RC 22393 (W0204-041) 4/9/2002
Mathematics 14 pages

Research Report

Multiplicative Adjustment of Class Probability: Educating Naïve Bayes

Se June Hong, Jonathan Hosking, Ramesh Natarajan

IBM Research Division
T. J. Watson Research Center
Yorktown Heights, NY 10598

Multiplicative Adjustment of Class Probability: Educating Naïve Bayes

Se June Hong, Jonathan Hosking, Ramesh Natarajan

IBM T.J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598
e-mail: sjhong@us.ibm.com, hosking@watson.ibm.com, nramesh@us.ibm.com

Abstract. Starting from the Naïve Bayes model, we develop a new concept for aggregating items of evidence in classification problems. We show that in Naïve Bayes, each feature variable contributes a multiplicative adjustment factor to the estimated class probability. We next introduce a way of controlling the importance of the feature variables by raising each adjustment factor to a different power. The powers are chosen so as to maximize the accuracy of estimated class probabilities on the training data, and their optimal values are obtained by fitting a logistic regression model whose explanatory variables are constructed from the feature variables of the classification problem. This optimization accomplishes more than what feature selection does for Naïve Bayes. We call this new model family the Adjusted Probability Model (APM). We also define a regularized version, APMR. Experiments demonstrate that APMR is surprisingly effective. Assigning different degrees of importance to the feature variables seems to remove much of the naïveté from Naïve Bayes.

Key words: Naïve Bayes, probabilistic model, classification, ensemble of models, adjusted probability model, evidence aggregation, maximum likelihood.

1. Introduction

The Naïve Bayes (NB) model for classification problems is attractive for its simplicity and its good model understandability. There have been several studies of how well the model performs as a classifier. Domingos and Pazzani (1997) explore theoretical conditions under which NB may be optimal even though its assumption of independence of the feature values given the class may not hold, and also supply empirical evidence. Hand and Yu (2001) give arguments on why the independence assumption is not so absurd. Garg and Roth (2001) consider all joint distributions and show that the number of these distributions goes down exponentially with their distance from the product distribution of NB, thereby explaining the power of NB beyond the independence assumption. These studies focus on classification error.

In many data mining applications, the desired model output is the class probability. Examples include marketing applications in which a mailing is sent out to consumers whose estimated probability of response to the mailing exceeds a given level; this level is chosen to maximize expected profit, based on a "lift curve" (e.g., Piatetsky-Shapiro and Steingold, 2000). We focus on class probability estimation, using a new model derived from a novel interpretation of NB. This derivation is given in section 2. The new model yields class probability estimates that are given by the prior probability of the class with successive multiplicative adjustments arising from the evidence supplied by each feature; we therefore call it the Adjusted Probability Model (APM). Each adjustment factor has an associated importance parameter that is estimated by fitting a logistic regression model. The estimation of class probability based on a single feature value can be construed as a simple model. Our formulation of APM is therefore a new way of aggregating the outputs of an ensemble of models. The aggregation uses multiplicative adjustments, in contrast to additive aggregation traditionally done by boosting or bagging. We also introduce a regularized version of APM that we call APMR.

In section 3, we present the straightforward process of "educating Naïve Bayes", or training the model parameters by the maximum likelihood criterion for both APM and APMR. Section 4 presents the results of experiments that investigate the performance of the new models on some UCI problems that have exclusively nominal features.

In section 5 we discuss APM and APMR in relation to other relevant techniques. We show that several other NB based techniques are syntactically close to our formulation of APM, but are not as natural or optimal for the goal of modeling the class probabilities. Section 6 contains some concluding remarks.

2. A new interpretation of Naïve Bayes and a new model

Suppose that the data consist of n examples. The ith example has class label c_i and feature values in vector X_i. Let x_{ij} be the feature value of feature j in example i. The column vector of values for feature j will be denoted by x_j. The probability that the class is k given feature values X is then

$$P(C = k|X) = P(C = k)\frac{P(X|C = k)}{P(X)}\,. \tag{1}$$

Naïve Bayes makes the assumption that the feature values are independent given the class, i.e. that

$$P(X|C = k) = \prod_j P(x_j|C = k)\,. \tag{2}$$

Thus when comparing the possible classes for a given example we have

$$P(C = k|X) \propto P(C = k)\prod_j P(x_j|C = k)\,; \tag{3}$$

this is the usual NB formulation.

