Naive Bayes for Text Classification with Unbalanced Classes

Eibe Frank[1] and Remco R. Bouckaert[1,2]

[1] Computer Science Department, University of Waikato, New Zealand
[2] Xtal Mountain Information Technology, Auckland, New Zealand
{eibe,remco}@cs.waikato.ac.nz, rrb@xm.co.nz

Abstract. Multinomial naive Bayes (MNB) is a popular method for document classification due to its computational efficiency and relatively good predictive performance. It has recently been established that predictive performance can be improved further by appropriate data transformations [1, 2]. In this paper we present another transformation that is designed to combat a potential problem with the application of MNB to unbalanced datasets. We propose an appropriate correction by adjusting attribute priors. This correction can be implemented as another data normalization step, and we show that it can significantly improve the area under the ROC curve. We also show that the modified version of MNB is very closely related to the simple centroid-based classifier and compare the two methods empirically.

1 Introduction

Multinomial naive Bayes (MNB) is the version of naive Bayes that is commonly used for text categorization problems. In this paper we identify a potential deficiency of MNB in the context of skewed class sizes. The standard practice of initializing word frequencies for all classes to the same value—normally, a value of one is used—biases predictions in favor of the larger class: initial word counts have a larger influence on the predicted probability when there is less data, as in the smaller classes in a text categorization problem. We investigate the use of distinct initial word counts for different-size classes, and propose a heuristic for choosing the initial word count for each class. This modification can be implemented as a pre-processing step that normalizes the word count vector associated with each class and can significantly improve predictive performance as measured by the area under the ROC curve. We also compare the modified version of MNB to the centroid classifier, to which it is closely related.

2 Naive Bayes for Text Classification

In the MNB classifier each document is viewed as a collection of words and the order of words is considered irrelevant. The probability of a class value c given

a test document d is computed as

$$P(c|d) = \frac{P(c)\prod_{w\in d} P(w|c)^{n_{wd}}}{P(d)}, \tag{1}$$

where n_{wd} is the number of times word w occurs in document d, $P(w|c)$ is the probability of observing word w given class c, $P(c)$ is the prior probability of class c, and $P(d)$ is a constant that makes the probabilities for the different classes sum to one. $P(c)$ is estimated by the proportion of training documents pertaining to class c and $P(w|c)$ is estimated as

$$P(w|c) = \frac{1 + \sum_{d\in D_c} n_{wd}}{k + \sum_{w'} \sum_{d\in D_c} n_{w'd}}, \tag{2}$$

where D_c is the collection of all training documents in class c, and k is the size of the vocabulary (i.e. the number of distinct words in all training documents). The additional one in the numerator is the so-called Laplace correction, and corresponds to initializing each word count to one instead of zero. It requires the addition of k in the denominator to obtain a probability distribution that sums to one. This kind of correction is necessary because of the zero-frequency problem: a single word in test document d that does not occur in any training document pertaining to a particular category c will otherwise render $P(c|d)$ zero.

3 Unbalanced Class Sizes

Many text categorization problems are unbalanced. This can cause problems because of the Laplace correction used in (2). Consider a word w in a two-class problem with classes c_1 and c_2, where w is completely irrelevant to the classification. This means the odds ratio for that particular word should be one, i.e. $\frac{p(w|c_1)}{p(w|c_2)} = 1$, so that this word does not influence the class probability.

Assume the word occurs with equal relative frequency 0.1 in the text of each of the two classes. Assume there are 20,000 words in the vocabulary ($k = 20,000$) and the total size of the corpus is 100,000 words. Figure 1 shows the estimated odds ratio, based on (2), as the relative size of the two classes changes. It has the desired value of one when both classes contain the same number of words. However, the situation changes dramatically as the class sizes become skewed. For example, when c_2 becomes very small and contains little text relative to the other class, the presence of the irrelevant word w in a test document substantially increases the estimated probability that the test document belongs to class c_1.

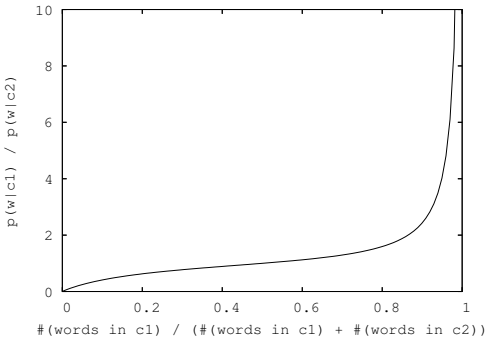

Fig. 1. Estimated odds ratio

Fortunately it turns out that there is a simple remedy: we can normalize the word counts in each class so that the total size of the classes is the same for both classes after normalization. To do this we replace n_{wd}, where d is in D_c, by

$$\alpha \times \frac{n_{wd}}{\sum_{w'} \sum_{d \in D_c} n_{w'd}}, \tag{3}$$

i.e. we normalize the vector of word counts for each class to have length α when measured according to L_1 norm. This ensures that the estimated odds ratio for our irrelevant word will be one, regardless of which particular value we choose for the normalization constant α.

