Mach Learn (2009) 74: 39–74
DOI 10.1007/s10994-008-5083-5

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

Ying Yang · Geoffrey I. Webb

Received: 4 July 2005 / Revised: 28 February 2008 / Accepted: 7 August 2008 /
Published online: 4 September 2008

Abstract Quantitative attributes are usually discretized in Naive-Bayes learning. We establish simple conditions under which discretization is equivalent to use of the true probability density function during 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 supply insights into managing discretization bias and variance by adjusting the number of intervals and the number of training instances contained in each interval. We accordingly propose *proportional discretization* and *fixed frequency discretization*, two efficient unsupervised discretization methods that are able to effectively manage discretization bias and variance. We evaluate our new techniques against four key discretization methods for naive-Bayes classifiers. The experimental results support our theoretical analyses by showing that with statistically significant frequency, 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 current established discretization methods.

Keywords Discretization · Naive-Bayes Learning · Bias · Variance

1 Introduction

When classifying an instance, naive-Bayes classifiers assume attributes conditionally independent of one another given the class; and then apply Bayes theorem to estimate

Editor: Dan Roth.

Y. Yang (✉)
Australian Taxation Office, 990 Whitehorse Road, Box Hill, Victoria 3128, Australia
e-mail: ying.yang@ato.gov.au

G.I. Webb
Faculty of Information Technology, Monash University, Clayton, Victoria 3800, Australia
e-mail: geoff.webb@infotech.monash.edu.au

the probability of each class given the instance. The class with the highest probability estimate is chosen as the class for the instance. Naive-Bayes classifiers are simple, effective, efficient and robust, as well as 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; Mitchell 1997; Lewis 1998). 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 complex classification algorithms (Kononenko, Langley et al. 1990, 1992; Domingos and Pazzani 1996, 1997). In consequence, naive-Bayes classifiers have widespread deployment in applications including medical diagnosis (Lavrac 1998; Lavrac et al. 2000; Kononenko 2001), email filtering (Androutsopoulos et al. 2000; Crawford et al. 2002), and recommender systems (Starr et al. 1996; Miyahara and Pazzani 2000; Mooney and Roy 2000). There has also been considerable interest in developing variants of naive-Bayes learning that weaken the attribute independence assumption (Langley and Sage 1994; Sahami 1996; Singh and Provan 1996; Friedman et al. 1997; Keogh and Pazzani 1999; Zheng and Webb 2000; Webb et al. 2005; Acid et al. 2005; Cerquides and Mántaras 2005) .

Classification tasks often involve quantitative attributes. For naive-Bayes classifiers, quantitative attributes are usually processed by discretization. This is because experience has shown that 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 probability density functions from which the quantitative attribute values are drawn. For instance, a conventional approach is to assume that a quantitative attribute's probability within a class has a normal distribution (Langley 1993; Langley and Sage 1994; Pazzani et al. 1994; Mitchell 1997). However, Pazzani (1995) argued that in many real-world applications the attribute data did not follow a normal distribution; and as a result, the probability estimation of naive-Bayes classifiers was not reliable and could lead to inferior classification performance. This argument was supported by Dougherty et al. (1995) who presented experimental results showing that naive-Bayes with discretization attained a large average increase in accuracy compared with naive-Bayes with normal distribution assumption. In contrast, discretization creates a qualitative attribute X_i^* from a quantitative attribute X_i. Each value of X_i^* corresponds to an interval of values of X_i. X_i^* is used instead of X_i for training a classifier. In contrast to parametric techniques for inference from quantitative attributes, such as probability density estimation, discretization avoids the need to assume the form of an attribute's underlying distribution. 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 classification bias and variance of generated naive-Bayes classifiers. Such effects are hereafter named *discretization bias* and *variance*. We believe that study of discretization bias and variance is illuminating. We investigate the impact of discretization bias and variance on the classification performance of naive-Bayes classifiers. We analyze the factors that can affect discretization bias and variance. The resulting insights motivate the development of two new heuristic discretization methods, *proportional discretization* and *fixed frequency 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.

In the rest of this paper, Sect. 2 prepares necessary background knowledge including terminology and naive Bayes learning. Section 3 defines discretization in naive-Bayes learning.

Section 4 discusses why discretization can be effective for naive-Bayes learning. In particular, it establishes specific conditions under which discretization will result in naive-Bayes classifiers delivering the same probability estimates as would be obtained if the true probability density function for each quantitative attribute were employed. Section 5 presents an analysis of the factors that can affect the effectiveness of discretization when learning from multiple attributes. It also introduces the bias-variance analysis of discretization outcomes. Much of this material has previously been covered in an earlier paper (Yang and Webb 2003), but it is included for completeness and ease of reference. Section 6 provides a review of previous key discretization methods for naive-Bayes learning with a focus on their discretization bias and variance profiles. To our knowledge, this is the first comprehensive review of this specialized field of research. Section 7 proposes our new heuristic discretization techniques, designed to manage discretization bias and variance. While much of the material in Sect. 7.1 has previously been covered in Yang and Webb (2001), it also is included here for completeness and ease of reference. Section 8 describes experimental evaluation. To our knowledge, this is the first extensive experimental comparison of techniques for this purpose. Section 9 presents conclusions.

2 Background knowledge

2.1 Terminology

There is an extensive literature addressing discretization, within which there is considerable variation in the terminology used to describe which type of data is transformed to which type of data by discretization, including 'quantitative' *vs.* 'qualitative', 'continuous' *vs.* 'discrete', 'ordinal' *vs.* 'nominal', or 'numeric' *vs.* 'categorical'. Turning to the authority of introductory statistical textbooks (Bluman 1992; Samuels and Witmer 1999), we believe that the 'quantitative' vs. 'qualitative' distinction is most applicable in the context of discretization, and hence choose them for use hereafter.

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** quantitative attribute assumes values that can be counted. The attribute cannot assume all values on the number line within its value range. An example is *number of children in a family*. A **continuous** quantitative 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 the *Fahrenheit temperature scale*.

2.2 Naive-Bayes classifiers

In naive-Bayes learning, we define:

- C as a random variable denoting the class of an instance,
- $\mathbf{X}\langle X_1, X_2, \ldots, X_k\rangle$ as a vector of random variables denoting the observed attribute values (an instance),
- c as a particular class label,

- $\mathbf{x}\langle x_1, x_2, \ldots, x_k\rangle$ as a particular observed attribute value vector (a particular instance),
- $\mathbf{X}=\mathbf{x}$ as shorthand for $X_1{=}x_1 \wedge X_2{=}x_2 \wedge \cdots \wedge X_k{=}x_k$.

The learner is asked to predict a test instance $\mathbf{x}$'s 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 $\mathbf{x}$ (Duda and Hart 1973). Bayes theorem can be used to calculate:

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

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 a previously unseen instance that 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 $X_1, X_2, \ldots, X_k$ 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:

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

Classifiers using (4) are *naive-Bayes classifiers*. The assumption embodied in (3) is the *attribute independence assumption*. The probability $p(C{=}c \,|\, \mathbf{X}{=}\mathbf{x})$ denotes the conditional probability of a class c given an instance $\mathbf{x}$. The probability $p(C{=}c)$ denotes the prior probability of a particular class c. The probability $p(X_i{=}x_i \,|\, C{=}c)$ denotes the conditional probability that an attribute X_i takes a particular value x_i given the class c.

3 The nature of discretization

For naive-Bayes learning, the class C is qualitative, and an attribute X_i can be either qualitative or quantitative. Since quantitative data have characteristics different from qualitative data, the practice of estimating probabilities in (4) when involving qualitative data is different from that when involving quantitative data.

Qualitative attributes, including the class, usually take a small number of values (Bluman 1992; Samuels and Witmer 1999). Thus there are usually many instances of each value in the training data. The probability $p(C{=}c)$ can be estimated from the frequency of instances with $C{=}c$. The probability $p(X_i{=}x_i \,|\, C{=}c)$, when X_i is qualitative, can be estimated from the frequency of instances with $C{=}c$ and the frequency of instances with $X_i{=}x_i \wedge C{=}c$.

These estimates are strong consistent estimates according to the strong law of large numbers (Casella and Berger 1990; John and Langley 1995).

When it is quantitative, X_i often has a large or even an infinite number of values (Bluman 1992; Samuels and Witmer 1999). Thus the probability of a particular value x_i given the class c, $p(X_i{=}x_i \mid C{=}c)$ can be infinitely small. Accordingly, there usually are very few training instances for any one value. Hence it is unlikely that reliable estimation of $p(X_i{=}x_i \mid C{=}c)$ can be derived from the observed frequency. 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^*. So long as there are sufficient training instances, probabilities of X_i^* can be reliably estimated from corresponding frequencies. However, because discretization loses the ability to differentiate between values within each interval, it might suffer information loss.

Two important concepts involved in our study of discretization are **interval frequency** and **interval number**. Interval frequency is the frequency of training instances in an interval formed by discretization. Interval number is the total number of intervals formed by discretization.

4 Why discretization can be effective

Dougherty et al. (1995) found empirical evidence that naive-Bayes classifiers using discretization achieved lower classification error than those using unsafe probability density assumptions. They suggested that discretization could be effective because it did not make assumptions about the form of the probability distribution from which the quantitative attribute values were drawn. Hsu et al. (2000, 2003) proposed a further analysis of 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 $p(X_i^*{=}x_i^* \mid C{=}c)$ simulated the role of $f(X_i{=}x_i \mid C{=}c)$ 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 $p(C{=}c \mid X_i^*{=}x_i^*)$ as an estimate of $p(C{=}c \mid X_i{=}x_i)$; and that discretization can be effective to the degree that $p(C{=}c \mid \mathbf{X}^*{=}\mathbf{x}^*)$ is an accurate estimate of $p(C{=}c \mid \mathbf{X}{=}\mathbf{x})$, where instance $\mathbf{x}^*$ is the discretized version of instance $\mathbf{x}$. Such an analysis was first proposed by Kononenko (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 conditional attribute independence assumption embodied in (3). Thus we present the following theorem that we suggest more accurately captures the mechanism by which discretization works in naive-Bayes learning than do previous theoretical analyses.

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

[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.

Proof According to Bayes theorem, we have:

$$p(C{=}c \mid \mathbf{X}{=}\mathbf{x}) = p(C{=}c)\frac{p(\mathbf{X}{=}\mathbf{x} \mid C{=}c)}{p(\mathbf{X}{=}\mathbf{x})};$$

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

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

using Bayes theorem:

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

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

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

$$\propto p(C{=}c)^{1-k}\prod_{i=1}^{k} p(C{=}c \mid X_i{=}x_i)$$

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

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

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

using Bayes theorem again:

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

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

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:

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

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

$$= p(C{=}c)p(\mathbf{X}^*{=}\mathbf{x}^* \mid C{=}c)$$

$$= p(C{=}c \mid \mathbf{X}^*{=}\mathbf{x}^*)p(\mathbf{X}^*{=}\mathbf{x}^*);$$

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

$$\propto p(C{=}c \mid \mathbf{X}^*{=}\mathbf{x}^*);$$

because we are talking about probability distributions, we can normalize $p(C \mid \mathbf{X}^*{=}\mathbf{x}^*)$ and obtain:

$$= p(C{=}c \mid \mathbf{X}^*{=}\mathbf{x}^*). \qquad \square$$

Theorem 1 assures us that so long as the attribute independence assumption holds, and discretization forms a qualitative X_i^* for each quantitative X_i such that $p(C{=}c \mid X_i^*{=}x_i^*) = p(C{=}c \mid X_i{=}x_i)$, discretization will result in naive-Bayes classifiers delivering the same probability estimates as would be obtained if the correct probability density function were employed. Theorem 1 suggests that the most important factor to influence the accuracy of the probability estimates will be the accuracy with which $p(C{=}c \mid X_i^*{=}x_i^*)$ serves as an estimate of $p(C{=}c \mid X_i{=}x_i)$. This leads us to the following section.

