Bayesian Model Averaging Naive Bayes (BMA-NB): Averaging over an Exponential Number of Feature Models in Linear Time

Ga Wu
Australian National University
Canberra, Australia
wuga214@gmail.com

Scott Sanner
NICTA & Australian National University
Canberra, Australia
ssanner@nicta.com.au

Rodrigo F.S.C. Oliveira
University of Pernambuco
Recife, Brazil
rfsantacruz@gmail.com

Abstract

Naive Bayes (NB) is well-known to be a simple but effective classifier, especially when combined with feature selection. Unfortunately, feature selection methods are often greedy and thus cannot guarantee an optimal feature set is selected. An alternative to feature selection is to use Bayesian model averaging (BMA), which computes a weighted average over multiple predictors; when the different predictor models correspond to different feature sets, BMA has the advantage over feature selection that its predictions tend to have lower variance on average in comparison to any single model. In this paper, we show for the first time that it is possible to exactly evaluate BMA over the exponentially-sized powerset of NB feature models in linear-time in the number of features; this yields an algorithm about as expensive to train as a single NB model with all features, but yet provably converges to the globally optimal feature subset in the asymptotic limit of data. We evaluate this novel BMA-NB classifier on a range of datasets showing that it never underperforms NB (as expected) and sometimes offers performance competitive (or superior) to classifiers such as SVMs and logistic regression while taking a fraction of the time to train.

Introduction

Naive Bayes (NB) is well-known to be a simple but effective classifier in practice owing to both theoretical justifications and extensive empirical studies (Langley, Iba, and Thompson 1992; Friedman 1997; Domingos and Pazzani 1997; Ng and Jordan 2001). At the same time, in domains such as text classification that tend to have many features, it is also noted that NB often performs best when combined with feature selection (Manning, Raghavan, and Schütze 2008; Fung, Morstatter, and Liu 2011).

Unfortunately, tractable feature selection methods are in general greedy (Guyon and Elisseeff 2003) and thus cannot guarantee that an optimal feature set is selected — even for the training data. An interesting alternative to feature selection is to use Bayesian model averaging (BMA) (Hoeting et al. 1999), which computes a weighted average over multiple predictors; when the different predictor models correspond to different feature sets, BMA has the advantage over feature selection that its predictions tend to have lower variance in comparison to any single model.

While BMA has had some notable successes for averaging over tree-based models such as the context-tree (CTW) weighting (Willems, Shtarkov, and Tjalkens 1995) algorithm for sequential prediction, or prunings of decision trees (Oliver and Dowe 1995), it has not been previously applied to averaging over all possible feature sets (i.e., the exponentially sized power set) for classifiers such as NB, which would correspond to averaging over all nodes of the feature subset lattice — a problem which defies the simpler recursions that can be used for BMA on tree structures.

In this paper, we show the somewhat surprising result that it is indeed possible to exactly evaluate BMA over the exponentially-sized powerset of NB feature models in *linear-time* in the number of features; this yields an algorithm no more expensive to train than a single NB model with all features, but yet provably converges to the globally optimal feature subset in the asymptotic limit of data.

This is the first-time we are aware of such an exact linear-time computation result for BMA over the feature subset lattice for *any* classifier or regressor. While previous work has proposed algorithms for BMA over the feature subset lattice in generalized linear models such as logistic regression and linear regression (Raftery 1996; Raftery, Madigan, and Hoeting 1997; Hoeting et al. 1999), all of these methods either rely on biased approximations or use MCMC methods to sample the model space with no *a priori* bound on the mixing time of such Markov Chains; in contrast our proposal for NB does not approximate and provides an *exact* answer in linear-time in the number of features.

We evaluate our novel BMA-NB classifier on a range of datasets showing that it never underperforms NB (as expected) and sometimes offers performance competitive (or superior) to classifiers such as SVMs and logistic regression while taking a fraction of the time to train.

Preliminaries

We begin by providing a graphical model perspective of Naive Bayes (NB) classifiers and Bayesian Model Averaging (BMA) that simplify the subsequent presentation and derivation of our BMA-NB algorithm.

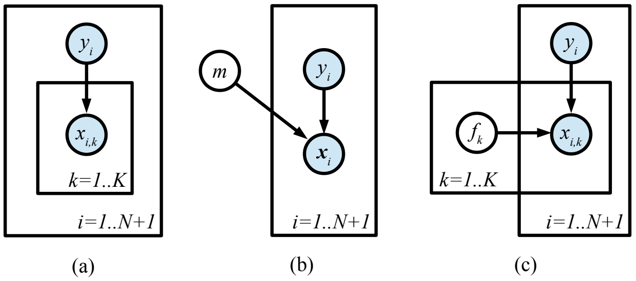

Figure 1: Bayesian graphical model representations of (a) Naive Bayes (NB), (b) Bayesian model averaging (BMA) for generative models such as NB, and (c) BMA-NB. Shaded nodes are observed.

