Pre-publication draft of a paper which appeared in the Proceedings of PAKDD'03 pp 501-512 Publisher: Springer

Weighted Proportional k-Interval Discretization for Naive-Bayes Classifiers

Ying Yang & Geoffrey I. Webb
yyang, geoff.webb@csse.monash.edu.au

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

Abstract. The use of different discretization techniques can be expected to affect the classification bias and variance of naive-Bayes classifiers. We call such an effect *discretization bias* and *variance*. Proportional k-interval discretization (PKID) tunes discretization bias and variance by adjusting discretized interval size and number proportional to the number of training instances. Theoretical analysis suggests that this is desirable for naive-Bayes classifiers. However PKID is sub-optimal when learning from training data of small size. We argue that this is because PKID equally weighs bias reduction and variance reduction. But for small data, variance reduction can contribute more to lower learning error and thus should be given greater weight than bias reduction. Accordingly we propose weighted proportional k-interval discretization (WPKID), which establishes a more suitable bias and variance trade-off for small data while allowing additional training data to be used to reduce both bias and variance. Our experiments demonstrate that for naive-Bayes classifiers, WPKID improves upon PKID for smaller datasets[1] with significant frequency; and WPKID delivers lower classification error significantly more often than not in comparison to three other leading alternative discretization techniques studied.

1 Introduction

Numeric attributes are often discretized for naive-Bayes classifiers [5, 9]. The use of different discretization techniques can be expected to affect the naive-Bayes classification bias and variance. We call such an effect *discretization bias* and *variance*. A number of previous authors have linked the number of discretized intervals to classification error. Pazzani [17] and Mora et al. [16] have mentioned that the interval number has a major effect on the naive-Bayes classification error. If it is too small, important distinctions are missed; if it is too big, the data are over-discretized and the probability estimation may become unreliable. The best interval number depends upon the size of the training data. Torgo and

[1] 'Small' is a relative rather than an absolute term. Of necessity, we here utilize an arbitrary definition, deeming datasets with size no larger than 1000 as 'smaller' datasets, otherwise as 'larger' datasets.

Gama [21] have noticed that an interesting effect of increasing the interval number is that after some threshold the learning algorithm's performance decreases. They suggested that it might be caused by the decrease of the number of training instances per class per interval leading to unreliable estimates due to overfitting the data. Gama et al. [8] have suggested that discretization with fewer number of intervals tends to have greater utility. By minimizing the number of intervals, the dependence of the set of intervals on the training data will be reduced. This fact will have positive influence on the variance of the generated classifiers. In contrast, if there are a large number of intervals, high variance tends to be produced since small variation on the training data will be propagated on to the set of intervals. Hussain et al. [10] have proposed that there is a trade-off between the interval number and its effect on the accuracy of classification tasks. A lower number can improve understanding of an attribute but lower learning accuracy. A higher number can degrade understanding but increase learning accuracy. Hsu et al. [9] have observed that as the interval number increases, the classification accuracy will improve and reach a plateau. When the interval number becomes very large, the accuracy will drop gradually. How fast the accuracy drops will depend on the size of the training data. The smaller the training data size, the earlier and faster the accuracy drops.

Our previous research [24] has proposed proportional k-interval discretization (PKID). To the best of our knowledge, PKID is the first discretization technique that adjusts discretization bias and variance by tuning interval size and interval number. We have argued on theoretical grounds that PKID suits naive-Bayes classifiers. Our experiments have demonstrated that when learning from large data, naive-Bayes classifiers trained by PKID can achieve lower classification error than those trained by alternative discretization methods. This is particularly desirable since large datasets with high dimensional attribute spaces and huge numbers of instances are increasingly used in real-world applications; and naive-Bayes classifiers are widely deployed for these applications because of their time and space efficiency. However, we have detected a serious limitation of PKID which reduces learning accuracy on small data. In this paper, we analyze the reasons of PKID's disadvantage. This analysis leads to *weighted proportional k-interval discretization* (WPKID), a more elegant approach to managing discretization bias and variance. We expect WPKID to achieve better performance than PKID on small data while retaining competitive performance on large data. Improving naive-Bayes classifiers' performance on small data is of particular importance as they have consistently demonstrated strong classification accuracy for small data. Thus, improving performance in this context is improving the performance of one of the methods-of-choice for this context.

