Pre-publication draft of a paper which appeared in the Proceedings of AI'03 LNAI 2903 pp 440 - 452 Publisher: Springer

On Why Discretization Works for Naive-Bayes Classifiers

Ying Yang YYANG@CSSE.MONASH.EDU.AU
Geoff I. Webb GEOFF.WEBB@CSSE.MONASH.EDU.AU
School of Computer Science and Software Engineering, Monash University, Melbourne, VIC 3800, Australia

Abstract

We investigate why discretization is effective in naive-Bayes learning. We prove a theorem that identifies particular conditions under which discretization will result in naive-Bayes classifiers delivering the same probability estimates as would be obtained if the correct probability density functions were employed. We discuss the factors that might affect naive-Bayes classification error under discretization. We suggest that the use of different discretization techniques can affect the classification bias and variance of the generated classifiers, an effect named *discretization bias* and *variance*. We argue that by properly managing discretization bias and variance, we can effectively reduce naive-Bayes classification error.

1. Introduction

Naive-Bayes classifiers are simple, effective, efficient, robust, and support incremental training. These merits have seen them deployed in numerous classification tasks. Naive-Bayes classifiers have long been a core technique in information retrieval (Maron & Kuhns, 1960; Lewis, 1992; Lewis & Gale, 1994; Larkey & Croft, 1996; Koller & Sahami, 1997; Mitchell, 1997; Pazzani & Billsus, 1997; Lewis, 1998; McCallum & Nigam, 1998; McCallum et al., 1998; Frasconi et al., 2001). They were first introduced into machine learning as a straw man, against which new algorithms were compared and evaluated (Cestnik et al., 1987; Clark & Niblett, 1989; Cestnik, 1990). But it was soon realized that their classification performance was surprisingly good compared with other more sophisticated classification algorithms (Kononenko, 1990; Langley et al., 1992; Domingos & Pazzani, 1997).

Holding the *attribute independence assumption*, naive-Bayes classifiers need only to estimate probabilities about individual attributes and the class instead of attribute combinations. An attribute can be either qualitative or quantitative. For a qualitative attribute, its probabilities can be estimated from corresponding frequencies. For a quantitative attribute, either probability density estimation or discretization can be employed to estimate its probabilities. Probability density estimation requires an assumption about the form of the probability distribution from which the quantitative attribute values are drawn. Discretization creates a qualitative attribute X_i^* from a quantitative attribute X_i. Each value of X_i^* corresponds to an interval of values of X_i. X_i^* is used instead of X_i for training a classifier.

With empirical evidence, Dougherty et al. (1995) suggested discretization to be effective because they did not make assumptions about the forms of quantitative attributes' probability distribution. Hsu et al. (2000) provided an analysis based on an assumption that each X_i^* has a Dirichlet prior. Because 'perfect aggregation' holds for Dirichlet distribution, the probability estimation of X_i^* can be estimated independent of the shape of the curve of X_i's probability density function. However, their analysis does not explain why different discretization methods have differing degrees of effectiveness. Instead, they suggested that 'well-known' discretization methods were unlikely to degrade the naive-Bayes classification performance. In contrast, we do not believe in this *unconditional* excellence. This motivates our research presented in this paper. In particular, we prove a theorem that explains why discretization can be effective. We argue that discretization for naive-Bayes learning should focus on the accuracy of the probability $p(C = c \,|\, X_i^* = x_i^*)$ as an estimate of $p(C = c \,|\, X_i = x_i)$ for each class c. Different discretization methods can have different accuracy of this estimation, which can affect the classification bias and variance of the generated naive-Bayes classifiers. We name this effect *discretization bias* and *variance*. We suggest that discretization methods that can well manage discretization bias and variance are of great utility.

The rest of this paper is organized as follows. Section 2 defines naive-Bayes classifiers. Section 3 proves a theorem that explains why discretization can be effective for naive-Bayes learning. It analyzes the factors that might affect the discretization effectiveness, and proposes the bias-variance characteristics of discretization. It introduces two new discretization tech-

niques that aim at managing discretization bias and variance. Section 4 discusses some related work. Section 5 presents the conclusion.

2. Naive-Bayes classifiers

In naive-Bayes learning, each *instance* is described by a vector of *attribute* values and its *class* can take any value from some predefined set of values. A set of instances with their classes, the *training data*, is provided. A *test instance* is presented. The learner is asked to predict its class according to the evidence provided by the training data. We define C as a random variable denoting the class of an instance; $\mathbf{X} < X_1, X_2, \cdots, X_k >$ as a vector of random variables denoting the observed attribute values (an instance); c as a particular class label; $\mathbf{x} < x_1, x_2, \cdots, x_k >$ as a particular observed attribute value vector (a particular instance); and $\mathbf{X} = \mathbf{x}$ as shorthand for $X_1 = x_1 \wedge X_2 = x_2 \wedge \cdots \wedge X_k = x_k$.