We make the important observation that

$$P(x_j|C = k) = \frac{P(x_j)P(C = k|x_j)}{P(C = k)}\,, \tag{4}$$

so the NB model may equivalently be written as

$$P(C = k|X) \propto P(C = k)\prod_j \frac{P(C = k|x_j)}{P(C = k)}\,. \tag{5}$$

This gives a new interpretation of NB: the class probability given the values of a set of features is proportional to the prior probability of the class adjusted multiplicatively by factors each of which reflects the influence of one of the individual features. Each adjustment factor is simply the ratio between the class probability given a feature value and the class prior probability. A natural extension of this interpretation is to permit the features to have different degrees of influence: importance parameters (weights) can be introduced as

$$P(C = k|X) \propto P(C = k)\prod_j \left(\frac{P(C = k|x_j)}{P(C = k)}\right)^{\alpha_j}\,. \tag{6}$$

Expression (6) defines our Adjusted Probability Model, APM.

The naïveté of NB arises from assumption (2), which ignores the effect of correlation between features. For example, if there are two nearly identical features they both contribute to the product in (5) although they are both based on the same information. Introducing the α coefficients in (6) enables the model to allow for such duplication: one would expect that in a properly trained model the α coefficients of nearly identical features would sum to 1 rather than each being equal to 1. Thus we might say that training APM to obtain an optimal set of α_j values amounts to "educating" Naïve Bayes to be more sophisticated.

We now show that the training process can be reduced to a logistic regression problem. From (6) we have

$$\frac{P(C|X)}{1-P(C|X)} = \frac{P(C)}{1-P(C)} \prod_j \left(\frac{P(C|x_j)\{1-P(C)\}}{\{1-P(C|x_j)\}P(C)} \right)^{\alpha_j}. \tag{7}$$

Rewriting for a given $X = X_i$, we have

$$\ln \frac{P(C|X_i)}{1-P(C|X_i)} = q_0 + \sum_j \alpha_j q_{ij} \tag{8}$$

where

$$q_0 = \ln \frac{P(C)}{1-P(C)}, \tag{9}$$

$$q_{ij} = \ln \frac{P(C|x_{ij})\{1-P(C)\}}{\{1-P(C|x_{ij})\}P(C)}. \tag{10}$$

Equation (8) can also be written as

$$P(C|X_i) = \frac{1}{1+e^{-q_0-\boldsymbol{\alpha}^{\mathrm{T}}Q_i}} \tag{11}$$

where $\boldsymbol{\alpha}$ is a vector with elements α_j and Q_i is a vector with elements q_{ij} defined as in (10).

Equations (8) and (11) display APM in the form of a logistic regression model. Note that the term q_0 does not have an α coefficient: in statistical terminology it is an offset rather than an intercept. For a two-class problem, it is straightforward

to estimate the model parameters α_j by the method of maximum likelihood: details are given in section 3. In a problem with three or more classes, equation (8) can be regarded as specifying a polytomous logistic regression. In practice it is more convenient to regard a multi-class problem as a succession of two-class problems (of the form one class vs. the rest): the estimation is easier and we gain the freedom of permitting the α_j values to be different in the different two-class problems. This enables a feature to have greater influence in distinguishing some classes as opposed to others, a situation that can plausibly occur in practice.

The probabilities needed to define the quantities q_0 and q_{ij} in equations (9)–(10) are naturally estimated from the data by

$$P(C=k) = \#\{i : c_i = k\}/n\,, \tag{12}$$

$$P(C=k|x_j=v) = \frac{\#\{i : x_{ij}=v \,\&\, c_i=k\}}{\#\{i : x_{ij}=v\}}\,. \tag{13}$$

Here one may use a small-sample adjustment, e.g. increasing the numerator by L and the denominator by $2L$ (this is the Laplace correction when $L=1$ and a modified Laplace correction when $L=1/n$ — see Kohavi et al., 1997).

The foregoing computations are for a discrete-valued feature. Features that take values in a continuous range may first be discretized, or some continuous relation between $P(C|x_j)$ and x_j can be developed for each feature. We do not discuss these possibilities further here.