The rest of this paper is organized as follows. Section 2 introduces naive-Bayes classifiers. Section 3 discusses discretization, bias and variance. Section 4 reviews PKID and three other leading discretization methods for naive-Bayes classifiers. Section 5 proposes WPKID. Section 6 empirically evaluates WPKID against previous methods. Section 7 provides a conclusion.

2 Naive-Bayes Classifiers

In classification learning, each instance is described by a vector of attribute values and its class can take any value from some predefined set of values. Training data, a set of instances with their classes, are provided. A test instance is presented. The learner is asked to predict the class of the test instance according to the evidence provided by the training data. We define:

– C as a random variable denoting the class of an instance,
– $X < X_1, X_2, \cdots, X_k >$ as a vector of random variables denoting observed attribute values (an instance),
– c as a particular class,
– $x < x_1, x_2, \cdots, x_k >$ as a particular observed attribute value vector (a particular instance),
– $X = x$ as shorthand for $X_1 = x_1 \wedge X_2 = x_2 \wedge \cdots \wedge X_k = x_k$.

Expected classification error can be minimized by choosing $argmax_c(p(C = c \,|\, X = x))$ for each x. Bayes' theorem can be used to calculate the probability:

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

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

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

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

$$p(X = x \,|\, C = c) = p(\wedge X_i = x_i \,|\, C = c) = \prod p(X_i = x_i \,|\, C = c). \tag{3}$$

Combining (2) and (3), one can further estimate the probability by:

$$p(C = c \,|\, X = x) \propto p(C = c) \prod p(X_i = x_i \,|\, C = c). \tag{4}$$

Classifiers using (4) are called naive-Bayes classifiers.

Naive-Bayes classifiers are simple, efficient, effective and robust to noisy data. One limitation, however, is that naive-Bayes classifiers utilize the attributes independence assumption embodied in (3) which is often violated in the real world. Domingos and Pazzani [4] suggest that this limitation is ameliorated by the fact that classification estimation under zero-one loss is only a function of the sign of the probability estimation. In consequence, the classification accuracy can remain high even while the probability estimation is poor.

3 Discretization, Bias and Variance

We here describe how discretization works in naive-Bayes learning, and introduce discretization bias and variance.

3.1 Discretization

An attribute is either categorical or numeric. Values of a categorical attribute are discrete. Values of a numeric attribute are either discrete or continuous [11].

A categorical attribute often takes a small number of values. So does the class label. Accordingly $p(C = c)$ and $p(X_i = x_i \,|\, C = c)$ can be estimated with reasonable accuracy from corresponding frequencies in the training data. Typically, the Laplace-estimate [3] is used to estimate $p(C = c)$: $\frac{n_c+k}{N+n\times k}$, where n_c is the number of instances satisfying $C = c$, N is the number of training instances, n is the number of classes and $k = 1$; and the M-estimate [3] is used to estimate $p(X_i = x_i \,|\, C = c)$: $\frac{n_{ci}+m\times p}{n_c+m}$, where n_{ci} is the number of instances satisfying $X_i = x_i \wedge C = c$, n_c is the number of instances satisfying $C = c$, p is the prior probability $p(X_i = x_i)$ (estimated by the Laplace-estimate) and $m = 2$.

A numeric attribute usually has a large or even an infinite number of values, thus for any particular value x_i, $p(X_i = x_i \,|\, C = c)$ might be arbitrarily close to 0. Suppose S_i is the value space of X_i within the class c, the probability distribution of $X_i \,|\, C = c$ is completely determined by a probability density function f which satisfies [19]:

1. $f(X_i = x_i \,|\, C = c) \geq 0, \forall x_i \in S_i$;
2. $\int_{S_i} f(X_i \,|\, C = c)\mathrm{d}X_i = 1$;
3. $\int_{a_i}^{b_i} f(X_i \,|\, C = c)\mathrm{d}X_i = p(a_i < X_i \leq b_i \,|\, C = c), \forall (a_i, b_i] \in S_i$.

Specifying f gives a description of the distribution of $X_i \,|\, C = c$, and allows associated probabilities to be found [20]. Unfortunately, f is usually unknown for real-world data. Thus it is often advisable to aggregate a range of values into a single value for the purpose of estimating probabilities [5, 9]. Under discretization, a categorical 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. X_i^* instead of X_i is employed for training classifiers. Since $p(X_i^* = x_i^* | C = c)$ is estimated as for categorical attributes, there is no need to assume the format of f. But the difference between X_i and X_i^* may cause information loss.

