A Regularized Linear Classifier for Effective Text Classification

Sharad Nandanwar and M. Narasimha Murty

Department of Computer Science and Automation,
Indian Institute of Science, Bangalore, 560012, Karnataka, India
{sharadnandanwar,mnm}@csa.iisc.ernet.in

Abstract. In document community support vector machines and naïve bayes classifier are known for their simplistic yet excellent performance. Normally the feature subsets used by these two approaches complement each other, however a little has been done to combine them. The essence of this paper is a linear classifier, very similar to these two. We propose a novel way of combining these two approaches, which synthesizes best of them into a hybrid model. We evaluate the proposed approach using 20ng dataset, and compare it with its counterparts. The efficacy of our results strongly corroborate the effectiveness of our approach.

Keywords: Support Vector Machine, Naïve Bayes Classifier, Regularization

1 Introduction

Text classification is the task of assigning documents to one or more of the predefined classes. It can be dated back to around 1200 BC when concept of libraries evolved, with an aim to organize document collections manually. However, the enormous growth in information available on Internet, in the form of blogs, micro-blogs, news-articles, e-mails, etc. has led us to explore the direction of automatic text classification.

Among various classification techniques Support Vector Machine (SVM) and Naïve Bayes Classifier (NBC) are widely used, because of their robustness and surprisingly good behavior in high dimensions. SVM being a discriminative model gives importance to features present in support vectors, while NBC gives higher weight to features prominent in dense region. The features given importance by one are generally not in harmony with those of other one. However while performing classification it seems intuitive to consider both such discriminative and density based features. In our work we look for a hybrid of these two models, which respects discriminative as well as density based features.

Section 2 gives a brief overview of SVM and NBC. We discuss state-of-the-art and related work in section 3. Section 4 describes our regularized linear classifier, followed by experiments and results in section 5. We conclude in section 7, after discussion in section 6.

2 Background Theory

2.1 Support Vector Machine

Support Vector Machine (SVM) [1] is a supervised learning algorithm which is based on the principle of structural risk minimization [2]. Given a finite set of patterns in two classes $\{+1, -1\}$, SVM learns a separating hyperplane which maximizes the margin between two classes. The width of maximum margin is inversely proportional to the norm of hyperplane. Thus an SVM objective can be stated as follows:

$$\min \quad \frac{1}{2} w^{\mathsf{T}} w$$

$$\text{subject to} \quad y_i[w^{\mathsf{T}} \cdot x_i + b] \geq 1, \quad i = 1, \ldots, l$$

However in case when patterns are not linearly separable, it is not feasible to find a hyperplane which satisfies all the above constraints together. To tackle this, SVM includes the error term in objective to account for misclassification penalty. The objective now is to find a hyperplane which maximizes the margin while simultaneously minimizing misclassification error, the modified SVM problem becomes:

$$\min \quad \frac{1}{2} w^{\mathsf{T}} w + C \sum_{i=1}^{l} \xi_i$$

$$\text{subject to} \quad y_i[w^{\mathsf{T}} \cdot x_i + b] \geq 1 - \xi_i, \quad i = 1, \ldots, l$$
$$\xi_i \geq 0, \quad i = 1, \ldots, l$$

In certain cases especially in low dimensions when a suitable linear boundary can not be found it is appropriate to look for non linear boundary. This is achieved by projecting the patterns into higher dimensions with the help of kernel functions, and then looking for a linear separator in the projected space. SVM inherently is a binary classifier but it can be used in multi-class classification by learning discriminant function corresponding to each class. This can be done by following either one-vs-one or one-vs-all approach [3].

2.2 Naïve Bayes Classification

Naïve Bayes Classifier (NBC)[4] is based on Bayes theorem with an assumption that different features in a pattern given a class are independent of each other. This naive assumption helps to overcome the effects of curse of dimensionality, which makes NBC practically useful even if limited amount of training data is available. Given a test pattern d, the task of a NBC is to assign that pattern to its probable class, based on the conditional probability $P(c_i|d)$. This conditional probability is computed as,

$$P(c_i|d) \propto P(c_i) \cdot P(d|c_i) = P(c_i) \cdot \prod_{w_j \in d} P(w_j|c_i)$$

where $P(c_i)$ is the prior probability of i^{th} class, and $P(w_j|c_i)$ is the likelihood of j^{th} feature given i^{th} class. In case of documents, the priors and likelihood are estimated from the training data as follows:

$$P(c_i) = \frac{|c_i|}{\sum_{c_j \in C} |c_j|}, \quad \forall c_i \in C$$

where $|c_i|$ denotes the number of documents in class c_i.

$$P(w_j|c_i) = \frac{tf_{c_i}(w_j)}{\sum_{w_k \in V} tf_{c_i}(w_k)}$$

$\forall w_j \in V$ and $\forall c_i \in C$, where $tf_{c_i}(w_j)$ denotes the frequency of term w_j in class c_i, and is given by,

$$tf_{c_i}(w_j) = \sum_{d \in c_i} tf(w_j, d)$$

$tf(w_j, d)$ is the frequency of the term w_j in document d.