We can also view this normalization step as a modification of the Laplace correction. Plugging the normalized counts into (2), it is easy to show that we have effectively replaced the standard initial word count of one by the class-specific initial word count $(\sum_{w'} \sum_{d \in D_c} n_{w'd})/\alpha$. This means that we are using asymmetric word count priors.

The value of α determines the amount of smoothing across word counts in the dictionary. Surprisingly, initial experiments on the Reuters data showed that a value of $\alpha = 1$ often works well, which corresponds to very heavy smoothing. As we will see, $\alpha = 1$ also results in good performance on other datasets included in our experimental comparison. In the next section we will show that our modified MNB with $\alpha = 1$ is closely related to the centroid classifier for text classification.

4 Relationship to Centroid Classifier

In the centroid classifier [3], each class is represented by its mean word vector, normalized to unit length using L_2 norm. The centroid $\mathbf{c}_c$ for class c is given by

$$\mathbf{c}_c = \{\frac{\sum_{d \in D_c} n_{w_1 d}}{\sqrt{\sum_w (\sum_{d \in D_c} n_{wd})^2}}, \frac{\sum_{d \in D_c} n_{w_2 d}}{\sqrt{\sum_w (\sum_{d \in D_c} n_{wd})^2}}, \ldots, \frac{\sum_{d \in D_c} n_{w_k d}}{\sqrt{\sum_w (\sum_{d \in D_c} n_{wd})^2}}\}.$$

Assuming $\mathbf{x}_d = \{n_{w_1 d}, n_{w_2 d}, \ldots, n_{w_k d}\}$ is the word vector representing a test document d, the scoring function for the centroid classifier is

$$\mathbf{x}_d \cdot \mathbf{c}_1 - \mathbf{x}_d \cdot \mathbf{c}_2. \tag{4}$$

Now consider MNB. The scoring function (log odds) for MNB can be written as

$$\left[\log P(c_1) + \sum_{i=1}^{k} n_{w_i d} \log(P(w_i|c_1))\right] - \left[\log P(c_2) + \sum_{i=1}^{k} n_{w_i d} \log(P(w_i|c_2))\right]. \tag{5}$$

The terms $\log P(c_1)$ and $\log P(c_2)$ are irrelevant when we rank documents. Comparing (4) and (5) we see that the only remaining difference is that $\log(P(w_i|c))$ is used instead of the corresponding vector component from the centroid. However, these two terms have a very similar effect if we set α to one in our modified

version of MNB. When $\alpha = 1$, using (3) in (2) means that

$$P(w|c) = \frac{1 + \frac{\sum_{d \in D_c} n_{wd}}{\sum_{w'} \sum_{d \in D_c} n_{w'd}}}{k+1}.$$

The denominator is constant and can be dropped. Furthermore,

$$\frac{\sum_{d \in D_c} n_{wd}}{\sum_{w'} \sum_{d \in D_c} n_{w'd}} << 1$$

for practical text datasets, and $\log(1+x) \approx x$ for $x << 1$. Hence the only remaining difference between the scoring functions in (4) and (5) is that L_2 norm is used in the former and L_1 norm in the latter.

As we will see in the next section, the modified MNB classifier with $\alpha = 1$ indeed gives very similar results to the centroid classifier on most of the datasets we investigated in our experiments.

5 Empirical Results

In this section we present experiments comparing MNB with and without our modification, and the centroid classifier. Weka was used for the experiments, and the area under the ROC curve (AUC) employed as the performance statistic. All results are averages from ten runs of the hold-out method. For each run 66% of the data was used for training and 34% for testing, with stratification to ensure that class proportions were preserved. The same runs were used for each of the learning schemes that we investigated. To test for significant differences we used the corrected resampled paired t-test [4], which has acceptable Type I error.

Our experiments were based on four well-known text classification datasets: `Reuters-21578`, `WebKB`, `Industry Sector`, and `20 Newsgroups`. For each dataset we created as many two-class classification problems as there were class values in the data, with the exception of the `Reuters-21578` data where we only used the 10 most frequent categories. All documents pertaining to a particular class value were put into one category and all the remaining documents in the other. This was necessary in order to use AUC for evaluation. Consequently we created 10 classification problems from the `Reuters-21578` data, four from the `WebKB` data, 105 from the `Industry Sector` data, and 20 from the `20 Newsgroups` data.

