Discretization for naive-Bayes learning: managing discretization bias and variance

Ying Yang Geoffrey I. Webb

School of Computer Science and Software Engineering
Monash University
Melbourne, VIC 3800, Australia

Abstract

Quantitative attributes are usually discretized in naive-Bayes learning. We prove a theorem that explains why discretization can be effective for naive-Bayes learning. The use of different discretization techniques can be expected to affect the classification bias and variance of generated naive-Bayes classifiers, effects we name *discretization bias* and *variance*. We argue that by properly managing discretization bias and variance, we can effectively reduce naive-Bayes classification error. In particular, we propose *proportional k-interval discretization* and *equal size discretization*, two efficient heuristic discretization methods that are able to effectively manage discretization bias and variance by tuning discretized interval size and interval number. We empirically evaluate our new techniques against five key discretization methods for naive-Bayes classifiers. The experimental results support our theoretical arguments by showing that naive-Bayes classifiers trained on data discretized by our new methods are able to achieve lower classification error than those trained on data discretized by alternative discretization methods.

Key words: Discretization, Naive-Bayes Learning, Bias, Variance

1 Introduction

When classifying an instance, naive-Bayes classifiers assume attributes conditionally independent of each other given the class; and then apply Bayes' theorem to calculate the probability of each class given this instance. The class with the highest probability is chosen as the class of this instance. Naive-Bayes

Email addresses: yyang@csse.monash.edu.au (Ying Yang),
Geoff.Webb@csse.monash.edu.au (Geoffrey I. Webb).

classifiers are simple, effective, efficient, robust and support incremental training. These merits have seen them deployed in numerous classification tasks. They have long been a core technique in information retrieval (Maron and Kuhns, 1960; Maron, 1961; Lewis, 1992; Guthrie and Walker, 1994; Lewis and Gale, 1994; Kalt, 1996; Larkey and Croft, 1996; Pazzani et al., 1996; Starr et al., 1996a; Joachims, 1997; Koller and Sahami, 1997; Li and Yamanishi, 1997; Mitchell, 1997; Pazzani and Billsus, 1997; Lewis, 1998; McCallum and Nigam, 1998; McCallum et al., 1998; Nigam 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 and 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 and Pazzani, 1996, 1997; Zhang et al., 2000). Thus they have often been chosen as the base algorithm for bagging, boosting, wrapper, voting or hybrid methodologies (Kohavi, 1996; Zheng, 1998; Bauer and Kohavi, 1999; Ting and Zheng, 1999; Gama, 2000; Kim et al., 2000; Tsymbal et al., 2002). In addition, naive-Bayes classifiers also have widespread deployment in medical diagnosis (Kononenko, 1993; Kohavi et al., 1997; Kukar et al., 1997; McSherry, 1997a,b; Zelic et al., 1997; Montani et al., 1998; Lavrac, 1998; Lavrac et al., 2000; Kononenko, 2001; Zupan et al., 2001), email filtering (Pantel and Lin, 1998; Provost, 1999; Androutsopoulos et al., 2000; Rennie, 2000; Crawford et al., 2002), and recommender systems (Starr et al., 1996b; Miyahara and Pazzani, 2000; Mooney and Roy, 2000).

Classification tasks often involve quantitative attributes. For naive-Bayes classifiers, quantitative attributes are usually processed by discretization, as the classification performance tends to be better when quantitative attributes are discretized than when their probabilities are estimated by making unsafe assumptions about the forms of the underlying distributions from which they are drawn (Dougherty et al., 1995). Discretization creates a qualitative attribute X^* from a quantitative attribute X. Each value of X^* corresponds to an interval of values of X. X^* is used instead of X for training a classifier. In contrast to parametric techniques for inference from quantitative attributes, such as probability density estimation, discretization can avoid assuming quantitative attributes' underlying distributions. However, because qualitative data have a lower level of measurement than quantitative data (Samuels and Witmer, 1999), discretization might suffer information loss. This information loss will affect the learning bias and variance of generated naive-Bayes classifiers. Such an effect is hereafter named *discretization bias* and *variance*. We believe that analysis of discretization bias and variance is illuminating. We investigate the impact of discretization bias and variance on the classification performance of naive-Bayes classifiers. The resulting insights motivate the development of two new heuristic discretization methods, *proportional k-interval discretization* and *equal size discretization*. Our goals are to improve both the classification efficacy and efficiency of naive-Bayes classifiers. These dual goals are of particular significance given naive-Bayes classifiers' widespread deployment, and in particular their deployment in time-sensitive interactive applications.

The rest of this paper is organized as follows. Section 2 presents an analysis on discretization's terminology and taxonomy. Section 3 defines naive-Bayes classifiers. In naive-Bayes learning, estimating probabilities for qualitative attributes is different from that for quantitative attributes. Section 4 introduces discretization in naive-Bayes learning. It includes a proof that states 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 function were employed. It analyzes the factors that might affect the discretization effectiveness in the multi-attribute learning context. It also proposes the bias-variance characteristic of discretization. Section 5 provides a review of previous key discretization methods, discussing their behaviors in terms of discretization bias and variance. Section 6 proposes two new heuristic discretization methods designed to manage discretization bias and variance, *proportional k-interval discretization* and *equal size discretization*. Section 7 compares the algorithm computational time complexities. Section 8 carries out experimental validation. Section 9 presents the conclusion.

2 Terminology and taxonomy

There is a large amount of literature addressing discretization, among which there is considerable variation in the terminology used to describe which type of data is transformed to which type of data by discretization. There also exist various proposals to taxonomize discretization methods, each taxonomy emphasizing different aspects of the distinctions among the methods. Here we clarify the difference among these variations and choose terms for use hereafter.

2.1 Terminology

Discretization transforms one type of data into another type. The two data types are variously referred to in previous literature, as 'quantitative' *vs.* 'qualitative', 'continuous' *vs.* 'discrete', 'nominal' *vs.* 'ordinal', or 'categorical' *vs.* 'numeric'. Turning to the authority of introductory statistical textbooks (Bluman, 1992; Samuels and Witmer, 1999), there are two parallel ways to classify data into different types. Data can be classified into either qualitative or quantitative. Data can also be classified into different levels of measurement scales.

Qualitative attributes, also often called **categorical** attributes, are attributes that can be placed into distinct categories, according to some characteristics. Some can be arrayed in a meaningful rank order. But no arithmetic operations can be applied to them. Examples are *blood type of a person: A, B, AB, O*; and *tenderness of beef: very tender, tender, slightly tough, tough.* **Quantitative** attributes are numerical in nature. They can be ranked in order. They also can be subjected to meaningful arithmetic operations. Quantitative attributes can be further classified into two groups, discrete or continuous. A **discrete** attribute assumes values that can be counted. The attribute can not assume all values on the number line within its value range. An example is *number of children in a family.* A **continuous** attribute can assume all values on the number line within the value range. The values are obtained by measuring rather than counting. An example is *Fahrenheit temperature scale.*

In addition to being classified as either qualitative or quantitative, attributes can also be classified by how they are categorized, counted or measured. This type of classification uses **measurement scales**, and four common types of scales are used: nominal, ordinal, interval and ratio. The **nominal level of measurement** classifies data into mutually exclusive (non overlapping), exhaustive categories in which no order or ranking can be imposed on the data, such as *blood type of a person.* The **ordinal level of measurement** classifies data into categories that can be ranked; however, the differences between the ranks can not be calculated meaningfully by arithmetic, such as *tenderness of beef.* The **interval level of measurement** ranks data, and the differences between units of measure can be calculated meaningfully by arithmetic. However, *zero* in the interval level of measurement does not mean ‘nil’ or ‘nothing’, such as *Fahrenheit temperature scale.* The **ratio level of measurement** possesses all the characteristics of interval measurement, and there exists a *zero* that means ‘nil’ or ‘nothing’. In consequence, true ratios exist between different units of measure. An example is *number of children in a family.*

We believe that ‘discretization’ is best described as the conversion of *quantitative* attributes to *qualitative* attributes. In consequence, we will address attributes as either qualitative or quantitative throughout this paper.

2.2 Taxonomy

Discretization methods can be classified into either *primary* or *composite.* Primary methods accomplish discretization without referring to any other discretization method. In contrast, composite methods are built on top of a primary method.

Primary methods can be classified as per the following taxonomies.

(1) **Supervised** *vs.* **Unsupervised** (Dougherty et al., 1995). Methods that use the class information of the training instances to select discretization cut points are supervised. Methods that do not use the class information are unsupervised. Supervised discretization can be further characterized as *error-based*, *entropy-based* or *statistics-based* according to whether intervals are selected using metrics based on error on the training data, entropy of the intervals, or some statistical measure.

(2) **Univariate** *vs.* **Multivariate** (Bay, 2000). Methods that discretize each attribute in isolation are univariate. Methods that take into consideration relationships among attributes during discretization are multivariate.