Naive Bayes (NB) Classifiers

Assume we have N i.i.d. data points $D = \{(y_i, \mathbf{x}_i)\}$ $(1 \leq i \leq N)$ where $y_i \in \{1..C\}$ is the discrete class label for datum i and $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,K})$ is the corresponding vector of K features for datum i. In a probabilistic setting for classification, given a new feature vector $\mathbf{x}_{N+1}$ (abbreviated $\mathbf{x}$), our objective is to predict the highest probability class label y_{N+1} (abbreviated y) based on our observed data. That is, for each y, we want to compute

$$P(y|\mathbf{x}, D) \propto P(y, \mathbf{x}, D) = \prod_{i=1}^{N+1} P(y_i, \mathbf{x}_i), \qquad (1)$$

where the first proportionality is due to the fact that D and $\mathbf{x}$ are fixed (so the proportionality term is a constant) and the $\prod_{i=1}^{N+1}$ owes to our i.i.d. assumption for D (we likewise assume the N+1st data point is also drawn i.i.d.) and the fact that we can absorb $(y, \mathbf{x}) = (y_{N+1}, \mathbf{x}_{N+1})$ into the product by changing the range of i to include $N + 1$.

The *Naive* Bayes independence assumption posits that all features $\mathbf{x}_{i,k}$ $(1 \leq k \leq K)$ are *conditionally independent* given the class y_i. If we additionally assume that $P(y_i)$ and $P(\mathbf{x}_{i,k}|y_i)$ are set to their maximum likelihood (empirical) estimates given D, we arrive at the following standard result for NB in (3):

$$\arg\max_y P(y|\mathbf{x}, D) = \arg\max_{y_{N+1}} \prod_{i=1}^{N+1} P(y_i) \prod_{k=1}^{K} P(\mathbf{x}_{i,k}|y_i) \quad (2)$$

$$= \arg\max_{y_{N+1}} P(y_{N+1}) \prod_{k=1}^{K} P(\mathbf{x}_{N+1,k}|y_{N+1}) \qquad (3)$$

In (3), we dropped all factors in the $\prod_{i=1}^{N}$ from (2) since these correspond to data constants in D that do not affect the $\arg\max_y$.

Later when we derive BMA-NB, we'll see that it is not possible to drop the $\prod_{i=1}^{N}$ since latent feature selection variables will critically depend on D. To get visual intuition for this perspective, we pause to provide a graphical model representation of the joint probability for (2) in Figure 1(a) that we extend later. We indicate all nodes as observed (shaded) since D and $\mathbf{x}$ are given and to evaluate $\arg\max_y$, we instantiate y to its possible values in turn.

Bayesian Model Averaging (BMA)

If we had a class of models $m \in \{1..M\}$, each of which specified a different way to generate the feature probabilities $P(\mathbf{x}_i|y_i, m)$, then we can write down a prediction for a *generative*[1] classification model m as follows:

$$P(y, \mathbf{x}|m, D) = P(\mathbf{x}|m, y, D)P(y|D). \qquad (4)$$

While we could select a *single* model m according to some predefined criteria, an effective way to combine all models to produce a lower variance prediction than any single model is to use Bayesian model averaging (Hoeting et al. 1999) and compute a weighted average over *all* models:

$$P(y, \mathbf{x}|D) = \sum_m P(y, \mathbf{x}|m, D)P(m|D)$$

$$= \sum_m P(\mathbf{x}|m, y, D)P(y|D)P(m|D). \qquad (5)$$

Here we can view $P(m|D)$ as the weight of model m and observe that it provides a convex combination of model predictions, since by definition: $\sum_m P(m|D) = 1$. BMA has the convenient property that in the asymptotic limit of data D, as $|D| \to \infty$, $P(m|D) \to 1$ for the single best model m (assuming no models are identical) meaning that asymptotically, BMA can be viewed as optimal model selection.

To compute $P(m|D)$, we can apply Bayes rule to obtain

$$P(m|D) \propto P(D|m)P(m) = \prod_{i=1}^{N} P(y_i, \mathbf{x}_i|m)$$

$$= \prod_{i=1}^{N} P(y_i)P(\mathbf{x}_i|m, y_i). \quad (6)$$

Substituting (6) into (5) we arrive at the final simple form, where we again absorb $P(\mathbf{x}|m, y_{N+1}, D)P(y|D)$ into the renamed $N + 1$st factor of $\prod_{i=1}^{N+1}$:

$$P(y, \mathbf{x}|m, D) \propto \sum_m \prod_{i=1}^{N+1} P(y_i)P(\mathbf{x}_i|m, y_i). \qquad (7)$$

We pause to make two simple observations on BMA before proceeding to our main result combining BMA and NB. First, as for NB, we can represent the underlying joint distribution for (7) as the graphical model in Figure 1(b) where in this case m is a latent (unobserved) variable that we marginalize over. Second, unlike the previous result for NB, we observe that with BMA, we cannot simply absorb the data likelihood into the proportionality constant since it depends on the unobserved m and is hence non-constant.