3.2 Bias and Variance

Error of a machine learning algorithm can be partitioned into a *bias* term, a *variance* term and an *irreducible* term [7, 12, 13, 22]. 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 the 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 x. Suppose we could draw a sequence of training sets $S_1, S_2, ..., S_l$, each of size m, and apply A to construct classifiers, the average error of A at x can be defined as: $Error(A, m, x) = Bias(A, m, x) + Variance(A, m, x) + Irreducible(A, m, x)$.

There is often a 'bias and variance trade-off' [12]. As one modifies some aspect of the learning algorithm, it will have opposite effects on bias and variance. For example, usually as one increases the number of degrees of freedom in the algorithm, the bias decreases but the variance increases. The optimal number of degrees of freedom (as far as the expected loss is concerned) is the number that optimizes this trade-off between bias and variance.

When discretization is employed to process numeric attributes in naive-Bayes learning, the use of different discretization techniques can be expected to affect the classification bias and variance. We call such an effect *discretization bias* and *variance*. Discretization bias and variance relate to *interval size* (the number of training instances in each interval) and *interval number* (the number of intervals formed). The larger the interval $(a_i, b_i]$ formed for a particular numeric value x_i, the more training instances in it, the lower the discretization variance, and thus the lower the probability estimation variance by substituting $(a_i, b_i]$ for x_i. However, the larger the interval, the less distinguishing information is obtained about x_i, the higher the discretization bias, and hence the higher the probability estimation bias. Low learning error can be achieved by tuning the interval size and interval number to find a good trade-off between the discretization bias and variance.

4 Rival Discretization Methods

We here review four discretization methods, each of which is either designed especially for, or is in practice often used by naive-Bayes classifiers. We believe that it is illuminating to analyze them in terms of discretization bias and variance.

4.1 Fixed k-Interval Discretization (FKID)

FKID [5] divides sorted values of a numeric attribute into k intervals, where (given n observed instances) each interval contains n/k instances. Since k is determined without reference to the properties of the training data, this method potentially suffers much attribute information loss. But although it may be deemed simplistic, FKID works surprisingly well for naive-Bayes classifiers. One reason suggested is that discretization approaches usually assume that discretized attributes have Dirichlet priors, and 'Perfect Aggregation' of Dirichlets can ensure that naive-Bayes with discretization appropriately approximates the distribution of a numeric attribute [9].

4.2 Fuzzy Discretization (FD)

FD [14, 15][2] initially discretizes the value range of X_i into k equal-width intervals $(a_i, b_i]$ $(1 \leq i \leq k)$, and then estimates $p(a_i < X_i \leq b_i \,|\, C = c)$ from all training

[2] There are three versions of fuzzy discretization proposed by Kononenko for naive-Bayes classifiers. They differ in how the estimation of $p(a_i < X_i \leq b_i \,|\, C = c)$ is obtained. Because of space limits, we present here only the version that, according to our experiments, best reduces the classification error.

instances rather than from instances that have values of X_i in $(a_i, b_i]$. The influence of a training instance with value v of X_i on $(a_i, b_i]$ is assumed to be normally distributed with the mean value equal to v and is proportional to $P(v, \sigma, i) = \int_{a_i}^{b_i} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-v}{\sigma})^2} \mathrm{d}x$. σ is a parameter to the algorithm and is used to control the 'fuzziness' of the interval bounds. σ is set equal to $0.7 \times \frac{max-min}{k}$ where max and min are the maximum value and minimum value of X_i respectively[3]. Suppose there are N training instances and there are N_c training instances with known value for X_i and with class c, each with influence $P(v_j, \sigma, i)$ on $(a_i, b_i]$ $(j = 1, \cdots, N_c)$: $p(a_i < X_i \leq b_i \,|\, C = c) = \frac{p(a_i < X_i \leq b_i \wedge C = c)}{p(C=c)} \approx \frac{\sum_{j=1}^{N_c} P(v_j, \sigma, i)}{N \times p(C=c)}$. The idea behind FD is that small variation of the value of a numeric attribute should have small effects on the attribute's probabilities, whereas under non-fuzzy discretization, a slight difference between two values, one above and on below the cut point can have drastic effects on the estimated probabilities. But when the training instances' influence on each interval does not follow the normal distribution, FD's performance can degrade.

Both FKID and FD fix the number of intervals to be produced (decided by the parameter k). When the training data are very small, intervals will have small size and tend to incur high variance. When the training data are very large, intervals will have large size and tend to incur high bias. Thus they control well neither discretization bias nor discretization variance.