(3) **Split** *vs.* **Merge** (Kerber, 1992) *vs.* **Single-scan**. Split discretization initially has the whole value range as an interval, then continues splitting it into sub-intervals until some threshold is met. Merge discretization initially puts each value into an interval, then continues merging adjacent intervals until some threshold is met. Some discretization methods utilize both split and merge processes. For example, intervals are initially formed by splitting, and then a merge process is performed to post-process the formed intervals. Other methods use neither split nor merge process. Instead, they scan the ordered values only once, sequentially forming the intervals, which we name single-scan.

(4) **Global** *vs.* **Local** (Dougherty et al., 1995). Global methods discretize with respect to the whole training data space. They perform discretization once only, using a single set of intervals throughout a single classification task. Local methods allow different sets of intervals to be formed for a single attribute, each set being applied in a different classification context. For example, different discretizations of a single attribute might be applied at different nodes of a decision tree (Quinlan, 1993).

(5) **Eager** *vs.* **Lazy** (Hsu et al., 2000). Eager methods perform discretization *prior* to classification time. Lazy methods perform discretization during the process of classification.

(6) **Disjoint** *vs.* **Non-disjoint**. Disjoint methods discretize the value range of the attribute under discretization into disjoint intervals. No intervals overlap. Non-disjoint methods discretize the value range into intervals that can overlap.

(7) **Parameterized** *vs.* **Unparameterized**. Parameterized discretization requires input from the user, such as the maximum number of discretized intervals. Unparameterized discretization only uses information from data and does not need input from the user.

A composite method first chooses some primary discretization method to form the initial intervals. It then focuses on how to adjust these initial intervals to achieve certain goals. The taxonomy of a composite method sometimes may depend on the taxonomy of its primary method.

To the best of our knowledge we are the first to propose the taxonomies ‘primary’ *vs.* ‘composite’, ‘disjoint’ *vs.* ‘non-disjoint’ and ‘parameterized’ *vs.* ‘unparameterized’; or to distinguish ‘single-scan’ from ‘split’ and ‘merge’.

3 Naive-Bayes classifiers

In naive-Bayes learning, 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),
- $\mathbf{X} = \mathbf{x}$ as shorthand for $X_1 = x_1 \wedge X_2 = x_2 \wedge \cdots \wedge X_k = x_k$.

Suppose a test instance $\mathbf{x}$ is presented. The learner is asked to predict its class according to the evidence provided by the training data. Expected classification error can be minimized by choosing $argmax_c(p(C = c \,|\, \mathbf{X} = \mathbf{x}))$ for each $\mathbf{x}$. Bayes’ theorem can be used to calculate:

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

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

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

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:

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

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

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

However, (4) is applicable only when X_i is qualitative. A qualitative attribute usually takes a small number of values (Bluman, 1992; Samuels and 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 and Berger, 1990; John and Langley, 1995).

When it is quantitative, X_i usually has a large or even an infinite number of values (Bluman, 1992; Samuels and 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 and Langley, 1995). Hence $p(X_i = x_i \,|\, C = c)$ is completely determined by a probability density function f, which satisfies:

(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 (Scheaffer and McClave, 1995).

When involving quantitative attributes, 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, $\lim\limits_{\Delta \to 0} \frac{p(x_i \leq X_i \leq x_i + \Delta \,|\, C = c)}{\Delta} = f(X_i = x_i \,|\, C = c)$. 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

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

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

$$
\begin{aligned}
& p(C = c \,|\, \mathbf{X} = \mathbf{x}) \propto p(C = c) \prod_{i=1}^{k} G(X_i = x_i \,|\, C = c), \\
& \quad \text{where} \quad G(X_i = x_i \,|\, C = c) \\
& = \begin{cases} p(X_i = x_i \,|\, C = c), & \text{if } X_i \text{ is qualitative;} \\ f(X_i = x_i \,|\, C = c), & \text{if } X_i \text{ is quantitative.} \end{cases}
\end{aligned} \tag{6}
$$

Classifiers using (6) are *naive-Bayes classifiers*. The assumption embodied in (3) is the *attribute independence assumption*.

3.1 Calculating frequency for qualitative data

A typical approach to estimating $p(C = c)$ is to use Laplace-estimate (Cestnik, 1990): $\frac{n_c+k}{N+n\times k}$, where n_c is the number of instances satisfying $C = c$, N is the number of training instances, n is the number of classes, and $k = 1$. A typical approach to estimating $p(X_i = x_i \,|\, C = c)$ is to use M-estimate (Cestnik, 1990): $\frac{n_{ci}+m\times p}{n_c+m}$, where n_{ci} is the number of instances satisfying $X_i = x_i \wedge C = c$, n_c is the number of instances satisfying $C = c$, p is $p(X_i = x_i)$ (estimated by the Laplace-estimate), and $m = 2$.

3.2 Probability density estimation for quantitative data

The density function f gives a description of the distribution of X_i within the class c, and allows probabilities associated with $X_i|C = c$ to be found (Silverman, 1986). Unfortunately however, f is usually unknown for real-world data. In consequence, *probability density estimation* is used to construct $\hat{f}$, an estimate of f from the training data.

A conventional approach to constructing $\hat{f}$ is to assume that the values of X_i within the class c are drawn from a normal (Gaussian) distribution (Dougherty et al., 1995; Mitchell, 1997). Thus $\hat{f} = N(X_i; \mu_c, \sigma_c) = \frac{1}{\sqrt{2\pi}\sigma_c} e^{-\frac{(X_i-\mu_c)^2}{2\sigma_c^2}}$, where μ_c is the *mean* and σ_c is the *standard deviation* of the attribute values from the training instances whose class equals c. In this case, training involves learning the parameters μ_c and σ_c from the training data. The normal distribution assumption is made because it may provide a reasonable approximation to many real-world distributions (John and Langley, 1995), or because it is perhaps the most well-studied probability distribution in statistics (Mitchell, 1997). This approach is *parametric*, that is, it assumes that the data are drawn from one of a known parametric family of distributions (Silverman, 1986). The major problem of this method is that when the attribute data do not follow a normal distribution, which is often the case in real-world data, the probability estimation of naive-Bayes classifiers is not reliable and thus can lead to inferior classification performance (Dougherty et al., 1995; Pazzani, 1995).

A second approach is *less parametric* in that it does not constrain $\hat{f}$ to fall in a given parametric family. Thus less rigid assumptions are made about the distribution of the observed data (Silverman, 1986). A typical approach is kernel density estimation (John and Langley, 1995). $\hat{f}$ is averaged over a large set of Gaussian kernels, $\hat{f} = \frac{1}{n_c} \sum\limits_i N(X_i; \mu_i, \sigma_c)$, where n_c is the total number of training instances with class c, μ_i is the ith value of X_i within class c, and $\sigma_c = \frac{1}{\sqrt{n_c}}$. It has been demonstrated that kernel density estimation results in higher naive-Bayes classification accuracy than the former method in domains that violate the normal distribution assumption, and causes only small decreases in accuracy in domains where the assumption holds. However, this approach tends to incur high computational memory and time. Whereas the former method can estimate μ_c and σ_c by storing only the sum of the observed x_i's and the sum of their squares, this approach must store every x_i. Whereas the former method only has to calculate $N(X_i)$ once for each $X_i = x_i \,|\, C = c$, this approach must perform this calculation n_c times. Thus it has a potential problem that undermines the efficiency of naive-Bayes learning.

3.3 Merits of naive-Bayes classifiers

Naive-Bayes classifiers are simple and efficient. They need only to collect information about individual attributes, which contrasts to most learning systems that must consider attribute combinations. Thus naive-Bayes' computational time complexity is only linear with respect to the amount of the training data. This is much more efficient than the exponential complexity of non-naive Bayes approaches (Yang and Liu, 1999; Zhang et al., 2000). They are also space efficient. They do not require training data be retained in memory during classification and can record all required information using only tables of no more than two dimensions.

Naive-Bayes classifiers are effective. They are optimal methods of supervised learning if the independence assumption holds and the estimates of the required probabilities are accurate. Even when the independence assumption is violated, their classification performance is still surprisingly good compared with other more sophisticated classifiers. One reason for this is because that the classification estimation under zero-one loss is only a function of the sign of the probability estimation. In consequence, the classification accuracy can remain high even while the probability estimation is poor (Domingos and Pazzani, 1997).

Naive-Bayes classifiers are robust to noisy data such as irrelevant attributes. They take all attributes into account simultaneously. Hence, the impact of a misleading attribute can be absorbed by other attributes under zero-one loss (Hsu et al., 2000).

Naive-Bayes classifiers support incremental training (Rennie, 2000; Roy and McCallum, 2001; Zaffalon and Hutter, 2002). One can map an existing naive-Bayes classifier and a new training instance to a new naive-Bayes classifier which is identical to the classifier that would have been learned from the original data augmented by the new instance. Thus it is trivial to update a naive-Bayes classifier whenever a new training instance becomes available. This contrasts to the non-incremental methods which must build a new classifier from scratch in order to utilize new training data. The cost of incremental update is far lower than that of retraining. Consequently, naive-Bayes classifiers are attractive for classification tasks where the training information updates frequently.

