RC 21994 (98372) 14 March 2001 Computer Science

# IBM Research Report

## An Analysis of Naive Bayes Classifier on Low-Entropy Distributions

**Irina Rish, Joseph Hellerstein, Jayram Thathachar**
IBM Research Division
Thomas J. Watson Research Center
P. O. Box 704
Yorktown Heights, NY 10598

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

# An analysis of naive Bayes classifier on low-entropy distributions

**I. Rish, J.L. Hellerstein, J.S. Thathachar**

T.J. Watson Research Center
30 Saw Mill River Road
Hawthorne, NY 10532

## Abstract

Naive Bayes classifier assumes that features are independent given class. Despite this unrealistic assumption, it is often highly competitive with more sophisticated learning techniques. Indeed, a classifier's optimality depends on its prediction of the most-likely class, rather than on the correct probability estimates. Although previous studies characterized certain cases of naive Bayes optimality, a better understanding of data characteristics that affect the performance of this classifier is still required. In this paper, we focus on problems including deterministic and almost-deterministic dependencies. Such dependencies are often present in practical problems, e.g., in error-correcting coding and in computer system performance management, to name a few. We derive error bounds for approximation of a joint distribution by the product of marginals for almost-deterministic (low-entropy) distributions, and demonstrate empirically how decreasing entropy of class-conditional feature distributions affects the error of naive Bayes classifier. We also prove that naive Bayes is optimal in certain cases of functionally dependent features. Then we relax those dependencies by adding noise, demonstrating empirically that naive Bayes reaches optimal performance in cases of completely independent and functionally dependent features, while it behaves worst between the two extremes.

## 1 Introduction

Classification is one of the central problems in machine learning, statistics, pattern recognition and data-mining. A variety of problems from different domains, such as image recognition, speech understanding, natural language processing, error-correcting coding, and medical diagnosis, can be viewed as classification problems, and a large variety of classification approaches has been proposed and tested in the past years. However, a better theoretical understanding of key factors that influence the classification accuracy of different learning algorithms, i.e. a better understanding of "what works well where", is still required.

One of the well-established approaches is Bayesian classification, which becomes increasingly popular in the recent years in AI community, especially due to recent developments in learning with Bayesian belief networks [Heckerman, 1995; Friedman *et al.*, 1997]. The simplest version of a Bayesian network classifier is a widely used *naive Bayes* classifier that assumes independence of features given class [Duda and Hart, 1973]. Although this assumption is often violated, naive Bayes is surprisingly successful in practice [Langley *et al.*, 1992; Domingos and Pazzani, 1997; Mitchell, 1997; Hellerstein *et al.*, 2000]. For example, naive Bayes is a state-of-the-art classifier in text classification [Mitchell, 1997]. Systems performance management [Hellerstein *et al.*, 2000] is another example, among many other applications. An explanation of those results is that naive Bayes can be optimal in terms of zero-one loss (classification error) even if its class probability estimates are wrong, as long as both true and estimated distributions agree on most-probable class [Domingos and Pazzani, 1997]. Domingos and Pazzani [Domingos and Pazzani, 1997] analyzed some cases of naive Bayes optimality (such as disjunctive and conjunctive concepts), and provided empirical studies on a set of UCI benchmark problems, many of which have high degree of feature dependencies. However, further analysis of data distributions that violate independence assumption but yield a good performance of naive Bayes, and characterizing naive Bayes accuracy as a function of distribution's parameters is still required.

In this paper, we focus on problems that include deterministic or close-to-deterministic dependencies. Note that such dependencies are often present in practical problems, such as error-correcting coding and computer system performance management, to name a few. We prove naive Bayes optimality in certain cases of functionally dependent features. Then we relax those dependencies by adding noise, demonstrating empirically that naive Bayes reaches optimal performance in two extreme cases of completely independent and functionally dependent features, while its worst performance appears in the middle. We also show that a joint distribution and its approximation by the product of marginals converge with decreasing entropy of the distribution (i.e., $P(a_1, ..., a_n) \approx \prod_{i=1}^{n} P(a_i)$ for low-entropy distributions), and demonstrate how decreasing entropy of class-conditional feature distributions affects the error of naive Bayes classifier.