5 What affects discretization effectiveness

When we talk about the effectiveness of a discretization method in naive-Bayes learning, we mean the classification performance of naive-Bayes classifiers that are trained on data pre-processed by this discretization method. There are numerous metrics on which classification performance might be assessed. In the current paper we focus on zero-one loss classification error.

Two influential factors with respect to performing discretization so as to minimize classification error 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, effects we name **discretization bias** and **discretization variance**. According to (4), 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.

5.1 Classification bias and variance

The performance of naive-Bayes classifiers discussed in our study is measured by their classification *error*. The error can be decomposed 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 S of training instances to produce a classifier to classify an instance $\mathbf{x}$. Suppose we could draw a sequence of training sets $S_1, S_2, \ldots, S_l$, 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

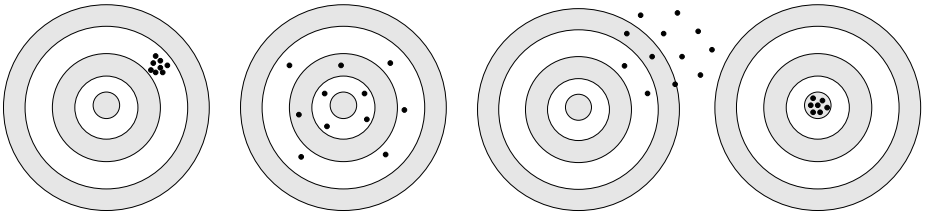


Fig. 1 Bias and variance in shooting arrows at a target. Bias means that the archer systematically misses in the same direction. Variance means that the arrows are scattered (Moore and McCabe 2002)

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

The use of different discretization techniques can be expected to affect the classification bias and variance of generated naive-Bayes classifiers. We name the effects *discretization bias* and *variance*.

5.2 Decision boundaries

Hsu et al. (2000, 2003) provided an interesting analysis of the discretization problem utilizing the notion of a *decision boundary*, relative to a probability density function $f(X_i \mid C{=}c)$ of a quantitative attribute X_i given each class c. They defined decision boundaries of X_i as intersection points of the curves of $f(X_i \mid C)$, where ties occurred among the largest conditional densities. They suggested that the optimal classification for an instance with $X_i{=}x_i$ was to pick the class c such that $f(X_i{=}x_i \mid C{=}c)$ was the largest, and observed that this class was different when x_i was on different sides of a decision boundary. Hsu et al.'s analysis only addressed one-attribute classification problems, and only suggested that the analysis could be extended to multi-attribute applications without indicating how this might be so.

In our analysis we employ a different definition of a decision boundary to that of Hsu et al.'s because:

1. Given Theorem 1, we believe that better insights are obtained by focusing on the values of X_i at which the class that maximizes $p(C{=}c \mid X_i{=}x_i)$ changes rather than those that maximize $f(X_i{=}x_i \mid C{=}c)$.
2. The condition that ties occur among the largest conditional probabilities is neither necessary nor sufficient for a decision boundary to occur. For example, suppose that we

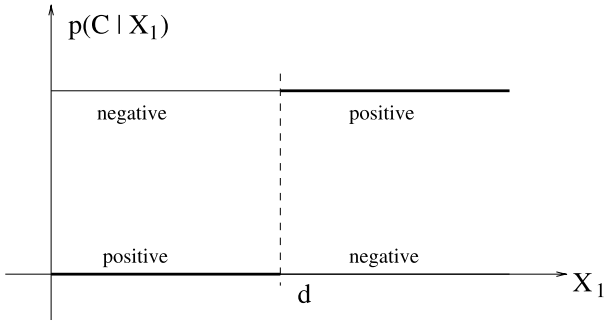


Fig. 2 A tie in conditional probabilities is not a necessary condition for a decision boundary to exist

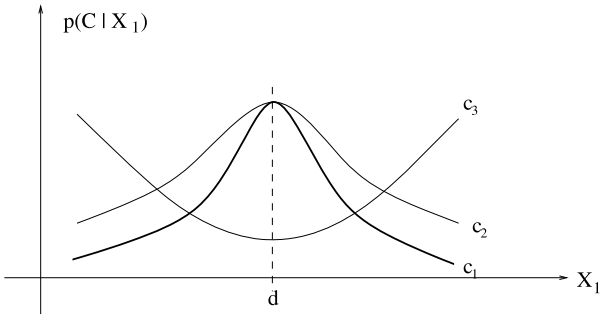


Fig. 3 A tie in conditional probabilities is not a sufficient condition for a decision boundary to exist

have probability distributions as plotted in Fig. 2 that depicts a domain with two classes (*positive* vs. *negative*) and one attribute X_1. We have $p(positive \mid X_1)$=1.0 (if $X_1 \geq d$); or 0.0 otherwise. X_1=d should be a decision boundary since the most probable class changes from *negative* to *positive* when X_i crosses the value d. However, there is no value of X_1 at which the probabilities of the two classes are equal. Thus the condition requiring ties is not necessary. Consider a second example as plotted in Fig. 3. The conditional probabilities for c_1 and c_2 are equal at X_1=d. However, d is not a decision boundary because c_2 is the most probable class on both sides of X_1=d. Thus the condition is not sufficient either.

3. It is possible that a decision boundary is not a single value, but a region of values. For example as plotted in Fig. 4, the two classes c_1 and c_2 are both most probable through the region $[d, e]$. In addition, the region's width can be *zero*, as illustrated in Fig. 2.
4. To extend the notion of decision boundaries to the case of multiple attributes, it is necessary to allow the decision boundaries of a given attribute X_i to vary from test instance to test instance, depending on the precise values of other attributes presented in the test instance, as we will explain later in this section. However, Hsu et al. defined the decision boundaries of a quantitative attribute in such a way that they were independent of other attributes.

In view of these issues we propose a new definition for decision boundaries. This new definition is central to our study of discretization effectiveness in naive-Bayes learning. As we have explained, motivated by Theorem 1, we focus on the probability $p(C$=$c \mid X_i)$ of each class c given a quantitative attribute X_i rather than on the density function $f(X_i$=$x_i \mid C$=$c)$.

To define a decision boundary of a quantitative attribute X_i, we first define a *most probable class*. When classifying an instance **x**, a most probable class c_m given **x** is the class that

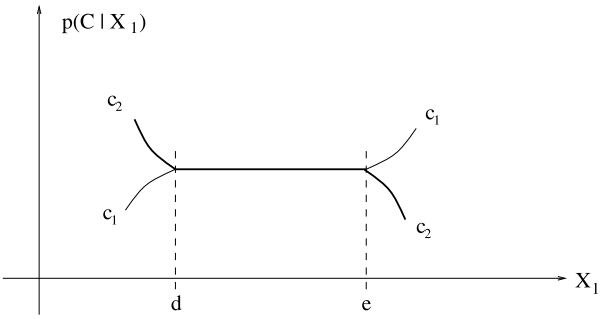


Fig. 4 Decision boundaries may be regions rather than points

satisfies $\forall c \in C, P(c \mid \mathbf{x}) \leq P(c_m \mid \mathbf{x})$. Note that there may be multiple most probable classes for a single $\mathbf{x}$ if the probabilities of those classes are equally the largest. In consequence, we define a *set of most probable classes*, $mpc(\mathbf{x})$, whose elements are all the most probable classes for a given instance $\mathbf{x}$. As a matter of notational convenience we define $\mathbf{x}\backslash X_i{=}v$ to represent an instance $\mathbf{x}'$ that is identical to $\mathbf{x}$ except that $X_i{=}v$ for $\mathbf{x}'$.

A *decision boundary* of a quantitative attribute X_i given an instance $\mathbf{x}$ in our analysis is an interval (l, r) of X_i (that may be of zero width) such that

$$\forall (w \in [l, r), u \in (l, r]), \neg(w{=}l \wedge u{=}r) \Rightarrow mpc(\mathbf{x}\backslash X_i{=}w) \cap mpc(\mathbf{x}\backslash X_i{=}u) \neq \emptyset$$

$$\wedge$$

$$mpc(\mathbf{x}\backslash X_i{=}l) \cap mpc(\mathbf{x}\backslash X_i{=}r) = \emptyset.$$

That is, a decision boundary is a range of values of an attribute throughout which the sets of most probable classes for every pair of values has one or more values in common and on either side of which the sets of most probable classes share no values in common.

5.3 How decision boundaries affect discretization bias and variance

When analyzing how decision boundaries affect discretization effectiveness, we suggest that the analysis involving only one attribute differs from that involving multiple attributes, since the final choice of the class is decided by the product of each attribute's probability in the later situation. 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 $p(C{=}c_1 \mid X_1) = 1 - (X_1 - 1)^2$ and $p(C{=}c_2 \mid X_1) = (X_1 - 1)^2$ respectively as plotted in Fig. 5.

The consequent decision boundaries are labeled DB_1 and DB_2 respectively in Fig. 5. The most probable class for an instance $\mathbf{x}{=}\langle x_1 \rangle$ changes each time x_1's location crosses a decision boundary. Assume a discretization method to create intervals I_i $(i{=}1, \ldots, 5)$ as in Fig. 5. I_2 and I_4 contain decision boundaries while the remaining intervals do not. For any two values in I_2 (or I_4) but on different sides of a decision boundary, the optimal naive-Bayes learner under zero-one loss should select a different class for each value.[2] But under discretization, all the values in the same interval cannot be differentiated and we will have

[2]Please note that some instances may be misclassified even when optimal classification is performed. An optimal classifier minimizes classification error under zero-one loss. Hence even though it is optimal, it may still misclassify instances on both sides of a decision boundary.

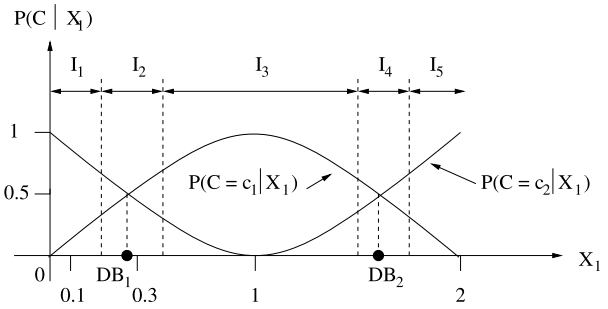


Fig. 5 Probability distribution in one-attribute problem

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. The larger the interval frequency, the more likely that the value range of the interval is larger, thus 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 $p(C \mid X_1)$ for each class. However, for a multi-attribute application, the decision boundaries of an attribute X_i are not only decided by the distribution of $p(C \mid X_i)$, 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 Fig. 6, 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 Fig. 5. However, because of the existence of X_2, those might not be decision boundaries any more. Consider a test instance $\mathbf{x}$ with $X_2 = 0.2$. Since $p(C{=}c_1 \mid X_2{=}0.2){=}0.8 > p(C{=}c_2 \mid X_2{=}0.2){=}0.2$, and $p(C{=}c \mid \mathbf{x}) \propto \prod_{i=1}^{2} p(C{=}c \mid X_i{=}x_i)$ for each class c according to Theorem 1, $p(C{=}c_1 \mid \mathbf{x})$ does not equal $p(C{=}c_2 \mid \mathbf{x})$ when X_1 falls on any of the single attribute decision boundaries as presented in Fig. 5. Instead X_1's decision boundaries change to be DB_1 and DB_4 as in Fig. 6. Now 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 Fig. 6.