4 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 and 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 and Witmer, 1999), discretization might suffer information loss.

4.1 Why discretization can be effective

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 a further analysis on this issue, based on an assumption that each X_i^* has a Dirichlet prior. Their analysis focused on the density function f, and suggested that discretization would achieve optimal effectiveness by forming x_i^* for x_i such that $p(X_i^* = x_i^* \,|\, C = c)$ simulated 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, as we will prove in Theorem 1, we believe that discretization for naive-Bayes learning should focus on the accuracy of $p(C = c \,|\, X_i^* = x_i^*)$ as an estimate of $p(C = c \,|\, X_i = x_i)$; and 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 $(1 \leq i \leq l)$. If $\forall i (p(C = c \,|\, X_i = x_i) = p(C = c \,|\, X_i^* = x_i^*))$, 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:

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

using Bayes theorem:

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

[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.

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

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

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:

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

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:

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

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 any form of the probability density.

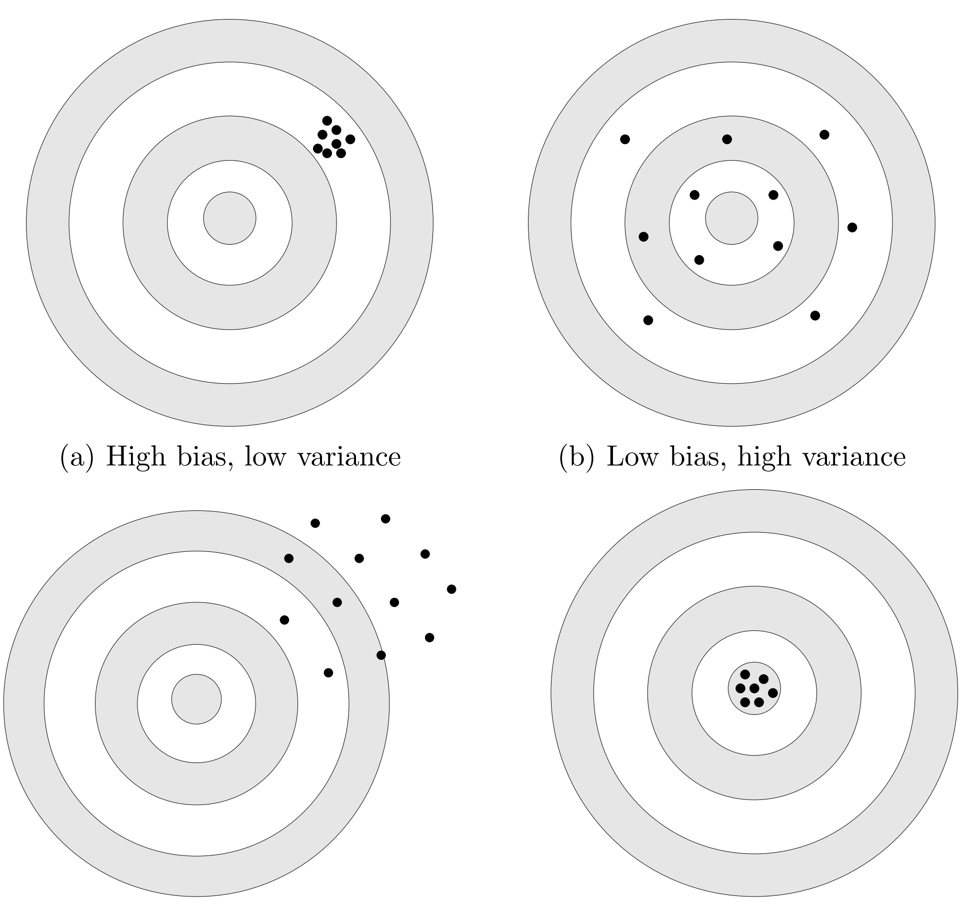
Our analysis that focuses on $p(C = c \,|\, X_i = x_i)$ instead of $f(X_i = x_i \,|\, C = c)$, is derived from Kononenko's 1992. However, Kononenko's analysis required that the attributes be assumed *unconditionally* independent of each other, which entitles $\prod_{i=1}^{k} p(X_i = x_i) = p(\mathbf{X} = \mathbf{x})$. 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 in naive-Bayes learning.

4.2 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 pre-processed 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 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, here we assume that each class has the same prior probability. Thus we can cancel the effect of $p(C = c)$. However, our analysis extends straightforwardly to non-uniform cases.

4.2.1 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 and Dietterich, 1995; Breiman, 1996; Kohavi and 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



(a) High bias, low variance

(b) Low bias, high variance

(c) High bias, high variance

(d) Low bias, low variance

Fig. 1. Bias and variance in shooting arrows at a target. Bias means that the archer systematically misses in the same direction. Variance means that the arrows are scattered.

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 and 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.

Moore and McCabe (2002) illustrated bias and variance through shooting arrows at a target, as reproduced in Figure 1. We can think of the perfect model as the bull's-eye on a target, and the algorithm learning from some set of training data as an arrow fired at the bull's-eye. Bias and variance describe what happens when an archer fires many arrows at the target. Bias means that the aim is off and the arrows land consistently off the bull's-eye in the same direction. The learned model does not center about the perfect model. Large variance means that repeated shots are widely scattered on the target. They do not give similar results but differ widely among themselves. A good

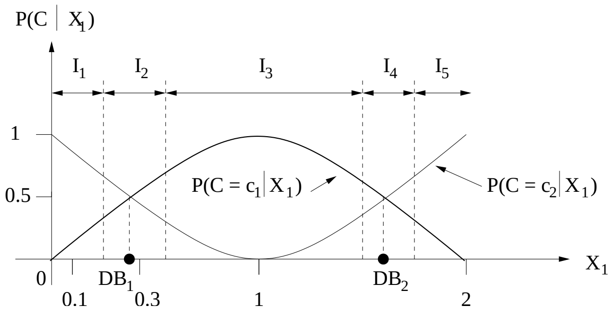


Fig. 2. Probability distribution in one-attribute problem

learning scheme, like a good archer, must have both low bias and low variance.

4.2.2 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, we are interested in the probability $p(C = c \,|\, X_i = x_i)$. Besides, Hsu et al.'s analysis only addressed one-attribute classification problems, [2] and only 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, since the final choice of the class is decided by the product of each attribute's probability in the later situation. 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 2. The consequent decision boundaries are labelled as DB_1 and DB_2 respectively in Figure 2. 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 2. 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. [3] But

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

[3] 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.

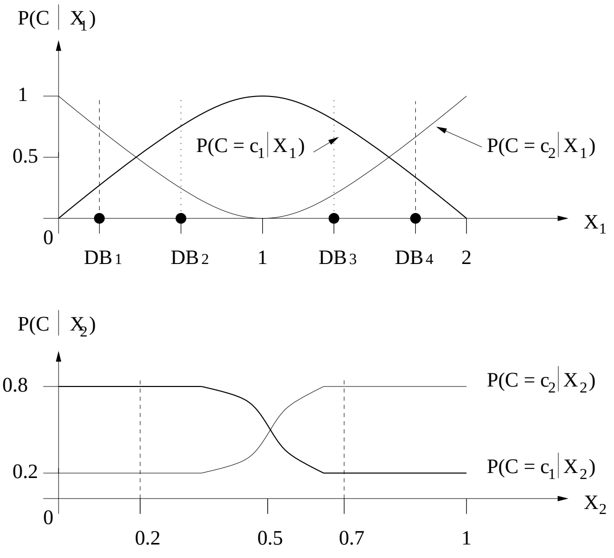


Fig. 3. Probability distribution in two-attribute problem

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, the locations of decision boundaries of the attribute X_1 depend on the distribution of $p(C \,|\, X_1)$ 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 $p(C \,|\, X_1)$, 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 3, 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 2. 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 $p(C = c_1 \,|\, X_2 = 0.2) = 0.8 > p(C = c_2 \,|\, X_2 = 0.2) = 0.2$, and $p(C = c \,|\, \mathbf{x}) \propto \prod_{i=1}^{2} p(C = c \,|\, X_i = x_i)$ for each class c according to Theorem 1, $p(C = c_1 \,|\, \mathbf{x})$ does not equal $p(C = c_2 \,|\, \mathbf{x})$ when X_1 falls on any of the single attribute decision boundaries as presented in Figure 2. Instead X_1's decision boundaries change to be DB_1 and DB_4 as in Figure 3. 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 3. 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 $p(C \,|\, X_1)$. Ironically, if we have already found $p(C \,|\, X_1)$, we can resolve the classification task directly; thus there is no need to consider discretization at all. Without knowing $p(C \,|\, X_1)$, 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 $p(C \,|\, X_1)$ 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 $p(C = c \,|\, X_1 = x_1)$ by $p(C = c \,|\, X_1^* = x_1^*)$, how much error can be tolerated without altering the classification. This motivates our following analysis.