We should emphasize that our error analysis only focuses

on the *bias* of naive Bayes classifier, not on its *variance*, i.e. we assume an infinite amount of data, or perfect knowledge of data distribution to be available and compare naive Bayes versus Bayes-optimal classifier.

Although it may seem counterintuitive, the boundary error and, subsequently, the naive Bayes error are not really correlated with class-conditional mutual information between the features. This phenomenon was observed before by several researchers (e.g., see empirical evaluations on UCI benchmarks in [Domingos and Pazzani, 1997]). Our simulations also demonstrate this fact on different problem generators.

Our ultimate objective is to understand how data characteristics (e.g., low entropy, almost-deterministic dependencies) affect the accuracy of approximate classification algorithms making simplifying independence assumptions, such as naive Bayes classifier. In particular, naive Bayes is optimal when features are independent given class; on the other hand, it can be optimal in case of functionally dependent features, so the class-conditional mutual information is not a good predictor of naive Bayes performance. Therefore, we need other parameters characterizing probability distributions that are insensitive to the independence assumption w.r.t. to classification task.

Our focus on almost-deterministic dependencies is also motivated by significant amount of empirical evidence suggesting that problems involving such dependencies often yield a good performance of approximate probabilistic inference algorithms based on independence assumptions. One of the most prominent examples is successful application of Pearl's belief propagation algorithm to probabilistic decoding [Frey and MacKay, 1998]: although belief propagation performs local inference ignoring long-range dependencies, its iterative variant applied to certain coding networks results into lower error rates than the state-of-the-art decoding algorithms. Another example of local inference algorithm that ignores some dependencies is the mini-bucket approach [Dechter, 1997]. When applied to finding most probable state, it demonstrates lower error on problems that involve close-to-deterministic dependencies [Rish, 1999]).

## 2 Definitions and Background

Let $\mathbf{X} = (X_1, ..., X_n)$ be a vector of observed random variables, called *features*, where each feature takes values from its *domain* $D_i$. A feature vector is also called an *example*, or a *state* (of nature), and the set of all possible examples, or the *state space*, is denoted $\Omega = D_1 \times ... \times D_n$. Let $C$ be an unobserved random variable denoting *class* of an example, where $C$ can take one of $m$ values $c \in \{0, ..., m-1\}$. Capital letters, such as $X_i$, will denote variables, while lowercase letters, such as $x_i$, will denote their values; boldface letters will denote vectors. Also, we will sometimes use shorter notation $P(\mathbf{x})$ and $P(x_1, ..., x_n)$ instead of $P(\mathbf{X} = \mathbf{x})$ and $P(X_1 = x_1, ..., X_n = x_n)$, respectively.

A classifier is a function $g : \Omega \rightarrow \{0, ..., m-1\}$, where that assigns class to a given example. A common approach is to associate each class $i$ with a discriminant function $f_i(\mathbf{x})$, $i = 0, ..., m-1$, and let the classifier select the class with maximum discriminant function on a given example[1]:

$$g(\mathbf{x}) = \arg \max_{i \in \{0,...,m-1\}} f_i(\mathbf{x}). \quad (1)$$

The *Bayes* classifier $g^*(\mathbf{x})$ (that we also call *Bayes-optimal* classifier and sometimes denote $BO(\mathbf{x})$), uses as discriminant functions the class posterior probabilities given a feature vector, i.e. $f_i^*(\mathbf{x}) = P(C = i|\mathbf{X} = \mathbf{x})$,. Applying the Bayes rule gives $P(C = i|\mathbf{X} = \mathbf{x}) = \frac{P(\mathbf{X}=\mathbf{x}|C=i)P(C=i)}{P(\mathbf{X}=\mathbf{x})}$, where $P(\mathbf{X} = \mathbf{x})$ is same for all classes, and therefore can be ignored, which yields Bayes discriminant functions

$$f_i^*(\mathbf{x}) = P(\mathbf{X} = \mathbf{x}|C = i)P(C = i). \quad (2)$$

Thus, the Bayes classifier

$$g^*(\mathbf{x}) = \arg \max_i P(\mathbf{X} = \mathbf{x}|C = i)P(C = i) \quad (3)$$