Main Result: BMA-NB Derivation

Consider a naive combination of the previous sections on BMA and NB if we consider each model m to correspond to a different subset of features. Given K features, there are 2^K possible feature subsets in the powerset of features yielding an exponential number of models m to sum over in (7).

[1] Models like NB *generate* features $\mathbf{x}_i$ given y_i and factorize the joint $P(y_i, \mathbf{x}_i)$ as $P(\mathbf{x}_i|y_i)P(y_i)$ (Ng and Jordan 2001).

However, if we assume feature selections are independent of each other (a strong assumption justified in the next section), we can explicitly factorize our representation of m and exploit this for computational efficiency. Namely, we can use $f_k \in \{0,1\}$ $(1 \leq k \leq K)$ to represent whether feature $x_{i,k}$ for each datum i is used (1) or not (0).

Considering our model class m to be $\mathbf{f} = (f_1, \ldots, f_K)$, we can now combine the naive Bayes factorization of (2) and BMA from (7) — and their respective graphical models in Figures 1(a,b) — into the BMA-NB model shown in Figure 1(c) represented by the following joint probability marginalized over the latent model class $\mathbf{f}$ and simplified by exploiting associative and reverse distributive laws:

$$\begin{aligned} P(y|\mathbf{x}, D) &= \sum_{\mathbf{f}} \left[\prod_{k=1}^{K} P(f_k) \right] \prod_{i=1}^{N+1} P(y_i) \prod_{k=1}^{K} P(\mathbf{x}_{i,k}|f_k, y_i) \\ &= \sum_{f_1} \sum_{f_2} \cdots \sum_{f_k} \left[\prod_{i=1}^{N+1} P(y_i) \right] \prod_{k=1}^{K} P(f_k) \prod_{i=1}^{N+1} P(\mathbf{x}_{i,k}|f_k, y_i) \\ &= \left[\prod_{i=1}^{N+1} P(y_i) \right] \prod_{k=1}^{K} \sum_{f_k} P(f_k) \prod_{i=1}^{N+1} P(\mathbf{x}_{i,k}|f_k, y_i) \end{aligned} \tag{8}$$

Before we proceed further, we need to define specifically what model prior $P(f_k)$ and feature distribution $P(\mathbf{x}_{i,k}|f_k, y_i)$ we wish to use for BMA-NB. For $P(f_k)$, we use the simple unnormalized prior

$$P(f_k) \propto \begin{cases} \frac{1}{\beta} & \text{if } f_k = 1 \\ 1 & \text{if } f_k = 0 \end{cases} \tag{9}$$

for some positive constant β.

In a generative model such as NB, what it means for a feature to be used is quite simple to define:

$$P(x_{i,k}|f_k, y_i) = \begin{cases} P(x_{i,k}|y_i) & \text{if } f_k = 1 \\ P(x_{i,k}) & \text{if } f_k = 0 \end{cases} \tag{10}$$

Then whenever $f_k = 0$, $P(x_{i,k}|f_k, y_i)$ effectively becomes a constant that can be ignored in the $\arg\max_y$ of (2) of NB.

Putting the pieces together and explicitly summing over $f_k \in \{0,1\}$, we can continue our derivation from (8):

$$P(y|\mathbf{x}, D) \propto \tag{11}$$

$$\left[\prod_{i=1}^{N+1} P(y_i) \right] \prod_{k=1}^{K} \left[\prod_{i=1}^{N+1} P(\mathbf{x}_{i,k}) + \frac{1}{\beta} \prod_{i=1}^{N+1} P(\mathbf{x}_{i,k}|y_i) \right]$$

And this brings us to a final form for BMA-NB that allows $P(y|\mathbf{x}, D)$ to be efficiently computed. Note that here we have averaged over an exponential number of models $\mathbf{f}$ in *linear time in the number of features K*. This result critically exploits the fact that NB models do not need to be retrained for different feature subsets due to the feature conditional independence assumption of NB. As standard for BMA, we note that in the limit of data D, $P(f_k|D) \to 1$ for the optimal feature selection choice $f_k = 1$ or $f_k = 0$ thus leading to a linear-time aymptotically optimal feature selection algorithm for NB. Before we proceed to an empirical evaluation, we pause to discuss a number of design choices and implementation details that arise in BMA-NB.