4.2.3 Error tolerance of probability estimation

To investigate this issue, we return to our example depicted in Figure 2. 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 $p(C = c_1 \,|\, \mathbf{x}) = p(C = c_1 \,|\, X_1 = 0.1) = 0.19$ and $p(C = c_2 \,|\, \mathbf{x}) = p(C = c_2 \,|\, X_1 = 0.1) = 0.81$. According to naive-Bayes learning, as long as $p(C = c_2 \,|\, X_1 = 0.1) > 0.50$, 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 $p(C = c_1 \mid \mathbf{x}) = p(C = c_1 \mid X_1 = 0.3) = 0.51$ and $p(C = c_2 \mid \mathbf{x}) = p(C = c_2 \mid X_1 = 0.3) = 0.49$. The error tolerance of estimating $p(C \mid X_1 = 0.3)$ 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 $p(C = c \mid X_i = x_i)$ 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.

4.3 Summary

By this line of reasoning, optimal discretization can only be performed if the probability distribution of $p(C = c \mid X_i = x_i)$ 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. On the other hand, however, we want to ensure that the intervals are sufficiently large to minimize the risk that the error of estimating $p(C = c \mid X_i^* = x_i^*)$ 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.

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.

5 Review of discretization methods

Here we review six key discretization methods, each of which was either designed especially for naive-Bayes classifiers or is in practice often used for naive-Bayes classifiers. We are particularly interested in analyzing each method's discretization bias and variance, which we believe illuminating.

5.1 Equal width discretization & Equal frequency discretization

Equal width discretization (EWD) (Catlett, 1991; Kerber, 1992; Dougherty et al., 1995) divides the number line between v_{min} and v_{max} into k intervals of equal width, where k is a user predefined parameter. Thus the intervals have width $w = (v_{max} - v_{min})/k$ and the cut points are at $v_{min} + w, v_{min} + 2w, \cdots, v_{min} + (k-1)w$.

Equal frequency discretization (EFD) (Catlett, 1991; Kerber, 1992; Dougherty et al., 1995) divides the sorted values into k intervals so that each interval contains approximately the same number of training instances, where k is a user predefined parameter. Thus each interval contains n/k training instances with adjacent (possibly identical) values. Note that training instances with identical values must be placed in the same interval. In consequence it is not always possible to generate k equal frequency intervals.

Both EWD and EFD are often used for naive-Bayes classifiers because of their simplicity and reasonably good performance (Hsu et al., 2000). However both EWD and EFD fix the number of intervals to be produced (decided by the parameter k). When the training data size is very small, intervals will have small size and thus tend to incur high variance. When the training data size is very large, intervals will have large size and thus tend to incur very high bias. Thus we anticipate that they control neither discretization bias nor discretization variance well.

5.2 Fuzzy learning discretization

Fuzzy learning discretization[4] (FLD) (Kononenko, 1992, 1993) initially discretizes X_i into k equal width intervals $(a_i, b_i]$ $(1 \leq i \leq k)$ using EWD, where k is a user predefined parameter. For each discretized value x_i^* corresponding to $(a_i, b_i]$, FLD estimates $p(X_i^* = x_i^* \,|\, C = c)$ from all training instances rather than from instances that have values of X_i in $(a_i, b_i]$. The influence of a training instance with value v of X_i on $(a_i, b_i]$ is assumed to be normally distributed with the mean value equal to v and is proportional to $P(v, \sigma, i) = \int_{a_i}^{b_i} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-v}{\sigma})^2} \mathrm{d}x$. The parameter σ is used to control the 'fuzziness' of the interval bounds, and is set to σ to $0.7 \times \frac{v_{max}-v_{min}}{k}$ because this setting achieved the best results in Kononenko's experiments (Kononenko, 1992). Suppose there are n_c training instances with known values for X_i and with class c, each with influence $P(v_j, \sigma, i)$ on $(a_i, b_i]$ $(j = 1, \cdots, n_c)$:$p(X_i^* = x_i^* \,|\, C = c) = \frac{p(a_i < X_i \leq b_i \wedge C = c)}{p(C=c)} \approx \frac{\frac{\sum_{j=1}^{n_c} P(v_j, \sigma, i)}{n}}{p(C=c)}$.

The idea behind fuzzy discretization is that small variation of the value of a quantitative attribute should have small effects on the attribute's probabilities, whereas under non-fuzzy discretization, a slight difference between two values, such that one is above and one is below the cut point, can have drastic effects on the estimated probabilities. But when the training instances' influence on each interval does not follow the normal distribution, FLD's performance can degrade. In terms of discretization bias and variance, FLD's effect is similar to its primary discretization method that forms the initial intervals, such as EWD.

5.3 Entropy minimization discretization

Entropy minimization discretization (EMD) (Fayyad and Irani, 1993) evaluates as a candidate cut point the midpoint between each successive pair of the sorted values. For evaluating each candidate cut point, the data are discretized into two intervals and the resulting class information entropy is calculated. A binary discretization is determined by selecting the cut point for which the entropy is minimal amongst all candidates. The binary discretization is applied recursively, always selecting the best cut point. A minimum description length criterion (MDL) is applied to decide when to stop discretization.

[4] This is one of the three versions of fuzzy discretization proposed by Kononenko (1992; 1993) for naive-Bayes classifiers. Because of space limits, we present here only the version that, according to our experiments, best reduces naive-Bayes classification error.

Although it has demonstrated strong performance for naive-Bayes (Dougherty et al., 1995; Perner and Trautzsch, 1998), EMD was developed in the context of top-down induction of decision trees. It uses MDL as the termination condition. According to An and Cercone (1999), this has an effect that tends to form qualitative attributes with few values. We thus anticipate that EMD focuses on reducing discretization variance, but does not control bias so successfully. This might work well for training data of small size, for which it is credible that variance reduction can contribute more to lower naive-Bayes learning error than bias reduction (Friedman, 1997). However, when training data size is large, it is very likely that the loss through bias increase will soon overshadow the gain through variance reduction, resulting in inferior learning performance. Besides since EMD discretizes a quantitative attribute by calculating the class information entropy as if the naive-Bayes classifiers only use that *single* attribute after discretization, EMD might be effective at identifying decision boundaries in the one-attribute learning context. But in the multi-attribute learning context, the resulting cut points can easily diverge from the true ones as we have explained in Section 4.2. If this happens, we anticipate that EMD controls neither bias nor variance well.

5.4 Iterative-improvement discretization

Iterative-improvement discretization (IID) (Pazzani, 1995) initially forms a set of intervals using EWD or EMD, and then iteratively adjusts the intervals to minimize naive-Bayes classification error on the training data. It defines two operators: merge two contiguous intervals, and split an interval into two intervals by introducing a new cut point that is midway between each pair of contiguous values in that interval. In each loop of the iteration, for each quantitative attribute, IID applies all operators in all possible ways to the current set of intervals and estimates the classification error of each adjustment using leave-one-out cross validation. The adjustment with the lowest error is retained. The loop stops when no adjustment further reduces the error.

A disadvantage of IID for naive-Bayes learning results from its iterative nature. When the training data size is large, the possible adjustments by applying the two operators will tend to be numerous. Consequently the repetitions of the leave-one-out cross validation will be prohibitive so that IID is infeasible in terms of computation time. This will impede IID from feasible implementation for classification tasks with large training data.

IID can split as well as merge discretized intervals. How many intervals will be formed and where the cut points are located are decided by the error of the cross validation. This is a case by case problem and thus it is not clear what IID's systematic impact on bias and variance is. However, if cross-validation is successful in finding a model that minimizing error, then we might infer that IID can obtain a good discretization bias-variance trade-off.

5.5 Lazy discretization

Lazy discretization (LD) (Hsu et al., 2000) defers discretization until classification time. It waits until a test instance is presented to determine the cut points and then estimates probabilities for each quantitative attribute of the test instance. For each quantitative value from the test instance, it selects a pair of cut points such that the value is in the middle of its corresponding interval and the interval size is equal to that produced by some other algorithm chosen by the user. In Hsu et al.'s implementation, the interval size is the same as created by EFD with $k = 10$. However, as already noted, 10 is an arbitrary value.

LD tends to have high memory and computational requirements because of its lazy methodology. Eager approaches carry out discretization at training time. Thus the training instances can be discarded before classification time. In contrast, LD needs to keep the training instances for use during classification time. This demands high memory when the training data size is large. Further, where a large number of instances need to be classified, LD will incur large computational overheads since it must estimate probabilities from the training data for each instance individually. Although LD achieves comparable accuracy to EFD and EMD (Hsu et al., 2000), the high memory and computational overheads have a potential to damage naive-Bayes classifiers' classification efficiency. We anticipate that LD's behavior on discretization bias and variance will mainly depend on the size strategy of its primary method, such as EFD or EMD.