finds the *maximum a posteriori probability* (MAP) hypothesis given example $\mathbf{x}$. However, direct estimation of $P(\mathbf{X} = \mathbf{x}|C = i)$ from a given set of training examples is hard when feature space is high-dimensional. Therefore, approximations are commonly used, such as using the simplifying assumption that features are independent given the class. This yields *n*aive Bayes classifier $NB(\mathbf{x})$ defined by discriminant functions

$$f_i^{NB}(\mathbf{x}) = \Pi_{j=1}^n P(X_j = x_j|C = i)P(C = i). \quad (4)$$

Subsequently, we will use the following shorter notation for class-conditional distributions: $P_i(A) \doteq P(A|C = i)$.

$R(g)$ denotes the error, or $risk$, of a classifier $g$, i.e.

$$R(g) = P(g(\mathbf{X}) \neq C) = \sum_{\mathbf{x} \in \Omega} P(g(\mathbf{x}) \neq C)P(\mathbf{X} = \mathbf{x}) = \\ = E_{\mathbf{x}}\{P(g(\mathbf{x}) \neq C)\},$$

where $E_{\mathbf{x}}$ is the expectation over $\mathbf{x}$. $R^* = R(g^*)$ denotes the Bayes error (Bayes risk). We will call a classifier *optimal* on a problem if its error probability coincides with Bayes risk on that problem.

As a measure of dependence between two features $X_k$ and $X_j$ we use the class-conditional mutual information [Cover and Thomas, 1991], which can be defined as

$$I(X_k; X_j|C) = H(X_k|C) + H(X_j|C) - H(X_k, X_j|C),$$

where $H(A|C)$ is the class-conditional entropy of $A$, defined as:

$$-\sum_i P(C = i) \sum_t P(A = t|C = i) \log P(A = t|C = i).$$

It can be shown [Cover and Thomas, 1991] that mutual information $I(X_k; X_j|C)$ equals the class-conditional *KL-divergence* between the joint distribution $P(X_k, X_j|C)$ and the product of marginals $P(X_k|C)P(X_j|C)$,

$$\sum_i P(C = i) \sum_{X_k = x_k, X_j = x_j} P(X_k = x_k, X_j = x_j|C = i) \\ \times \log \frac{P(X_k = x_k, X_j = x_j|C = i)}{P(X_k = x_k|C = i)P(X_j = x_j|C = i)},$$

[1] Clearly, discriminant functions are not unique, since classifier does not change if a monotone function (e.g., $\log$) is applied to all $f_i$'s.

which is zero when $X_k$ and $X_j$ are mutually independent given class $C$, and increases with increasing level of dependence, reaching the maximum when one feature is a deterministic function of the other.

## 3 Naive Bayes on low-entropy distributions

We start with analysis of naive Bayes on domains with almost deterministic, i.e. low-entropy, or "extreme", probability distributions, i.e. distributions having almost all the probability mass concentrated in one state. We show that approximations to such distributions based on independence assumption become more accurate with decreasing entropy, and therefore yield asymptotically optimal performance of naive Bayes.

The following lemma states that if a joint distribution over a set of variables is "extreme", than the marginal distributions of those variables are also "extreme".

**Lemma 1** *Given a joint probability distribution $P(X_1, ..., X_n)$ such that $P(\mathbf{x}^*) \geq 1 - \delta$ for some state $\mathbf{x}^* = (x_1^*, ..., x_n^*)$, $\mathbf{x}^* \in \Omega$,then for each $i$,*

$$P(X_i = x_i^*) \geq 1 - \delta.$$

**Proof.** Let $S_i^* = \{x = (x_{1,...,}x_n) | x_i = x_i^*, x \in \Omega\}$. Since $P(X_i = x_i^*) = \sum_{\mathbf{x} \in S_i^*} P(\mathbf{x})$, and since $S_i^*$ includes the point $(x_1^*, ..., x_n^*)$ that has $1 - \delta$ of all probability mass, we get $P(X_i = x_i^*) \geq 1 - \delta$. ■

The opposite result is also true: if all marginal distributions are "extreme", the joint distribution is "extreme".