2.3 Zipf's Law

Zipf's law [5] is an empirical law commonly used to model distribution of terms in a document collection. It states that, the frequency f of a term in a collection is inversely proportional to its rank r.

$$f = \frac{k}{r}$$

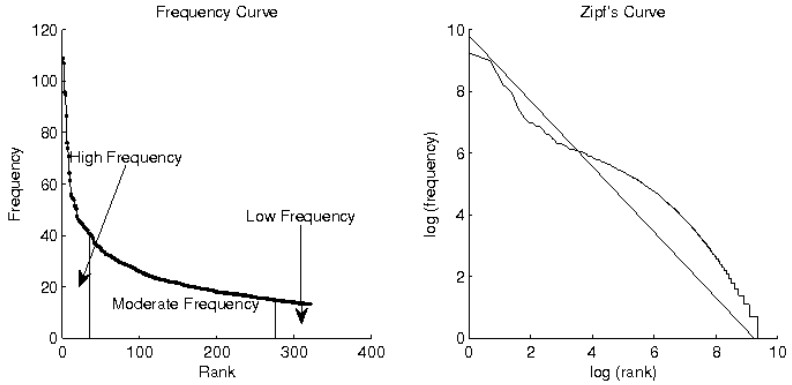
where k is constant of proportionality for the corpus.

$\log(f)$ drawn against $\log(r)$ as shown in Figure 1 *(right)*, is known as zipf's curve. Based on frequencies, the curve in Figure 1 *(left)* can be divided into three regions as rare, moderate, and high frequency zones. It has been observed that terms occurring in rare and high frequency zones does not contribute to classification much. Thus zipf's law offers a simple but powerful way to filter out terms irrelevant to classification.

3 State-of-the-Art

In classification scenario both discriminative models and generative models have been studied extensively, but substantially less amount of work has been done on coalescing these two approaches. However, the complementary behavior of two has provoked some researchers to blend their advantages.

Fig. 1: Zipf's Curve



In [6] Bishop et al. argue, that under sufficiently large number of training patterns, discriminative models perform better than generative. But, if size of training set is limited, generative models can aid their discriminative counterparts. They further talk of discriminatively trained generative models in their work. Generative and discriminative models correspond to specific choice of prior over parameters. A soft constraint introduced amongst the parameters governs the balance between two. In [7] WBC_{SVM} has been introduced, to perform weighted Bayesian classification with a SVM. To find the maximum margin hyperplane, authors use a kernel function which depends on the distribution of data, and class information. The kernel function corresponds to weighted naive bayes classification with weights estimated in accordance with NBC hyperplane. Raina et al. argues in [8] that independence assumption in NBC is too strong, and they propose a modified algorithm which assigns different weights to different regions, normalized according to region length. After some manipulations the modified algorithm is shown to be very similar to logistic regression, however, it outperforms both NBC and logistic regression. [9] shows that logistic regression when modified can be approximated to SVM.

4 Regularized Linear Classifier

We aim to weigh the features in such a manner, so that the strengths of both NBC and SVM are embodied in our hybrid classification approach. SVM weights are determined as the coefficients of normalized separating hyperplane i.e.

$$\frac{(w^T x + b)}{\sum_{w_j \in w} |w_j|}.$$

Similarly NBC weights are coefficients of normalized hyperplane separating two class distributions. According to NBC we have,

$$P(c_+) \cdot \prod_{w_j \in d} P(w_j|c_+)^{x_j} - P(c_-) \cdot \prod_{w_j \in d} P(w_j|c_-)^{x_j} \begin{cases} > 0, & d \in c_+ \\ < 0, & d \in c_- \end{cases}.$$

Where x_j is the frequency count of w_j in document d. Since log is monotonically increasing we can write above as,

$$\ln P(c_+) - \ln P(c_-) + \sum_{w_j \in d} x_j \{\ln P(w_j|c_+) - \ln P(w_j|c_-)\} \begin{cases} > 0, & d \in c_+ \\ < 0, & d \in c_- \end{cases}.$$

After appropriate normalization the weight of i^{th} feature will be,

$$\frac{\ln P(w_i|c_+) - \ln P(w_i|c_-)}{\sum_{j=1}^{l} |\ln P(w_j|c_+) - \ln P(w_j|c_-)|}$$