Expected classification error under zero-one loss can be minimized by choosing $argmax_c(p(C = c \,|\, \mathbf{X} = \mathbf{x}))$ for each $\mathbf{x}$ (Duda & Hart, 1973). Bayes' theorem can be used to calculate:

$$p(C = c \,|\, \mathbf{X} = \mathbf{x}) = \frac{p(C = c)p(\mathbf{X} = \mathbf{x} \,|\, C = c)}{p(\mathbf{X} = \mathbf{x})}. \quad (1)$$

Since the denominator in (1) is invariant across classes, it does not affect the final choice and can be dropped:

$$p(C = c \,|\, \mathbf{X} = \mathbf{x}) \propto p(C = c)p(\mathbf{X} = \mathbf{x} \,|\, C = c). \quad (2)$$

The probabilities $p(C = c)$ and $p(\mathbf{X} = \mathbf{x} \,|\, C = c)$ need to be estimated from the training data. Unfortunately, since $\mathbf{x}$ is usually an unseen instance which does not appear in the training data, it may not be possible to directly estimate $p(\mathbf{X} = \mathbf{x} \,|\, C = c)$. So a simplification is made: if attributes $X_1, X_2, \cdots, X_k$ are conditionally independent of each other given the class, then:

$$\begin{aligned} p(\mathbf{X} = \mathbf{x} \,|\, C = c) &= p(\wedge_{i=1}^{k} X_i = x_i \,|\, C = c) \\ &= \prod_{i=1}^{k} p(X_i = x_i \,|\, C = c). \end{aligned} \quad (3)$$

Combining (2) and (3), one can further estimate the most probable class by using:

$$p(C = c \,|\, \mathbf{X} = \mathbf{x}) \propto p(C = c) \prod_{i=1}^{k} p(X_i = x_i \,|\, C = c). \quad (4)$$

However, (4) is applicable only when X_i is qualitative. A qualitative attribute usually takes a small number of values (Bluman, 1992; Samuels & Witmer, 1999). Thus each value tends to have sufficient representative data. The probability $p(X_i = x_i \,|\, C = c)$ can be estimated from the frequency of instances with $C = c$ and the frequency of instances with $X_i = x_i \wedge C = c$. This estimate is a strong consistent estimate of $p(X_i = x_i \,|\, C = c)$ according to the strong law of large numbers (Casella & Berger, 1990; John & Langley, 1995).

When it is quantitative, X_i usually has a large or even an infinite number of values (Bluman, 1992; Samuels & Witmer, 1999). Since it denotes the probability that X_i will take the particular value x_i when the class is c, $p(X_i = x_i \,|\, C = c)$ might be arbitrarily close to zero. Accordingly, there usually are very few training instances for any one value. Hence it is unlikely that reliable estimation of $p(X_i = x_i \,|\, C = c)$ can be derived from the observed frequency. Consequently, in contrast to qualitative attributes, each quantitative attribute is modelled by some continuous probability distribution over the range of its values (John & Langley, 1995). Thus $p(X_i = x_i \,|\, C = c)$ is completely determined by a probability density function f, which satisfies (Scheaffer & McClave, 1995): (1) $f(X_i = x_i \,|\, C = c) \geq 0, \forall x_i \in S_i$; (2) $\int_{S_i} f(X_i \,|\, C = c)\mathrm{d}X_i = 1$; (3) $\int_{a_i}^{b_i} f(X_i \,|\, C = c)\mathrm{d}X_i = p(a_i \leq X_i \leq b_i \,|\, C = c), \forall [a_i, b_i] \in S_i$, where S_i is the value space of X_i. Naive-Bayes classifiers manipulate $f(X_i = x_i \,|\, C = c)$ instead of $p(X_i = x_i \,|\, C = c)$. According to John and Langley (1995), supposing X_i lying within some interval $[x_i, x_i + \Delta]$, we have $p(x_i \leq X_i \leq x_i + \Delta \,|\, C = c) = \int_{x_i}^{x_i+\Delta} f(X_i \,|\, C = c)\mathrm{d}X_i$. By the definition of a derivative, $\lim_{\Delta \to 0} \frac{p(x_i \leq X_i \leq x_i + \Delta \,|\, C = c)}{\Delta} = f(X_i = x_i \,|\, C = c)$. Thus for very small constant Δ, $p(X_i = x_i \,|\, C = c) \approx p(x_i \leq X_i \leq x_i + \Delta \,|\, C = c) \approx f(X_i = x_i \,|\, C = c) \times \Delta$. The factor Δ then appears in the numerator of (4) for each class. They cancel out when normalization is performed. Thus

$$p(X_i = x_i \,|\, C = c) \widetilde{\propto} f(X_i = x_i \,|\, C = c). \quad (5)$$

Combining (4) and (5), naive-Bayes classifiers estimate the probability of a class c given an instance $\mathbf{x}$ by

$$p(C = c \,|\, \mathbf{X} = \mathbf{x}) \propto p(C = c) \prod_{i=1}^{k} G(X_i = x_i \,|\, C = c),$$

$$\text{where} \quad G(X_i = x_i \,|\, C = c) = \begin{cases} p(X_i = x_i \,|\, C = c), & \text{if } X_i \text{ is qualitative;} \\ f(X_i = x_i \,|\, C = c), & \text{if } X_i \text{ is quantitative.} \end{cases} \quad (6)$$

Classifiers using (6) are *naive-Bayes classifiers*. The assumption embodied in (3) is the *attribute independence assumption*. In practice, typical approaches to estimating $p(C = c)$ and $p(X_i = x_i \,|\, C = c)$ are the Laplace-estimate and M-estimate respectively (Cestnik, 1990). Typical approaches to estimating $f(X_i = x_i \,|\, C = c)$ are assuming f to have a Gaussian distribution (Dougherty et al., 1995; Mitchell, 1997) or kernel density estimation (John & Langley, 1995).