**Lemma 2** *Given marginal probability distributions $P(X_1), ..., P(X_n)$ such that for each $i$ $P(X_i = x_i^*) \geq 1 - \delta$ for some $x_i^*$, then*

$$P(x_1^*, ..., x_n^*) \geq 1 - n\delta.$$

**Proof.** Since $P(X_i \neq x_i^*) \leq \delta$ for all $i$,

$$P(X_1 \neq x_1^* \vee \cdots \vee X_n \neq x_n^*) \leq \sum_i P(X_i \neq x_i^*) \leq n\delta,$$

using the simple union bound. The claimed bound follows by taking the complement of the event in the left hand side of the above inequality. ■

These results allow to compute a bound on approximation error when using independence assumption with "extreme" distributions, i.e. when the joint distribution is replaced by the product of marginals:

**Theorem 3** *Given a joint probability distribution $P(X_1, ..., X_n)$ such that $P(x_1^*, ..., x_n^*) \geq 1 - \delta$ for some state $\mathbf{x}^* = (x_1^*, ..., x_n^*)$, then*

$$|P(x_1, ..., x_n) - \prod_{i=1}^{n} P(X_i = x_i)| \leq n\delta.$$

**Proof.** From lemma 1 it follows that $P(X_i = x_i^*) \geq 1 - \delta$ for any $i = 1, ..., n$. Since $1 - n\delta \leq (1 - \delta)^n$ for $0 \leq \delta \leq 1$, we get

$$|P(\mathbf{x}^*) - \prod_{i=1}^{n} P(X_i = x_i^*)| \leq 1 - (1 - \delta)^n \leq n\delta.$$

On the other hand, if $\mathbf{x} = (x_1, ..., x_n)$ and $x_i \neq x_i^*$ for some $i$, then $P(X_i = x_i) \leq \delta$, and $P(\mathbf{x}) \leq \delta$, so that

$$|P(\mathbf{x}) - \prod_{i=1}^{n} P(X_i = x_i)| \leq \delta \leq n\delta,$$

which concludes the proof. ■

Similarly, it can be shown that

**Theorem 4** *Given a set of marginal probability distributions $P(X_1), ..., P(X_n)$ such that for each $i$ $P(X_i = x_i^*) \geq 1 - \delta$ for some $x_i^*$, then*

$$|P(x_1, ..., x_n) - \prod_{i=1}^{n} P(X_i = x_i)| \leq n\delta.$$

**Proof.** Let $\mathbf{x}^* = (x_1^*, ..., x_n^*)$. From lemma 2 it follows that $P(\mathbf{x}^*) \geq 1 - n\delta$, therefore, since $1 - n\delta \leq (1 - \delta)^n$, we get

$$|P(X = x^*) - \Pi_{i=1}^{n} P(X_i = x_i^*)| \leq n\delta.$$

On the other hand, if $\mathbf{x} = (x_1, ..., x_n)$ and $x_i \neq x_i^*$ for some $i$, then $P(\mathbf{x}) \leq n\delta$, and $P(X_i = x_i^*) \leq \delta$, so that

$$|P(\mathbf{x}) - \prod_{i=1}^{n} P(X_i = x_i)| \leq \delta \leq n\delta,$$

which concludes the proof. ■

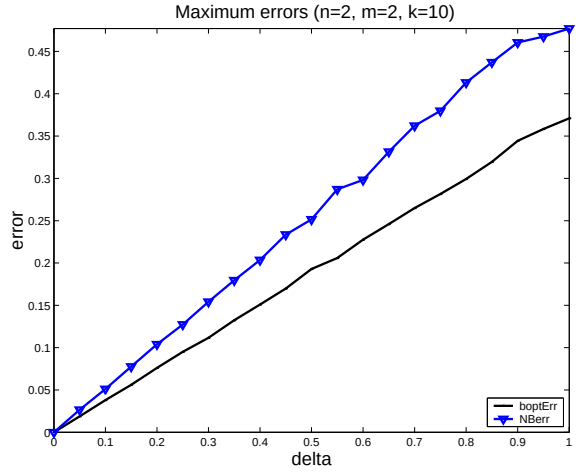
### 3.1 Empirical results

Clearly, if the difference between the joint feature distribution and its approximation by the product of marginals (all conditioned on class) vanishes with $\delta \to 0$, we would expect the naive Bayes error to vanish as well. Indeed, this is demonstrated by simulations on randomly generated problems.