However, from Figure 3, we can conclude that not every feature is necessarily required for classification. For both models, we can safely ignore some of the features without affecting their performance. This reduction in dimensionality helps to control the complexity of model. We observe that removing some of the features from either of the model is equally good as considering all of them. But anyhow it doesn't help to perform beyond what allowing all the feature will do, as shown in Figure 3. In our regularized classifier we regularize the NBC model with SVM as shown below,

$$(1 - \alpha) \cdot \text{NBC} + \alpha \cdot \text{SVM}$$

where $\alpha \in [0, 1]$, and is the parameter which determines the balance between SVM and NBC in resulting model. Because of the complementary behavior of SVM and NBC, and feature subset selection which we do to reduce the complexity of these models, the difference of these two feature subsets will be finite in most of the cases. Therefore while regularizing, we are increasing the complexity of resulting model as we are accounting for more features now. However as evident from experimental results shown in Table 1, this increased complexity results in a lower error on test patterns.

5 Experiments and Results

For measuring the effectiveness of our approach, we performed binary classification on different possible subsets of 20ng datasets. At the preprocessing stage itself we removed some of the features based on the Zipf's curve as mentioned in section 2.3, i.e. those features which are highly frequent or very rare. We divided the data into three subsets, training, validation and testing, for learning SVM and NBC model, for determining best possible α, and for measuring the

Table 1: Experimental Results

Dataset	**Accuracy**		
	SVM	**NBC**	**Hybrid**
alt.atheism vs soc.religion.christian	89.79 %	91.42 %	95.13 %
alt.atheism vs talk.politics.guns	94.96 %	97.48 %	98.49 %
alt.atheism vs talk.politics.mideast	93.75 %	95.50 %	96.50 %
alt.atheism vs talk.religion.misc	67.55 %	76.78 %	77.31 %
comp.graphics vs comp.sys.mac.hardware	85.54 %	89.88 %	91.33 %
comp.graphics vs comp.windows.x	80.93 %	86.34 %	87.89 %
comp.sys.ibm.pc.hardware vs misc.forsale	86.67 %	90.71 %	92.62 %
comp.sys.ibm.pc.hardware vs sci.crypt	94.91 %	95.60 %	96.99 %
comp.windows.x vs sci.crypt	95.05 %	95.83 %	96.88 %
comp.windows.x vs sci.electronics	90.21 %	94.07 %	95.62 %
rec.autos vs rec.motorcycles	90.38 %	90.89 %	93.42 %
rec.motorcycles vs sci.med	94.85 %	95.10 %	97.16 %
sci.space vs soc.religion.christian	92.66 %	96.89 %	98.31 %
talk.politics.guns vs talk.religion.misc	83.91 %	86.33 %	87.67 %
talk.politics.mideast vs talk.religion.misc	90.52 %	94.01 %	95.01 %

Fig. 2: Performance of proposed model *(solid line)* with variation in α, compared against SVM *(dotted line)* and NBC *(dashed line)*

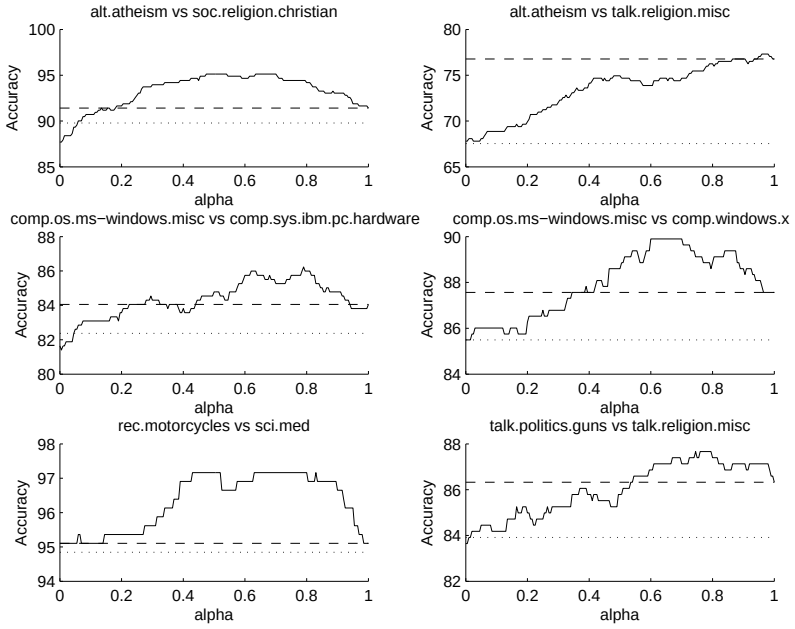
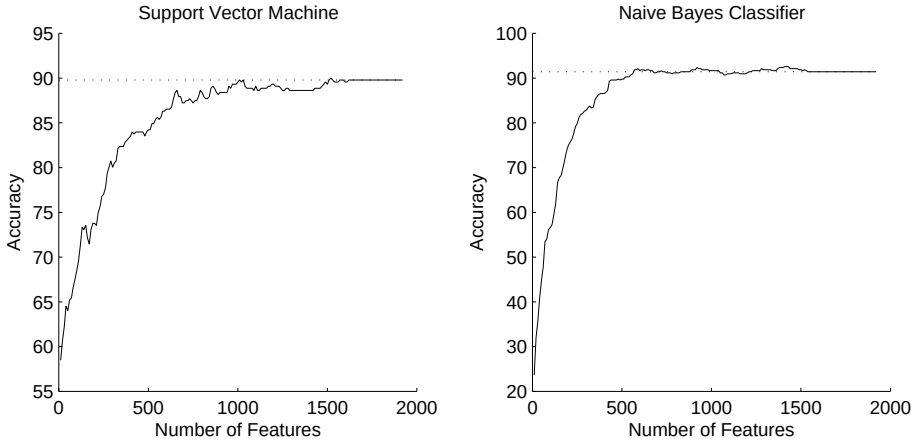