Feature Model Justification

One may question exactly why implementing $P(x_{i,k}|f_k, y_i)$ as in (10) corresponds to a feature selection approach in Naive Bayes. As already outlined, then $f_k = 0$, this is equivalent to making $P(x_{i,k}|f_k, y_i)$ a constant w.r.t. y_i and hence this feature can be ignored when determining the most probable class y_i. However there is slightly deeper second justification for this model based on the following feature and class independence analysis:

$$\frac{\prod_{i=1}^{N} P(x_{i,k}, y_i)}{\prod_{i=1}^{N} P(x_{i,k}) P(y_i)} = \frac{\prod_{i=1}^{N} P(x_{i,k}|y_i)}{\prod_{i=1}^{N} P(x_{i,k})} \geq 1 \tag{12}$$

On the LHS, we show the ratio of the joint distribution of feature $x_{i,k}$ and class label y_i to the product of their marginal distributions — a ratio which would equal 1 if the two variables were independent, but which would be greater than 1 if the variables were dependent — the joint would carry more information than the product of the marginals. Simply by dividing the LHS through by $\prod_{i=1}^{N} (P(y_i)/P(y_i)) = 1$, we arrive at the equivalent middle term which shows the two terms used in (10).

From this we can conclude that when feature $x_{i,k}$ and class label y_i are independent (i.e., $x_{i,k}$ carries no information as a predictor of y_i) then $P(x_{i,k}|f_k, y_i) = P(x_{i,k})$ regardless of f_k. Only when $P(x_{i,k}|f_k, y_i)$ is predictive would we expect $P(f_k = 1|x_{i,k}, y_i) > P(f_k = 0|x_{i,k}, y_i)$ and hence BMA-NB would tend to include feature k (i.e., models including feature k would have a higher likelihood and hence a higher model weight), which is precisely the BMA behavior we desire to weight models with more predictive features more highly than those with less predictive features.

BMA vs. Feature Selection

The analysis in the last subsection suggests that features which have a higher mutual information (MI) with the class label will lead to higher weights for models that include those features. This is encouraging since it is consistent with the observation that MI is often a good criterion for feature selection (Manning, Raghavan, and Schütze 2008).

However one may ask: why then bother with BMA-NB if one could simply use MI as a feature selection criterion instead? Aside from the tendency of BMA to reduce predictor variance on average compared to the choice of any single model along with the asymptotic convergence of BMA to the optimal feature subset, there is an additional important reason why BMA might be preferred. Feature selection requires choosing a threshold on the given criterion (e.g., MI), which is another classifier hyperparameter that must be properly tuned on the training data via cross-validation for optimal performance. Since this expensive threshold tuning substantially slows down the training time for naive Bayes with feature selection, it (somewhat surprisingly) turns out to be much faster to simply average over all exponential models in linear time using BMA-NB.

Hyperparameter Tuning

Our key hyperparameter to tune in BMA-NB is the constant β used in the feature inclusion prior (9) for f_k. The purpose of this constant is simply to weight the *a priori* preference for models with the feature and without it. Clearly a $\beta = 1$ places no prior preference on either model, while $\beta > 1$ prefers the simpler model in the absence of any data and corresponds to a type of Occam's razor assumption — prefer the simplest model unless the evidence (Bayesian posterior for f_k) suggests otherwise.

In experimentation we found that the optimal value of β is highly sensitive to the amount of data N since the likelihood terms $\prod_{i=1}^{N+1}$ of (11) scale proportional to N. A stable way to tune the hyperparameter β across a range of data is to instead tune Γ defined as

$$\beta = \Gamma^{N+1} \tag{13}$$

where N is size of training data and we evaluated $\Gamma \in \{0.5, 0.6, 0.7,\ 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5\}$ since, empirically, wider ranges did not lead to better performance.

Conditional Probability Estimation

As usual for NB and other generative classifiers, it is critical to smooth probability estimates for discrete variables to avoid 0 probabilities (or otherwise low probabilities) for low frequency features. As often done for NB, we use Dirichlet prior smoothing for discrete-valued features taking M possible feature values:

$$P(x_k|y) = \frac{\#D\{x_k \bigwedge y\} + \alpha}{\#D\{y\} + \alpha M} \tag{14}$$

When the Dirichlet prior $\alpha = 1$, then this approach is called Laplace smoothing (Mitchell 2010). Note that while α is also a potential hyperparameter in this work, we keep $\alpha = 1$ as performance for BMA-NB and NB did not differ significantly for other values of α.

For continuous features x_k, we instead model the conditional distribution $P(x_k|y) = \mathcal{N}(\mu_{k,y}, \sigma^2_{k,y})$ as a Gaussian where for each continuous feature k and class label y we estimate the empirical mean $\mu_{k,y}$ and variance $\sigma^2_{k,y}$ for the corresponding subset of feature data $\{x_{i,k}|y_i = y\}_i$ for class label y.