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 the one-attribute context, intervals containing decision boundaries have potential negative impact on discretization bias.

The above expectation has been verified on real-world data, taking the benchmark data 'Balance-Scale' from the UCI machine learning repository (Blake and Merz 1998) as an example. We chose 'Balance-Scale' because it is a relatively large data set with the class and quantitative attributes both having relatively few values. This is important in order to derive clear plots of the probability density functions (pdf). The data have four attributes, 'left

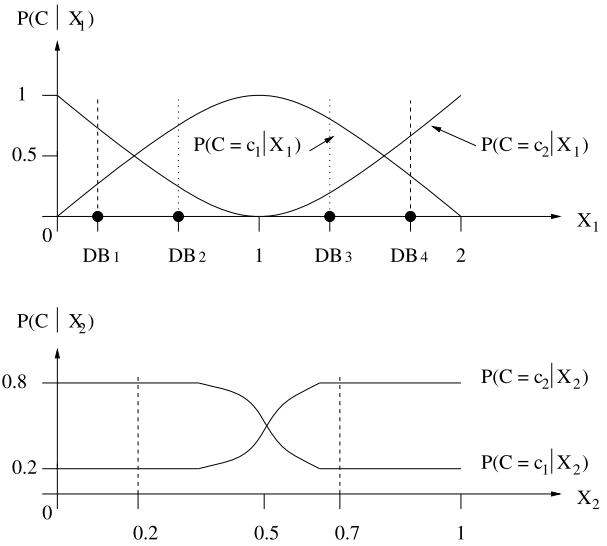


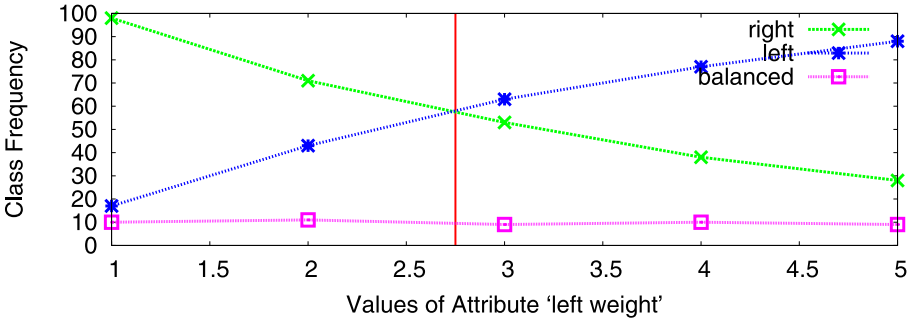
Fig. 6 Probability distribution in two-attribute problem

weight', 'left distance', 'right weight', and 'right distance'. If (left-distance $\times$ left-weight $>$ right-distance $\times$ right-weight), the class is 'left'; if (left-distance $\times$ left-weight $<$ right-distance $\times$ right-weight), the class is 'right'; otherwise the class is 'balanced'. Hence given a class label, there is strong interdependency among attributes. For example, Figs. 7a to 7c illustrate how the decision boundaries of 'left weight' move depending on the values of 'right weight'. Figure 7a depicts the pdf of each class[3] for the attribute 'left weight' according to the whole data set. We then increasingly sort all instances by the attribute 'right weight', and partition them into two equal-size sets. Figure 7b depicts the class pdf curves on the attribute 'left weight' in the first half instances while Fig. 7c in the second half. It is clearly shown that the decision boundary of 'left weight' changes its location among those three figures.

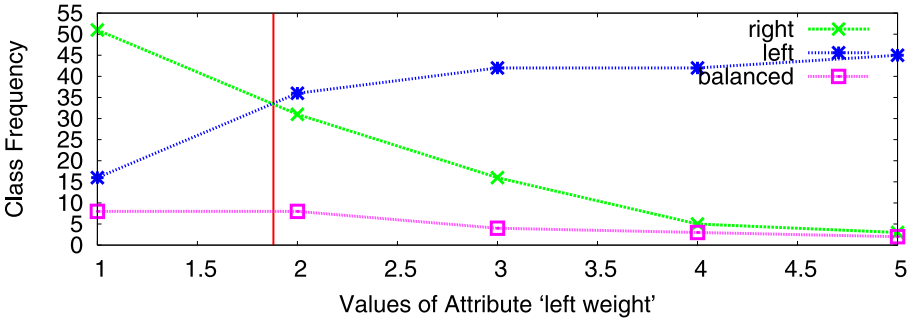
According to the above understandings, discretization bias can be reduced by identifying the decision boundaries and setting the interval boundaries close to them. However, identifying the correct decision boundaries depends on finding the true form of $p(C \mid X)$. Ironically, if we have already found $p(C \mid X)$, we can resolve the classification task directly; thus there is no need to consider discretization at all. Without knowing $p(C \mid X)$, an extreme solution is to set each value as an interval. Although this most likely guarantees that no interval contains a decision boundary, it usually results in very few instances per interval. As a result, the estimation of $p(C \mid X)$ might be so unreliable that we cannot identify the truly most probable class even if there is no decision boundary in the interval. The smaller the interval frequency, the less training instances per interval for probability estimation, thus the more likely that the variance of the generated classifiers increases since even a small change of the training data might totally change the probability estimation.

A possible solution to this problem is to require that the interval frequency should be sufficient to ensure stability in the probability estimated therefrom. This raises the question, how reliable must the probability be? That is, when estimating $p(C{=}c \mid X{=}x)$ by

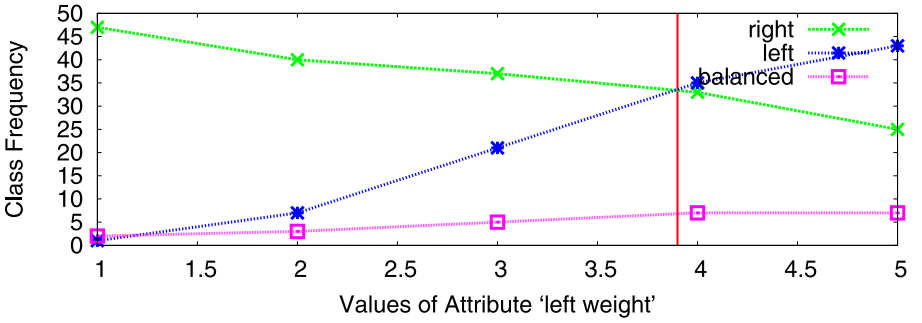
[3]Strictly speaking, the curves depict frequencies of classes from which the pdf can be derived.



(a) Decision boundary of the attribute 'left weight' in all instances



(b) Decision boundary of the attribute 'left weight' in the 1st half of instances sorted by the attribute 'right weight'



(c) Decision boundary of the attribute 'left weight' in the 2nd half of instances sorted by the attribute 'right weight'

Fig. 7 Decision boundary of the attribute 'left weight' moves according to values of the attribute 'right weight' in the UCI benchmark data set 'Balance-Scale'

$p(C{=}c \mid X^{*}{=}x^{*})$, how much error can be tolerated without altering the classification. This motivates our following analysis.

5.4 Error tolerance of probability estimation

To investigate this factor, we return to our example depicted in Fig. 5. We suggest that different values have different error tolerance with respect to their probability estimation. For example, for a test instance $\mathbf{x}\langle X_1{=}0.1\rangle$ and thus of class c_2, its true class probability distribution is $p(C{=}c_1 \mid \mathbf{x}){=}p(C{=}c_1 \mid X_1{=}0.1) = 0.19$ and $p(C{=}c_2 \mid \mathbf{x}){=}p(C{=}c_2 \mid X_1{=}0.1) = 0.81$. According to naive-Bayes learning, so long as $p(C{=}c_2 \mid X_1{=}0.1) > 0.50$, c_2 will be correctly assigned as the class and the classification is optimal under zero-one loss. This means, the error tolerance of estimating $p(C \mid X_1{=}0.1)$ can be as large as $0.81 - 0.50 = 0.31$. However, for another test instance $\mathbf{x}\langle X_1{=}0.3\rangle$ 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 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 larger an interval's frequency, the lower the expected error of probability estimates pertaining to that interval. Hence, the lower the error tolerance for a value, the larger the ideal frequency for the interval from which its probabilities are estimated. Since all factors affecting error tolerance vary from case to case, there cannot be a universal, or even a domain-wide constant that represents the ideal interval frequency, 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 unknown, the best we can hope for is heuristic approaches to managing error tolerance that work well in practice.

5.5 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 $\langle c, x_i\rangle$ given each particular test instance is known; and thus the decision boundaries are known. If the decision boundaries are unknown, 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. Further, if we want to have a single discretization of an attribute that applies to every instance to be classified, as the decision boundaries may move from instance to instance, it is desirable to minimize the size of each interval so as to minimize the total extent of the number range falling within an interval on the wrong size of a decision boundary. By this means we expect to reduce the discretization bias. On the other hand, we want to ensure that each interval frequency is 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 frequency and interval number. That is, the larger the interval frequency, the smaller the interval number, and vice versa. Low learning error can be achieved by tuning interval frequency and interval number to find a good trade-off between discretization bias and variance. We have argued that there is no universal solution to this problem, that the optimal

trade-off between interval frequency 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.

6 Existing discretization methods

Here we review four 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.

6.1 Equal width discretization and 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 $v_{min} + w, v_{min} + 2w, \ldots, v_{min} + (k-1)w$.

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, 2003). 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 frequency and thus tend to incur high variance. When the training data size becomes large, more and more instances are added into each interval. This can reduce variance. However successive increases to an interval's size have decreasing effect on reducing variance and hence have decreasing effect on reducing classification error. Our study suggests it might be more effective to use additional data to increase interval numbers so as to further decrease bias, as reasoned in Sect. 5.

6.2 Entropy minimization discretization

EWD and EFD are *unsupervised* discretization techniques. That is, they take no account of the class information when selecting cut points. In contrast, entropy minimization discretization (EMD) (Fayyad and Irani 1993) is a *supervised* technique. It 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.

Although EMD has demonstrated strong performance for naive-Bayes (Dougherty et al. 1995; Perner and Trautzsch 1998), it 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 so as to help avoid the fragmentation problem in decision tree learning. For the same reasoning as employed with respect to EWD and EFD, we thus anticipate that EMD will fail to fully utilize available data to reduce bias when the data are large. Further, 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 when the values of other attributes change, as we have explained in Sect. 5.

6.3 Lazy discretization

Lazy discretization (LD) (Hsu et al. 2000, 2003) 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 width is equal to that produced by some other algorithm chosen by the user, such as EWD or EMD. In Hsu et al.'s implementation, the interval frequency is the same as created by EWD 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 all 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 EWD and EMD (Hsu et al. 2000, 2003), the high memory and computational overheads have a potential to damage naive-Bayes classifiers' classification efficiency. We anticipate LD will attain low discretization variance because it always puts the value in question at the middle of an interval. We also anticipate that its behavior on controlling bias will be affected by its adopted interval frequency strategy.

