RC23358 (W0410-017) October 1, 2004
Computer Science

IBM Research Report

Discriminative Training of Naïve Bayes Classifiers for Natural Language Call Routing

Pengfei Liu, Hui Jiang
Department of Computer Science and Engineering
York University
4700 Keele Street
Toronto, Ontario M3J 1P3
Canada

Imed Zitouni
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

DISCRIMINATIVE TRAINING OF NAIVE BAYES CLASSIFIERS FOR NATURAL LANGUAGE CALL ROUTING

Pengfei Liu[†]*, Hui Jiang*[†]*, Imed Zitouni*[‡]

[†] Department of Computer Science and Engineering
York University, 4700 Keele Street, Toronto, Ont. M3J 1P3, CANADA
[‡] IBM T.J. Watson Research Center, Yorktown Heights, NY, USA 10598
{pfliu,hj}@cs.yorku.ca izitouni@us.ibm.com

Abstract

In this paper, we propose to use a discriminative training(DT) method to improve naive Bayes classifiers in context of natural language call routing. As opposed to the traditional maximum likelihood estimation, all conditional probabilties in Naive Bayes classifers (NBC) are estimated discriminatively based on the minimum classification error (MCE) criterion. A smoothed classification error rate in training set is formulated as an objective function and the GPD (generalized probabilistic descent) method is used to minimize the objective function with respect to all conditional probabilities in NBCs. Two versions of NBC are used in this work. In the first version all NBCs corresponding to various destinations use the same word feature set while destination-dependent feature set is chosen for each destination in the second version. Experimental results on a banking call routing task show that the discriminative training method can achieve up to about 30% error reduction over our best ML-trained system. The proposed method is also compared with the vector-based method used previously by others in call routing task. The comparative results clearly show that NBCs after DT can outperform the vector-based technique.

1. Introduction

Natural language call router is the next generation system designed to replace the currently prevalent touch-tone menus in call centers. It offers a better interaction between callers and the system by prompting "How may I direct your call", replacing the cumbersome "Please listen to the following 9 options...". Call routing based on spoken utterances aims at an accurate analysis of what the callers say and transfers the call to the correct department automatically [1]. In previous work [1, 2], a vector-based information retrieval technique was introduced to perform call routing. A vector-based technique was used to measure the similarity between a new user request and the underlying routing matrix. The Naive Bayes classifiers (NBC) are widely used in many pattern classification problems due to its simplicity and effectiveness. In this paper, we investigate the use of Naive Bayes Classifiers trained with discriminative training techniques to achieve a better classification performance in call routing tasks.

The Naive Bayes Classifier technique is based on the Bayes theorem and particularly suits to the classification tasks where the dimensionality of the inputs is high. For example, if C_j is a class and $\vec{x}_j = (x_{1j}, \ldots, x_{nj})$ is the feature vector for C_j, the NBC assumes the features are independent given the class id C_j, i.e., $Pr(\vec{x}|C) = \prod_i Pr(x_i|C)$. Despite its simplicity, NBC can often outperform other more sophisticated classification methods in practice[3]. In this paper, we propose the use of minimum classification error criterion in discriminative training of NBCs. Minimization of a smoothed classification error rate in the training set can be achieved by adjusting the model dynamically to increase the separation of the correct class from its competitors. Experimental results show that the accuracy and robustness is improved considerably: up to 31% error rate reduction is observed over the ML-trained models. This is the first study of building a natural language call router by combining the NBC with the MCE-based discriminative training for call routing task. Due to the popularity and proven effectiveness of NBC, the proposed formulation can be extended to other algorithms addressing a wide range of tasks, such as topic identification, information retrieval, speech understanding and medical diagnosis, etc.

The remainder of the paper is organized as follows: In section 2, we first show how to apply naive Bayes classifiers to call routing tasks, especially how to calculate normalization term when various features are used for different destinations. In section 3, the MCE/GPD based discriminative training algorithm is derived to estimate NBC conditional probabilities. Experimental results on the USAA banking data set are reported in section 4. Finally, we conclude the paper with our discussions and findings in section 5.