5.6 Summary

We summarize in Table 1 each previous discretization method in terms of the taxonomy in Section 2.2.

6 Managing discretization bias and variance

We have argued that the interval size (the number of training instances in an interval) and interval number (the number of intervals) formed by a discretization method can affect the method's discretization bias and variance. Also, a

Table 1
Discretization Methods Review

Method	Taxonomy
EWD	primary,unsupervised,univariate,single-scan,global,eager,disjoint,parameterized
EFD	primary,unsupervised,univariate,single-scan,global,eager,disjoint,parameterized
FLD	composite,univariate,global,eager,non-disjoint
EMD	primary,supervised,univariate,split,global,eager,disjoint,unparameterized
IID	composite,supervised,multivariate,split&merge,global,eager,disjoint
LD	composite,univariate,global,lazy,non-disjoint

number of previous authors have mentioned that the interval number has a major effect on the naive-Bayes classification error (Pazzani, 1995; Torgo and Gama, 1997; Gama et al., 1998; Hussain et al., 1999; Mora et al., 2000; Hsu et al., 2000).. Thus we anticipate that one way to manage discretization bias and variance is to adjust interval size and interval number.

However, many previous discretization methods, according to Section 5, are sub-optimal in terms of manipulating interval size and interval number, thus are sub-optimal at managing discretization bias and variance. One weakness of some methods is that they produce interval numbers without reference to the training data size. However, it has been observed by previous authors that there is a link between the ideal interval number and the training data size (Torgo and Gama, 1997; Hsu et al., 2000). That is, as the number of discretized interval increases, the classification accuracy will improve and reach a plateau. When the interval number becomes very large, the accuracy will drop gradually. How fast the accuracy drops will depend on the size of the training data. The smaller the training data size, the earlier and faster the accuracy drops (Hsu et al., 2000). We also have argued that when the true distribution of $p(C \mid X_i)$, and thus the true decision boundaries for X_i, are unknown, it is advisable to form as many intervals as constraints on adequate probability estimation accuracy allow. Consequently it is not appropriate to fix or minimize the interval number. Instead, discretization techniques that react more sensitively to training data size can be of greater utility. Another usual weakness, as a consequence of the first one, is that when the interval size is sufficiently large, and thus the discretization variance is under control, the increase of the training data will mainly contribute to increasing discretization bias. As a result, it is credible that as the amount of data increases, learning error will also increase. This contradicts our normal expectation that the more data we have, the better we learn; and is of particular disadvantage since large data are increasingly common. Concerned by these weaknesses, we propose two new heuristic discretization techniques, *proportional k-interval discretization* and *equal size discretization*. To the best of our knowledge, these are the first techniques that explicitly manage discretization bias and variance by tuning interval size and interval number.

6.1 Proportional k-interval discretization

According to our analysis in Section 4.2, increasing interval size (decreasing interval number) will decrease variance but increase bias. Conversely, decreasing interval size (increasing interval number) will decrease bias but increase variance. PKID aims to resolve this conflict by setting interval size equal to interval number, and setting both proportional to the number of training instances. When discretizing a quantitative attribute for which there are n training instances with known values, supposing the desired interval size is s and the desired interval number is t, PKID employs (7) to calculate s and t. It then sorts the quantitative values in ascending order and discretizes them into intervals of size s. Thus each interval contains approximately s training instances with adjacent (possibly identical) values.

$$\begin{aligned} s \times t &= n \\ s &= t. \end{aligned} \tag{7}$$

By setting interval size and interval number equal, PKID equally weighs discretization bias and variance. By setting them proportional to the training data size, PKID can use an increase in training data to lower both discretization bias and variance. 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. This means that PKID has greater capacity to take advantage of the additional information inherent in large volumes of training data than previous methods.

6.2 Equal size discretization

An alternative approach to managing discretization bias and variance is *equal size discretization* (ESD). ESD first controls variance. It then uses additional data to decrease bias. To discretize a quantitative attribute, ESD sets a *safe interval size m*. Then it discretizes the ascendingly sorted values into intervals of size m. Thus each interval has approximately the same number m of training instances with adjacent (possibly identical) values.

By introducing m, ESD aims to ensure that in general 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. As we have explained in Section 4.2, the optimal interval size varies from instance to instance and from domain to domain. Nonetheless, we have to choose a size so that we can implement and evaluate ESD. In our study, we choose the size as 30 since it is commonly held to be the minimum sample size from which one should draw statistical inferences (Weiss, 2002).

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. Thus intervals of high bias are not associated with large datasets any more. In this way, ESD can prevent both high bias and high variance.

According to our taxonomy in Section 2.2, both PKID and ESD are unsupervised, univariate, single-scan, global, eager and disjoint primary discretizations. PKID is unparameterized, while ESD is parameterized.

7 Time complexity comparison

Here we calculate the time complexities of our new discretization methods as well as the previous ones discussed in Section 5. Naive-Bayes classifiers are very attractive to applications with large data because of their computational efficiency. Thus it will often be important that the discretization methods are efficient so that they can scale to large data.

To discretize a quantitative attribute, suppose the number of training instances,[5] test instances, attributes and classes are n, l, v and m respectively.

- EWD, EFD, FLD, PKID and ESD are dominated by sorting. Their complexities are of order $O(n \log n)$.
- EMD does sorting first, an operation of complexity $O(n \log n)$. It then goes through all the training instances a maximum of $\log n$ times, recursively applying 'binary division' to find out at most $n-1$ cut points. Each time, it will estimate $n-1$ candidate cut points. For each candidate point, probabilities of each of m classes are estimated. The complexity of that operation is $O(mn \log n)$, which dominates the complexity of the sorting, resulting in complexity of order $O(mn \log n)$.
- IID's operators have $O(n)$ possible ways to adjust cut points in each iteration. For each adjustment, the leave-one-out cross validation has complexity of order $O(nmv)$. The number of times that the iteration will repeat depends on the initial discretization as well the error estimation. It varies from case

[5] We only consider instances with known value of the quantitative attribute.

to case, which we denote by u here. Thus the complexity of IID is of order $O(n^2mvu)$.

- LD sorts the attribute values once and performs discretization separately for each test instance and hence its complexity is $O(n \log n) + O(nl)$.

Thus EWD, EFD, FLD, PKID and ESD have complexity lower than EMD. LD tends to have high complexity when the training or testing data size is large. IID's complexity is prohibitively high when the training data size is large.

8 Experimental evaluation

We evaluate whether PKID and ESD can better reduce naive-Bayes classification error by better managing discretization bias and variance, compared with previous discretization methods, EWD, EFD, FLD, EMD and LD. [6] Since IID tends to have high computational complexity, while our experiments include large training data (up to 166 quantitative attributes and up to half million training instances), we do not implement it for the sake of feasibility.

8.1 Data

We ran our experiments on 29 natural datasets from UCI machine learning repository (Blake and Merz, 1998) and KDD archive (Bay, 1999). This experimental suite comprises 3 parts. The first part is composed of all the UCI datasets used by Fayyad and Irani (1993) when publishing the entropy minimization heuristic for discretization. The second part is composed of all the UCI datasets with quantitative attributes used by Domingos and Pazzani (1996) for studying naive-Bayes classification. In addition, as discretization bias and variance responds to the training data size and the first two parts are mainly confined to small size, we further augment this collection with datasets that we can identify containing numeric attributes, with emphasis on those having more than 5000 instances. Table 2 describes each dataset, including the number of instances (Size), quantitative attributes (Qa.), qualitative attributes (Ql.) and classes (C.). The datasets are ordered increasingly by size.

[6] EWD, EFD and FLD are implemented with the parameter $k = 10$. LD is implemented with the interval size equal to that produced by EFD with $k = 10$.