7 New discretization techniques that manage discretization bias and variance

We have argued that the interval frequency and interval number formed by a discretization method can affect its discretization bias and variance. Such a relationship has been hypothesized also by a number of previous authors ((Pazzani 1995; Torgo and Gama 1997; Gama et al. 1998; Hussain et al. 1999; Mora et al. 2000); Hsu et al. 2000, 2003). Thus we anticipate that one way to manage discretization bias and variance is to adjust interval frequency and interval number. Consequently, we propose two new heuristic discretization techniques, *proportional discretization* and *fixed frequency discretization*. To the best of our knowledge, these are the first techniques that explicitly manage discretization bias and variance by tuning interval frequency and interval number.

7.1 Proportional discretization

Since a good learning scheme should have both low bias and low variance (Moore and McCabe 2002), it would be advisable to equally weigh discretization bias reduction and variance reduction. As we have analyzed in Sect. 5, discretization resulting in large interval frequency tends to have low variance; conversely, discretization resulting in large interval number tends to have low bias. To achieve this, as the amount of training data increases we should increase both the interval frequency and number and as it decreases we should reduce both. One credible manner to achieve this is to set interval frequency and interval number equally proportional to the amount of training data. This leads to a new discretization method, *proportional discretization* (PD).

When discretizing a quantitative attribute for which there are n training instances with known values, supposing that the desired interval frequency is s and the desired interval number is t, PD employs (5) to calculate s and t. It then sorts the quantitative values in ascending order and discretizes them into intervals of frequency s. Thus each interval contains approximately s training instances with adjacent (possibly identical) values.

$$\begin{aligned} s \times t &= n, \\ s &= t. \end{aligned} \tag{5}$$

By setting interval frequency and interval number equal, PD can use any increase in training data to lower both discretization bias and variance. Bias can decrease because the interval number increases, thus any given interval is less likely to include a decision boundary of the original quantitative value. Variance can decrease because the interval frequency increases, thus the naive-Bayes probability estimation is more stable and reliable. This means that PD has greater potential to take advantage of the additional information inherent in large volumes of training data than previous methods.

7.2 Fixed frequency discretization

An alternative approach to managing discretization bias and variance is *fixed frequency discretization* (FFD). As we have explained in Sect. 5, ideal discretization for naive-Bayes learning should first ensure that the interval frequency is sufficiently large so that the error of the probability estimate falls within the quantitative data's error tolerance of probability estimation. In addition, ideal discretization should maximize the interval number so that the formed intervals are less likely to contain decision boundaries. This understanding leads to the development of FFD.

To discretize a quantitative attribute, FFD sets a *sufficient interval frequency*, m. Then it discretizes the ascendingly sorted values into intervals of frequency m. Thus each interval has approximately the same number m of training instances with adjacent (possibly identical) values.

By introducing m, FFD aims to ensure that in general the interval frequency is sufficient so that there are enough training instances in each interval to reliably estimate the naive-Bayes probabilities. Thus FFD can control discretization variance by preventing it from being very high. As we have explained in Sect. 5, the optimal interval frequency varies from instance to instance and from domain to domain. Nonetheless, we have to choose a frequency so that we can implement and evaluate FFD. In our study, we choose the frequency 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, more intervals can be formed as the training data increase. This means that FFD can make use of extra data to reduce discretization bias. In this way, where there are sufficient data, FFD can prevent both high bias and high variance.

It is important to distinguish our new method, fixed frequency discretization (FFD) from equal frequency discretization (EFD) (Catlett 1991; Kerber 1992; Dougherty et al. 1995), both of which form intervals of equal frequency. EFD fixes the interval number. It arbitrarily chooses the interval number k and then discretizes a quantitative attribute into k intervals such that each interval has the same number of training instances. Since it does not control the interval frequency, EFD is not good at managing discretization bias and variance. In contrast, FFD fixes the interval frequency. It sets an interval frequency m that is sufficient for the naive-Bayes probability estimation. It then sets cut points so that each interval contains m training instances. By setting m, FFD can control discretization variance. On top of that, FFD forms as many intervals as constraints on adequate probability estimation accuracy allow, which is advisable for reducing discretization bias.

7.3 Time complexity analysis

We have proposed two new discretization methods as well as reviewed four previous key ones. We here analyze the computational time complexity of each method. 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. For each method to discretize a quantitative attribute, supposing the number of training instances,[4] test instances, attributes and classes are n, l, v and m respectively, its time complexity is analyzed as follows.

- EWD, EFD, PD and FFD 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 $O(mn \log n)$.
- 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, PD and FFD have complexity lower than EMD. LD tends to have high complexity when the training or testing data size is large.

8 Experimental evaluation

We evaluate whether PD and FFD can better reduce naive-Bayes classification error by better managing discretization bias and variance, compared with previous discretization methods, EWD, EFD, EMD and LD. EWD and EFD are implemented with the parameter k=10. The original LD in Hsu et al.'s implementation (2000, 2003) chose EWD with k=10 to decide its interval. That is, it formed *interval width* equal to that produced by EWD with k=10.

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

Table 1 Experimental data sets

Data set	Size	Qn.	Ql.	C.	Data set	Size	Qn.	Ql.	C.
LaborNegotiations	57	8	8	2	Annealing	898	6	32	6
Echocardiogram	74	5	1	2	German	1000	7	13	2
Iris	150	4	0	3	MultipleFeatures	2000	3	3	10
Hepatitis	155	6	13	2	Hypothyroid	3163	7	18	2
WineRecognition	178	13	0	3	Satimage	6435	36	0	6
Sonar	208	60	0	2	Musk	6598	166	0	2
Glass	214	9	0	6	PioneerMobileRobot	9150	29	7	57
HeartCleveland	270	7	6	2	HandwrittenDigits	10992	16	0	10
LiverDisorders	345	6	0	2	SignLanguage	12546	8	0	3
Ionosphere	351	34	0	2	LetterRecognition	20000	16	0	26
HorseColic	368	7	14	2	Adult	48842	6	8	2
CreditScreening	690	6	9	2	IpumsLa99	88443	20	40	13
BreastCancer	699	9	0	2	CensusIncome	299285	8	33	2
PimaIndiansDiabetes	768	8	0	2	ForestCovertype	581012	10	44	7
Vehicle	846	18	0	4					

Since we manage discretization bias and variance through interval frequency (and interval number), which is relevant but not identical to interval width, we implement LD with EFD being its interval frequency strategy. That is, LD forms *interval frequency* equal to that produced by EFD with k=10. We clarify again that training instances with identical values must be placed in the same interval under each and every discretization scheme.

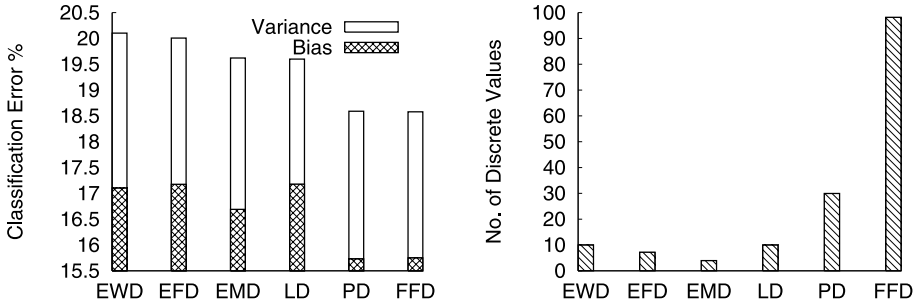
8.1 Data

We run our experiments on 29 benchmark data sets 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 data sets used by Fayyad and Irani when publishing the entropy minimization heuristic for discretization. The second part is composed of all the UCI data sets with quantitative attributes used by Domingos and Pazzani 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 data sets that we can identify containing numeric attributes, with emphasis on those having more than 5000 instances. Table 1 describes each data set, including the number of instances (Size), quantitative attributes (Qn.), qualitative attributes (Ql.) and classes (C.). The data sets are increasingly ordered by the size.

8.2 Design

To evaluate a discretization method, for each data set, 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 following experimental results are recorded.

- **Classification error**. Listed in Table 3 in Appendix is the percentage of incorrect classifications of naive-Bayes classifiers in the test averaged across all folds of the cross validation.



(a) Mean error decomposed into bias and variance

(b) Averaged number of discrete values formed by each method

Fig. 8 Comparing alternative discretization methods

- **Classification bias and variance**. Listed respectively in Table 4 and Table 5 in Appendix are bias and variance 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.
- **Number of discrete values**. Each discretization method discretizes a quantitative attribute into a set of discrete values (intervals), the number of which as we have suggested relates to discretization bias and variance. The number of intervals formed by each discretization method, averaged across all quantitative attributes is also recorded and illustrated in Fig. 8b.

8.3 Statistics

Various statistics are employed to evaluate the experimental results.

- **Mean error**. This is the arithmetic mean of a discretization's errors across all data sets. It provides a gross indication of the relative performance of competing methods. It is debatable whether errors in different data sets are commensurable, and hence whether averaging errors across data sets is very meaningful. Nonetheless, a low average error is indicative of a tendency towards low errors for individual data sets.
- **Win/lose/tie record (w/l/t)**. Each record comprises three values that are respectively the number of data sets for which the naive-Bayes classifier trained with one discretization method obtains lower, higher or equal classification error, compared with the naive-Bayes classifier trained with another discretization method.
- **Mean rank**. Following the practice of the Friedman test (Friedman 1937, 1940), for each data set, we rank competing algorithms. The one that leads to the best naive Bayes classification accuracy is ranked 1, the second best ranked 2, so on and so forth. A method's

mean rank is obtained by averaging its ranks across all data sets. The mean rank is less susceptible to distortion by outliers than is the mean error.

- **Nemenyi test**. As recommended by Demsar (2006), to compare multiple algorithms across multiple data sets, the Nemenyi test can be applied to mean ranks of competing algorithms and indicates the absolute difference in mean ranks that is required for the performance of two alternative algorithms to be assessed as significantly different (here we use the 0.05 critical level).

8.4 Observations and analyses

Experimental results are presented and analyzed in this section.

8.4.1 Mean error and average number of formed intervals

Figure 8a depicts the mean error of each discretization method across all data sets, which is further decomposed into bias and variance. It is observed that both PD and FFD achieve the lowest mean error among alternative methods. PD attains the lowest mean bias and FFD the second lowest. LD acquires the lowest mean variance.

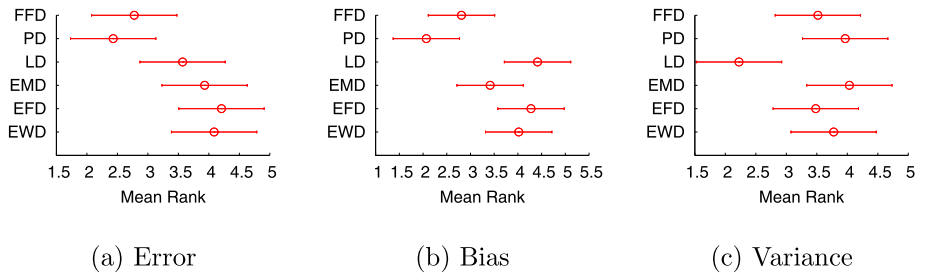
Figure 8b depicts the average number of discrete values formed by each discretization method across all data sets. It reveals that on average, EMD forms the least number of discrete values while FFD forms the most. This partially explains why FFD achieves lower bias than EMD in general. The same reasoning applies to PD against EMD. Note that training instances with identical values are always placed in the same interval. In consequence EFD is not always possible to generate 10 equal frequency intervals.