Fig. 3: Regularization of SVM *(left)* and NBC *(right)*



performance of resulting model respectively. After training the SVM and NBC model we further removed some of the features from the models based on their weights in separating hyperplanes. We experimentally determined the number of features that can be removed safely, and found it to be approximately half of the total number of features, for both SVM and NBC. On validation data we record the performance of our hybrid model, for different values of α. The α corresponding to which best performance is recorded is used for prediction of test patterns. Table 1 lists the performance of our naive approach against SVM and NBC.

6 Discussion

When we regularize NBC using SVM, how do we deal with bias of SVM? In our approach we are actually ignoring SVM bias. When we talk of high dimensional space, considering bias becomes insignificant since it is only one degree of freedom which we are ignoring [1]. Our regularized hybrid model outperforms SVM and NBC in most of the cases, but in some cases it may not go beyond NBC. To answer this we looked at the feature subset given by SVM, and weights of features in it. Not much to our surprise, we found the higher weighted features of SVM to be having much lower weights in these cases compared to other cases. We also found the number of support vectors to be reasonably large in these cases. This indicates that SVM is unable to identify dominant discriminative features here. Thus, the complementary features which we expect to be provided by SVM are not helping NBC much, because of their low weights.

7 Conclusion

In this paper our main goal was to amalgamate the complementary behavior of linear SVM and NBC. Specifically it is well known, that the top ranked (based on weights in separating hyperplane) features of NBC are descriptive and those in SVM are discriminative. We have introduced a new linear classifier obtained by regularizing NBC using SVM, which improves upon both Regularized SVM and Regularized NBC at the cost of increased model complexity, which still remains less than the complexity of standard SVM or NBC in terms of number of features. Experimental results of NBC, SVM and our hybrid approach, on some subsets of 20ng dataset have been presented. The performance of the proposed regularized classifier is better than that of either NBC or SVM with or without regularization. One possible direction for future work may be, to look for a principled theoretical formulation of the proposed model.

References

1. Burges, C.: A tutorial on Support Vector Machines for pattern recognition. Data Mining and Knowledge Discovery (1998)
2. Vapnik, V., Chervonenkis, A.: About Structural Risk Minimization principle. Automation Remote Control (1974)
3. Manning, C., Raghavan, P., Schutze, H.: Introduction to Information Retrieval. Cambridge University Press Cambridge (2008)
4. Murphy, K.: Naive Bayes Classifiers (2006)
5. Zipf, G.: Human behavior and the principle of least effort. (1949)
6. Lasserre, J., Bishop, C., Minka, T.: Principled Hybrids of Generative and Discriminative Models. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE (2006)
7. Thomas, T., Flach, P.: WBC_{svm}: Weighted Bayesian Classification based on Support Vector Machines. In: Proceedings of the Eighteenth International Conference on Machine Learning, Citeseer (2001)
8. Raina, R., Shen, Y., Ng, A., McCallum, A.: Classification with Hybrid Generative/Discriminative Models. Advances in Neural Information Processing Systems (2003)
9. Zhang, J., Jin, R., Yang, Y., Hauptmann, A.: Modified Logistic Regression: An approximation to SVM and its applications in large-scale Text Categorization. In: International Conference on Machine Learning. (2003)