Overfitting to the training set is a definite possibility in this scheme. We therefore define a regularized version of APM, APMR, by restricting the α_j coefficients in (8) to satisfy $\sum_j \alpha_j^2 = m$ where m is chosen to maximize the accuracy of out-of-sample predictions of class probabilities. Finding the optimal value of m is a form of structural risk minimization. We use an internal cross validation: in each fold the training set is divided into train-train and train-validate sets. The value $\hat{m}$ is found that produces the smallest median (or mean) loss over the train-validate sets of all folds. (The loss is the criterion function used to fit the logistic regression model (8), and is formally defined in equation (15) below.) Model (8) is then fitted to the entire training set, with the restriction that $\sum_j \alpha_j^2 = \hat{m}$. This yields the final APMR classifier.

In detail, to find the optimal value of m, we proceed as follows. Let the $\sum_j \alpha_j^2$ value from the unrestricted APM model be m_{u}. Denote by APM(m) the APM model with the restriction $\sum_j \alpha_j^2 = m$. To find the optimum value of m, search for a local minimum loss within the interval of m values between 0 and m_{u}. This is effective

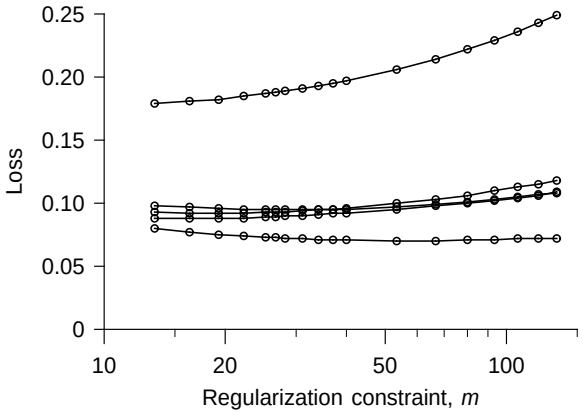

Figure 1. Loss, as a function of m, for the five train-validate sets in the internal cross-validation used in training the APMR model for the SDNA example from section 4.

because in our experiments the average loss over the examples in the train-validate set is generally concave in m. An example is given in Figure 1. For 10 values of m, equally spaced between 0 and m_{u}, generate APM(m) models for all the folds in a 5-fold cross-validation. In these APM(m) models, the probability estimates q_0 and q_{ij} in equations (9) and (10) are computed from the entire training set (i.e. they are the values that were computed for the unrestricted APM model), rather than being computed separately for the train-train sets of each fold of the cross-validation. Find the value m' that produces the smallest median (or mean) average train-validate set loss across all folds. Define a new search interval bounded by the two adjacent values to m'. Define 10 equally spaced values in this interval (one of them being m'). Again find the m value that produces the smallest median (or mean) average train-validate set loss across all folds. In all this procedure requires $5 \times (10 + 9) = 95$ runs of the APM(m) algorithm. The model APM($\hat{m}$) is finally fitted to the entire training set to obtain APMR.

3. Optimization of the importance parameters

We assume a two-class problem with class labels 0 and 1. The likelihood of the data under the APM model is

$$\prod_i \{P(C|X_i)\}^{c_i} \{1 - P(C|X_i)\}^{1-c_i} \tag{14}$$

where $P(C|X_i)$ is given by equation (11). Equivalently to maximizing (14), we can minimize the average loss per example,

$$\text{Loss} = -n^{-1} \sum_i [c_i \log P(C|X_i) + (1 - c_i) \log\{1 - P(C|X_i)\}] . \quad (15)$$

This loss measure directly penalizes incorrect estimation of class probabilities: the loss is zero when the actual class has estimated probability 1, and increases as this estimated probability decreases (the loss is essentially the "fair fee" of Good, 1952; see also Dawid, 1982). We use logarithms to base 2 in (15), so the loss is measured in bits.

The minimization of (15) can be achieved by the Newton-Raphson method, which we implement as the APM algorithm. We denote by $\mathbf{p}$ a vector with elements $p_i = P(C|X_i)$, by $\mathbf{D}$ a diagonal matrix with entries $p_i(1-p_i)$, by $\boldsymbol{\alpha}$ a vector with elements α_j, and by $\mathbf{Q}$ a matrix with elements q_{ij}.