8.4.2 Win/lose/tie records on error, bias and variance

The win/lose/tie records, which compare each pair of competing methods on classification error, bias and variance respectively, are listed in Table 2. It shows that in terms of reducing bias, both PD and FFD win more often than not compared with every single previous discretization method. PD and FFD do not dominate other methods in reducing variance. Nonetheless, very frequently their gains in bias reductions overwhelm their losses in variance reduction. The end effect is that both PD and FFD win more often than not compared with every single alternative method.

8.4.3 Mean rank and Nemenyi test

Figure 9 illustrates the mean rank of each discretization method as well as applying Nemenyi test to mean ranks. In each subgraph, the mean rank of a method is depicted by a circle. The horizontal bar across each circle indicates the 'critical difference'. The performance of two methods is significantly different if their corresponding mean ranks differ by at least the critical difference. That is, two methods are significantly different if their horizontal bars are not overlapping. Accordingly, it is observed in Fig. 9b that in terms of reducing bias, PD is ranked the best and FFD the second best. Furthermore, PD is statistically significantly better than EWD, EFD and LD. It also wins (although not significantly) against EMD (w/l/t record being 22/4/3 as in Table 2a). FFD is statistically significantly better than LD and EFD. It also wins (although not significantly) against EWD and EMD (w/l/t records being 19/8/2 and 16/11/2 respectively as in Table 2a). Figure 9c suggests that as for variance reduction, there is no significant difference between PD, FFD and alternative methods, except for LD



(a) Error (b) Bias (c) Variance

Fig. 9 Friedman test and Nemenyi test

Table 2 Win/lose/tie records on error, bias and variance for each pair of competing methods

w/l/t	EWD	EFD	EMD	LD	PD
(a) error					
EFD	11/16/2				
EMD	17/12/0	13/15/1			
LD	17/10/2	19/8/2	15/14/0		
PD	22/7/0	22/6/1	21/5/3	20/8/1	
FFD	20/8/1	19/8/2	20/9/0	19/8/2	12/15/2
(b) bias					
EFD	12/16/1				
EMD	18/10/1	19/9/1			
LD	10/14/5	14/12/3	6/16/7		
PD	24/5/0	23/3/3	22/4/3	27/1/1	
FFD	19/8/2	19/10/0	16/11/2	22/7/0	14/14/1
(c) variance					
EFD	13/12/4				
EMD	9/15/5	9/15/5			
LD	21/5/3	23/3/3	22/5/2		
PD	12/14/3	6/17/6	12/12/5	5/20/4	
FFD	17/10/2	11/14/4	15/13/1	11/17/1	13/14/2

which is the most effective method. However, LD's bias reduction is adversely affected by employing EFD to decide its interval frequency. Hence it does not achieve good classification accuracy overall. In contrast, PD and FFD reduce bias as well as control variance. In consequence, as shown in Fig. 9a, they are ranked the best for reducing error, where from the most effective to the least are PD, FFD, LD, EMD, EWD and EFD.

8.4.4 PD and FFD's performance relative to EFD and EMD

We now focus on analyzing PD and FFD's performance relative to EFD and EMD because the latter two are currently the most frequently used discretization methods in machine learning community. Among papers published in 2005 and so far in 2006 by the journal "Machine Learning" and the proceedings of "International Conference on Machine Learning", there are no less than 15 papers on Bayesian classifiers, among which 2 papers assume all

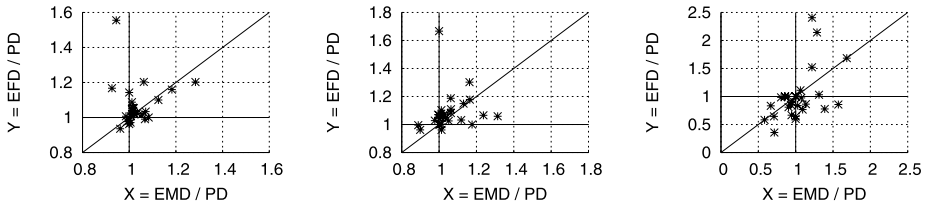


Fig. 10 PD's performance relative to EFD and EMD

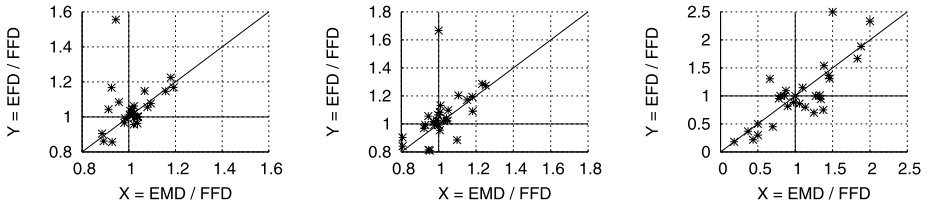


Fig. 11 FFD's performance relative to EFD and EMD

variables being discrete, 6 papers use EFD with $k = 5$ or 10, and 7 papers use EMD. The comparison results are illustrated in Figs. 10, 11 and 12. In each subgraph of Fig. 10, the values on the Y axis are the outcome for EFD divided by that for PD. The values of the X axis are the outcome for EMD divided by that for PD. Each point on the graph represents one of the 29 data sets. Points on the right of the vertical line at $X = 1$ in each subgraph are those for which PD outperforms EMD. Points above the horizontal line at $Y = 1$ indicate that PD outperforms EFD. Points above the diagonal line $Y = X$ represent that EMD outperforms EFD. It is observed that PD is more effective in reducing bias compared with EFD and EMD as the majority of points fall beyond the boundaries $X = 1$ and $Y = 1$ in Fig. 10b. On the other hand, PD is less effective in reducing variance than EFD and EMD as more points fall within the boundaries $X = 1$ and $Y = 1$ in Fig. 10c. Nonetheless, PD's gain in bias reduction dominates. The end effect is that PD outperforms both EFD and EMD in reducing error as the majority of points fall beyond the boundaries $X = 1$ and $Y = 1$ in Fig. 10a. The same lines of reasoning apply to FFD in Fig. 11 as well.

8.4.5 Rival algorithms' performance relative to data set size

Figure 12 depicts PD, FFD, EFD and EMD's classification error, bias and variance respectively with regard to the increase of data set size. The horizontal axis corresponds to data sets whose sizes are increasingly ordered as in Table 1, where the size values are treated as 'nominal' instead of 'numeric'. Please be noted that although it is not justified to connect points with lines since data sets are independent of each other, we do it because we need differentiate among alternative discretization methods. The Y axis represents the classification error obtained by a discretization method on a data set that is normalized by the mean error of all methods on this data set. It is observed that when the data set size becomes large,

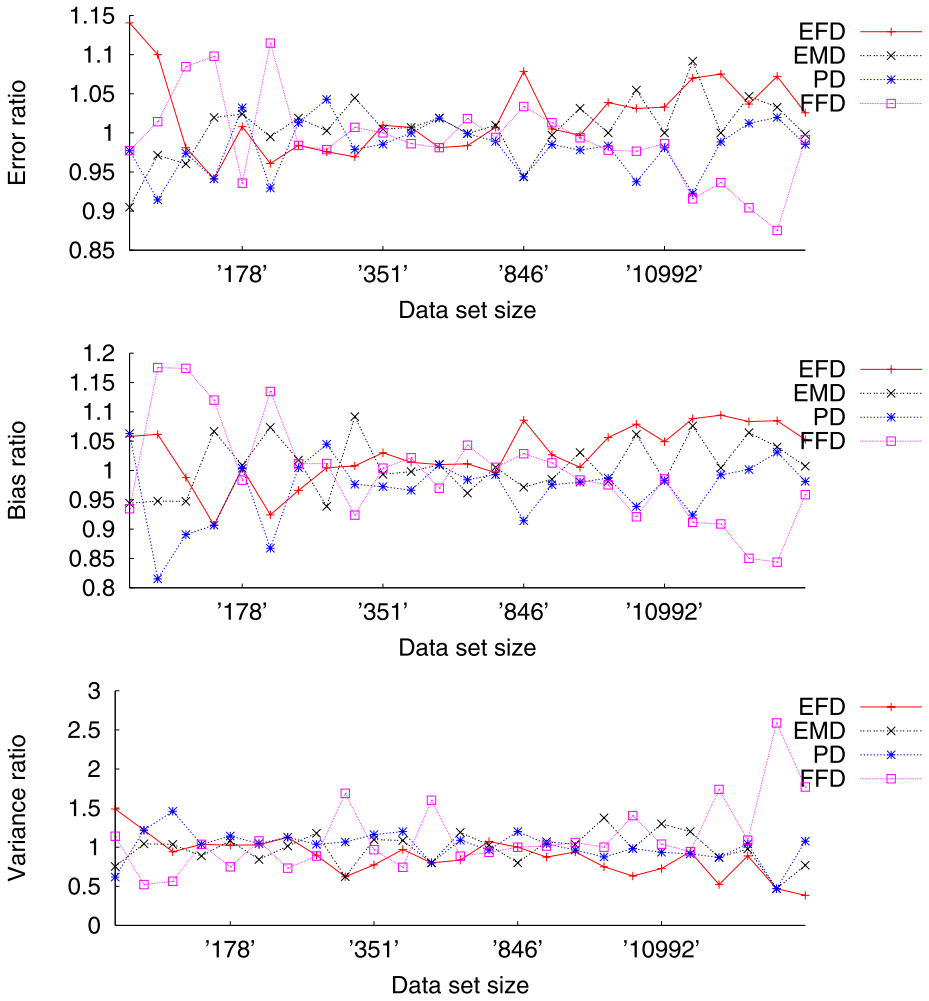


Fig. 12 Classification errors, bias and variance along data set size change

PD and FFD can consistently reduce classification error relative to EFD and EMD. This is very welcome because modern classification applications very often involve large amounts of data. This empirical observation also confirms our theoretical analysis that with training data increasing, in order to reduce classification error, contributing extra data to reducing bias is more effective than to reducing variance.

8.4.6 FFD's bias and variance relative to m

FFD involves a parameter m, the sufficient interval frequency. In this particular paper, we set m as 30 since it is commonly held to be the minimum sample size from which one should draw statistical inferences (Weiss 2002). The statistical inference here is to estimate $p(C{=}c \mid X_i{=}x_i)$ from $p(C{=}c \mid X_i^*{=}x_i^*)$ where the attribute X_i^* is the discretized version of the original quantitative attribute X_i. We have argued that by using m FFD can control

variance while use additional data to decrease bias. It is interesting to explore the effect of different m values on bias and variance. Figure 13 in the Appendix illustrates for each data set NB's classification bias when using FFD with alternative m values (varying from 10 to 100). It is observed that bias monotonically increases with m increasing in many data sets such as Hepatitis, Glass, Satimage, Musk, PioneerMobileRobot, IpumsLa99, CensusIncome and ForestCovertype; and bias zigzags in other data sets such as HeartCleveland, LiverDisorders, CreditScreening and SignLanguage. Nonetheless, the general trend is that bias increases while m increases. The bias when $m = 100$ is higher than the bias when $m = 10$ in 27 data sets out of all 29 data sets. This frequency is statistically significant at the 0.05 critical level according to the one-tailed binomial sign test. Note that for very small data sets such as LaborNegotiations and Echocardiogram, the curves reach a plateau very early. This is because if the number of training instances n is less than or equal to $2m$, FFD simply forms two intervals, each containing approximately $\frac{n}{2}$ instances. For example, LaborNegotiations has 57 instances and thus 38 training instances under 3-fold cross validation. When m becomes equal to or larger than 20, FFD always conducts the same binary discretization. Hence the bias becomes a constant and is no longer dependent on the m value. This limitation is more and more relieved in the succeeding data sets whose sizes become bigger and bigger.