Experiments

We evaluated NB, BMA-NB (shortened to just BMA here), SVM, and Logistic Regression (LR) on a variety of real-world datasets from the UCI machine learning repository (Asuncion and Newman 2007). Characteristics of the datasets we evaluated are outlined in Table 1.

We compare to the NB classifier with the full feature set to determine whether BMA always provides better performance than NB as we might expect (since it should place high weight on the full feature set used by NB if that feature set offers best performance). We choose both SVM and LR as discriminative linear classifiers known for their typically stronger performance but slower training time compared to NB. While we would not expect generatively trained BMA to beat disriminatively trained SVM and LR, we are interested to see (a) where BMA falls in the range between SVM and LR, (b) whether it is significantly faster to train than SVM and LR, and (c) if there are cases where BMA is competitive with SVM and LR.

Table 1: UCI datasets for experimentation.

Problem	K	$\|D\|$	C	$K/\|D\|$	Missing?
anneal	38	798	6	4.76%	Yes
autos	26	205	7	12.68%	Yes
vote	16	435	2	3.68%	Yes
sick	30	3772	2	0.80%	Yes
crx	16	690	2	2.32%	No
mushroom	22	8124	2	0.27%	Yes
heart-statlog	13	270	2	4.81%	No
segment	19	2310	7	0.82%	No
labor	16	57	2	28.07%	No
vowel	14	990	11	1.41%	No
audiology	69	226	23	30.53%	Yes
iris	5	150	3	3.33%	No
zoo	18	101	7	17.82%	No
lymph	19	148	4	12.84%	No
soybean	35	683	19	5.12%	Yes
balance-scale	4	625	3	0.64%	No
glass	10	214	7	4.67%	No
hepatitis	19	155	2	12.25%	No
haberman	4	306	2	1.31%	No

Comparison Methodology

In our experimentation, we compare classifiers using a nested random resampling cross-validation (CV) model to ensure all classifier hyperparameters were properly tuned via CV on the training data prior to evaluation on the test data. All algorithms — NB, BMA, SVM, and Logistic Regression (LR) — require hyperparameter tuning during nested cross-valiation as outlined below along with other training details for each algorithm.

The implementation of SVM and LR are from Liblinear API package (Fan et al. 2008). Both toolkits optimize the regularized error function

$$\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi\left(\mathbf{w};\mathbf{x_i};y_i\right) \tag{15}$$

where SVM uses hinge loss for ξ and LR uses log loss for ξ. Both use a regularization parameter denoted λ, typically tuned in reciprocal form $C = \frac{1}{\lambda}$. C is regarded as penalty factor or cost factor.[2] With large C, the classifiers are more sensitive to the empirical loss ξ and tend to overfit for high C, while small C simply prevents learning altogether.

Another parameter of Liblinear package is the constant bias term B. The following equation corresponds to separating hyperplane for general linear classifiers like SVM and LR

$$y\left(\mathbf{x},\mathbf{w}\right) = \sum_{k=1}^{K} w_k\phi_k\left(\mathbf{x}\right) + B\left(w_0\right) \tag{16}$$

where ϕ_k is base function, $\mathbf{w}$ is parameter vector and B is the bias.[3]

[2] We use C to be consistent with the Liblinear software package we use, but this is not to be confused with the number of classes C; the intended usage should be clear from context.

[3] Note it is not a good idea to regularize B; it is often better to tune it via cross-validation.

Table 2: (left) Error ±95% confidence interval for each classifier with the best performance average performance shown in bold — lower is better; (right) training time (ms). Datasets that show a 0.00 error rate appear to be linearly separable.

Problems	BMA	NB	LR	SVM
anneal	7.87±2.00	7.87±2.00	**6.74±1.86**	17.98±2.86
autos	30.00±7.13	40.00±7.62	**25.00±6.74**	55.00±7.74
vote	**4.65±2.25**	11.63±3.42	**4.65±2.25**	**4.65±2.25**
sick	6.37±0.89	7.69±0.97	**2.65±0.58**	4.51±0.75
crx	23.19±3.58	23.19±3.58	**18.84±3.32**	21.74±3.50
mushroom	**1.35±0.29**	3.82±0.47	2.96±0.42	4.06±0.49
heart-statlog	**7.41±3.55**	**7.41±3.55**	18.52±5.27	11.11±4.26
segment	23.81±1.97	23.81±1.97	**10.82±1.44**	14.72±1.64
labor	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**
vowel	**33.33±3.34**	42.42±3.50	44.44±3.52	47.47±3.54
audiology	22.73±6.21	27.27±6.60	**9.09±4.26**	13.64±5.09
iris	**6.67±4.54**	13.33±6.18	**6.67±4.54**	**6.67±4.54**
zoo	**0.00±f0.00**	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**
lymph	**7.14±4.72**	**7.14±4.72**	14.29±6.41	14.29±6.41
soybean	11.76±2.75	11.76±2.75	**10.29±2.59**	11.76±2.75
balance-scale	**9.68±2.63**	**9.68±2.63**	16.13±3.28	12.90±2.99
glass	52.38±7.61	52.38±7.61	**33.33±7.18**	47.62±7.61
hepatitis	**26.67±7.91**	**26.67±7.91**	33.00±8.77	46.67±8.93
haberman	**33.33±6.00**	**33.33±6.00**	36.67±6.14	**33.33±6.00**