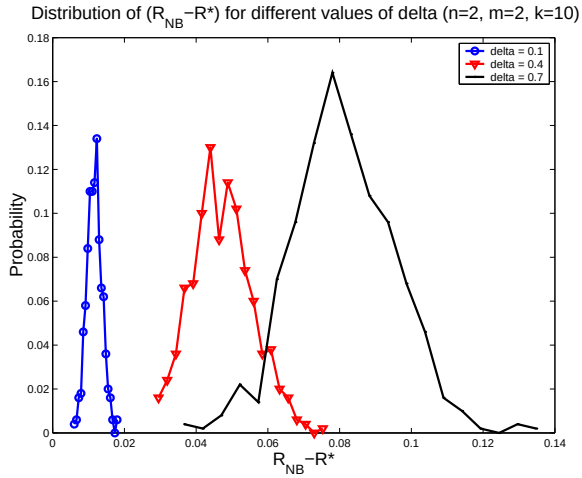
The problem generator, called **EXTREME**, takes the number of classes, $m$, number of features, $n$, number of values per feature, $k$, and the parameter $\delta$, and creates $m$ class-conditional feature distributions, each satisfying the condition $P(\mathbf{x}|C = c) = 1 - \delta$ if $\mathbf{x} = \mathbf{x}^c$, where $\mathbf{x}^c$ are $m$ different states randomly selected from $k^n$ possible states. For each class $i$, the remaining probability mass $\delta$ in $P(\mathbf{x}|C = i)$ is randomly distributed among the remaining $k^n - 1$ states. Class prior distributions are uniform. Once $P(\mathbf{X}|C)$ is generated, we compare naive Bayes classifier (NB) versus Bayes-optimal classifier (BO), assuming that both classifiers have perfect knowledge of data distribution (i.e., infinite amount of data).

Simulation results on a set of 500 problems with $n = 2$, $m = 2$, $k = 10$, and $\delta$ varying from 0 to 1 are shown in Figures 1a-1c. The maximum Bayes error (boptErr) and the maximum naive Bayes (NBerr) decrease to zero with $\delta \to 0$, as shown in Figure 1a. Figure 1b shows the distributions of the difference between Bayes and naive Bayes errors, $R_{NB} - R^*$, as a function of $\delta$. As expected, the distributions shift to the smaller values with decreasing $\delta$.

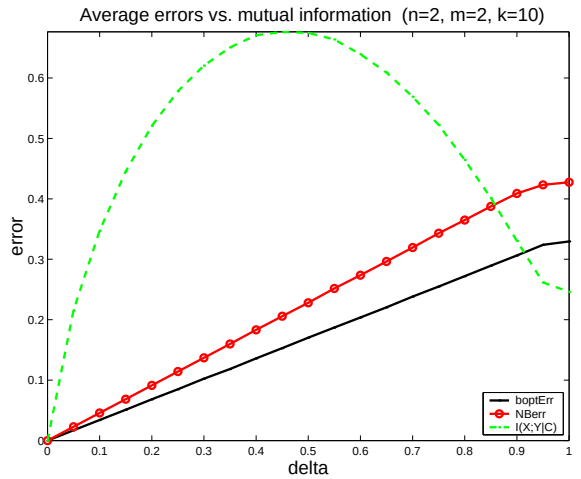
It is interesting to observe that the strength of dependencies among features (for the sake of simplicity, we consider only two features here) is not correlated with naive Bayes error, as we see in figure 1c which plots Bayes error and naive Bayes error versus average mutual information between the




(a)




(b)




(c)

Figure 1: Results for the generator **EXTREME**: (a) Maximum Bayes and naive Bayes errors; (b) Distribution of $R_{NB} - R^*$, the difference between average naive Bayes and Bayes errors; (c) Average Bayes and naive Bayes errors versus average class-conditional mutual information.

two features, as functions of $\delta$. The errors are monotone functions increasing with $\delta$, while mutual information is a concave function reaching its maximum at intermediate value of $\delta$ (approximately between 0.45 and 0.5). Note that empirical studies on UCI benchmark problems by [Domingos and Pazzani, 1997]) also revealed low correlation between the degree of feature dependence and relative performance of naive Bayes with respect to other state-of-the art classifiers, such as C4.5, CN2, and PEBLS.