Figure 14 in the Appendix illustrates for each data set NB's classification variance when using FFD with alternative m values (varying from 10 to 100). It is observed that variance monotonically decreases with m increasing in some data sets such as Echocardiogram, Hepatitis, HandwrittenDigits, Adult, CensusIncome and ForestCovertype; and variance zigzags in other data sets such as WineRecognition, Ionosphere, BreastCancer and Annealing. Nonetheless, the general trend is that variance decreases while m increases. The variance when $m = 100$ is lower than the variance when $m = 10$ in 22 data sets out of all 29 data sets. This frequency is statistically significant at the 0.05 critical level according to the one-tailed binomial sign test. Again, small data sets reach a plateau early as explained for bias in the above paragraph.

Because NB's final classification error is a combination of bias and variance, and because bias and variance often present opposite trends with m increasing, how to dynamically choose m to achieve the best trade-off between bias and variance is a domain-dependent problem and is a topic for future research.

8.4.7 Summary

The above observations suggest that

- PD and FFD enjoy an advantage in terms of classification error reduction over the suite of data sets studied in this research.
- PD and FFD better reduce classification bias than alternative methods. Their advantage in bias reduction grows more apparent with the training data size increasing. This supports our expectation that PD and FFD can use additional data to decrease discretization bias, and thus high bias is less likely to attach to large training data any more.
- Although not able to minimize variance, PD and FFD control variance in a way competitive to most existent methods. However, PD tends to have higher variance especially in small data sets. This indicates that among smaller data sets where naive-Bayes probability estimation has a higher risk to suffer insufficient training data, controlling variance by ensuring sufficient interval frequency should have a higher weight than controlling bias. That is why FFD is often more successful at preventing discretization variance from being very high among smaller data sets. Meanwhile, we have also observed that FFD does have higher variance especially in some very large data sets. We suggest the reason is that

m=30 might not be the optimal frequency for those data sets. Nonetheless, the loss is often compensated by their outstanding capability of reducing bias. Hence PD and FFD still achieve lower naive-Bayes classification error more often than not compared with previous discretization methods.

- Although PD and FFD manage discretization bias and variance from two different perspectives, they attain classification accuracy competitive with each other. The win/lose/tie record of PD compared with FFD is 15/12/2.

9 Conclusion

We have proved a theorem that provides a new explanation of why discretization can be effective for naive-Bayes learning. Theorem 1 states that so 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. We have analyzed two factors, decision boundaries and the error tolerance of probability estimation for each quantitative attribute, which can affect discretization's effectiveness. In the process, we have presented a new definition of the useful concept of a decision boundary. We have also analyzed the effect of multiple attributes on these factors. Accordingly, we have proposed the bias-variance analysis of discretization performance. 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 frequency 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 discretization (PD) and fixed frequency discretization (FFD). Both are able to manage discretization bias and variance by tuning interval frequency and interval number. Both are also able to actively take advantage of increasing information in large data to reduce discretization bias as well as control variance. Thus they are expected to outperform previous methods especially when learning from large data. It is desirable that a machine learning algorithm maximize the information that it derives from large data sets, since increasing the size of a data set can provide a *domain-independent* way of achieving higher accuracy (Freitas and Lavington 1996; Provost and Aronis 1996). This is especially important since large data sets 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. The results have demonstrated that our new methods frequently reduce naive-Bayes classification error when compared to previous alternatives. Another interesting issue arising from our empirical study is that simple unsupervised discretization methods (PD and FFD) are able to outperform a commonly-used supervised one (EMD) in our experiments in the context of naive-Bayes learning. This contradicts the previous understanding that EMD tends to have an advantage over unsupervised methods (Dougherty et al. 1995; Hsu et al. 2000; Hsu et al. 2003). Our study suggests it is because EMD was designed for decision tree learning and can be sub-optimal for naive-Bayes learning.

Appendix

Table 3 Naive Bayes' classification error (%) under alternative discretization methods

Data set	EWD	EFD	EMD	LD	PD	FFD
LaborNegotiations	12.3	8.9	9.5	9.6	7.4	9.3
Echocardiogram	29.6	30.0	23.8	29.1	25.7	25.7
Iris	5.7	7.7	6.8	6.7	6.4	7.1
Hepatitis	14.3	14.2	13.9	13.7	14.1	15.7
WineRecognition	3.3	2.4	2.6	2.9	2.4	2.8
Sonar	25.6	25.1	25.5	25.8	25.7	23.3
Glass	39.3	33.7	34.9	32.0	32.6	39.1
HeartCleveland	18.3	16.9	17.5	17.6	17.4	16.9
LiverDisorders	37.1	36.4	37.4	37.0	38.9	36.5
Ionosphere	9.4	10.3	11.1	10.8	10.4	10.7
HorseColic	20.5	20.8	20.7	20.8	20.3	20.6
CreditScreening	15.6	14.5	14.5	13.9	14.4	14.2
BreastCancer	2.5	2.6	2.7	2.6	2.7	2.6
PimaIndiansDiabetes	24.9	25.6	26.0	25.4	26.0	26.5
Vehicle	38.7	38.8	38.9	38.1	38.1	38.3
Annealing	3.8	2.4	2.1	2.3	2.1	2.3
German	25.1	25.2	25.0	25.1	24.7	25.4
MultipleFeatures	31.0	31.8	32.9	31.0	31.2	31.7
Hypothyroid	3.6	2.8	1.7	2.4	1.8	1.8
Satimage	18.8	18.8	18.1	18.4	17.8	17.7
Musk	13.7	18.4	9.4	15.4	8.2	6.9
PioneerMobileRobot	13.5	15.0	19.3	15.3	4.6	3.2
HandwrittenDigits	12.5	13.2	13.5	12.8	12.0	12.5
SignLanguage	38.3	37.7	36.5	36.4	35.8	36.0
LetterRecognition	29.5	29.8	30.4	27.9	25.7	25.5
Adult	18.2	18.6	17.3	18.1	17.1	16.2
IpumsLa99	21.0	21.1	21.3	20.4	20.6	18.4
CensusIncome	24.5	24.5	23.6	24.6	23.3	20.0
ForestCovertype	32.4	33.0	32.1	32.3	31.7	31.9

Table 4 Naive Bayes' classification bias (%) under alternative discretization methods

Data set	EWD	EFD	EMD	LD	PD	FFD
LaborNegotiations	7.7	5.4	6.7	6.3	5.1	6.1
Echocardiogram	22.7	22.3	19.9	22.3	22.4	19.7
Iris	4.2	5.6	5.0	4.8	4.3	6.2
Hepatitis	13.1	12.2	11.7	11.8	11.0	14.5
WineRecognition	2.4	1.7	2.0	2.0	1.7	2.1
Sonar	20.6	19.9	20.0	20.6	19.9	19.5
Glass	24.6	21.1	24.5	21.8	19.8	25.9
HeartCleveland	15.6	14.9	15.7	16.1	15.5	15.6
LiverDisorders	27.6	27.5	25.7	29.6	28.6	27.7
Ionosphere	8.7	9.6	10.4	10.4	9.3	8.8
HorseColic	18.8	19.6	18.9	19.2	18.5	19.1
CreditScreening	14.0	12.8	12.6	12.6	12.2	12.9
BreastCancer	2.4	2.5	2.5	2.5	2.5	2.4
PimaIndiansDiabetes	21.5	22.3	21.2	22.8	21.7	23.0
Vehicle	31.9	31.9	32.2	32.4	31.8	32.2
Annealing	2.9	1.9	1.7	1.7	1.6	1.8
German	21.9	22.1	21.2	22.3	21.0	21.8
MultipleFeatures	27.6	27.9	28.6	27.9	27.2	27.3
Hypothyroid	2.7	2.5	1.5	2.2	1.5	1.5
Satimage	18.0	18.3	17.0	18.0	17.1	16.9
Musk	13.1	16.9	8.5	14.6	7.6	6.2
PioneerMobileRobot	11.0	11.8	16.1	12.9	2.8	1.6
HandwrittenDigits	12.0	12.3	12.1	12.1	10.7	10.5
SignLanguage	35.8	36.3	34.0	35.4	34.0	34.1
LetterRecognition	23.9	26.5	26.2	24.7	22.5	22.2
Adult	18.0	18.3	16.8	17.9	16.6	15.2
IpumsLa99	16.9	17.2	16.9	16.9	15.9	13.5
CensusIncome	24.4	24.3	23.3	24.4	23.1	18.9
ForestCovertype	32.0	32.5	31.1	32.0	30.3	29.6

Table 5 Naive Bayes' classification variance (%) under alternative discretization methods

Data set	EWD	EFD	EMD	LD	PD	FFD
LaborNegotiations	4.6	3.5	2.8	3.3	2.3	3.2
Echocardiogram	6.9	7.7	3.9	6.8	3.2	5.9
Iris	1.5	2.1	1.8	1.9	2.1	0.9
Hepatitis	1.2	2.0	2.2	1.9	3.1	1.2
WineRecognition	1.0	0.7	0.6	0.9	0.7	0.7
Sonar	5.0	5.2	5.5	5.2	5.8	3.8
Glass	14.7	12.6	10.3	10.2	12.8	13.2
HeartCleveland	2.7	2.0	1.8	1.5	2.0	1.3
LiverDisorders	9.5	8.9	11.7	7.3	10.3	8.8
Ionosphere	0.7	0.7	0.7	0.5	1.2	1.9
HorseColic	1.7	1.2	1.7	1.6	1.8	1.5
CreditScreening	1.6	1.7	1.9	1.3	2.1	1.3
BreastCancer	0.1	0.1	0.1	0.1	0.1	0.2
PimaIndiansDiabetes	3.4	3.3	4.7	2.6	4.3	3.5
Vehicle	6.9	7.0	6.7	5.7	6.3	6.1
Annealing	0.8	0.5	0.4	0.6	0.6	0.5
German	3.1	3.1	3.8	2.9	3.7	3.6
MultipleFeatures	3.4	3.9	4.3	3.1	4.0	4.4
Hypothyroid	0.8	0.3	0.3	0.2	0.3	0.3
Satimage	0.8	0.6	1.1	0.4	0.7	0.8
Musk	0.7	1.5	0.9	0.8	0.7	0.6
PioneerMobileRobot	2.5	3.2	3.2	2.4	1.9	1.7
HandwrittenDigits	0.5	0.9	1.4	0.6	1.4	2.0
SignLanguage	2.5	1.4	2.5	1.0	1.8	2.0
LetterRecognition	5.5	3.3	4.2	3.2	3.2	3.3
Adult	0.2	0.3	0.5	0.2	0.5	1.0
IpumsLa99	4.1	4.0	4.4	3.5	4.7	4.9
CensusIncome	0.2	0.2	0.2	0.2	0.2	1.1
ForestCovertype	0.4	0.5	1.0	0.3	1.4	2.3