APM algorithm:

1) Initialize: $\boldsymbol{\alpha} = \mathbf{0}$.
2) Compute $\mathbf{p}$ and form $\mathbf{D}$.
3) Compute $\Delta\boldsymbol{\alpha} = (\mathbf{Q}^{\mathrm{T}}\mathbf{D}\mathbf{Q})^{-1}\mathbf{Q}^{\mathrm{T}}(C - \mathbf{p})$.
4) Update $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}$.
5) If all elements of $\Delta\boldsymbol{\alpha}$ are less in absolute value than some small value, say 10^{-10}, stop; otherwise, continue iteration from step 2).

The optimization for the APM(m) model makes use of a Lagrangian multiplier λ and minimizes

$$\text{Loss} + \lambda\left(\sum_i \alpha_j^2 - m\right) . \quad (16)$$

Again, the minimization is performed via the Newton-Raphson method. After some algebra, we obtain the following algorithm. $\mathbf{I}$ denotes the identity matrix.

APM(m) algorithm:

1) Run the APM algorithm, obtaining the solution $\boldsymbol{\alpha}_{\mathrm{u}}$; let $m_{\mathrm{u}} = \boldsymbol{\alpha}_{\mathrm{u}}^{\mathrm{T}}\boldsymbol{\alpha}_{\mathrm{u}}$.
2) Initialize: $\boldsymbol{\alpha} = \boldsymbol{\alpha}_{\mathrm{u}} \times (m/m_{\mathrm{u}})^{1/2}$.
3) Compute $\mathbf{p}$ and $\mathbf{D}$.
4) Compute $\mathbf{r} = \mathbf{Q}^{\mathrm{T}}(\mathbf{p} - C) + 2\lambda\boldsymbol{\alpha}$.
5) Compute $s = \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\alpha} - m$.
6) Compute $\Delta\lambda = [4\boldsymbol{\alpha}^{\mathrm{T}}(\mathbf{Q}^{\mathrm{T}}\mathbf{D}\mathbf{Q} + 2\lambda\mathbf{I})^{-1}\boldsymbol{\alpha}]^{-1}[s - 2\boldsymbol{\alpha}^{\mathrm{T}}(\mathbf{Q}^{\mathrm{T}}\mathbf{D}\mathbf{Q} + 2\lambda\mathbf{I})^{-1}\mathbf{r}]$.

7) Compute $\Delta\boldsymbol{\alpha} = -(\mathbf{Q}^{\mathrm{T}}\mathbf{D}\mathbf{Q} + 2\lambda\mathbf{I})^{-1}(2\boldsymbol{\alpha}\Delta\lambda + \mathbf{r})$.
8) Update $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}$ and $\lambda \leftarrow \lambda + \Delta\lambda$.
9) If all elements of $\Delta\boldsymbol{\alpha}$ have absolute value less than some small value, say 10^{-10}, stop; otherwise, continue iteration from step 3).

The APM and APM(m) algorithms can occasionally be numerically unstable. More robust optimization algorithms will be developed in the future. The experiments reported here used some ad hoc modifications, as follows. In both APM and APM(m) algorithms some elements of $\mathbf{p}$ may become 0 or 1 to machine precision, causing a matrix inversion problem. We replaced such values by 10^{-10} and $1 - 10^{-10}$ respectively. Also, after step 5 of the APM(m) algorithm, we terminated the loop if the loss, which we also computed, was small (say less than 0.001) and s/m was similarly small. The run time is dominated by the matrix inversion.

4. Experiments

For our experiments we took most of the classification problems in the UCI data repository that do not have numerical features. We used the following problems: DNA (3 classes, 60 features, 2000 training examples and a prescribed test set of 1186 examples), SDNA (3190 combined "original" examples of DNA, of which 15 examples have partially unknown feature values and were discarded), mushroom (2 classes, 23 features, 8124 examples), breast (2 classes, 9 features, 286 examples), vote (2 classes, 16 features, 435 examples), and lymph (4 classes, 18 features, 148 examples). Following most other reported experiments on these problems we treated "missing" as a separate value of each feature.