3. Discretization

Discretization provides an alternative to probability density estimation when naive-Bayes learning involves quantitative attributes. Under probability density estimation, if the assumed density is not a proper estimate of the true density, the naive-Bayes classification performance tends to degrade (Dougherty et al., 1995; John & Langley, 1995). Since the true density is usually unknown for real-world data, unsafe assumptions unfortunately often occur. Discretization can circumvent this problem. Under discretization, a qualitative attribute X_i^* is formed for X_i. Each value x_i^* of X_i^* corresponds to an interval $(a_i, b_i]$ of X_i. Any original quantitative value $x_i \in (a_i, b_i]$ is replaced by x_i^*. All relevant probabilities are estimated with respect to x_i^*. Since probabilities of X_i^* can be properly estimated from corresponding frequencies as long as there are enough training instances, there is no need to assume the probability density function any more. However, because qualitative data have a lower level of measurement than quantitative data (Samuels & Witmer, 1999), discretization might suffer information loss.

3.1. Why discretization can be effective

We here prove Theorem 1 that suggests that discretization can be effective to the degree that $p(C = c \,|\, \mathbf{X}^* = \mathbf{x}^*)$ is an accurate estimate of $p(C = c \,|\, \mathbf{X} = \mathbf{x})$, where instance $\mathbf{x}^*$ is the discretized version of instance $\mathbf{x}$.

Theorem 1 *Assume the first l of k attributes are quantitative and the remaining attributes are qualitative*[1]*. Suppose instance $\mathbf{X}^* = \mathbf{x}^*$ is the discretized version of instance $\mathbf{X} = \mathbf{x}$, resulting from substituting qualitative attribute X_i^* for quantitative attribute X_i $(1 \le i \le l)$. If $\forall_{i=1}^{l}(p(C = c \,|\, X_i = x_i) = p(C = c \,|\, X_i^* = x_i^*))$, and the naive-Bayes attribute independence assumption (3) holds, we have $p(C = c \,|\, \mathbf{X} = \mathbf{x}) \propto p(C = c \,|\, \mathbf{X}^* = \mathbf{x}^*)$.*

Proof: According to Bayes theorem, we have:

$$\begin{aligned} & p(C = c \,|\, \mathbf{X} = \mathbf{x}) \\ = \;& p(C = c) \frac{p(\mathbf{X} = \mathbf{x} \,|\, C = c)}{p(\mathbf{X} = \mathbf{x})}; \end{aligned}$$

since the naive-Bayes attribute independence assumption (3) holds, we continue:

$$= \quad \frac{p(C = c)}{p(\mathbf{X} = \mathbf{x})} \prod_{i=1}^{k} p(X_i = x_i \,|\, C = c);$$

using Bayes theorem:

$$\begin{aligned} = \;& \frac{p(C = c)}{p(\mathbf{X} = \mathbf{x})} \prod_{i=1}^{k} \frac{p(X_i = x_i) p(C = c \,|\, X_i = x_i)}{p(C = c)} \\ = \;& \frac{p(C = c)}{p(C = c)^k} \frac{\prod_{i=1}^{k} p(X_i = x_i)}{p(\mathbf{X} = \mathbf{x})} \prod_{i=1}^{k} p(C = c \,|\, X_i = x_i); \end{aligned}$$

since the factor $\frac{\prod_{i=1}^{k} p(X_i = x_i)}{p(\mathbf{X} = \mathbf{x})}$ is invariant across classes:

$$\begin{aligned} \propto \;& p(C = c)^{1-k} \prod_{i=1}^{k} p(C = c \,|\, X_i = x_i) \\ = \;& p(C = c)^{1-k} \prod_{i=1}^{l} p(C = c \,|\, X_i = x_i) \prod_{j=l+1}^{k} p(C = c \,|\, X_j = x_j); \end{aligned}$$

since $\forall_{i=1}^{l}(p(C = c \,|\, X_i = x_i) = p(C = c \,|\, X_i^* = x_i^*))$:

$$= \quad p(C = c)^{1-k} \prod_{i=1}^{l} p(C = c \,|\, X_i^* = x_i^*) \prod_{j=l+1}^{k} p(C = c \,|\, X_j = x_j);$$

using Bayes theorem again:

$$\begin{aligned} = \;& p(C = c)^{1-k} \prod_{i=1}^{l} \frac{p(C = c) p(X_i^* = x_i^* \,|\, C = c)}{p(X_i^* = x_i^*)} \\ & \prod_{j=l+1}^{k} \frac{p(C = c) p(X_j = x_j \,|\, C = c)}{p(X_j = x_j)} \\ = \;& p(C = c) \frac{\prod_{i=1}^{l} p(X_i^* = x_i^* \,|\, C = c) \prod_{j=l+1}^{k} p(X_j = x_j \,|\, C = c)}{\prod_{i=1}^{l} p(X_i^* = x_i^*) \prod_{j=l+1}^{k} p(X_j = x_j)}; \end{aligned}$$