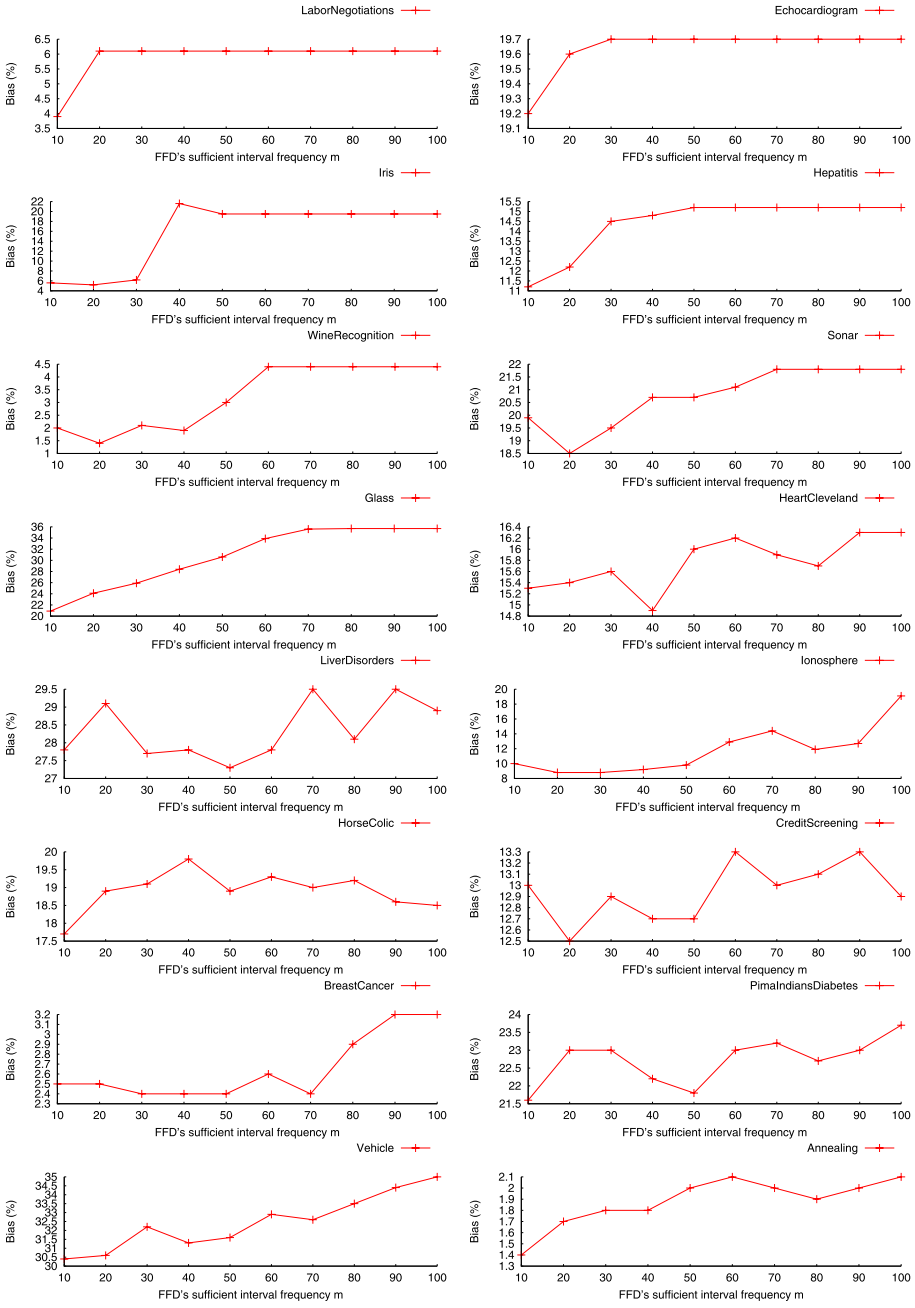


Fig. 13 NB's classification bias when using FFD with alternative m values. Note that very small data sets reach a plateau very early because FFD simply performs binary discretization when the number of training instances n is less than or equal to $2m$

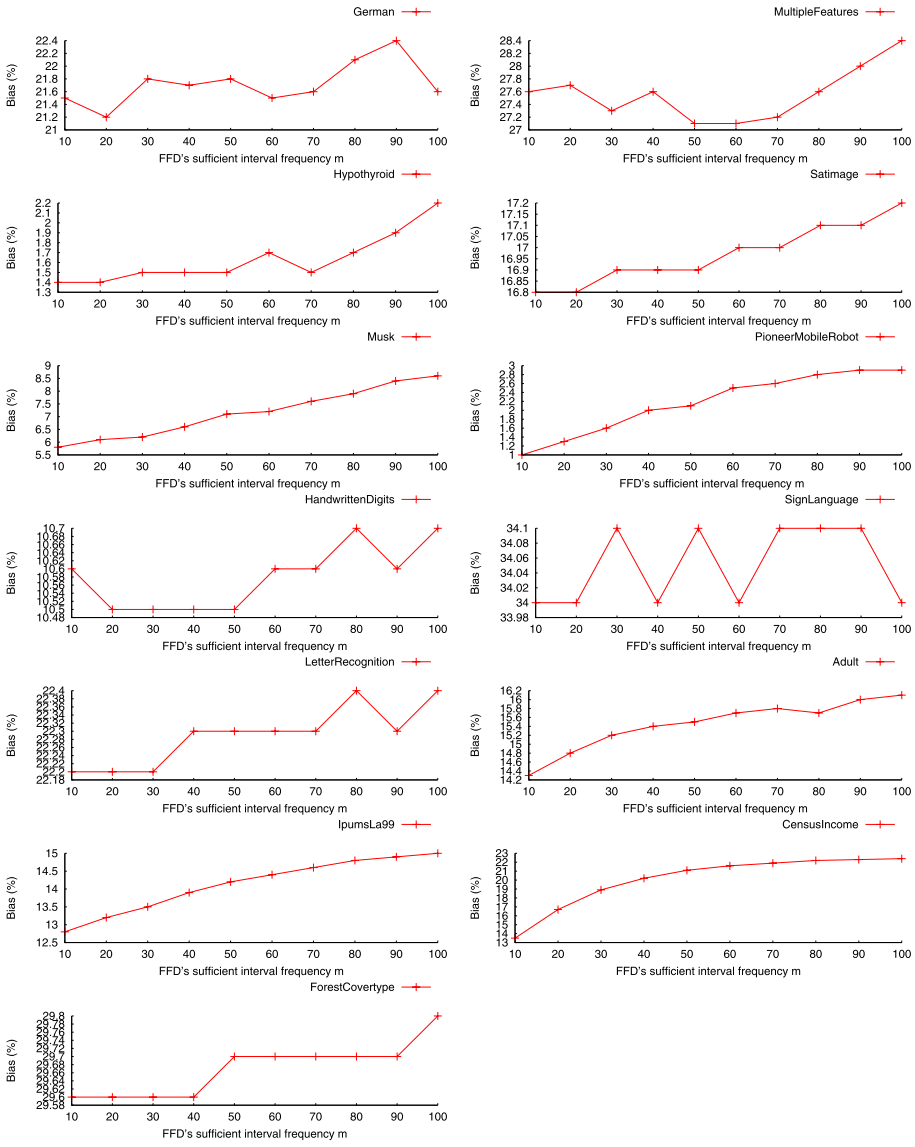


Fig. 13 (*Continued*)

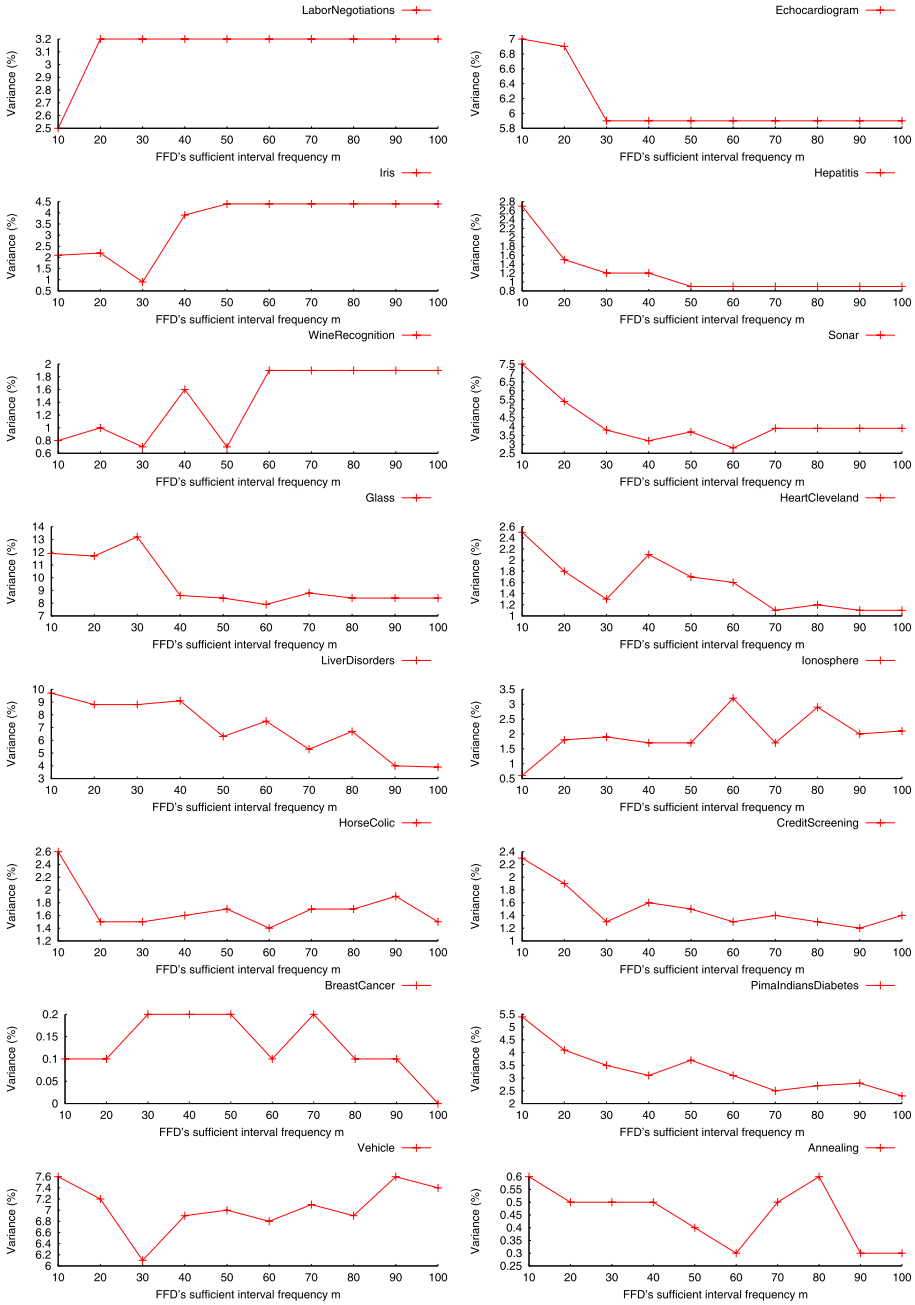


Fig. 14 NB's classification variance when using FFD with alternative m values. Note that very small data sets reach a plateau very early because FFD simply performs binary discretization when the number of training instances n is less than or equal to $2m$