The results are summarized in Table 1. The reported results are means and standard deviations over the 100 test sets obtained from 10 repetitions of 10-fold cross-validation (except for the DNA problem, which has a fixed train/test split). In the table we indicate the small-sample adjustment that we used. Only problems with a small number of examples, viz., breast and lymph, are sensitive to the magnitude of the modifying factor L in the Laplace correction. It was not clear whether a small-sample adjustment was used in the results reported by others. All runs were done in interpreted APL on a 133MHz IBM RS/6000 workstation. Run times are reasonable, e.g. one 10-fold cross validation run on the SDNA data generating NB, APM and APMR models took about two hours. (For this 3-class problem, this involved running the APM(m) algorithm a total of $10 \times 95 \times 3 = 2850$ times.)

For problems with more than two classes we report the results for each class (class k vs. rest is denoted as k for short in Table 1) and for "All" classes. The all-classes results were computed by combining the class probabilities from the individual class models, although completely different APMR models were generated for different classes. We report the classification error rate and the loss measure (15), which directly measures the inaccuracy of estimated class probabilities. In the loss measure, to avoid infinities estimated class probabilities of 0 and 1 were replaced by 10^{-10} and $1 - 10^{-10}$ respectively. For the APM and APMR methods, we also report m or $\hat{m}$ values and the number of negative α_j values in the final classifier.

Our loss measure has not been reported in other papers, but we can compare its value across the NB, APM, and APMR methods. In all cases, APMR produces a smaller loss than NB, often by a significant margin. What is surprising is that our optimization was done with the goal of maximizing the accuracy of class probability, i.e. minimum loss, and yet the classification error rates are also often superior to NB and to many other methods whose results have been reported in the literature, as follows.

DNA: Gärtner and Flach (2001) report error rates for class 1 ranging from 2.11% to 3.88% depending on the SVM parameter setting for SVM$^{\text{light}}$ and their BSVM. The APMR rate for this case is 2.50%. The all-class error rate of APMR is 3.49% which compares well with the rate of 5.0% for bagged CART reported by Breiman (1996).

SDNA: For "all classes", the APMR error rate is 3.9%. Freund and Schapire (1996) reported error rates for C4.5 boosted and bagged of 4.9% and 5.2%. Freund and Mason (1999) reported error rates for boosted C5.0, boosted stumps and AD-Tree of 3.7%, 6.0% and 4.1%. Quinlan (1996) reported error rates for bagged C4.5 and boosted C4.5 of 5.58% and 5.43%.

Mushroom: Gärtner and Flach (2001) report an error rate of $4.23 \pm 0.75\%$ for NB and zero error for both linear SVM and WBC_{SVM}. Zero error is also attained by both APM and APMR.

Breast(-cancer): The APMR error rate of 27.97% is slightly higher than that of NB, 27.59%. However, APMR's error rate has much smaller standard deviation, and its average loss is significantly better. Hall and Holmes (2000) report an error rate of 26.88% for NB and 26.46% for NB with selection of best features. Gärtner and Flach (2001) report error rates of $26.88 \pm 7.48\%$ for NB, $31.1 \pm 7.85\%$ for WBC_{SVM}, and $30.12 \pm 6.57\%$ for a linear SVM.

Table 1. Comparison of NB, APM and APMR. "Error" is the misclassification rate. "Loss" is the loss measure of equation (15). "# neg." is the number of negative α_j coefficients in the final model. Table entries are means, and numbers in parentheses are standard deviations, over 10 repetitions of the folds of a 10-fold cross-validation.