2. Naive Bayes Classifiers for call routing

In NBC-based natural language call routing, each NBC for one destination is trained by a collection of labeled documents (transcriptions of calls routed to that destination). The test document (a new caller request) is classified by measuring the relevance to each destination. A single NBC is required for each destination. Then we estimate all conditional probabilities in all NBCs which correspond to all known destinations. In this paper, we study two kinds of NBC based on two different assumptions. In one version we assume that every class has the same features, i.e., all NBCs have the same child nodes but different conditional probabilities. In the second version, we choose different features for each NBC, which are the most informative to that destination. In this case, different NBCs have various number of child nodes and a proper normalization term becomes critical when we compare across various destinations.

Regarding data preparation, each document (customer utterances) is cleaned by performing the morphological process to remove a list of ignore words. Then the *mutual information* between each pair of word and destination is calculated. The

This work was done when Dr. Zitouni was in Bell Labs

top N words with the highest mutual information are chosen to be the features for that destination. In the NBC corresponding to j-th destination C_j, we need to estimate a conditional probability $p_{ji} = Pr(x_i|C_j)$ for each feature x_i. In our baseline system, all conditional probabilities are estimated from training data based on the ML method.

In testing stage, given any new user request, assume $\vec{y}$ is its feature vector, the request will be classified according to the *Maximum A Posterior* (MAP) decision rule as:

$$\begin{aligned} C^* &= \arg\max_j Pr(C_j = 1|\vec{y}) \\ &= \arg\max_j \frac{\Pr(\vec{y}|C_j = 1) \cdot \Pr(C_j = 1)}{\Pr(\vec{y})} \\ &\equiv \arg\max_j \Pr(\vec{y}|C_j = 1) \cdot \Pr(C_j = 1) \end{aligned} \tag{1}$$

However, if we use various feature set for different destination, we need to properly calculate the above normalization term $\Pr(\vec{y})$ to compare across various destinations. The MAP decision rule in this case becomes:

$$\begin{aligned} C^* &= \arg\max_j \frac{\Pr(\vec{y}|C_j = 1) \cdot \Pr(C_j = 1)}{\Pr(\vec{y})} \\ &= \arg\max_j \frac{\Pr(\vec{y}|C_j = 1) \cdot \Pr(C_j = 1)}{\Pr(C_j = 1)\Pr(\vec{y}|C_j = 1) + \Pr(C_j = 0)\Pr(\vec{y}|C_j = 0)} \end{aligned} \tag{2}$$

3. Discriminative training of NBC's

Discriminative training has recently been proposed for natural language call routing[4]. The generalized probabilistic descent(GPD) algorithm[5] is used here for optimization purpose. In [4], the routing matrix R is viewed as the model parameters to be estimated. The algorithm adjusts the model parameters based on each training sample to minimize the total error rate in the entire training set.

Four steps are involved in the MCE/GPD algorithm, namely constructions of:

1. *discriminant function* for each class.
2. *misclassification function* for the target destination k.
3. *loss function*, which represents the smoothed total errors in training data set.
4. *adjustment rule* which guides the DT algorithm to update the parameter set.

First of all, let's denote all parameters in all NBCs as $\alpha = Pr(C_j = 1)$, $\beta = Pr(C_j = 0)$, $p_{ji} = Pr(x_i = 1|C_j = 1)$ and $q_{ji} = Pr(x_i = 1|C_j = 0)$.

3.1. NBC with same features for each class

In this section, we first derive the MCE/GPD algorithm for NBCs with the same feature set, where we use the MAP decision rule in eq.(1) for decision.