For each dataset we used the same steps to extract word features. All characters were converted to lowercase, only alphabetic tokens were considered, stopwords and *hapax legomena* were removed, and the *full* resulting vocabulary was used (i.e. no feature selection was performed).

We applied the following pre-processing steps to the raw word counts [1, 2]. First, TF×IDF was used to transform n_{wd} into $n'_{wd} = \log(1+n_{wd}) \times \log(m/m_w)$, where m is the total number of training documents and m_w is the number of training documents that contain w. Secondly, following this transformation, the vector of transformed word frequencies for each document was normalized to

length l using L_2 norm to counteract the effect of varying document lengths. The vector length l was set to the average vector length in the training documents before normalization. The transformed and normalized frequencies were then used in both versions of MNB and the centroid classifier.

We will first look at the impact of our proposed modification using $\alpha = 1$, which is equivalent to normalizing the word vector for each class to length one in L_1 norm before applying standard MNB. Figure 2 shows the results for both standard MNB and MNB with per-class normalization (MNB_{PCN}). Bars that are striped mark differences that are statistically significant at the 5% level (in these figures and all other figures in this paper).

The results show that our modification significantly improves the performance of MNB on three of the ten `Reuters-21578` categories. It significantly reduces AUC on two categories. However, not taking the significance of the individual differences into account, there are eight wins and only two losses. This win/loss ratio has a p-value of 0.11 according to a two-sided sign test.

On the `Web KB` data the results are clear cut. MNB_{PCN} achieves significant gains on all four categories. Although there is only one significant win on the `Industry Sector` data, there is no significant loss. Moreover, not taking significance on individual `Industry Sector` categories into account, the win/loss ratio is 81/24 in favor of the modified version, which is highly significant according to a two-sided sign test (p-value $= 4.64 \times 10^{-8}$).

The `20 Newsgroups` data is the outlier in this collection of results. Here MNB outperforms MNB_{PCN}: there are 18 significant wins for MNB, and no significant losses (without considering significance, the win/loss ratio is 20/0). However, it turns out that the performance of MNB_{PCN} can be improved by using a different value for α when normalizing the word vectors for each class. Figure 3 compares MNB to MNB with per-class normalization when the length of the word vector for each class (i.e. α) is set to the minimum of the two class vector lengths before normalization (MNB_{PCNmin}). Using this value of α gives a much lower influence to the attribute priors, and the results show that this is beneficial. MNB_{PCNmin} is on par with standard MNB on the `20 Newsgroups` categories: there is one significant win and one significant loss. The win/loss ratio is 10/10 when significance of individual differences is not taken into account. Hence it appears that less smoothing is beneficial for the `20 Newsgroups` data.

Unfortunately the new value for α is no silver bullet. Although it improves performance further on the `Industry Sector` data, it results in significant degradation on `Reuters-21578` and `WebKB`.[3] Hence the right amount of smoothing for $P(w|c)$ depends on the domain, and, for best performance, an appropriate value of α should be chosen using a validation set or cross-validation.

In Section 4 we have shown that MNB with per-class normalization and $\alpha = 1$ is closely related to the centroid classifier. In the following we investigate this relationship empirically. Figure 4 shows the AUC for the centroid classifier and the modified version of MNB using $\alpha = 1$.

[3] These results are not included here due to space constraints.