since the denominator $\prod_{i=1}^{l} p(X_i^* = x_i^*) \prod_{j=l+1}^{k} p(X_j = x_j)$ is invariant across classes:

$$\propto \quad p(C = c) \prod_{i=1}^{l} p(X_i^* = x_i^* \,|\, C = c) \prod_{j=l+1}^{k} p(X_j = x_j \,|\, C = c);$$

since the naive-Bayes attribute independence assumption (3) holds:

$$\begin{aligned} = \;& p(C = c) p(\mathbf{X}^* = \mathbf{x}^* \,|\, C = c) \\ = \;& p(C = c \,|\, \mathbf{X}^* = \mathbf{x}^*) p(\mathbf{X}^* = \mathbf{x}^*); \end{aligned}$$

since $p(\mathbf{X}^* = \mathbf{x}^*)$ is invariant across classes:

$$\propto p(C = c \,|\, \mathbf{X}^* = \mathbf{x}^*). \quad \square$$

Theorem 1 assures us that as long as the attribute independence assumption holds, and discretization forms a qualitative X_i^* for each quantitative X_i such that $p(C = c \,|\, X_i^* = x_i^*) = p(C = c \,|\, X_i = x_i)$, discretization will result in naive-Bayes classifiers delivering the same probability estimates as would be obtained if the correct probability density function were employed. Since X_i^* is qualitative, naive-Bayes classifiers can estimate $p(C = c \,|\, \mathbf{X} = \mathbf{x})$ without assuming

[1] In naive-Bayes learning, the order of attributes does not matter. We make this assumption only to simplify the expression of our proof. This does not at all affect the theoretical analysis.

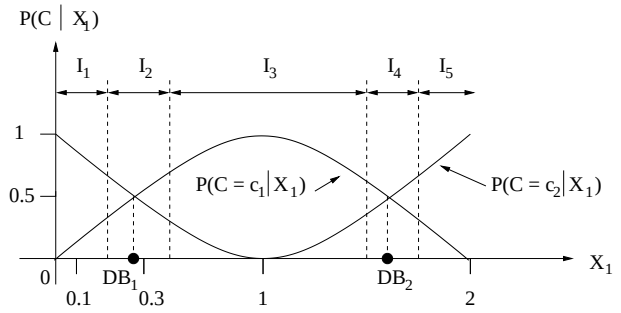

Figure 1. Probability distribution in one-attribute problem

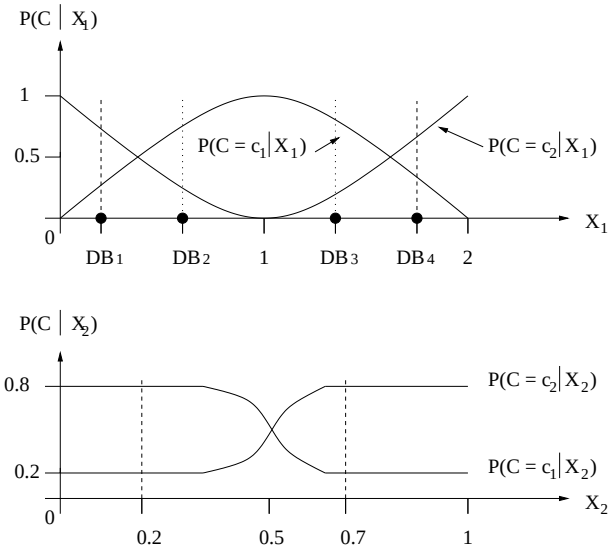

Figure 2. Probability distribution in two-attribute problem

any form of the probability density.

3.2. What can affect discretization effectiveness

When we talk about the effectiveness of a discretization method, we mean the classification performance of naive-Bayes classifiers that are trained on data preprocessed by this discretization method. According to Theorem 1, we believe that the accuracy of estimating $p(C = c \,|\, X_i = x_i)$ by $p(C = c \,|\, X_i^* = x_i^*)$ takes a key role in this issue. Two influential factors are *decision boundaries* and the *error tolerance of probability estimation*. How discretization deals with these factors can affect the classification bias and variance of the generated classifiers, an effect we name *discretization bias* and *variance*. According to (6), the prior probability of each class $p(C = c)$ also affects the final choice of the class. To simplify our analysis, we here assume that each class has the same prior probability. That is, $p(C = c)$ is identical for each c. Thus we can cancel the effect of $p(C = c)$. However, our analysis extends straightforwardly to non-uniform cases.