Let p_j be the decision score between the observation vector $\vec{y}$ and the j-th destination C_j as

$$p_j = Pr(C_j|\vec{y}) = \alpha \cdot \prod_{i=1}^{N} (p_{ji})^{M_{ji}} (1 - p_{ji})^{\delta(M_{ji})} \tag{3}$$

where N is the number of features for each destination C_j. $M_{ji} (= 0, 1, 2...N)$ is frequency of i-th feature occurring in the observation $\vec{y}$, and $\delta(M_{ji})$ is defined as follows,

$$\delta(M_{ji}) = \begin{cases} 0 & \text{when } M_{ji} > 0, \\ 1 & \text{when } M_{ji} = 0. \end{cases}$$

Note the value of p_j in eq.(3) is a real number between 0 and 1. We define the *discriminant function* for j-th class and observation vector $\vec{x}$ as negative logarithm of p_j:

$$\begin{aligned} g_j(\vec{y}, p) &= -\ln(p_j) \\ &= -\Big\{ \ln \alpha + \sum_{i=1}^{N} [M_{ji} \cdot \ln p_{ji} + \delta(M_{ji}) \cdot \ln(1 - p_{ji})] \Big\} \end{aligned} \tag{4}$$

Assuming the correct target destination for $\vec{y}$ is k, the *misclassification measure* is defined as

$$\begin{aligned} d_k(\vec{y}, p) &= g_k(\vec{y}, p) - \left[\sum_{j \neq k} g_j(\vec{y}, p)^{\eta} \right]^{\frac{1}{\eta}} \\ &= -\ln(p_k) - \left[\sum_{j \neq k} [-\ln(p_j)]^{\eta} \right]^{\frac{1}{\eta}} \end{aligned} \tag{5}$$

where η is a *negative* number. Note that as $\eta \to -\infty$, the second term in eq.(5) converges the the score from the most competing class. Intuitively, $d_k() > 0$ indicates a misclassification and $d_k() < 0$ implies a correct classification.

Then the above misclassification measure is plugged into a sigmoid function to obtain the following *loss function*:

$$l(p) = \sum_{\vec{y}} \frac{1}{1 + e^{-\gamma d_k(\vec{y}, p) + \theta}} \tag{6}$$

where the summation is done over the entire training set, and γ and θ are positive constants to control the slope and shift of the sigmoid function. The function $l(p)$ is a smoothed measure of total errors in the entire training set.

Finally we define the *adjustment rule* to update NBC parameters to minimize the above loss function as follows:

$$p_v(t + 1) = p_v(t) - \epsilon_t \nabla l_k(\vec{y}, p_v(t)) \tag{7}$$

where $p_v(t)$ is the feature vector for v-th destination at iteration t and ϵ_t is the step size in the GPD algorithm. The gradient can be calculated as:

$$\nabla l_k(\vec{x}, p_v) = \frac{\partial l_k(\vec{y}, p_v)}{\partial p_v} = \frac{\partial l_k}{\partial d_k} \cdot \frac{\partial d_k(\vec{y}, p_v)}{\partial p_{vw}} \tag{8}$$

We know

$$\frac{\partial l_k}{\partial d_k} = \gamma l_k(d_k)(1 - l_k(d_k)) \tag{9}$$

Finally, we derive the updating formula for all NBC parameters $\{p_{ji}\}$ as follows:

$$p_{vw}(t+1) = \begin{cases} p_{vw}(t) + \epsilon_t \cdot \frac{\partial l_k}{\partial d_k} \cdot \\ \left[M_{kw} \cdot \frac{1}{p_{kw}} + \delta(M_{kw}) \cdot \frac{-1}{1-p_{kw}} \right] & \text{if } v = k, \\ p_{vw}(t) - \epsilon_t \cdot \frac{\partial l_k}{\partial d_k} \cdot Z^{\frac{1}{\eta}-1} \cdot (-\ln p_v)^{\eta-1} \\ \cdot \left[M_{vw} \cdot \frac{1}{p_{vw}} + \delta(M_{vw}) \cdot \frac{-1}{1-p_{vw}} \right] & \text{if } v \neq k. \end{cases} \tag{10}$$

where $Z = \sum_{v \neq k} (g_v)^\eta = \sum_{v \neq k} (-\ln p_v)^\eta$. In each iteration, we explicitly check every conditional probability value to make sure it never goes out of the valid range $[0, 1]$.