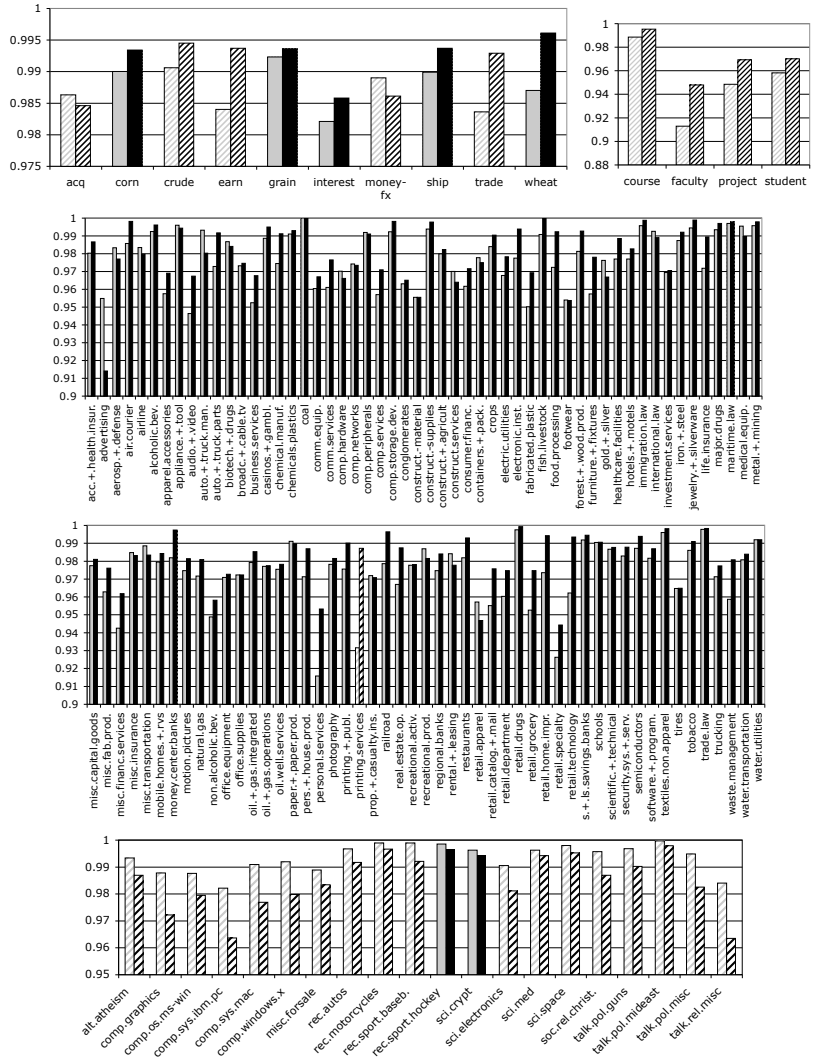

Fig. 2. AUC for MNB (grey) and MNB$_{PCN}$ (black) on Reuters (top left), WebKB (top right), Industry Sector (two middle graphs), and 20 Newsgroups (bottom)