3.2.1. Classification bias and variance

The performance of naive-Bayes classifiers discussed in our study is measured by their classification *error*. The error can be partitioned into a *bias* term, a *variance* term and an *irreducible* term (Kong & Dietterich, 1995; Breiman, 1996; Kohavi & Wolpert, 1996; Friedman, 1997; Webb, 2000). Bias describes the component of error that results from systematic error of the learning algorithm. Variance describes the component of error that results from random variation in the training data and from random behavior in the learning algorithm, and thus measures how sensitive an algorithm is to changes in the training data. As the algorithm becomes more sensitive, the variance increases. Irreducible error describes the error of an optimal algorithm (the level of noise in the data). Consider a classification learning algorithm A applied to a set S of training instances to produce a classifier to classify an instance $\mathbf{x}$. Suppose we could draw a sequence of training sets $S_1, S_2, ..., S_l$, each of size m, and apply A to construct classifiers. The error of A at $\mathbf{x}$ can be defined as: $Error(A, m, \mathbf{x}) = Bias(A, m, \mathbf{x}) + Variance(A, m, \mathbf{x}) + Irreducible(A, m, \mathbf{x})$. There is often a 'bias and variance trade-off' (Kohavi & Wolpert, 1996). All other things being equal, as one modifies some aspect of the learning algorithm, it will have opposite effects on bias and variance. A good learning scheme must have both low bias and low variance (Moore & McCabe, 2002).

3.2.2. Decision boundary

This factor in our analysis is inspired by Hsu et al.'s study on discretization (2000). However, Hsu et al.'s analysis focused on the curve of $f(X_i = x_i \,|\, C = c)$. Instead, a decision boundary of a quantitative attribute X_i in our analysis is the value that makes ties among the largest probabilities of $p(C \,|\, \mathbf{X} = \mathbf{x})$ for a test instance $\mathbf{x}$, given the precise values of other attributes presented in $\mathbf{x}$.

Consider a simple learning task with one quantitative attribute X_1 and two classes c_1 and c_2. Suppose $X_1 \in [0, 2]$, and suppose that the probability distribution function for each class is $p(C = c_1 \,|\, X_1) = 1 - (X_1 - 1)^2$ and $p(C = c_2 \,|\, X_1) = (X_1 - 1)^2$ respectively, which are plotted in Figure 1. The consequent decision boundaries are labelled as DB_1 and DB_2 respectively in Figure 1. The most-probable class for an instance $\mathbf{x} =< x_1 >$ changes each time x_1's location crosses a decision boundary. Assume a discretization method to create intervals I_i $(i = 1, \cdots, 5)$ as in Figure 1. I_2 and I_4 contain decision boundaries while the remaining intervals do not. For any two values in I_2 (or I_4) but on different sides of a decision

boundary, the optimal naive-Bayes learner under zero-one loss should select a different class for each value[2]. But under discretization, all the values in the same interval can not be differentiated and we will have the same class probability estimate for all of them. Consequently, naive-Bayes classifiers with discretization will assign the same class to all of them, and thus values at one of the two sides of the decision boundary will be misclassified. This effect is expected to affect the bias of the generated classifiers, and thus is named hereafter *discretization bias*. The larger the *interval size* (the number of training instances in the interval), the more likely that the interval contains a decision boundary. The larger the interval containing a decision boundary, the more instances to be misclassified, thus the higher the discretization bias.

In one-attribute problems[3], the locations of decision boundaries of the attribute X_1 depend on the distribution of $p(C \mid X_1)$ for each class. However, for a multi-attribute application, the decision boundaries of an attribute, say X_1, are not only decided by the distribution of $p(C \mid X_1)$, but also vary from test instance to test instance depending upon the precise values of other attributes. Consider another learning task with two quantitative attributes X_1 and X_2, and two classes c_1 and c_2. The probability distribution of each class given each attribute is depicted in Figure 2, of which the probability distribution of each class given X_1 is identical with that in the above one-attribute context. We assume that the attribute independence assumption holds. We analyze the decision boundaries of X_1 for an example. If X_2 does not exist, X_1 has decision boundaries as depicted in Figure 1. However, because of the existence of X_2, those might not be decision boundaries any more. Consider a test instance $\mathbf{x}$ with $X_2 = 0.2$. Since $p(C = c_1 \mid X_2 = 0.2) = 0.8 > p(C = c_2 \mid X_2 = 0.2) = 0.2$, and $p(C = c \mid \mathbf{x}) \propto \prod_{i=1}^{2} p(C = c \mid X_i = x_i)$ for each class c according to Theorem 1, $p(C = c_1 \mid \mathbf{x})$ does not equal $p(C = c_2 \mid \mathbf{x})$ when X_1 falls on any of the single attribute decision boundaries as presented in Figure 1. Instead X_1's decision boundaries change to be DB_1 and DB_4 as in Figure 2. Suppose another test instance with $X_2 = 0.7$. By the same reasoning X_1's decision boundaries change to be DB_2 and DB_3 as in Figure 2. When there are more than two attributes, each combination of values of the attributes other than X_1 will result in corresponding decision boundaries of X_1. Thus in multi-attribute applications the decision boundaries of one attribute can only be identified with respect to each specific combination of values of the other attributes. Increasing either the number of attributes or the number of values of an attribute will increase the number of combinations of attribute values, and thus the number of decision boundaries. In consequence, each attribute may have a very large number of potential decision boundaries. Nevertheless, for the same reason as we have discussed in one-attribute context, intervals containing decision boundaries have the potential negative impact on discretization bias.