It can be seen from (10) that the feature vector for the correct destination and competing destination is updated differently. It confirms the idea that the minimization of classification error is achieved by increasing the score for the correct class while punishing the competing classes.

3.2. NBC with different features for each class

In this section, we study the MCE/GPD formulation for NBC's with different feathers for each class, where we use the MAP decision rule in eq.(2) to make classification decision. Due to the normalization term, we need to update not only conditional probabilities $\{p_{ji}\}$ but also $\{q_{ji}\}$. For brevity, here we omit the details and give only the main results.

The *Discriminant function* for jth NBC is calculated as:

$$\begin{aligned} g_j(\vec{y}, p) &= -\Big\{ \ln \alpha + \sum_{i=1}^{N_j} [M_{ji} \cdot \ln p_{ji} + \delta(M_{ji}) \ln(1 - p_{ji})] \\ &- \ln \Big[\alpha \cdot \prod_{i=1}^{N_j} (p_{ji})^{M_{ji}} \cdot (1 - p_{ji})^{\delta(M_{ji})} \\ &+ \beta \cdot \prod_{i=1}^{N_j} (q_{ji})^{M_{ji}} \cdot (1 - q_{ji})^{\delta(M_{ji})} \Big] \Big\} \end{aligned} \tag{11}$$

where N_j denotes the number of features for j-th destination.

The *Misclassification measure* for the target class

$$d_k(\vec{y}, p) = g_k(\vec{y}, p) - \left[\sum_{j \neq k} g_j(\vec{y}, p)^\eta \right]^{\frac{1}{\eta}} \tag{12}$$

where $\eta < 0$.

The *Loss function* is formulated similarly as in eq.(6). And the NBC parameters, both $\{p_{ji}\}$ and $\{q_{ji}\}$, are updated based on the GPD algorithm as shown in eq.(7). At last, we derive the formula to update $\{p_{ji}\}$ and $\{q_{ji}\}$ as follows:

$$p_{vw}(t+1) = \begin{cases} p_{vw}(t) + \epsilon_t \cdot \frac{\partial l_k}{\partial d_k} \cdot \\ [M_{kw} \frac{1}{p_{kw}} + \delta(M_{kw})(\frac{-1}{1-p_{kw}}) - \\ \frac{[\alpha \prod_{i \neq w}^{N_k} (p_{ki})^{M_{ki}} (1-p_{ki})^{\delta(M_{ki})}](M_{kw} - \delta(M_{kw}))}{D}] \\ \qquad \text{if } v = k, \\ p_{vw}(t) - \epsilon_t \cdot \frac{\partial l_k}{\partial d_k} \cdot Z^{\frac{1}{\eta}-1} \cdot (-\ln p_v)^{\eta-1} \cdot \\ \Big[M_{vw} \frac{1}{p_{vw}} + \delta(M_{vw})(\frac{-1}{1-p_{vw}}) - \\ \frac{[\alpha \prod_{i \neq w}^{N_v} (p_{vi})^{M_{vi}} (1-p_{vi})^{\delta(M_{vi})}](M_{vw} - \delta(M_{vw}))}{D}\Big] \\ \qquad \text{if } v \neq k. \end{cases} \tag{13}$$

$$q_{vw}(t+1) = \begin{cases} q_{vw}(t) - \epsilon_t \cdot \frac{\partial l_k}{\partial d_k} \cdot \\ \frac{[\beta \prod_{i \neq w}^{N_k} (q_{ki})^{M_{ki}} (1-q_{ki})^{\delta(M_{ki})}](M_{kw} - \delta(M_{kw}))}{D} \\ \qquad \text{if } v = k, \\ q_{vw}(t) + \epsilon_t \cdot \frac{\partial l_k}{\partial d_k} \cdot Z^{\frac{1}{\eta}-1} \cdot (-\ln p_v)^{\eta-1} \cdot \\ \frac{[\beta \prod_{i \neq w}^{N_v} (q_{vi})^{M_{vi}} (1-q_{vi})^{\delta(M_{vi})}](M_{vw} - \delta(M_{vw}))}{D} \\ \qquad \text{if } v \neq k. \end{cases} \tag{14}$$

where the denominator D stands for

$$\begin{aligned} D = \alpha \cdot \prod_{i=1}^{N_v} (p_{vi})^{M_{vi}} (1 - p_{vi})^{\delta(M_{vi})} + \\ \beta \cdot \prod_{i=1}^{N_v} (q_{vi})^{M_{vi}} (1 - q_{vi})^{\delta(M_{vi})} \end{aligned}$$