Table 2
Experimental Datasets and Classification Error

Dataset	**Size**	**Qa.**	**Ql.**	**C.**	**Classification Error %**						
					EWD	EFD	FLD	EMD	LD	PKID	ESD
Labor-Negotiations	57	8	8	2	12.3	8.9	12.8	9.5	9.6	7.4	9.3
Echocardiogram	74	5	1	2	29.6	30.0	26.5	23.8	29.1	25.7	25.7
Iris	150	4	0	3	5.7	7.7	5.4	6.8	6.7	6.4	7.1
Hepatitis	155	6	13	2	14.3	14.2	14.3	13.9	13.7	14.1	15.7
Wine-Recognition	178	13	0	3	3.3	2.4	3.2	2.6	2.9	2.4	2.8
Sonar	208	60	0	2	25.6	25.1	25.8	25.5	25.8	25.7	23.3
Glass-Identification	214	9	0	6	39.3	33.7	40.7	34.9	32.0	32.6	39.1
Heart-Disease(Cleveland)	270	7	6	2	18.3	16.9	15.8	17.5	17.6	17.4	16.9
Liver-Disorders	345	6	0	2	37.1	36.4	37.6	37.4	37.0	38.9	36.5
Ionosphere	351	34	0	2	9.4	10.3	8.7	11.1	10.8	10.4	10.7
Horse-Colic	368	7	14	2	20.5	20.8	20.8	20.7	20.8	20.3	20.6
Credit-Screening(Australia)	690	6	9	2	15.6	14.5	15.2	14.5	13.9	14.4	14.2
Breast-Cancer(Wisconsin)	699	9	0	2	2.5	2.6	2.8	2.7	2.6	2.7	2.6
Pima-Indians-Diabetes	768	8	0	2	24.9	25.6	25.2	26.0	25.4	26.0	26.5
Vehicle	846	18	0	4	38.7	38.8	42.4	38.9	38.1	38.1	38.3
Annealing	898	6	32	6	3.8	2.4	3.7	2.1	2.3	2.1	2.3
German	1000	7	13	2	25.1	25.2	25.2	25.0	25.1	24.7	25.4
Multiple-Features	2000	3	3	10	31.0	31.8	30.9	32.9	31.0	31.2	31.7
Hypothyroid	3163	7	18	2	3.6	2.8	2.7	1.7	2.4	1.8	1.8
Satimage	6435	36	0	6	18.8	18.8	20.2	18.1	18.4	17.8	17.7
Musk	6598	166	0	2	13.7	18.4	23.0	9.4	15.4	8.2	6.9
Pioneer-MobileRobot	9150	29	7	57	13.5	15.0	21.2	19.3	15.3	4.6	3.2
Handwritten-Digits	10992	16	0	10	12.5	13.2	13.3	13.5	12.8	12.0	12.5
Australian-Sign-Language	12546	8	0	3	38.3	37.7	38.7	36.5	36.4	35.8	36.0
Letter-Recognition	20000	16	0	26	29.5	29.8	34.7	30.4	27.9	25.7	25.5
Adult	48842	6	8	2	18.2	18.6	18.5	17.3	18.1	17.1	16.2
Ipums-la-99	88443	20	40	13	21.0	21.1	37.8	21.3	20.4	20.6	18.4
Census-Income	299285	8	33	2	24.5	24.5	24.7	23.6	24.6	23.3	20.0
Forest-Covertype	581012	10	44	7	32.4	33.0	32.2	32.1	32.3	31.7	31.9
ME	-	-	-	-	20.1	20.0	21.5	19.6	19.6	18.6	18.6
GE	-	-	-	-	16.0	15.6	16.8	15.0	15.3	13.8	13.7

8.2 Design

To evaluate a discretization method, for each dataset, we implement naive-Bayes learning by conducting a 10-trial, 3-fold cross validation. For each fold, the training data are discretized by this method. The intervals so formed are applied to the test data. The experimental results are recorded as follows.

- **Classification error**, listed under the column 'Error' in Table 2, is the percentage of incorrect classifications of naive-Bayes classifiers in the test averaged across all folds of the cross validation.
- **Classification bias and variance**, listed under the columns 'Bias' and 'Variance' respectively in Table 3, are estimated by the method described by Webb (2000). They equate to the bias and variance defined by Breiman (1996), except that irreducible error is aggregated into bias and variance. An instance is classified once in each trial and hence ten times in all. The central tendency of the learning algorithm is the most frequent classification of an instance. Total error is the proportional of classifications across the 10 trials that are incorrect. Bias is that portion of the total error that is due to errors committed by the central tendency of the learning algorithm. This is the portion of classifications that are both incorrect and equal to the central tendency. Variance is that portion of the total error that is due to errors that are deviations from the central tendency of the learning algorithm. This is the portion of classifications that are both incorrect and unequal to the central tendency. Bias and variance sum to the total error.

8.3 Statistics

Three statistics are employed to evaluate the experimental results.

- **Mean error** is the arithmetic mean of the errors across all datasets. It provides a gross indication of the relative performance of competitive methods. It is debatable whether errors in different datasets are commensurable, and hence whether averaging errors across datasets is very meaningful. Nonetheless, a low average error is indicative of a tendency towards low errors for individual datasets. For each discretization method, the 'ME' row in each table presents the arithmetic means of the classification error, bias and variance respectively.
- **Geometric mean error** is the geometric mean of the errors across all datasets. Webb (2000) suggested that geometric mean error ratio is a more useful statistic than mean error ratio across datasets. Geometric mean error functions the same as geometric mean error ratio to indicate the relative performance of two methods. For two methods X and Y, the ratio of their geometric mean errors is their geometric mean error ratio. That X's geometric mean error is smaller than Y's is equivalent to that the geometric mean error ratio of X against Y is smaller than 1, and vice versa. For each discretization method, the 'GE' row of each table presents the geometric means of the classification error, bias and variance respectively.
- **Win/lose/tie record** comprises three values that are respectively the number of datasets for which the naive-Bayes classifier trained with one discretization method obtains lower, higher or equal classification error,

Table 3
Classification Bias and Variance

Dataset	Size	Classification Bias %							Classification Variance %						
		EWD	EFD	FLD	EMD	LD	PKID	ESD	EWD	EFD	FLD	EMD	LD	PKID	ESD
Labor-Negotiations	57	7.7	5.4	9.6	6.7	6.3	5.1	6.1	4.6	3.5	3.2	2.8	3.3	2.3	3.2
Echocardiogram	74	22.7	22.3	22.0	19.9	22.3	22.4	19.7	6.9	7.7	4.5	3.9	6.8	3.2	5.9
Iris	150	4.2	5.6	4.0	5.0	4.8	4.3	6.2	1.5	2.1	1.4	1.8	1.9	2.1	0.9
Hepatitis	155	13.1	12.2	13.5	11.7	11.8	11.0	14.5	1.2	2.0	0.8	2.2	1.9	3.1	1.2
Wine-Recognition	178	2.4	1.7	2.2	2.0	2.0	1.7	2.1	1.0	0.7	1.0	0.6	0.9	0.7	0.7
Sonar	208	20.6	19.9	21.0	20.0	20.6	19.9	19.5	5.0	5.2	4.9	5.5	5.2	5.8	3.8
Glass-Identification	214	24.6	21.1	29.0	24.5	21.8	19.8	25.9	14.7	12.6	11..7	10.3	10.2	12.8	13.2
Heart-Disease(Cleveland)	270	15.6	14.9	13.9	15.7	16.1	15.5	15.6	2.7	2.0	1.9	1.8	1.5	2.0	1.3
Liver-Disorders	345	27.6	27.5	30.2	25.7	29.6	28.6	27.7	9.5	8.9	7.4	11.7	7.3	10.3	8.8
Ionosphere	351	8.7	9.6	8.2	10.4	10.4	9.3	8.8	0.7	0.7	0.5	0.7	0.5	1.2	1.9
Horse-Colic	368	18.8	19.6	19.3	18.9	19.2	18.5	19.1	1.7	1.2	1.5	1.7	1.6	1..8	1.5
Credit-Screening(Australia)	690	14.0	12.8	13.8	12.6	12.6	12.2	12.9	1.6	1..7	1.5	1.9	1.3	2.1	1.3
Breast-Cancer(Wisconsin)	699	2.4	2.5	2.7	2.5	2.5	2.5	2.4	0.1	0.1	0.1	0.1	0.1	0.1	0.2
Pima-Indians-Diabetes	768	21.5	22.3	23.4	21.2	22.8	21.7	23.0	3.4	3.3	1.8	4.7	2.6	4.3	3.5
Vehicle	846	31.9	31.9	36.0	32.2	32.4	31.8	32.2	6.9	7.0	6.4	6.7	5.7	6.3	6..1
Annealing	898	2.9	1.9	3.2	1.7	1.7	1.6	1.8	0.8	0.5	0.4	0.4	0.6	0.6	0.5
German	1000	21.9	22.1	22.3	21.2	22.3	21.0	21.8	3.1	3.1	2.9	3.8	2.9	3.7	3..6
Multiple-Features	2000	27.6	27.9	27.5	28.6	27.9	27.2	27.3	3.4	3.9	3.4	4.3	3.1	4.0	4.4
Hypothyroid	3163	2.7	2.5	2.5	1.5	2.2	1.5	1.5	0.8	0.3	0.2	0.3	0.2	0.3	0.3
Satimage	6435	18.0	18.3	19.4	17.0	18.0	17.1	16.9	0.8	0.6	0.7	1.1	0.4	0.7	0.8
Musk	6598	13.1	16.9	22.4	8.5	14.6	7.6	6.2	0.7	1.5	0.6	0.9	0.8	0.7	0.6
Pioneer-MobileRobot	9150	11.0	11.8	18.0	16.1	12.9	2.8	1.6	2.5	3.2	3.2	3.2	2.4	1.9	1.7
Handwritten-Digits	10992	12.0	12.3	12.9	12.1	12.1	10.7	10.5	0.5	0.9	0.4	1.4	0.6	1.4	2.0
Australian-Sign-Language	12546	35.8	36.3	36.9	34.0	35.4	34.0	34.1	2.5	1.4	1.8	2.5	1.0	1.8	2.0
Letter-Recognition	20000	23.9	26.5	30.3	26.2	24.7	22.5	22.2	5.5	3.3	4.5	4.2	3.2	3.2	3.3
Adult	48842	18.0	18.3	18.3	16.8	17.9	16.6	15.2	0.2	0.3	0.2	0.5	0.2	0.5	1..0
Ipums-la-99	88443	16.9	17.2	25.1	16.9	16.9	15.9	13.5	4.1	4.0	12.7	4.1	3.5	4.7	4.9
Census-Income	299285	24.4	24.3	24.6	23.3	24.4	23.1	18.9	0.2	0.2	0.2	0.2	0.2	0.2	1.1
Forest-Covertype	581012	32.0	32.5	31.9	31.1	32.0	30.3	29.6	0.4	0.5	0.3	1..0	0.3	1.4	2.3
ME	-	17.1	17.2	18.8	16.7	17.2	15.7	15.8	3.0	2.8	2.8	2.9	2.4	2.9	2.8
GE	-	13.5	13.2	14.6	12.7	13.2	11.4	11.4	1.7	1.6	1.4	1.7	1.3	1.7	1.8