Consequently, discretization bias can be reduced by identifying the decision boundaries and setting the interval boundaries close to them. However, identifying the correct decision boundaries depends on finding the true form of $p(C \mid X_1)$. Ironically, if we have already found $p(C \mid X_1)$, we can resolve the classification task directly; thus there is no need to consider discretization at all. Without knowing $p(C \mid X_1)$, an extreme solution is to set each value as an interval. Although this most likely guarantees that no interval contains a decision boundary, it usually results in very few instances per interval. As a result, the estimation of $p(C \mid X_1)$ might be so unreliable that we can not identify the truly most probable class even if there is no decision boundary in the interval. This will affect the classification variance of the generated classifiers. The less training instances per interval for probability estimation, the more likely that it increases the variance of the generated classifiers since even a small change of the training data might totally change the probability estimation. Thus we name this effect *discretization variance*. A possible solution to this problem is to require that the size of an interval should be sufficient to ensure stability in the probability estimated therefrom. This raises the question, how reliable must the probability be? That is, when estimating $p(C = c \mid X_1 = x_1)$ by $p(C = c \mid X_1^* = x_1^*)$, how much error can be tolerated without altering the classification. This motivates our following analysis.

3.2.3. Error tolerance of probability estimation

To investigate this issue, we return to our example depicted in Figure 1. We suggest that different values have different error tolerance of their probability estimation. For example, for a test instance $\mathbf{x} < X_1 = 0.1 >$ and thus of class c_2, its true class probability distribution is $p(C = c_1 \mid \mathbf{x}) = p(C = c_1 \mid X_1 = 0.1) = 0.19$ and $p(C = c_2 \mid \mathbf{x}) = p(C = c_2 \mid X_1 = 0.1) = 0.81$. According to naive-Bayes learning, as long as $p(C = c_2 \mid X_1 = 0.1) > 0.50$, c_2 will be correctly assigned as the class and the classification is optimal under zero-one loss. This means, the error tolerance of estimating $p(C \mid X_1 = 0.1)$ can be as big as $0.81 - 0.50 = 0.31$. However, for another test instance $\mathbf{x} < X_1 = 0.3 >$ and thus of class c_1, its probability distribution is

[2]Please note that since naive-Bayes classification is a probabilistic problem, some instances will be misclassified even when optimal classification is performed. An optimal classifier is such that minimizes the naive-Bayes classification error under zero-one loss. Hence even though it is optimal, it still can misclassify instances on both sides of a decision boundary.

[3]By default, we talk about quantitative attributes.

$p(C = c_1 \,|\, \mathbf{x}) = p(C = c_1 \,|\, X_1 = 0.3) = 0.51$ and $p(C = c_2 \,|\, \mathbf{x}) = p(C = c_2 \,|\, X_1 = 0.3) = 0.49$. The error tolerance of estimating $p(C \,|\, X_1 = 0.3)$ is only $0.51 - 0.50 = 0.01$. In the learning context of multi-attribute applications, the analysis of the tolerance of probability estimation error is even more complicated. The error tolerance of a value of an attribute affects as well as is affected by those of the values of the other attributes since it is the multiplication of $p(C = c \,|\, X_i = x_i)$ of each x_i that decides the final probability of each class.

The lower the error tolerance a value has, the larger its interval size is preferred for the purpose of reliable probability estimation. Since all the factors that affect error tolerance vary from case to case, there can not be a universal, or even a domain-wide constant that represents the ideal interval size, which thus will vary from case to case. Further, the error tolerance can only be calculated if the true probability distribution of the training data is known. If it is not known, then the best we can hope for is heuristic approaches to managing error tolerance that work well in practice.

3.3. Summary

By this line of reasoning, optimal discretization can only be performed if the probability distribution of $p(C = c \,|\, X_i = x_i)$ for each pair of x_i and c, given each particular test instance, is known; and thus the decision boundaries are known. If the decision boundaries are not known, which is often the case for real-world data, we want to have as many intervals as possible so as to minimize the risk that an instance is classified using an interval containing a decision boundary. By this means we expect to reduce the discretization bias. Also, a number of previous authors have mentioned that the *interval number* (the number of intervals formed) has a major effect on the naive-Bayes classification error (Pazzani, 1995; Torgo & Gama, 1997; Gama et al., 1998; Hussain et al., 1999; Mora et al., 2000; Hsu et al., 2000). On the other hand, however, we want to ensure that the intervals are sufficiently large to minimize the risk that the error of estimating $p(C = c \,|\, X_i^* = x_i^*)$ will exceed the current error tolerance. By this means we expect to reduce the discretization variance.

However, when the number of the training instances is fixed, there is a trade-off between interval size and interval number. That is, the larger the interval size, the smaller the interval number, and vice versa. Because larger interval size can result in lower discretization variance but higher discretization bias, while larger interval number can result in lower discretization bias but higher discretization variance, low learning error can be achieved by tuning interval size and interval number to find a good trade-off between discretization bias and variance. We argue that there is no universal solution to this problem, that the optimal trade-off between interval size and interval number will vary greatly from test instance to test instance.