4. Experiments

Experiments were conducted on a banking call routing task called USAA to evaluate the effectiveness of the methods. The training data consists of 23 different departments and 3749 calls. It is used to estimate conditional probabilities in all NBCs. In the baseline system, conditional probabilities are estimated based on maximum likelihood criterion. After morphological processing, for NBC with the same feature set as described in section 3.1, we choose top 391 words with highest overall *mutual information* averaged across all destination as the common feature set for all 23 destinations, denoted as NCB-v1 for short hereafter. For NBCs with various feature sets for different destinations as described in section 3.2, we choose at most 120 features for each destination based on the *mutual information* of all feature words versus that particular destination. All features chosen for both versions must have frequency greater than 2 in training set. Experimental results are reported on both human transcription (Bank-HT) and ASR recognition output results (Bank-ASR). In the MCE/GPD training, we always use the ML-trained NBCs as initial models in the GPD algorithm.

4.1. Selection of parameters in MCE/GPD

First of all, we study how to determine the appropriate parameters for the MCE/GPD algorithm, such as η, γ, θ, ϵ_t, etc. They must be chosen based on experimental results. For each parameter, we initially predict a reasonable range and then fine-tune it through a series of experiments. For step size ϵ_t, we use a fixed value for simplicity. A lot of experiments were conducted to get a good parameter set. For NBC-v1, we choose the following parameter set: $\eta = -20, \gamma = 5.0, \theta = 1.0, \epsilon_t = 0.00001$. Another set of parameters, $\eta = -20, \gamma = 5.0, \theta = 4.0, \epsilon_t = 0.000001$, is used for NBC-v2.

Secondly, we study the behavior of the MCE/GPD algorithm as training iterations proceed. Here we take NBC-v2 as an example. In Figure 1, we draw the value of the objective function as shown in eq.(6), i.e., the smoothed error counts in training set, as a function of training iterations in MCE/GPD training. In Figure 2, we draw the actual 0-1 count of errors in training set as a function of GPD iterations. From these two figures, we clearly see that the MCE/GPD algorithm can significantly reduce the training error rate. After 150 iterations, the actual classification error rate in training set is reduced from 9.2% (with ML-trained NBCs) down to 8.1%. Meanwhile, we also see that the value of the objective function in eq.(6) is dragged

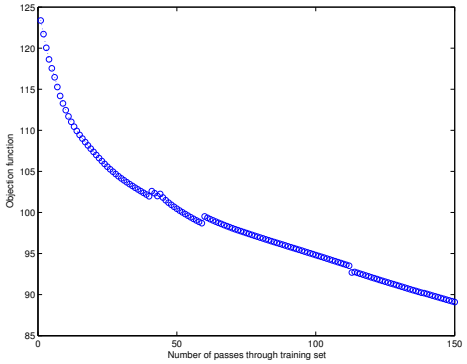

Figure 1: The value of the objective function in eq.(6) is shown as a function of DT iterations. (for NBC-v2 in Bank-HT training data)