Similar results were obtained for a different class of problems that mixes low-entropy distributions with an arbitrary ones. For one class, our generator called **MIX** creates a low-entropy class-conditional distribution as described before, and for the other one, it generates $k^n$ random entries in the feature probability table, and then normalizes the probabilities. The Naive Bayes error converges slower than in the case of two low-entropy distributions, since for same value of $\delta$ there is more noise in the problems due to random (instead of low-entropy) nature of one of the class-conditional feature distributions. The plot for average errors versus average mutual information (omitted here due to space limitations) looks very similar to the one we showed before, except that the errors are higher, and the average mutual information does not reach zero with $\delta \to 0$ since now the features do not become independent in the limit.

## 4 Functional and almost-functional dependencies among features

Surprisingly, naive Bayes can be optimal in cases just opposite to the class-conditional feature independence (when mutual information is at minimum) - namely, in cases of completely deterministic dependence among the features (when mutual information achieves its maximum).

**Theorem 5** *Given equal class priors, Naive Bayes is optimal if $X_i = f_i(X_1)$ for every feature $X_i$, $i = 2, ..., n$, where $f_i(\cdot)$ is a one-to-one mapping.*

**Proof.** Clearly, for every class $c$ and feature $i$, $P_c(X_i = x_i|X_1 = x_1)$ is 1 if $x_i = f_i(x_1)$ and 0 otherwise (assuming that $f_1(x_1) = x_1$). Let $\mathbf{x}^* = (x_1, f_2(x), ..., f_n(x))$. Then this is the only non-zero-probability state. Therefore, by the theorem of total probability

$$P_c(x_1) = \sum_{\mathbf{x}:\mathbf{x}\neq\mathbf{x}^*} P_c(\mathbf{x}) + P_c(\mathbf{x}^*) = P_c(\mathbf{x}^*).$$

Then the Bayes-optimal for selecting class $c$ can be written as

$$P_c(X_1 = x_1) > P_{c'}(X_1 = x_1) \forall c' \neq c.$$

On the other hand, the naive Bayes rule for selecting class $c$ can be written as

$$\prod_{i=1}^{n} P_i(f_i(x_1)) > \prod_{i=1}^{n} P_{c'}(f_i(x_1)) \; \forall c' \neq c,$$

and, since $\forall c$, $P_c(f_i(x_1)) = P_c(x_1)$, we get

$$P_c(X_1 = x_1)^n > P_{c'}(X_1 = x_1)^n \; \forall c' \neq c.$$

Clearly, those two classification rules agree for every value of $\mathbf{x}$ that has nonzero probability, i.e. for every $\mathbf{x} = (x_1, f_2(x_1), ..., f_m(x_1))$. Thus naive Bayes is optimal. ■

### 4.1 Empirical results

Our next question is how does naive Bayes behave with increasing noise in functional dependencies between the features. To answer this question, we used two random problem generators that relax functional dependencies using a noise parameter $\delta$ in a way similar to the low-entropy distribution generators. As before, we assume uniform class priors. Consider a simple case of two classes and two variables ($n = 2$ and $m = 2$) with $k$ values each. Our almost-functional distribution generator called **FUNC1** selects a random permutation of $k$ numbers, which corresponds to a one-to-one function $f$ that binds the two features: $X_2 = f(X_1)$ $(1 - \delta)$. Then it generates randomly two class-conditional (marginal) distributions for the $X_1$ feature, $P_0(X_1)$ and $P_1(X_1)$, for class 0 and class 1, respectively. Finally, it creates class-conditional joint feature distributions satisfying the following conditions:

$$P_c(x_1, x_2 = f(x_1)) = P_c(x_1)(1 - \delta), \text{and}$$
$$P_c(x_1, x_2 \neq f(x_1)) = P_c(x_1)\frac{\delta}{k-1}, c = 0, 1.$$