Class	Method	Error (%)	Loss	m or $\hat{m}$	# neg.
DNA: (APMR results are averaged over 10 runs; no Laplace correction)					
1	NB	3.63	0.13		
	APM	2.70	0.12	156	11
	APMR	2.50	0.10	21	8.0
2	NB	3.12	0.19		
	APM	4.22	0.20	209	12
	APMR	3.51	0.17	18	7.8
3	NB	8.09	0.32		
	APM	6.49	0.24	114	11
	APMR	6.41	0.23	27	10.0
All	NB	5.40	0.25		
	APM	3.96	0.24		
	APMR	3.49	0.24 (0.00)		
SDNA (no Laplace correction)					
1	NB	3.12 (0.97)	0.13 (0.05)		
	APM	2.65 (0.93)	0.13 (0.06)	127 (57)	10.9 (1.5)
	APMR	2.50 (0.75)	0.12 (0.05)	23 (9)	7.5 (1.6)
2	NB	2.99 (0.89)	0.18 (0.08)		
	APM	3.27 (0.85)	0.17 (0.08)	77 (27)	11.5 (1.6)
	APMR	2.87 (0.77)	0.16 (0.07)	14 (4)	9.0 (1.5)
3	NB	7.78 (1.46)	0.30 (0.05)		
	APM	6.28 (1.48)	0.24 (0.05)	100 (20)	11.7 (1.6)
	APMR	6.12 (1.36)	0.23 (0.03)	29 (8)	9.6 (1.8)
All	NB	4.36 (1.05)	0.25 (0.09)		
	APM	4.27 (1.09)	0.25 (0.09)		
	APMR	3.90 (1.09)	0.24 (0.08)		
Mushroom (modified Laplace correction, $L = 1/n$)					
	NB	0.45 (0.23)	0.02 (0.01)		
	APM	0	0	19816 (927)	7.4 (0.9)
	APMR	0	0	5714 (2732)	5.1 (0.4)
Breast (full Laplace correction, $L = 1$)					
	NB	27.59 (7.36)	0.93 (0.23)		
	APM	28.53 (7.54)	0.85 (0.13)	6.0 (1.3)	0.04 (0.2)
	APMR	27.97 (6.27)	0.82 (0.11)	2.2 (1.4)	0
Vote (modified Laplace correction, $L = 1/n$)					
	NB	9.72 (4.27)	0.94 (0.48)		
	APM	4.66 (2.89)	0.29 (0.33)	1002 (3425)	6.6 (0.8)
	APMR	4.25 (2.75)	0.20 (0.16)	446 (3280)	6.1 (0.9)
Lymph (full Laplace correction, $L = 1$)					
1	NB	2.00 (4.50)	0.13 (0.25)		
	APM	0.67 (2.11)	0.16 (0.47)	26 (12)	7.5 (1.6)
	APMR	0.67 (2.11)	0.02 (0.04)	17 (8)	11.6 (1.1)
2	NB	15.52 (11.58)	0.58 (0.45)		
	APM	19.62 (7.49)	0.93 (0.81)	278 (191)	4.9 (0.9)
	APMR	18.24 (11.90)	0.53 (0.30)	16 (10)	3.5 (1.4)
3	NB	12.14 (8.36)	0.57 (0.46)		
	APM	16.33 (16.25)	0.77 (0.79)	8893 (22211)	3.9 (1.0)
	APMR	14.24 (14.59)	0.57 (0.42)	1069 (2799)	3.8 (1.8)
4	NB	2.00 (3.22)	0.07 (0.12)		
	APM	0.67 (2.11)	0.02 (0.04)	7.3 (3.3)	6.4 (0.7)
	APMR	0.00 (0.00)	0.01 (0.02)	5.1 (2.2)	8.6 (1.7)
All	NB	12.81 (11.64)	0.64 (0.55)		
	APM	19.62 (10.94)	0.88 (0.66)		
	APMR	14.90 (11.01)	0.57 (0.36)		

Vote: The APMR error rate is 4.25%. Quinlan (1996) reports error rates for C4.5 bagged and boosted of 4.37% and 5.29%. Freund and Schapire (1996) report error rates for C4.5 bagged and boosted of 3.6% and 5.1%. Freund and Mason (1999) report error rates of 4.5% for boosted C5.0, 4.4% for boosted stumps, and 3.7% for AD-Tree. Hall and Holmes (2000) report error rates of 9.81% for NB (compared with our 9.72%) and 4.07% for NB with selection of best features. Gärtner and Flach (2001) report error rates of $9.81 \pm 3.92\%$ for NB, $4.14 \pm 3.13\%$ for WBC_{SVM}, and $3.77 \pm 2.77\%$ for a linear SVM.

Lymph: APMR was worse than NB with a modified Laplace correction. Even with the full correction APMR is comparable to NB only for the loss. Quinlan (1996) reports error rates for bagged and boosted C4.5 of 20.41% and 17.43%. Hall and Holmes (2000) report error rates of 16.76% for NB and 15.89% for NB with selection of best features. We infer that these results must have been obtained with a small Laplace correction. This problem has a fairly small number of examples and many features have values that are present in only one or two examples.

It is rather surprising that the test error rate of APMR is often much better than that of NB and also competitive with more sophisticated SVM or bagged/boosted models, even though loss, not the classification error rate, was the optimization target.