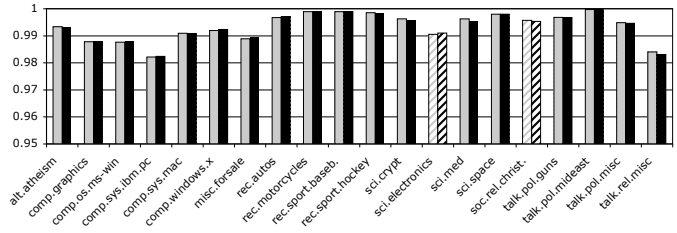

Fig. 3. AUC for MNB (grey) and MNB$_{PCNmin}$ (black) on 20 Newsgroups

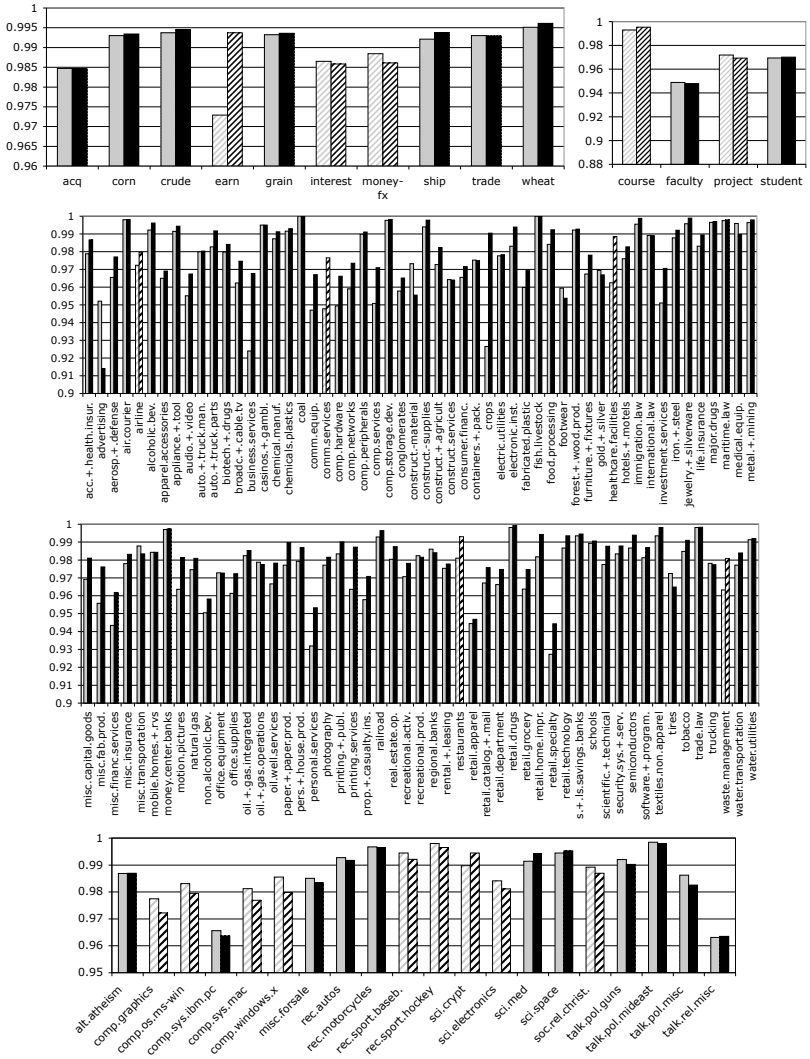

Fig. 4. AUC for centroid classifier (grey) and MNB$_{PCN}$ (black) on Reuters (top left), WebKB (top right), Industry Sector (two middle graphs), and 20 Newsgroups (bottom)

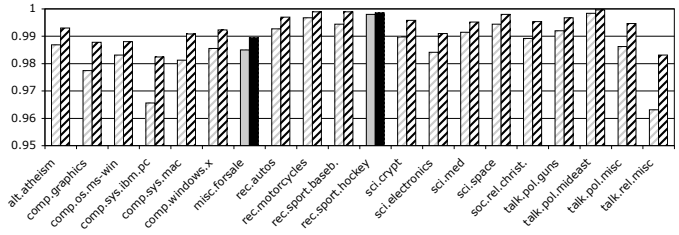

Fig. 5. AUC for centroid classifier (grey) and MNB$_{PCNmin}$ (black) on 20 Newsgroups

Performance of the two methods is indeed very similar in most cases. Ignoring the magnitude of the differences there is no clear winner on the `Reuters-21578` data (two significant wins and one significant loss for the centroid classifier) and no clear winner on the `Web KB` data (one significant win and one significant loss).

However, MNB_{PCN} has an edge on the 105 `Industry Sector` categories. There are five significant losses for the centroid classifier and no significant win, and it almost always yields a lower AUC value. On the other hand, the centroid classifier performs better than MNB_{PCN} on the `20 Newsgroups` data, with eight significant wins and only one significant loss.

Although the AUC of MNB_{PCN} and the centroid classifier is often very similar, there are some noticeable differences, e.g. on the category `earn` in the `Reuters-21578` data. The obvious question is which of the differences discussed in Section 4 is responsible for these discrepancies. To this end we performed a further experiment where L_1 normalization was used to normalize the centroids in the centroid classifier. The AUC scores for this modified centroid classifier were virtually indistinguishable from those obtained using MNB_{PCN}: they did not differ at all when rounded to the fourth decimal place. Hence the differences in Figure 4 are almost exclusively due to L_1 normalization vs. L_2 normalization.

Note that MNB with per-class normalization outperforms the standard centroid classifier with L_2 normalization if α is set to the minimum vector length before normalization (as in Figure 3). This is shown in Figure 5. There are eighteen significant wins for MNB and no significant losses (the win/loss ratio is 20/0 if significance is not taken into account). A similar win/loss ratio holds for standard MNB, which also outperforms the centroid classifier on this data.

6 Conclusions

In this paper we have identified a potential deficiency of MNB in the context of unbalanced datasets and shown that per-class word vector normalization presents a way to address the problem. Our empirical results show that normalization can indeed significantly improve performance. We have also shown that MNB with class vector normalization is very closely related to the standard centroid classifier for text classification if the class vectors are normalized to unit length, and verified the relationship empirically.

References

1. Rennie, J.D., Shih, L., Teevan, J., Karger, D.R.: Tackling the poor assumptions of naive Bayes text classifiers. In: Proc Int Conf on Machine Learning. (2003) 616–623
2. Kibriya, A.M., Frank, E., Pfahringer, B., Holmes, G.: Multinomial naive Bayes for text categorization revisited. In: Proc Australian Conf on AI. (2004) 488–499
3. Han, E.H., Karypis, G.: Centroid-based document classification: Analysis and experimental results. In: Proc European Conf on Principles and Practice of Knowledge Discovery in Databases. (2000) 424–431
4. Nadeau, C., Bengio, Y.: Inference for the generalization error. Machine Learning **52**(3) (2003) 239–281