This way the states satisfying functional dependence obtain $1 - \delta$ probability mass, so that by controlling $\delta$ we can get as close as we want to the functional dependence described before, i.e. the generator relaxes the conditions of theorem 5. Note that, on the other hand, $\delta = \frac{k-1}{k}$ gives us uniform distributions over the second feature $P_c(x_2) = \sum_{x_1} P_c(x_1, x_2) = \frac{1}{k}$, which makes it independent of $X_1$ (given class $c$). Thus varying $\delta$ from 0 to 1 explores the whole range from deterministic dependence to complete independence between the features given class. Figure 2 summarizes the results for 500 problems with $n = 2$, $m = 2$ and $k = 10$. In Figure 2a we show maximum errors for each value of $\delta$ (note that Bayes error is constant). In Figure 2b we show the average errors; as we can see, naive Bayes is optimal when $\delta = 0$ and when $\delta = 0.9$, while the maximum error is reached between the two extremes. Note again, that errors are not correlated with the average class-conditional mutual information between the features (see Figure 2c), which decreases monotonically with $\delta$ going from 0 to 0.9. So, we can conclude that naive Bayes is close to the optimal not only for close-to-independent features (low mutual information), but also for close to one-to-one functional dependence.

Similar results are obtained for a modification of generator **FUNC1**, called **FUNC2**, where the probability mass $\delta$ is distributed over the rest of states not uniformly but randomly. As expected, in this case we do not get complete independence, and therefore average naive Bayes error only increases with $\delta$. Again, no (linear) correlation was observed between average error and average mutual information.

## 5 Conclusions

Naive Bayes classifier assumes that features are independence given class. Despite this unrealistic assumption, it is often highly competitive with more sophisticated learning techniques [Domingos and Pazzani, 1997; Mitchell, 1997; Friedman *et al.*, 1997]. Indeed, naive Bayes can be optimal in terms of zero-one loss (classification error) even if its class