4.3 Fayyad & Irani's Entropy Minimization Discretization (FID)

FID [6] evaluates as a candidate cut point the midpoint between each successive pair of the sorted values for a numeric attribute. For each evaluation of a 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 candidate cut points. 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. FID was developed in the particular context of top-down induction of decision trees. It uses MDL as the termination condition. This has an effect to tend to minimize the number of resulting intervals, which is desirable for avoiding the fragmentation problem in the decision tree learning [18]. As a result, FID always focuses on reducing discretization variance, but does not control bias. This might work well for training data of small size, for which it is credible that variance reduction can contribute more to lower naive-Bayes learning error than bias reduction [7]. However, when training data size is large, it is very possible that the loss through bias increase will soon overshadow the gain through variance reduction, resulting in inferior learning performance.

[3] This setting of σ is chosen because it achieved the best performance in Kononenko's experiments [14].

4.4 Proportional k-Interval Discretization (PKID)

PKID [24] adjusts discretization bias and variance by tuning the interval size and number, and further adjusts the naive-Bayes' probability estimation bias and variance to achieve lower classification error. As described in Section 3.2, increasing interval size (decreasing interval number) will decrease variance but increase bias. Conversely, decreasing interval size (increasing interval number) will decrease bias but increase variance. PKID aims to resolve this conflict by setting the interval size and number proportional to the number of training instances. With the number of training instances increasing, both discretization bias and variance tend to decrease. Bias can decrease because the interval number increases. Variance can decrease because the interval size increases. This means that PKID has greater capacity to take advantage of the additional information inherent in large volumes of training data than previous methods. Given a numeric attribute, supposing there are N training instances with known values for this attribute, the desired interval size is s and the desired interval number is t, PKID employs (5) to calculate s and t:

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

PKID discretizes the ascendingly sorted values into intervals with size s. Experiments have shown that although it significantly reduced classification error in comparison to previous methods on larger datasets, PKID was sub-optimal on smaller datasets. We here suggest the reason. Naive-Bayes learning is probabilistic learning. It estimates probabilities from the training data. According to (5), PKID gives equal weight to discretization bias reduction and variance reduction by setting the interval size equal to the interval number. When N is small, PKID tends to produce a number of intervals with small size. In particular, small interval size should result in high variance. Thus fewer intervals each containing more instances would be of greater utility.

5 Weighted Proportional k-Interval Discretization

The above analysis leads to *weighted proportional k-interval discretization* (WPKID). This new discretization techniques sets a minimum interval size $\mathbf{m}$. As the training data increase, both the interval size *above* the minimum and the interval number increase. Given the same definitions of N, s and t as in (5), we calculate s and t by:

$$\begin{aligned} s \times t &= N \\ s - \mathbf{m} &= t \\ \mathbf{m} &= 30. \end{aligned} \tag{6}$$

We set $\mathbf{m}$ as 30 since it is commonly held to be the minimum sample from which one should draw statistical inferences [23]. WPKID should establish a more suitable bias and variance trade-off for training data of small size. By introducing $\mathbf{m} = 30$, we ensure that in general each interval has enough instances for reliable probability estimation for naive-Bayes classifiers. Thus WPKID can be expected to improve upon PKID by preventing intervals of high variance. For example, with 100 training instances, WPKID will produce 3 intervals containing approximately 33 instances each, while PKID will produce 10 intervals containing only 10 instances each. At the same time, WPKID still allows additional training data to be used to reduce both bias and variance as PKID does.

6 Experiments

We want to evaluate whether WPKID can better reduce classification errors of naive-Bayes classifiers compared with FKID, FD, FID and PKID.

6.1 Experimental Design

We run experiments on 35 natural datasets from UCI machine learning repository [2] and KDD archive [1]. This experimental suit comprises two parts. One is all the 29 datasets used by PKID [24]. The other is an addition of 6 datasets[4] with size smaller than 1000, since WPKID is expected to improve upon PKID for small data. Table 1 describes each dataset, including the number of instances (Size), numeric attributes (Num.), categorical attributes (Cat.) and classes (Class). Datasets are ascendingly ordered by their sizes and broken down to smaller and larger ones. ‘Small’ is a relative rather than an absolute term. Of necessity, we here utilize an arbitrary definition, deeming datasets with size no larger than 1000 as ‘smaller’ datasets, otherwise as ‘larger’ datasets. For each dataset, we implement naive-Bayes learning by conducting a 10-trial, 3-fold cross validation. For each fold, the training data is separately discretized by FKID ($k = 10$), FD ($k = 10$), FID, PKID and WPKID. The intervals so formed are separately applied to the test data. The experimental results are recorded in Table 1 as: **classification error** is the percentage of incorrect predictions of naive-Bayes classifiers in the test averaged across all folds in the cross validation; and **classification bias and variance** are defined and calculated using the method of Webb [22].