5. Relation to other modifications to NB

That NB is like logistic regression has been observed by many authors. Elkan (1997) made this observation, without introducing the α coefficients of APM, and showed that boosted NB is related to a neural network with one hidden layer. Hand and Adams (2000) also show the relationship and use it for discretization of numerical features. Ridgeway et al. (1998) showed that a linearized form of boosted NB yields a probability model similar to (8) without the α coefficients but with the values of q_0 and q_{ij} replaced by appropriately weighted values over several iterations of boosting. A "weighted NB" method has been used in text categorization with fixed exponentiating weights based on the frequency of observing a feature word in the text example (Rennie, 1999).

A modified NB model that is syntactically close to APM is the weighted NB, called WBC_{SVM}, of Gärtner and Flach (2001). In our notation, this model can be

formulated for the two-class case corresponding to model (6) as

$$P(C|X) \propto \{P(C)\}^{\alpha_0} \prod_j \{P(X_j|C)\}^{\alpha_j} . \tag{17}$$

Gärtner and Flach use a kernel derived from a "linear decision function" that is defined similarly to APM's q values in equation (10). However, the model is optimized for classification error rate as is usual for SVMs. Overfitting is avoided by reliance on SVM. Beside the difference in the optimization goal, there is one syntactically small but important difference between APM and WBC_{SVM}: APM does not fit an importance measure α_0. This reflects our intention of using the importance weights only to compensate for the absence of class-conditional correlations among features in the NB model.

It is true that one could take the original NB formulation (3), generalize it to

$$P(C|X) \propto P(C) \prod_j \{P(X_j|C)\}^{\alpha_j} , \tag{18}$$

and still arrive at the same logistic regression, equation (8). For (18) can be written as

$$\frac{P(C|X)}{1-P(C|X)} \propto \frac{P(C)}{1-P(C)} \prod_j \left(\frac{P(X_j|C)}{P(X_j|\bar{C})} \right)^{\alpha_j} , \tag{19}$$

where $\bar{C}$ is the event "not-C", and Bayes's theorem shows that

$$\frac{P(x_j|C)}{P(x_j|\bar{C})} = \frac{P(C|x_j)\{1-P(C)\}}{\{1-P(C|x_j)\}P(C)} . \tag{20}$$

However, when weighting the importance of different features it seems more natural to exponentiate the probability ratios in (6) rather than the odds ratios in (19). Another feature of our generalization of NB is that missing values and unseen values, which happen often during 10-fold modeling runs, are correctly handled. When in the test set a value of a feature, v, is encountered that is not present in the training examples, a sensible assignment is to set $P(C|v) = P(C)$. APM does this, whereas NB as usually implemented "ignores v": this amounts to assigning $P(v|C) = 1$, a strange choice that nonetheless is effective in practice.

APMR can also be viewed as a generalization of feature selection, which may be regarded as the case where the alpha values are all constrained to be either 0 or 1 and their squared norm m is the cardinality of the optimum feature subset. In

APMR, the alpha values are arbitrary (positive as well as negative) and their sum m can take positive non-integer values: this is similar to the relationship between ridge techniques and feature selection in regression.

The importance exponents of APM, α_j, can become negative. As indicated in Table 1, this is not unusual. This phenomenon can also occur in $\mathrm{WBC}_{\mathrm{SVM}}$, for which Gärtner and Flach (2001) suggest that "counter-productive attributes might be assigned negative weights". However, in a limited experiment with our APM we removed those features whose α_j values were negative; the resulting models were degraded by varying degrees. We interpret negative α_j values in a different light. In an APM model with a single feature, the α value for that feature should be $+1$ in order for the two sides of expression (6) to be in agreement. Therefore the occurrence of a negative α value for a feature suggests that the relationship between that feature's values and the class probability has become reversed in the presence of additional features. This situation is closely related to Simpson's paradox, a phenomenon much studied by statisticians (e.g., Blyth, 1972; Samuels, 1993). Indeed, when we trained APM on the example of Simpson's paradox given by Pearl (2000, sec. 6.1) we did obtain a negative α value for the relevant feature (drug/no-drug).

6. Concluding remarks

APM is derived from a new interpretation of Naïve Bayes. While there have been many similar modifications to NB, they either aim at reducing the error rate or are crucially different in the training procedure: one distinctive feature of APM is having q_0 fixed. The usual NB approach is known to be quite effective in predicting class membership; the APM/APMR approach provides the same or better accuracy and yields much better estimates of the class probability. We conclude that APMR is well suited for many applications where estimation of the class probability is of prime importance. We plan to perform more experiments, including problems that have numerical features.