Our analysis has been supported by the success of two new discretization techniques that we have recently developed: *proportional k-interval discretization* (PKID) and *equal size discretization* (ESD) (Yang & Webb, 2001; Yang & Webb, 2003). To discretize a quantitative attribute, PKID equally weighs discretization bias and variance by setting interval size and interval number equal. It uses an increase in training data to lower both discretization bias and variance by setting them proportional to the training data size. As the number of training instances increases, both discretization bias and variance tend to decrease. Bias can decrease because the interval number increases, thus the decision boundaries of the original quantitative values are less likely to be included in intervals. Variance can decrease because the interval size increases, thus the naive-Bayes probability estimation is more stable and reliable. ESD sets a *safe interval size* m. It discretizes the values into intervals of size m. By introducing m, ESD aims to ensure that the interval size is sufficient so that there are enough training instances in each interval to reliably estimate the naive-Bayes probabilities. Thus ESD can control discretization variance by preventing it from being very high. By not limiting the number of intervals formed, more intervals can be formed as the training data increases. This means that ESD can make use of extra data to reduce discretization bias. Our experimental results have demonstrated that with frequency significant at the 0.05 level, PKID and ESD each better reduce naive-Bayes classification error than previous key discretization methods.

4. Related work

Our analysis in Theorem 1, which focuses on $p(C = c \,|\, X_i = x_i)$ instead of $f(X_i = x_i \,|\, C = c)$, is derived from Kononenko's (1992). However, Kononenko's analysis required that the attributes be assumed *unconditionally* independent of each other, which entitles $\prod_{i=1}^{k} p(X_i = x_i) = p(\mathbf{X} = \mathbf{x})$. This assumption is much stronger than the naive-Bayes attribute independence assumption embodied in (3). Thus we suggest that our deduction in Theorem 1 more accurately captures the mechanism by which discretization works.

Dougherty et al. (1995) conducted an empirical study to show that naive-Bayes classifiers resulting from discretization achieved lower classification error than those resulting from unsafe probability density assumptions. With these empirical supports, Dougherty et al. suggested that discretization could be effective because they did not make assumptions about the form of the probability distribution from which the quantitative attribute values were drawn.

Hsu et al. (2000) proposed an analysis of discretization with more theoretical supports. Different from our approach, they were interested in analyzing the density function f. In particular, they suggested that discretization would achieve optimal effectiveness by forming x_i^* for x_i such that $p(X_i^* = x_i^* \,|\, C = c)$ simulates the role of $f(X_i = x_i \,|\, C = c)$ by distinguishing the class that gives x_i high density from the class that gives x_i low density. In contrast, our study focuses on the probability $p(C = c \,|\, X_i = x_i)$. Besides, Hsu et al.'s analysis only addressed one-attribute classification problems, and suggested that the analysis could be extended to multi-attribute applications without indicating how this might be so. In contrast, we argue that the analysis involving only one attribute differs from that involving multi-attributes. Furthermore, Hsu et al.'s analysis suggested that 'well-known' discretization methods would have similar effectiveness and would be unlikely to degrade the probability estimation. Instead we supply an insight into why there exists different degrees of effectiveness among difference methods, and discuss this difference in terms of discretization bias and variance.

5. Conclusion

In this paper, we supply the proof of a theorem that provides a new explanation of why discretization can be effective for naive-Bayes classifiers by showing that discretization will not alter the naive-Bayes estimate as long as $p(C = c \,|\, X_i^* = x_i^*) = p(C = c \,|\, X_i = x_i)$. We explore the factors that can affect the discretization effectiveness in terms of the classification bias and variance of the generated classifiers. We name this effect discretization bias and variance. We have argued that the analysis of the bias-variance characteristics of discretization provides insights into discretization effectiveness. In particular, we have obtained new understandings of how discretization bias and variance can be manipulated by adjusting interval size and interval number. In short, we want to maximize the number of intervals in order to minimize discretization bias, but at the same time ensure that each interval contains sufficient training instances in order to obtain low discretization variance.

Another illuminating issue arising from our study is that since the decision boundaries of a quantitative attribute value depend on the values of other quantitative attributes given a particular test instance, we can not develop optimal discretization by any apriori methodology, that is, by forming intervals prior to the classification time. However, even if we adopt a lazy methodology (Zheng & Webb, 2000), that is, taking into account the values of other attributes during classification time, we still cannot guarantee optimal discretization unless we know the true probability distribution of the quantitative attributes. These insights reveal that, while discretization is desirable when the true underlying probability density function is not available, practical discretization techniques are necessarily heuristic in nature. The holy grail of an optimal universal discretization strategy for naive-Bayes learning is unobtainable.

References

Bluman, A. G. (1992). *Elementary statistics, a step by step approach.* Wm.C.Brown Publishers. page5-8.

Breiman, L. (1996). Bias, variance and arcing classifiers. *Technical Report, Statistics Department, University of California, Berkerley.*

Casella, G., & Berger, R. L. (1990). *Statistical inference.* Pacific Grove, Calif. the strong law of larger number.

Cestnik, B. (1990). Estimating probabilities: A crucial task in machine learning. *Proc. of the Ninth European Conf. on Artificial Intelligence* (pp. 147–149).

Cestnik, B., Kononenko, I., & Bratko, I. (1987). Assistant 86: A knowledge-elicitation tool for sophisticated users. *Proc. of the Second European Working Session on Learning* (pp. 31–45).