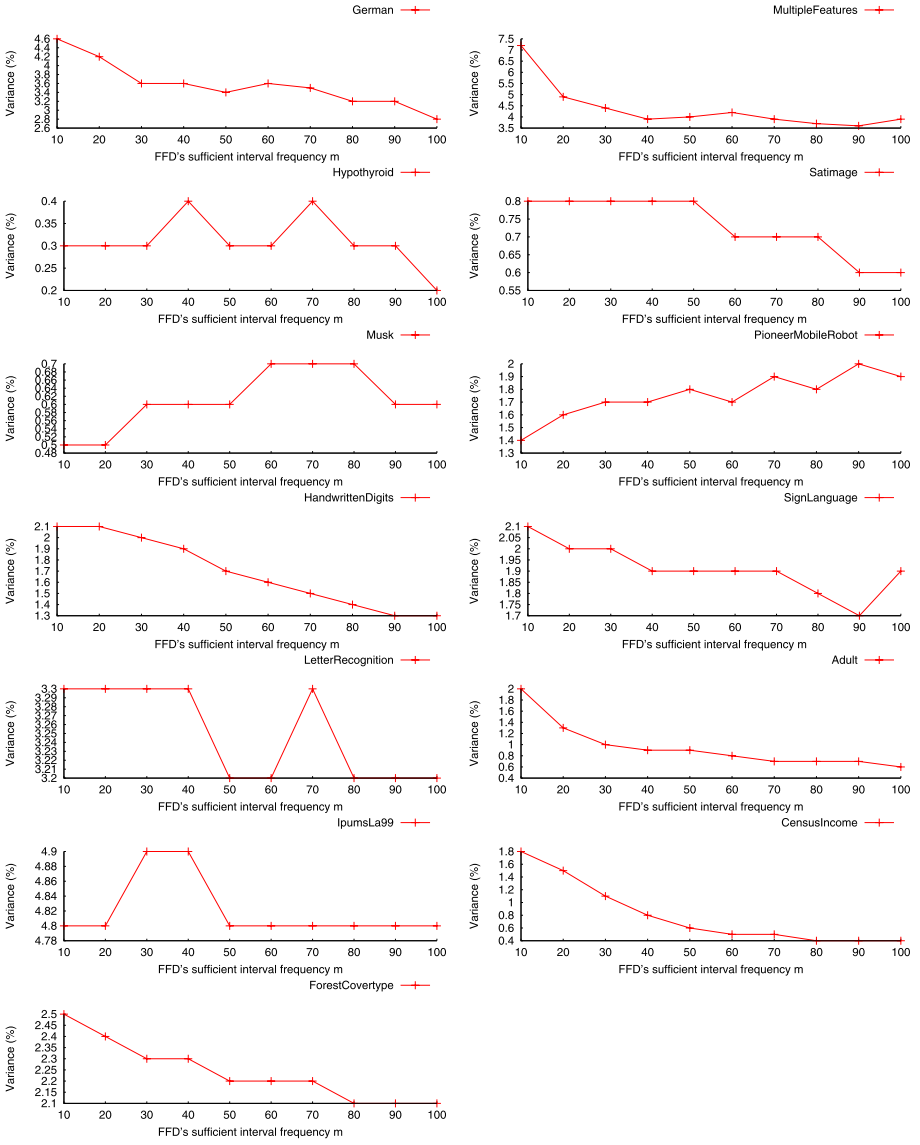


Fig. 14 (*Continued*)

References

Acid, S., Campos, L. M. D., & Castellano, J. G. (2005). Learning Bayesian network classifiers: searching in a space of partially directed acyclic graphs. *Machine Learning*, *59*(3), 213–235.

An, A., & Cercone, N. (1999). Discretization of continuous attributes for learning classification rules. In *Proceedings of the 3rd Pacific-Asia conference on methodologies for knowledge discovery and data mining* (pp. 509–514), 1999.

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), 2000.

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

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

Bluman, A. G. (1992). *Elementary statistics, a step by step approach*. Dubuque: Wm.C. Brown Publishers.

Breiman, L. (1996). *Bias, variance and arcing classifiers* (Technical report 460). Statistics Department, University of California, Berkeley.

Casella, G., & Berger, R. L. (1990). *Statistical inference*. Pacific Grove.

Catlett, J. (1991). On changing continuous attributes into ordered discrete attributes. In *Proceedings of the European working session on learning* (pp. 164–178), 1991.

Cerquides, J., & Mántaras, R. L. D. (2005). TAN classifiers based on decomposable distributions. *Machine Learning*, *59*(3), 323–354.

Cestnik, B. (1990). Estimating probabilities: a crucial task in machine learning. In *Proceedings of the 9th European conference on artificial intelligence* (pp. 147–149), 1990.

Cestnik, B., Kononenko, I., & Bratko, I. (1987). Assistant 86: a knowledge-elicitation tool for sophisticated users. In *Proceedings of the 2nd European working session on learning* (pp. 31–45), 1987.

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 19th international conference on machine learning* (pp. 83–90), 2002.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

Domingos, P., & Pazzani, M. J. (1996). Beyond independence: conditions for the optimality of the simple Bayesian classifier. In *Proceedings of the 13th international conference on machine learning* (pp. 105–112). San Mateo: Morgan Kaufmann Publishers.

Domingos, P., & Pazzani, M. J. (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 12th international conference on machine learning* (pp. 194–202), 1995.

Duda, R., & Hart, P. (1973). *Pattern classification and scene analysis*. New York: Wiley.

Fayyad, U. M., & Irani, K. B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th international joint conference on artificial intelligence* (pp. 1022–1027), 1993.

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 14th British national conference on databases* (pp. 124–133), 1996.

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

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, *32*, 675–701.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics*, *11*, 86–92.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, *29*(2), 131–163.

Gama, J., Torgo, L., & Soares, C. (1998). Dynamic discretization of continuous attributes. In *Proceedings of the 6th Ibero-American conference on AI* (pp. 160–169), 1998.

Hsu, C.-N., Huang, H.-J., & Wong, T.-T. (2000). Why discretization works for naive Bayesian classifiers. In *Proceedings of the 17th international conference on machine learning* (pp. 309–406), 2000.

Hsu, C.-N., Huang, H.-J., & Wong, T.-T. (2003). Implications of the Dirichlet assumption for discretization of continuous variables in naive Bayesian classifiers. *Machine Learning*, *53*(3), 235–263.

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

John, G. H., & Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the 11th conference on uncertainty in artificial intelligence* (pp. 338–345), 1995.

Keogh, E., & Pazzani, M. J. (1999). Learning augmented Bayesian classifiers: a comparison of distribution-based and classification-based approaches. In *Proceedings of international workshop on artificial intelligence and statistics* (pp. 225–230), 1999.

Kerber, R. (1992). Chimerge: Discretization for numeric attributes. In *National conference on artificial intelligence* (pp. 123–128). Menlo Park: AAAI Press.

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), 1996.

Kong, E. B., & Dietterich, T. G. (1995). Error-correcting output coding corrects bias and variance. In *Proceedings of the 12th international conference on machine learning* (pp. 313–321), 1995.

Kononenko, I. (1990). *Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition*. Amsterdam: IOS Press.

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

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

Langley, P. (1993). Induction of recursive Bayesian classifiers. In *Proceedings of the European conference on machine learning* (pp. 153–164), 1993.

Langley, P., & Sage, S. (1994). Induction of selective Bayesian classifiers. In *Proceedings of the 10th conference on uncertainty in artificial intelligence* (pp. 399–406), 1994.

Langley, P., Iba, W., & Thompson, K. (1992). An analysis of Bayesian classifiers. In *Proceedings of the 10th national conference on artificial intelligence* (pp. 223–228), 1992.

Lavrac, N. (1998). Data mining in medicine: selected techniques and applications. In *Proceedings of the 2nd international conference on the practical applications of knowledge discovery and data mining* (pp. 11–31), 1998.

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. (1998). Naive (Bayes) at forty: the independence assumption in information retrieval. In *Proceedings of the 10th European conference on machine learning* (pp. 4–15), 1998.

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

Mitchell, T. M. (1997). *Machine learning*. New York: McGraw-Hill.

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), 2000.

Mooney, R. J., & Roy, L. (2000). Content-based book recommending using learning for text categorization. In *Proceedings of the 5th ACM conference on digital libraries* (pp. 195–204). New York: ACM Press.

Moore, D. S., & McCabe, G. P. (2002). *Introduction to the practice of statistics* (4th ed.). San Francisco: Michelle Julet.

Mora, L., Fortes, I., Morales, R., & Triguero, F. (2000). Dynamic discretization of continuous values from time series. In *Proceedings of the 11th European conference on machine learning* (pp. 280–291), 2000.

Pazzani, M. J. (1995). An iterative improvement approach for the discretization of numeric attributes in Bayesian classifiers. In *Proceedings of the 1st international conference on knowledge discovery and data mining* (pp. 228–233), 1995.

Pazzani, M. J., Merz, C., Murphy, P., Ali, K., Hume, T., & Brunk, C. (1994). Reducing misclassification costs. In *Proceedings of the 11th international conference on machine learning* (pp. 217–225). San Mateo: Morgan Kaufmann.

Perner, P., & Trautzsch, S. (1998). Multi-interval discretization methods for decision tree learning. In *Proceedings of advances in pattern recognition, joint IAPR international workshops SSPR98 and SPR98* (pp. 475–482), 1998.

Provost, F., & Aronis, J. (1996). Scaling up machine learning with massive parallelism. *Machine Learning*, *23*(1), 33–46.

Sahami, M. (1996). Learning limited dependence Bayesian classifiers. In *Proceedings of the 2nd international conference on knowledge discovery and data mining* (pp. 334–338), 1996.

Samuels, M. L., & Witmer, J. A. (1999). *Statistics for the life sciences* (2nd ed.). New York: Prentice-Hall.

Singh, M., & Provan, G. M. (1996). Efficient learning of selective Bayesian network classifiers. In *Proceedings of the 13th international conference on machine learning* (pp. 453–461), 1996.

Starr, B., Ackerman, M. S., & Pazzani, M. J. (1996). Do-I-care: a collaborative web agent. In *Proceedings of the ACM conference on human factors in computing systems* (pp. 273–274), 1996.

Torgo, L., & Gama, J. (1997). Search-based class discretization. In *Proceedings of the 9th European conference on machine learning* (pp. 266–273), 1997.

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

Webb, G. I., Boughton, J., & Wang, Z. (2005). Not so naive Bayes: averaged one-dependence estimators. *Machine Learning*, *58*(1), 5–24.

Weiss, N. A. (2002). *Introductory statistics* (6th ed.). Greg Tobin.
Yang, Y., & Webb, G. I. (2001). Proportional k-interval discretization for naive-Bayes classifiers. In *Proceedings of the 12th European conference on machine learning* (pp. 564–575), 2001.
Yang, Y., & Webb, G. I. (2003). On why discretization works for naive-Bayes classifiers. In *Proceedings of the 16th Australian joint conference on artificial intelligence* (pp. 440–452), 2003.
Zheng, Z., & Webb, G. I. (2000). Lazy learning of Bayesian rules. *Machine Learning*, *41*(1), 53–84.