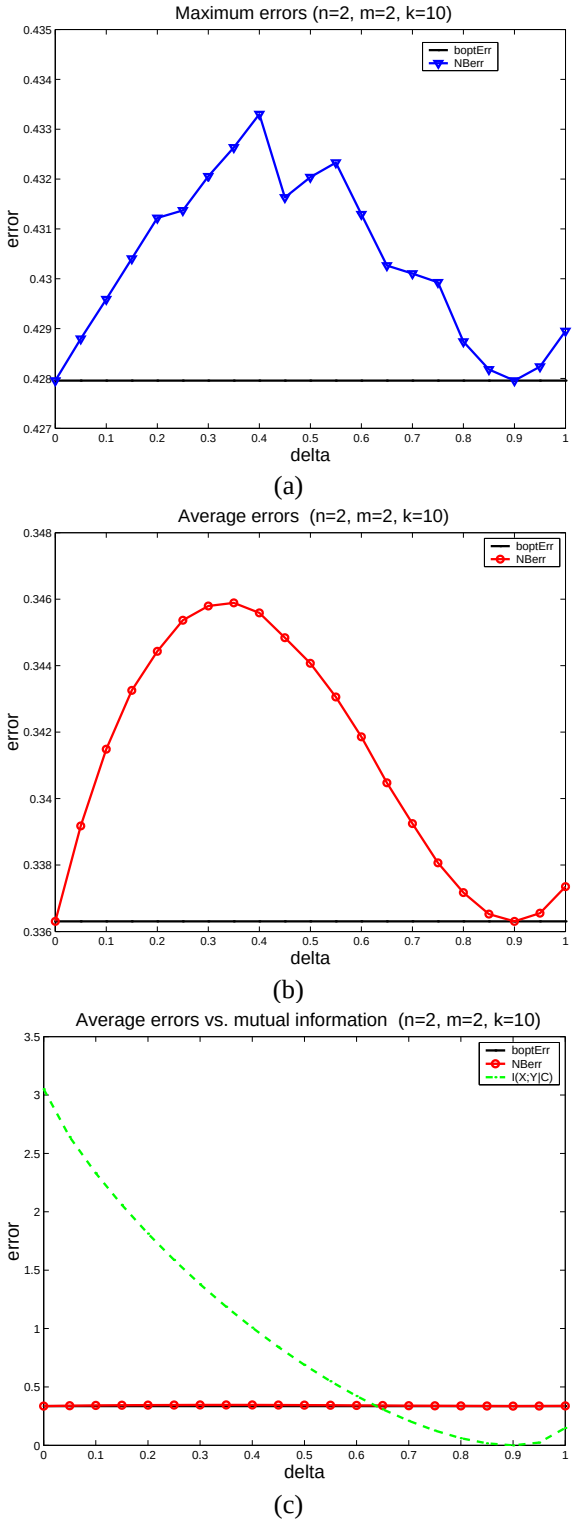



Figure 2: Results for the generator **FUNC1**: (a) Maximum Bayes and naive Bayes errors; (b) Average Bayes and naive Bayes errors ; (c) Average Bayes and naive Bayes errors versus average class-conditional mutual information.

probability estimates are wrong, as long as both true and estimated distributions agree on most-probable class [Domingos and Pazzani, 1997]. Although some optimality conditions of naive Bayes (such as conjunctive and disjunctive concepts) have been already identified in the past [Domingos and Pazzani, 1997], a deeper understanding of data characteristics that affect the performance of naive Bayes is still required. In this paper, we focus on problems including deterministic and close to deterministic dependencies. Such dependencies are often present in practical problems, e.g., in error-correcting coding [Frey and MacKay, 1998] and in computer system performance management [Hellerstein *et al.*, 2000], to name a few. We prove that naive Bayes is optimal in certain cases of functionally dependent features. Then we relax those dependencies by adding noise, demonstrating empirically that naive Bayes reaches optimal performance in cases of completely independent and functionally dependent features, while it behaves worst between the two extremes. We also derive bounds on the absolute error between a joint distribution and its approximation by the product of marginals for almost-deterministic, or low-entropy distributions, and demonstrate empirically how decreasing entropy of class-conditional feature distributions affects the error of naive Bayes classifier.

Directions for future work include analysis of naive Bayes on more complex problem classes that include almost-deterministic dependencies, including practical application, and further investigation of cases when ignoring dependence among features still yields a good classification accuracy. An ultimate goal is characterizing probability distributions that are insensitive to the independence assumption w.r.t. to classification task. This approach aims directly at learning probabilistic models that result into better classification accuracy, even though such models may not provide good approximations with respect to other criteria, such as KL-distance between true and estimated probability distributions, or such as MDL criterion. Empirical results also suggest that such general model selection criteria do not always lead to better classifiers [Friedman *et al.*, 1997].

## References


[Cover and Thomas, 1991] T.M. Cover and J.A. Thomas. *Elements of information theory*. New York:John Wiley & Sons, 1991.

[Dechter, 1997] R. Dechter. Mini-buckets: A general scheme for generating approximations in automated reasoning. In *Proc. Fifteenth International Joint Conference of Artificial Intelligence (IJCAI-97), Japan*, pages 1297–1302, 1997.

[Domingos and Pazzani, 1997] P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

[Duda and Hart, 1973] R.O. Duda and P.E. Hart. *Pattern classification and scene analysis*. New York: John Wiley and Sons, 1973.

[Frey and MacKay, 1998] B.J. Frey and D.J.C. MacKay. A revolution: Belief propagation in graphs with cycles. *Advances in Neural Information Processing Systems*, 10, 1998.

[Friedman *et al.*, 1997] N. Friedman, D. Geiger, and Goldszmidt M. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[Heckerman, 1995] D. Heckerman. A tutorial on learning Bayesian networks, technical report msr-tr-95-06. Technical report, Microsoft Research, 1995.

[Hellerstein *et al.*, 2000] J. Hellerstein, Jayram Thathachar, and I. Rish. Recognizing end-user transactions in performance management. In *Proceedings of AAAI-2000*, pages 596–602, Austin, Texas, 2000.

[Langley *et al.*, 1992] P. Langley, W. Iba, and K. Thompson. An analysis of Bayesian classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 399–406, San Jose, CA, 1992. AAAI Press.

[Mitchell, 1997] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[Rish, 1999] I. Rish. *Efficient reasoning in graphical models*. PhD thesis, 1999.