6.2 Experimental Statistics

Three statistics are employed to evaluate the experimental results.

Mean error is the arithmetic mean of errors across all datasets, listed in ‘ME’ rows of Table 1. It provides a gross indication of relative performance. It is debatable whether errors in different datasets are commensurable, and hence whether averaging errors across datasets is very meaningful. Nonetheless, a low average error is indicative of a tendency toward low errors for individual datasets.

[4] They are Pittsburgh-Bridges-Material, Flag-Landmass, Haberman, Ecoli, Dermatology and Vowel-Context.

Table 1. Experimental datasets; and classification error, bias and variance (%)

	Dataset	Size	Num.	Cat.	Class	Error					Bias		Variance	
						WPKID	PKID	FID	FD	FKID	WPKID	PKID	WPKID	PKID
	Labor-Negotiations	57	8	8	2	8.6	7.2	9.5	12.8	8.9	4.6	5.3	4.0	1.9
	Echocardiogram	74	5	1	2	25.7	25.3	23.8	27.7	29.2	19.7	20.8	5.9	4.5
	Pittsburgh-Bridges-Material	106	3	8	3	11.9	13.0	12.6	10.5	12.1	9.9	10.2	2.0	2.8
S	Iris	150	4	0	3	6.9	7.5	6.8	5.3	7.5	4.7	5.5	2.1	1.9
M	Hepatitis	155	6	13	2	15.9	14.6	14.5	13.4	14.7	14.7	12.7	1.2	1.9
A	Wine-Recognition	178	13	0	3	2.0	2.2	2.6	3.3	2.1	1.3	1.1	0.7	1.1
L	Flag-Landmass	194	10	18	6	29.0	30.7	29.9	32.0	30.5	21.5	21.5	7.5	9.2
L	Sonar	208	60	0	2	23.7	25.7	26.3	26.8	25.2	19.3	19.6	4.4	6.2
E	Glass-Identification	214	9	0	3	38.4	33.6	34.9	40.7	34.1	29.7	21.3	8.7	12.3
R	Heart-Disease(Cleveland)	270	7	6	2	16.7	17.5	17.5	16.3	17.1	14.9	15.6	1.8	2.0
	Haberman	306	3	0	2	25.8	27.7	26.5	25.1	27.1	22.3	23.3	3.5	4.4
	Ecoli	336	5	2	8	17.6	19.0	17.9	16.0	19.0	12.3	11.5	5.2	7.4
	Liver-Disorders	345	6	0	2	35.5	38.0	37.4	37.9	37.1	26.3	28.8	9.2	9.2
D	Ionosphere	351	34	0	2	10.3	10.6	11.1	8.5	10.2	8.5	9.8	1.8	0.8
A	Dermatology	366	1	33	6	1.9	2.2	2.0	1.9	2.2	1.5	1.5	0.4	0.7
T	Horse-Colic	368	7	14	2	20.7	20.9	20.7	20.7	20.9	18.9	19.0	1.7	1.8
A	Credit-Screening(Australia)	690	6	9	2	14.3	14.2	14.5	15.2	14.5	12.8	12.1	1.5	2.1
S	Breast-Cancer(Wisconsin)	699	9	0	2	2.7	2.7	2.7	2.8	2.6	2.6	2.6	0.1	0.1
E	Pima-Indians-Diabetes	768	8	0	2	25.5	26.3	26.0	24.8	25.9	22.0	21.8	3.5	4.5
T	Vehicle	846	18	0	4	38.2	38.2	38.9	42.4	40.5	31.5	31.4	6.7	6.8
S	Annealing	898	6	32	6	2.2	2.2	1.9	3.9	2.3	1.9	1.6	0.2	0.5
	Vowel-Context	990	10	1	11	39.2	43.0	41.4	38.0	38.4	20.1	19.2	19.1	23.9
	German	1000	7	13	2	25.4	25.5	25.1	25.2	25.4	22.0	21.7	3.4	3.7
	ME	-	-	-	-	19.0	19.5	19.3	19.6	19.5	14.9	14.7	4.1	4.8
	GM	-	-	-	-	1.00	1.02	1.02	1.05	1.03	1.00	0.99	1.00	1.16
	Multiple-Features	2000	3	3	10	31.4	31.5	32.6	30.8	31.9	27.6	27.3	3.8	4.2
	Hypothyroid	3163	7	18	2	2.1	1.8	1.7	2.6	2.8	1.8	1.6	0.3	0.3
	Satimage	6435	36	0	6	17.7	17.8	18.1	20.1	18.9	16.9	17.0	0.8	0.7
L	Musk	6598	166	0	2	8.5	8.3	9.4	21.2	19.2	7.9	7.6	0.6	0.7
A	Pioneer-MobileRobot	9150	29	7	57	1.8	1.7	14.8	18.2	10.8	0.9	0.8	0.9	0.9
R	Handwritten-Digits	10992	16	0	10	12.2	12.0	13.5	13.2	13.2	10.7	10.7	1.5	1.4
G	Australian-Sign-Language	12546	8	0	3	36.0	35.8	36.5	38.7	38.2	34.2	34.1	1.9	1.8
E	Letter-Recognition	20000	16	0	26	25.7	25.8	30.4	34.7	30.7	22.4	22.5	3.2	3.2
R	Adult	48842	6	8	2	17.0	17.1	17.2	18.5	19.2	16.4	16.6	0.6	0.5
	Ipums-la-99	88443	20	40	13	19.9	19.9	20.1	32.0	20.5	15.3	15.3	4.6	4.6
	Census-Income	299285	8	33	2	23.3	23.3	23.6	24.7	24.5	23.1	23.1	0.2	0.2
	Forest-Covertype	581012	10	44	7	31.7	31.7	32.1	32.2	32.9	30.3	30.3	1.4	1.4
	ME	-	-	-	-	18.9	18.9	20.8	23.9	21.9	17.3	17.2	1.7	1.7
	GM	-	-	-	-	1.00	0.98	1.22	1.47	1.34	1.00	0.98	1.00	0.98

