

Secondly, the rates of performance increase are different for different selection methods. We can roughly divide the selection methods into several groups. The best group contains Document Frequency, Count Difference, and Optimal Orthogonal Centroid. They start off with relatively high performance values even with just 100 features, and reach the highest performance values between 1000 and 2000 features. Our own feature selection method, Count Difference, gets the best performance of 0.795 with 1000 features. The group with the highest overlaps of featuers, including 춘춘. Ranking, Likelihood Ratio, and Information Gain, start with low performance values and improve slowly to reach their peaks. Note that Mutual Information (used in Nigam et al. [5]) has the worst performance since it tends to favor rare features.

Finally, we see that the performance between different selection methods become close as we reach 1000 features and above (except for Mutual Information). This leads us to conclude that maximum entropy modeling is not too sensitive to feature selection as long as most important features are included in the pool of features, since the weights for log-linear combination provide further differentiations between the features used for text categorization.

4.2 Comparison with Other Methods

Reuters RCV1 data set provides benchmark results for several text categorization methods, including Support Vector Machines, K-Nearest Neighbors, and Rocchio-style classifiers.

Table 3. Comparison with Other Methods

	SVMs	K-NN	Rocchio	MaxEnt
Macro-avg	.619	.560	.504	.472
Micro-avg	.816	.765	.693	.794

Table 3 summarizes the classification results for the entire topic scheme in both macro- and micro-averages. We see that MaxEnt (maximum entropy modeling) is a competitive method for text categorization: its performance is better than that for K-Nearest Neighbors and Rocchio-style classifiers in terms of micro-averages, although the performance is still not as good as that for Support Vector Machines. In terms of macro-averages, MaxEnt is the lowest among the four methods. This leads us to conclude that MaxEnt tends to perform better when a category is adequately covered by training documents. Otherwise, the performance will be affected considerably. This is perhaps not surprising since the feature functions are essentially defined in terms of maximum likelihood estimate (N(d,w)/N(d)). Future work remains to smooth this probability with data from parent categories.

5 Conclusions

We compared eight different feature selection methods for text categorization using maximum entropy modeling. We showed that feature selection is an effective way of reducing the computational cost while at the same time improving the classification performance. We demonstrated that our own feature selection method, Count Difference, is promising for text categorization, not only achieving the best performance but also working reasonably well for very aggressive feature cutoffs. We further illustrated that maximum entropy modeling is a competitive method for text categorization and has potential for further improvements.

References

1. Joachims, T.: Text categorization with Support Vector Machines: Learning with many relevant features. In: Tenth European Conference on Machine Learning, pp. 137–142 (1998)
2. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: RCV1: A new benchmark collection for text categorization research. Journal of Machine Learning Research 5, 361–397 (2004)
3. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge (1999)
4. Mitchell, T.: Machine Learning. The McGraw-Hill Companies, Inc., New York (1997)
5. Nigam, K., Lafferty, J., McCallum, A.: Using maximum entropy for text classification. In: IJCAI 1999 Workshop on Machine Learning for Information Filtering, pp. 61–67 (1999)
6. Salton, G.: Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley, Reading (1989)
7. Yan, J., Liu, N., Zhang, B., Yan, S., Chen, Z., Cheng, Q., Fan, W., Ma, W.-Y.: OCFS: Optimal Orthogonal Centroid Feature Selection for text categorization. In: 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 122–129 (2005)
8. Yang, Y., Chute, C.G.: An example-based mapping method for text categorization and retrieval. ACM Transactions on Information Systems 12(3), 252–277 (1994)
9. Yang, Y., Pedersen, J.O.: A comparative study of feature selection in text categorization. In: Fisher, J.D.H. (ed.) The Fourteenth International Conference on Machine Learning, pp. 412–420 (1997)