compared with the naive-Bayes classifier trained with another discretization method. A one-tailed sign test can be applied to each record. If the test result is significantly low (here we use the 0.05 critical level), it is reasonable to conclude that the outcome is unlikely to be obtained by chance and hence the record of wins to losses represents a systematic underlying advantage to the winning discretization method with respect to the type of datasets studied. Table 4 presents the records on classification error of PKID and ESD respectively compared with all the previous discretization methods.

Table 4
Win/lose/tie records of classification error

Methods	**PKID**				**ESD**			
	Win	**Lose**	**Tie**	**Sign Test**	**Win**	**Lose**	**Tie**	**Sign Test**
vs. **EWD**	22	7	0	<0.01	20	8	1	0.02
vs. **EFD**	22	6	1	<0.01	19	8	2	0.03
vs. **FLD**	23	6	0	<0.01	22	7	0	<0.01
vs. **EMD**	21	5	3	<0.01	20	9	0	0.03
vs. **LD**	20	8	1	0.02	19	8	2	0.03

8.4 Result analysis

According to Table 2, 3 and 4, we have following observations and analysis.

- Both PKID and ESD achieve lower mean and geometric mean of classification error than all previous methods. With respect to the win/lose/tie records, both achieve lower classification error than all previous methods with frequency significant at the 0.05 level. These observations suggest that PKID and ESD enjoy an advantage in terms of classification error reduction over the type of datasets studied in this research.
- PKID and ESD better reduce classification bias than alternative methods. They both achieve lower mean and geometric mean of classification bias. Their advantage in bias reduction grows more apparent with the training data size increasing. This supports our suggestion that PKID and ESD can use additional data to decrease discretization bias, and thus high bias does not attach to large training data any more.
- PKID's outstanding performance in bias reduction is achieved without incurring high variance. This is because PKID equally weighs bias reduction and variance reduction. ESD also demonstrates a good control on discretization variance. Especially among smaller datasets where naive-Bayes probability estimation has a higher risk to suffer insufficient training data, ESD usually achieves lower variance than alternative methods. This supports our suggestion that by controlling the size of the interval, ESD can prevent the discretization variance from being very high. However, we have also observed that ESD does have higher variance especially in some very large datasets. We suggest the reason is that $m = 30$ might not be the optimal size for those datasets. Nonetheless, the gain through ESD's bias reduction is more than the loss through its variance increase, thus ESD still achieves lower naive-Bayes classification error in large datasets.
- Although PKID and ESD manage discretization bias and variance from two different perspectives, they obtain effectiveness competing with each other. The win/lose/tie record of PKID compared with ESD is 15/12/2, which is insignificant with sign test = 0.35.

9 Conclusion

We have proved a theorem that provides a new explanation on why discretization can be effective for naive-Bayes learning. The theorem states that as long as discretization preserves the conditional probability of each class given each quantitative attribute value for each test instance, discretization will result in naive-Bayes classifiers delivering the same probability estimates as would be obtained if the correct probability density functions were employed. Two factors that affect discretization's effectiveness are decision boundaries and the error tolerance of probability estimation for each quantitative attribute. We have analyzed the effect of multiple attributes on these factors, and proposed the bias-variance characterisation of discretization. We have demonstrated that it is unrealistic to expect a single discretization to provide optimal classification performance for multiple instances. Rather, an ideal discretization scheme would discretize separately for each instance to be classified. Where this is not feasible, heuristics that manage discretization bias and variance should be employed. In particular, we have obtained new insights into 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.

These insights have motivated our new heuristic discretization methods, proportional k-interval discretization (PKID) and equal size discretization (ESD). Both PKID and ESD are able to actively take advantage of increasing information in large data to reduce discretization bias and variance. Thus they are expected to demonstrate greater advantage than previous methods especially when learning from large data.

It is desirable that a machine learning algorithm maximizes the information that it derives from large datasets, since increasing the size of a dataset can provide a *domain-independent* way of achieving higher accuracy (Freitas and Lavington, 1996; Provost and Aronis, 1996). This is especially important since large datasets with high dimensional attribute spaces and huge numbers of instances are increasingly used in real-world applications; and naive-Bayes classifiers are particularly attractive to theses applications because of their space and time efficiency. Our experimental results have supported our theoretical analysis. They have demonstrated that with significant frequency, our new methods reduce naive-Bayes classification error when compared to previous alternatives. Another interesting issue arising from our study is that unsupervised discretization methods are able to outperform supervised ones, in our experiments the entropy minimization discretization (Fayyad and Irani, 1993), in the context of naive-Bayes learning. This contradicts the previous understanding that supervised methods tend to have an advantage over unsupervised methods (Dougherty et al., 1995; Hsu et al., 2000).

References

An, A., Cercone, N., 1999. Discretization of continuous attributes for learning classification rules. In: Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining. pp. 509–514.

Androutsopoulos, I., Koutsias, J., Chandrinos, K., Spyropoulos, C., 2000. An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with encrypted personal e-mail messages. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 160–167.

Bauer, E., Kohavi, R., 1999. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning 36 (1-2), 105–139.

Bay, S. D., 1999. The UCI KDD archive [`http://kdd.ics.uci.edu`]. Department of Information and Computer Science, University of California, Irvine.

Bay, S. D., 2000. Multivariate discretization of continuous variables for set mining. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 315–319.

Blake, C. L., Merz, C. J., 1998. UCI repository of machine learning databases [`http://www.ics.uci.edu/~mlearn/MLRepository.html`]. Department of Information and Computer Science, University of California, Irvine.

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, Berkeley .

Casella, G., Berger, R. L., 1990. Statistical Inference. Pacific Grove, Calif.

Catlett, J., 1991. On changing continuous attributes into ordered discrete attributes. In: Proceedings of the European Working Session on Learning. pp. 164–178.

Cestnik, B., 1990. Estimating probabilities: A crucial task in machine learning. In: Proceedings of the Ninth European Conference on Artificial Intelligence. pp. 147–149.

Cestnik, B., Kononenko, I., Bratko, I., 1987. Assistant 86: A knowledge-elicitation tool for sophisticated users. In: Proceedings 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.

Crawford, E., Kay, J., Eric, M., 2002. Iems - the intelligent email sorter. In: Proceedings of the Nineteenth International Conference on Machine Learning. pp. 83–90.

Domingos, P., Pazzani, M., 1996. Beyond independence: Conditions for the optimality of the simple Bayesian classifier. In: Proceedings of the Thirteenth International Conference on Machine Learning (ICML'96). Morgan Kaufmann Publishers, pp. 105–112.

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. In: Proceedings of the Twelfth International Conference on Machine Learning. pp. 194–202.

Fayyad, U. M., Irani, K. B., 1993. Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence. pp. 1022–1027.

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

Freitas, A. A., Lavington, S. H., 1996. Speeding up knowledge discovery in large relational databases by means of a new discretization algorithm. In: Advances in Databases, Proceedings of the Fourteenth British National Conferenc on Databases. pp. 124–133.

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

Gama, J., 2000. Iterative Bayes. Intelligent Data Analysis 4, 475–488.

Gama, J., Torgo, L., Soares, C., 1998. Dynamic discretization of continuous attributes. In: Proceedings of the Sixth Ibero-American Conference on AI. pp. 160–169.

Guthrie, L., Walker, E., 1994. Document classification by machine: Theory and practice. In: Proceedings of COLING-94.