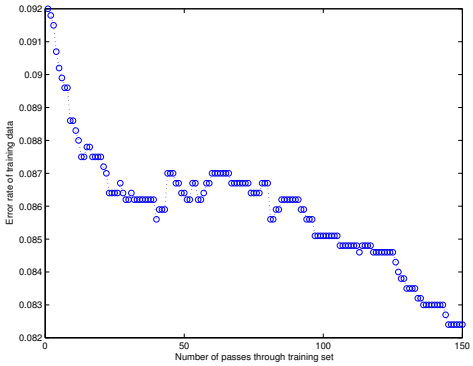

Figure 2: The classification error rate is shown as a function of DT iterations. (for NBC-v2 in Bank-HT training data)

down from 125.0 (with ML-trained NBCs) to 89.0. In Figure 3, we draw the classification error rate in the HT test data set as a function of the GPD iterations. From the figure, we can see that the best performance of the MCE/GPD is achieved after about 50 iterations. The classification error rate drops from 7.17% (with ML-trained NBCs) to 5.54%, which is a relative error reduction of about 22.7%. After 50 iterations, the error rate in test data begins to increase which indicates over-fitting in discriminative training after 50 iterations. Thus, the NBCs after 50 iterations are chosen as the best DT models for the following experiments.

4.2. Comparison with the vector-based methods

In order to prove the effectiveness of NBC with DT, the approaches proposed in this paper are compared with the vector-based call routers as reported in [4]. In Table 1, we first list the performance of the vector-based method (before and after DT) as reported in [4]. Then we give the best performance achieved by NBC-v1 and NBC-v2 (before and after DT). From the results, it is clear that DT is also very effective to improve the performance of NBCs. For example, when evaluated in HT test data, NBC-v2 after DT achieves 5.54% classification error rate, which is a significant improvement from 7.17% before DT.

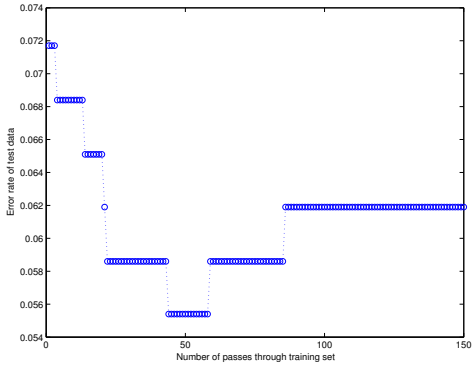

Figure 3: The classification error rate in test data is shown as a function of DT iterations. (for NBC-v2 in Bank-HT test data)

Table 1: *Comparison of error rate between vector-based and NBC call routers*

	Bank-HT	Bank-ASR
vector-based	7.82%	10.42%
vector-based + DT	6.84%	8.47%
NBC-v1	8.47%	10.71%
NBC-v1 + DT	5.86%	**9.09%**
NBC-v2	7.17%	10.71%
NBC-v2 + DT	**5.54%**	9.74%

5. Discussion and conclusion

In this paper, We have demonstrated that the MCE/GPD based discriminative training can be used to minimize the classification error for NBC in natural language call routing tasks. We propose two versions of NBC armed with DT retraining process. Both of them can outperform the vector-based call routers. The effectiveness of the simple NBC is proven.

To our knowledge, this is the first study of its kind to apply discriminative training to Naive Bayes classifiers (NBC). Due to the popularity and effectiveness of NBC, we believe it will play an increasingly important role in the areas, such as speech understanding, topic identification and information retrieval.

6. References

[1] B. Carpenter and J. Chu-Carroll, "Natural language call routing: A robust, self-organizing approach", ICSLP-98(Sydney, Australia),pp.2059-2062, Dec. 1998.

[2] J.Chu-Carroll and B. Carpenter, "Vector-based natural language call routing", Computational Linguistics, vol.25, no.3,pp.361-388,1999.

[3] P. Langley,W. Iba, and K. Thompson, "An analysis of Bayesian classifiers", Proceedings of the Tenth National Conference on Artificial Intelligence, pp. 399C406, San Jose, CA, 1992.

[4] H.-K.J.Kuo and C.-H.Lee, "Discriminative training of natural language call routers", IEEE Transaction on Speech and Audio Processing, vol.11,no.1,pp.24-35, Jan. 2003.

[5] S.Katagiri, C.-H.Lee and B.-H.Juang, "Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method",Proceeding of the IEEE,vol.86,pp.2345-2373,Nov.1998.