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, *3*, 261–283.

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, *29*, 103–130.

Dougherty, J., Kohavi, R., & Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. *Proc. of the Twelfth International Conf. on Machine Learning* (pp. 194–202).

Duda, R., & Hart, P. (1973). *Pattern classification and scene analysis.* John Wiley & Sons.

Frasconi, P., Soda, G., & Vullo, A. (2001). Text categorization for multi-page documents: a hybrid naive bayes hmm approach. *Proc. of the ACM/IEEE Joint Conf. on Digital Libraries* (pp. 11–20).

Friedman, J. H. (1997). On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, *1*, 55–77.

Gama, J., Torgo, L., & Soares, C. (1998). Dynamic discretization of continuous attributes. *Proc. of the Sixth Ibero-American Conf. on AI* (pp. 160–169).

Hsu, C. N., Huang, H. J., & Wong, T. T. (2000). Why discretization works for naive Bayesian classifiers. *Proc. of the Seventeenth International Conf. on Machine Learning* (pp. 309–406).

Hussain, F., Liu, H., Tan, C. L., & Dash, M. (1999). Discretization: An enabling technique. Technical Report, TRC6/99, School of Computing, National University of Singapore.

John, G. H., & Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. *Proc. of the Eleventh Conf. on Uncertainty in Artificial Intelligence* (pp. 338–345).

Kohavi, R., & Wolpert, D. (1996). Bias plus variance decomposition for zero-one loss functions. *Proc. of the 13th International Conf. on Machine Learning* (pp. 275–283).

Koller, D., & Sahami, M. (1997). Hierarchically classifying documents using very few words. *Proc. of the Fourteenth International Conf. on Machine Learning* (pp. 170–178).

Kong, E. B., & Dietterich, T. G. (1995). Error-correcting output coding corrects bias and variance. *Proc. of the Twelfth International Conf. on Machine Learning* (pp. 313–321).

Kononenko, I. (1990). Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition.

Kononenko, I. (1992). Naive Bayesian classifier and continuous attributes. *Informatica*, *16*, 1–8.

Langley, P., Iba, W., & Thompson, K. (1992). An analysis of bayesian classifiers. *Proc. of the Tenth National Conf. on Artificial Intelligence* (pp. 223–228).

Larkey, L. S., & Croft, W. B. (1996). Combining classifiers in text categorization. *Proc. of the 19th Annual International Conf. on Research and Development in Information Retrieval* (pp. 289–297).

Lewis, D. D. (1992). An evaluation of phrasal and clustered representations on a text categorization task. *Proc. of the 15th Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval* (pp. 37–50).

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. *In European Conf. on Machine Learning*.

Lewis, D. D., & Gale, W. A. (1994). A sequential algorithm for training text classifiers. *Proc. of the 17th Annual International ACM-SIGIR Conf. on Research and Development in Information Retrieval* (pp. 3–12).

Maron, M., & Kuhns, J. (1960). On relevance, probabilistic indexing, and information retrieval. *Journal of the Association for Computing Machinery*, *7*, 216–244.

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive bayes text classification. *Proc. of the AAAI-98 Workshop on Learning for Text Categorization*.

McCallum, A., Rosenfeld, R., Mitchell, T. M., & Ng, A. (1998). Improving text classification by shrinkage in a hierarchy of classes. *Proc. of the 15th International Conf. on Machine Learning*.

Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill Companies.

Moore, D. S., & McCabe, G. P. (2002). *Introduction to the practice of statistics*. Michelle Julet. Fourth Edition.

Mora, L., Fortes, I., Morales, R., & Triguero, F. (2000). Dynamic discretization of continuous values from time series. *Proc. of the Eleventh European Conf. on Machine Learning* (pp. 280–291).

Pazzani, M., & Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine Learning 27*, 313–331.

Pazzani, M. J. (1995). An iterative improvement approach for the discretization of numeric attributes in Bayesian classifiers. *Proc. of the First International Conf. on Knowledge Discovery and Data Mining*.

Samuels, M. L., & Witmer, J. A. (1999). *Statistics for the life sciences, second edition*. Prentice-Hall. page10-11.

Scheaffer, R. L., & McClave, J. T. (1995). *Probability and statistics for engineers*. Duxbury Press. Fourth edition.

Torgo, L., & Gama, J. (1997). Search-based class discretization. *Proc. of the Ninth European Conf. on Machine Learning* (pp. 266–273).

Webb, G. I. (2000). Multiboosting: A technique for combining boosting and wagging. *Machine Learning*, *40*, 159–196.

Yang, Y., & Webb, G. I. (2001). Proportional k-interval discretization for naive-Bayes classifiers. *Proc. of the Twelfth European Conf. on Machine Learning* (pp. 564–575).

Yang, Y., & Webb, G. I. (2003). Discretization for naive-Bayes learning: Managing discretization bias and variance. *Technical Report 2003/131 (submitted for journal publication), School of Computer Science and Software Engineering, Monash University*.

Zheng, Z., & Webb, G. I. (2000). Lazy learning of Bayesian rules. *Machine Learning*, *41*, 53–84.