Problems	BMA	NB	LR	SVM
anneal	526	147	618	2070
autos	81	40	183	1278
vote	10	6	22	33
sick	202	79	434	2475
crx	36	9	61	540
mushroom	248	98	1072	5785
heart-statlog	18	5	22	213
segment	435	120	2606	6637
labor	3	3	6	27
vowel	105	35	557	5266
audiology	81	14	172	260
iris	4	2	10	102
zoo	6	3	31	222
lymph	7	3	22	175
soybean	99	18	559	818
balance-scale	15	5	40	542
glass	14	5	50	621
hepatitis	7	3	16	120
haberman	4	2	11	157

Since parameter tuning of SVM and LR is very time consuming, we limited our hyperparameter search to a combination of 36 joint values of B and C as shown below:

$$C \in \{0.01, 0.1, 1, 10, 20, 100\} \tag{17}$$

$$B \in \{0.01, 0.1, 1, 10, 20, 100\} \tag{18}$$

We implement the Naive Bayes classifier ourselves to guarantee that NB is exactly the special case of BMA where weight 1 is placed on the model with $\mathbf{f} = \mathbb{1}$ and weight 0 on other models. As described previously, while β is tuned for BMA, $\alpha = 1$ yielded best results for both NB and BMA and setting $\alpha = 1$ for both classifiers allows us to evaluate the performance of BMA's model averaging vs. the same full model that NB uses.

Experimental Results

Now we proceed to compare each of the classifiers on the UCI datasets as shown in Table 2.

Here we observe a few general trends:

- BMA *never* underperforms NB as expected. LR seems to offer the best performance, edging out SVM since we generally found that LR was more robust during hyperparameter tuning in comparison to SVM.
- In the range of performance between NB and LR/SVM, BMA twice outperforms LR (when NB performance is significantly worse) and outperforms SVM even more often! When BMA does worse than LR and SVM, it often comes close to the best performance in many cases.
- BMA is up to 4 times slower than NB, but it's significant performance improvement over BN seems worth this tradeoff.
- BMA is always faster than LR and SVM — up to 5 times faster than LR in some cases and up to 50 times faster than the SVM for *glass*!

To continue with our experimentation, we now perform some specialized experiments for text classification with the UCI *newsgroups* dataset to experiment with BMA performance relative to other classifiers as the amount of data and number of features changes. We choose text classification for this analysis since both data and features are abundant in such tasks and they are also a special case where NB is known to perform relatively well, especially with feature selection (Manning, Raghavan, and Schütze 2008; Fung, Morstatter, and Liu 2011).

We focus on a binary classification problem which is abstracted from the newsgroup dataset. To construct D for our first task, we chose 1000 data entries selected from two categories of *newsgroups* — it is an exactly balanced dataset. The $K = 1000$ features are selected from 20,000 words with high frequency of occurrence after removing English stopwords. Results for this text classification analysis are shown in Figure 2. We average this plot over a large number of runs to obtain relatively smoothed average performance results, but omit dense confidence intervals to keep the plots readable.

In Figure 2(left), we observe that after around 150 features selected, BMA, SVM and LR show significantly better performance than NB. Furthermore, the accuracy of BMA, SVM and Logistic Regression also asymptote earlier with BMA showing competitive performance with SVM and LR for this text classification task with a large number of (correlated) word features.

To show whether training dataset size can significantly impact the accuracy of news classification task, we refer to Figure 2(right), which is similar to the setup for Fig-