Hsu, C. N., Huang, H. J., Wong, T. T., 2000. Why discretization works for naive Bayesian classifiers. In: Proceedings of the Seventeenth International Conference 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.

Joachims, T., 1997. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In: Proceedings of the International Conference on Machine Learning.

John, G. H., Langley, P., 1995. Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. pp. 338–345.

Kalt, T., 1996. A new probabilistic model of text classification and retrieval. Technical Report IR-78, Center for Intelligent Information Retrieval, University of Massachusetts.

Kerber, R., 1992. Chimerge: Discretization for numeric attributes. In: National Conference on Artificial Intelligence. AAAI Press, pp. 123–128.

Kim, Y.-H., Hahn, S.-Y., Zhang, B.-T., 2000. Text filtering by boosting naive Bayes classifiers. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 168–175.

Kohavi, R., 1996. Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining.

Kohavi, R., Sommerfield, D., Dougherty, J., 1997. Data mining using MLC++: A machine learning library in C++. International Journal on Artificial Intelligence Tools 6 (4), 537–566.

Kohavi, R., Wolpert, D., 1996. Bias plus variance decomposition for zero-one loss functions. In: Proceedings of the 13th International Conference on Machine Learning. pp. 275–283.

Koller, D., Sahami, M., 1997. Hierarchically classifying documents using very few words. In: Proceedings of the Fourteenth International Conference on Machine Learning. pp. 170–178.

Kong, E. B., Dietterich, T. G., 1995. Error-correcting output coding corrects bias and variance. In: Proceedings of the Twelfth International Conference 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), 1–8.

Kononenko, I., 1993. Inductive and Bayesian learning in medical diagnosis. Applied Artificial Intelligence 7, 317–337.

Kononenko, I., 2001. Machine learning for medical diagnosis: history, state of the art and perspective. Artificial Intelligence in Medicine 23 (1), 89–109.

Kukar, M., Groselj, C., Kononenko, I., Fettich, J., 1997. An application of machine learning in the diagnosis of ischaemic heart disease. In: Proceedings of the 6th Conference on Artificial Intelligence in Medicine Europe. pp. 461–464.

Langley, P., Iba, W., Thompson, K., 1992. An analysis of bayesian classifiers. In: Proceedings of the Tenth National Conference on Artificial Intelligence. pp. 223–228.

Larkey, L. S., Croft, W. B., 1996. Combining classifiers in text categorization. In: Proceedings of the 19th Annual International Conference on Research and Development in Information Retrieval. pp. 289–297.

Lavrac, N., 1998. Data mining in medicine: Selected techniques and applications. In: Proceedings of the Second International Conference on The Practical Applications of Knowledge Discovery and Data Mining. pp. 11–31.

Lavrac, N., Keravnou, E., Zupan, B., 2000. Intelligent data analysis in medicine. Encyclopedia of Computer Science and Technology 42 (9), 113–157.

Lewis, D. D., 1992. An evaluation of phrasal and clustered representations on a text categorization task. In: Proceedings of the 15th Annual International

ACM SIGIR Conference 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: In European Conf. on Machine Learning.

Lewis, D. D., Gale, W. A., 1994. A sequential algorithm for training text classifiers. In: Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. pp. 3–12.

Li, H., Yamanishi, K., 1997. Document classification using a finite mixture model. In: Proceedings of the 8th Conference of the European Chapter of the Association for Computational Linguistics. pp. 39–47.

Maron, M., 1961. Automatic indexing: An experimental inquiry. Journal of the Association for Computing Machinery 8, 404–417.

Maron, M., Kuhns, J., 1960. On relevance, probabilistic indexing, and information retrieval. Journal of the Association for Computing Machinery 7 (3), 216–244.

McCallum, A., Nigam, K., 1998. A comparison of event models for naive bayes text classification. In: Proceedings 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. In: Proceedings of the 15th International Conference on Machine Learning.

McSherry, D., 1997a. Avoiding premature closure in sequential diagnosis. Artificial Intelligence in Medicine 10 (3), 269–283.

McSherry, D., 1997b. Hypothesist: A development environment for intelligent diagnostic systems. In: Proceedings of the 6th Conference on Artificial Intelligence in Medicine in Europe. pp. 223–234.

Mitchell, T. M., 1997. Machine Learning. McGraw-Hill Companies.

Miyahara, K., Pazzani, M. J., 2000. Collaborative filtering with the simple bayesian classifier. In: Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence. pp. 679–689.

Montani, S., Bellazzi, R., Portinale, L., Fiocchi, S., Stefanelli, M., 1998. A casebbased retrieval system for diabetic patient therapy. IDAMAP 98 working notes.

Mooney, R. J., Roy, L., 2000. Content-based book recommending using learning for text categorization. In: Proceedings of DL-00, 5th ACM Conference on Digital Libraries. ACM Press, New York, US, San Antonio, US, pp. 195–204.

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. In: Proceedings of the Eleventh European Conference on Machine Learning. pp. 280–291.

Nigam, K., McCallum, A., Thrun, S., Mitchell, T. M., 1998. Learning to classify text from labeled and unlabeled documents. In: Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference, AAAI 98, IAAI 98. pp. 792–799.

Pantel, P., Lin, D., 1998. Spamcop a spam classification & organization program. In: Proceedings of AAAI-98 Workshop on Learning for Text Categorization. pp. 95–98.

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

Pazzani, M., Muramatsu, J., Billsus, D., 1996. Syskill & webert: Identifying interesting web sites. In: Proceedings of the National Conference on Artificial Intelligence.

Pazzani, M. J., 1995. An iterative improvement approach for the discretization of numeric attributes in Bayesian classifiers. In: Proceedings of the First International Conference on Knowledge Discovery and Data Mining.

Perner, P., Trautzsch, S., 1998. Multi-interval discretization methods for decision tree learning. In: Advances in Pattern Recognition, Joint IAPR International Workshops SSPR '98 and SPR '98. pp. 475–482.

Provost, F. J., Aronis, J. M., 1996. Scaling up machine learning with massive parallelism. Machine Learning 23.

Provost, J., 1999. Naive-Bayes vs. rule-learning in classification of email. Technical Report AI-TR-99-284, Artificial Intelligence Lab, The University of Texas at Austin.

Quinlan, J. R., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers.

Rennie, J., 2000. ifile: An application of machine learning to mail filtering. In: Proceedings of the KDD-2000 Workshop on Text Mining.

Roy, N., McCallum, A., 2001. Toward optimal active learning through sampling estimation of error reduction. In: Proceedings of the Eighteenth International Conference on Machine Learning. pp. 441–448.

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, 4th Edition. Duxbury Press.

Silverman, B., 1986. Density Estimation for Statistics and Data Analysis. Chapman and Hall Ltd.

Starr, B., Ackerman, M. S., Pazzani, M., 1996a. Do i care? – tell me what's changed on the web. AAAI Spring Symposium. Stanford, CA.

Starr, B., Ackerman, M. S., Pazzani, M., 1996b. Do-i-care: A collaborative web agent. In: Proceedings of the ACM Conference on Human Factors in Computing Systems. pp. 273–274.

Ting, K., Zheng, Z., 1999. Improving the performance of boosting for naive Bayesian classification. In: Proceedings of the Third Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 296–305.

Torgo, L., Gama, J., 1997. Search-based class discretization. In: Proceedings of the Ninth European Conference on Machine Learning. pp. 266–273.

Tsymbal, A., Puuronen, S., Patterson, D., 2002. Feature selection for ensembles of simple bayesian classifiers. In: Proceedings of the 13th International Symposium on Foundations of Intelligent Systems. pp. 592–600.

Webb, G. I., 2000. Multiboosting: A technique for combining boosting and wagging. Machine Learning 40 (2), 159–196.

Weiss, N. A., 2002. Introductory Statistics. Vol. Sixth Edition. Greg Tobin.

Yang, Y., Liu, X., 1999. A re-examination of text categorization methods. In: Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR1999). pp. 42–49.

Zaffalon, M., Hutter, M., 2002. Robust feature selection using distributions of mutual information. In: Proceedings of the Fourteenth International Conference on Uncertainty in Artificial Intelligence.

Zelic, I., Kononenko, I., Lavrac, N., Vuga, V., 1997. Induction of decision trees and bayesian classification applied to diagnosis of sport injuries. Journal of Medical Systems 21 (6), 429–444.

Zhang, H., Ling, C. X., Zhao, Z., 2000. The learnability of naive Bayes. In: Proceedings of Canadian Artificial Intelligence Conference. pp. 432–441.

Zheng, Z., 1998. Naive Bayesian classifier committees. In: Proceedings of the 10th European Conference on Machine Learning. pp. 196–207.

Zupan, B., Demsar, J., Kattan, M. W., Ohori, M., Graefen, M., Bohanec, M., Beck, J. R., 2001. Orange and decisions-at-hand: Bridging predictive data mining and decision support. In: Proceedings of IDDM2001: ECML/PKDD-2001 Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning. pp. 151–162.