Although APMR performs well, its basic model (6) still does not completely account for dependence between features. One approach to dealing with this is to add new features based on combinations of the original features. For example, one can include outputs of other models as input features to APM. This further extends the viewpoint that APM is a means of combining the outputs of an ensemble of models.

When there are more than two classes, we took the class probabilities from different APMR models for each class vs. the rest and computed error and loss mea-

sures for the all-class problem. Collins et al. (2002) suggest using Bregman distance to reformulate logistic regression in the context of boosting; it is possible that this idea can also be used to generate a multiclass APM by training a polytomous logistic regression. We plan further investigation of this approach.

Acknowledgements

We are grateful for useful discussions with Kishore Papineni and Edwin Pednault.

References

C. R. Blyth, "On Simpson's paradox and the sure thing principle," *Journal of the American Statistical Association*, **67**, 364–366, 1972.

L. Breiman, "Bagging Predictors," *Machine Learning*, **24**, 123–140, 1996.

M. Collins, R. E. Shapire and Y. Singer, "Logistic Regression, AdaBoost and Bregman Distances," *Machine Learning*, **48**, 253–285, 2002.

A. P. Dawid, "Probability Forecasting," *Encyclopedia of Statistical Sciences, vol. 6*, eds. S. Kotz, N. L. Johnson and C. B. Read, Wiley, New York, 210–218, 1982.

P. Domingos and M. Pazzani, "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss," *Machine Learning*, **29**, 103–130, 1997.

C. Elkan, "Boosting and Naïve Bayesian Learning," *Technical Report CS97-557*, University of California, San Diego, 1997.

Y. Freund and R. Shapire, "Experiments with a New Boosting Algorithm," *Proceedings of ICML-96*, 1996.

Y. Freund and L. Mason, "The Alternating Decision Tree Learning Algorithm," *Proceedings of ICML-99*, 1999.

T. Gärtner and P. A. Flach, "$\mathrm{WBC}_{\mathrm{SVM}}$: Weighted Bayesian Classification based on Support Vector Machines," *Proceedings of ICML-2001*, 154–161, 2001.

A. Garg and D. Roth, "Understanding Probabilistic Classifiers," *Proceedings of ECML-2001*, 2001.

I. J. Good, "Rational Decisions," *Journal of the Royal Statistical Society, Series B*, **14**, 107–114, 1952.

M. A. Hall and G. Holmes, "Benchmarking Attribute Selection Techniques for Data Mining," Technical report, Dept. of Comp. Sci., Univ. of Waikato, Hamilton, New Zealand, 2000.

D. J. Hand and N. M. Adams, "Defining Attributes for Scorecard Construction," *Journal of Applied Statistics*, **27**, 527–540, 2000.

D. J. Hand and K. Yu, "Idiot's Bayes—Not so Stupid After All," *International Statistical Review*, **69**, 385–398, 2001.

R. Kohavi, B. Becker and D. Sommerfield, "Improving Simple Bayes," Technical report, Data Mining and Visualization group, Silicon Graphics Inc., Mountain View, Calif., 1997.

P. Langley and S. Sage, "Induction of Selective Bayesian Classifiers," *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufman, Seattle, Wash., 399–406, 1994.

J. Pearl, *Causality*, Cambridge University Press, New York, 2000.

G. Piatetsky-Shapiro and S. Steingold, "Measuring Lift Quality in Database Marketing," *SIGKDD Explorations*, **2**, 76–80, 2000.

J. D. M. Rennie, "Improving Multi-Class Text Classification with Naive Bayes," M.S. thesis, Dept. of Elec. Eng. and Comp. Sci., Carnegie-Mellon University, Pittsburgh, 1999.

G. Ridgeway, D. Madigan, T. Richardson, and J. O'Kane, "Interpretable Boosted Naïve Bayes Classification," *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, New York, 101–104, 1998.

J. R. Quinlan, "Bagging Boosting and C4.5," *Proceedings of AAAI-96*, 1996.

M. L. Samuels, "Simpson's paradox and related phenomena," *Journal of the American Statistical Association*, **88**, 81–88, 1993.