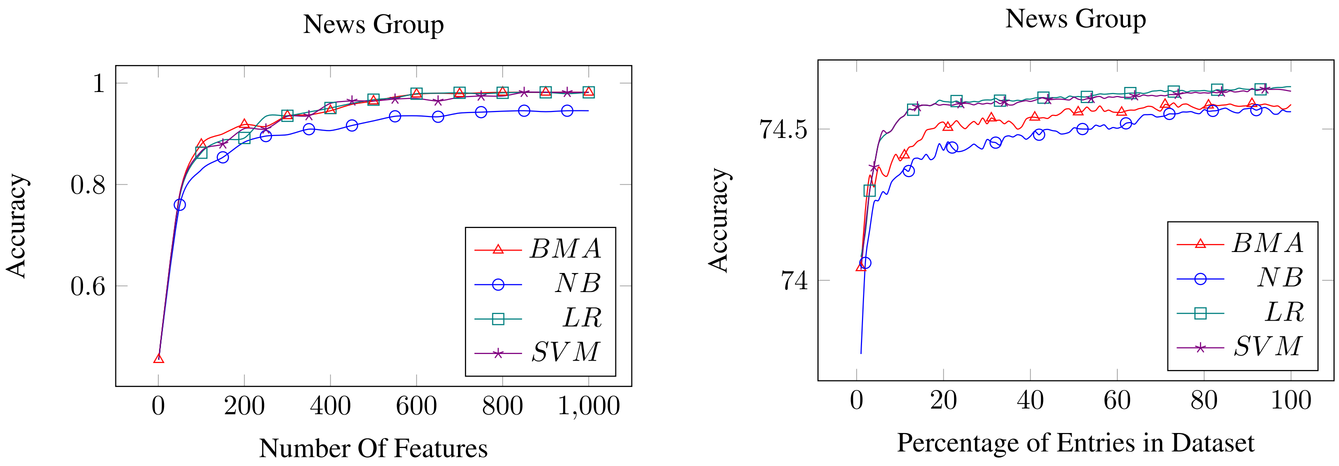

Figure 2: (left) Accuracy vs. number of features, (right) accuracy vs. number of training data.

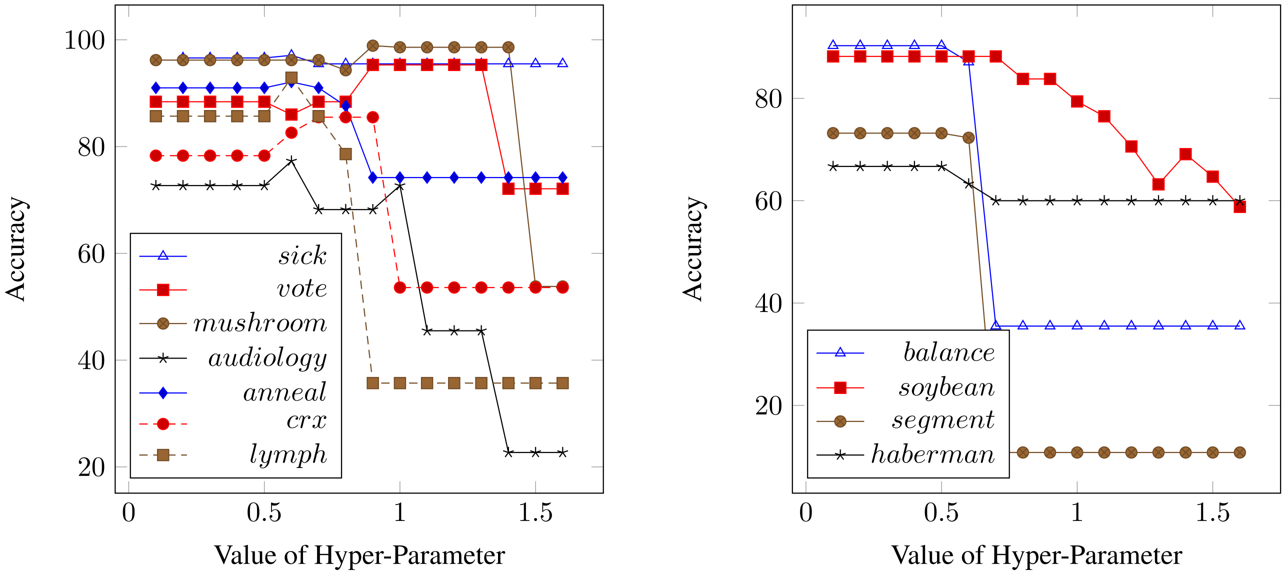

Figure 3: Effect of tuning hyperparameter Γ for BMA on the UCI datasets.

ure 2(left) except with $|D| = 4000$ and $K = 500$. These results show the discriminative classifiers SVM and LR performing better than BMA, but overall asymptoting more quickly than NB indicating an ability of BMA to place heavier weight on the most useful features when data is limited and many features are noisy.

Finally we show results for hyperparameter tuning of BMA across the UCI datasets in Figure 3. We break the results into two sets — Figure 3(left) where there seems to be an optimal choice for β and Figure 3(right) where the optimal choice seems to be a small $\beta < 1$ which places a strong prior on using all features.

If we examine the dataset characteristics in Table 1, we see all of the problems in Figure 3(right) have small numbers of features and large amounts of data; hence it seems that in these settings all features are useful and thus the hyperparameter tuning is largest for a prior which selects all features. This is further corroborated by the performance results in Table 2 where we see that BMA performance matches NB performance (which uses all features) for the problems in Figure 3(right).

Conversely, we might infer that in Figure 3(left), not all features in these problems are useful, thus a peak value for prior $\beta > 1$ is best which indicates a preference to exclude features unless the data likelihood overrides this prior. For many of these problems in Figure 3(left) we similarly note that the performance of BMA was better than NB indicating that BMA may have placed low weight on models with certain (noisy) features, whereas NB was constrained to use all features and hence performed worse.

Conclusion

In this work we presented a novel derivation of BMA for NB that averaged over the exponential powerset of feature models in linear time in the number of features. This tends to yield a lower variance NB predictor (a result of averaging multiple estimators) that converges to the selection of the best feature subset model in the asymptotic limit of data. This is the first such result we are aware of for exact BMA over the powerset of features that does not require approximations or sampling.

Empirically, we observed strong performance from BMA-NB in comparison to NB, SVMs, and logistic regression (LR). These results suggest that (1) in return for a fairly constant 4 times slow-down (no matter what dataset size or feature set size), BMA-NB never underperforms NB; (2) sometimes it performs as well as or better than LR or SVM; (3) it trains faster than LR and SVM and in the best case is up to 50 times faster than SVM training. These results suggest that BMA-NB may be a superior general replacement for NB, and in some cases BMA-NB may even be a reasonable replacement for LR and SVM, especially when efficient online training is required.

Future work should examine whether the derivational approach used in this paper can extend linear-time BMA over all feature subsets to other problems. It may be difficult to do this for the class of discriminative probabilistic classifiers such as logistic regression since each feature subset leads to a different set of optimal parameters (unlike NB where the learned parameters for each feature are independent thus preventing the need to retrain NB for each model). However, BMA has also been applied beyond classification to other problems such as unsupervised learning (Dean and Raftery 2010; Raftery and Dean 2006) and sequential models (Raftery, Karny, and Ettler 2010), where techniques developed in this work may apply. For example, hidden Markov models (HMMs) are an important generative sequential model with observation independence (as in NB) that may admit extensions like that developed in BMA-NB; it would be interesting to explore how such HMM extensions compare to state-of-the-art linear chain conditional random fields (CRFs) (Lafferty, McCallum, and Pereira 2001).

Acknowledgements

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program. Rodrigo F.S.C. Oliveira was sponsored by CNPq - Brazil.

References

Asuncion, A., and Newman, D. J. 2007. UCI machine learning repository.

Dean, N., and Raftery, A. 2010. Latent class analysis variable selection. *Annals of the Institute of Statistical Mathematics* 62:11–35.

Domingos, P., and Pazzani, M. 1997. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning* 29:103130.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear - a library for large linear classification. The Weka classifier works with version 1.33 of LIBLINEAR.

Friedman, J. H. 1997. On bias, variance, 0/1 - loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery* 1:5577.

Fung, P. C. G.; Morstatter, F.; and Liu, H. 2011. Feature selection strategy in text classification. In *Advances in Knowledge Discovery and Data Mining*. Springer. 26–37.

Guyon, I., and Elisseeff, A. 2003. An introduction to variable and feature selection. *J. Mach. Learn. Res.* 3:1157–1182.

Hoeting, J. A.; Madigan, D.; Raftery, A. E.; and Volinsky, C. T. 1999. Bayesian model averaging: A tutorial. *Statistical Science* 14(4):382–417.

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML-01)*, ICML '01, 282–289.

Langley, P.; Iba, W.; and Thompson, K. 1992. An analysis of bayesian classifiers. In *AAAI-92*.

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.

Mitchell, T. M. 2010. Generative and discriminative classifier: Naive bayes and logistic regression.

Ng, A. Y., and Jordan, M. I. 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *NIPS*, 841–848.

Oliver, J. J., and Dowe, D. L. 1995. On pruning and averaging decision trees. In *ICML-95*, 430–437. 340 Pine Street, 6th Floor San Francisco, California 94104 U.S.A.: Morgan Kaufmann.

Raftery, A., and Dean, N. 2006. Variable selection for model-based clustering. *Journal of the American Statistical Assocation* 101:168–178.

Raftery, A.; Karny, M.; and Ettler, P. 2010. Online prediction under model uncertainty via dynamic model averaging: Application to a cold rolling mill. *Technometrics* 52:52–66.

Raftery, A. E.; Madigan, D.; and Hoeting, J. A. 1997. Bayesian model averaging for linear regression models. *Journal of the American Statistical Association* 92(437):179–191.

Raftery, A. E. 1996. Approximate Bayes Factors and Accounting for Model Uncertainty in Generalized Linear Models. *Biometrika* 83(2):251–266.

Willems, F. M. J.; Shtarkov, Y. M.; and Tjalkens, T. J. 1995. The context-tree weighting method: basic properties. *IEEE Transactions on Information Theory* 41(3):653–664.