Geometric mean error ratio has been explained by Webb [22]. It allows for the relative difficulty of error reduction in different datasets and can be more reliable than the mean ratio of errors across datasets. The 'GE' rows of Table 1 lists the results of alternative methods against WPKID.

Win/lose/tie record comprises three values that are respectively the number of datasets for which WPKID obtains lower, higher or equal classification error, compared with alternative algorithms. Table 2 shows the results of WPKID compared with alternatives on smaller datasets, larger datasets and all datasets respectively. A one-tailed sign test can be applied to each record. If the test result is significantly low (here we use the 0.05 critical level), it is reasonable to conclude that the outcome is unlikely to be obtained by chance and hence the record of wins to losses represents a systematic underlying advantage to WPKID with respect to the type of datasets studied.

Table 2. Win/Lose/Tie Records of WPKID against Alternatives

Datasets	**Smaller**				**Larger**				**All**			
WPKID	**Win**	**Lose**	**Tie**	**Sign Test**	**Win**	**Lose**	**Tie**	**Sign Test**	**Win**	**Lose**	**Tie**	**Sign Test**
PKID	15	5	3	0.02	4	5	3	0.50	19	10	6	0.07
FID	15	6	2	0.04	11	1	0	< 0.01	26	7	2	< 0.01
FD	11	10	2	0.50	11	1	0	< 0.01	22	11	2	0.04
FKID	17	5	1	< 0.01	12	0	0	< 0.01	29	5	1	< 0.01

6.3 Experimental Results Analysis

WPKID is devised to overcome PKID's disadvantage for small data while retaining PKID's advantage for large data. It is expected that naive-Bayes classifiers trained on data preprocessed by WPKID are able to achieve lower classification error, compared with those trained on data preprocessed by FKID, FD, FID or PKID. The experimental results support this expectation.

1. For Smaller Datasets
   - WPKID achieves the lowest mean error among all the methods.
   - The geometric mean error ratios of the alternatives against WPKID are all larger than 1. This suggests that WPKID enjoys an advantage in terms of error reduction on smaller datasets.
   - With respect to the win/lose/tie records, WPKID achieves lower classification error than FKID, FID and PKID with frequency significant at 0.05 level. WPKID and FD have competitive performance.
2. For Larger Datasets
   - WPKID achieves mean error the same as PKID and lower than FKID, FD and FID.
   - The geometric mean error ratio of PKID against WPKID is close to 1, while those of other methods are all larger than 1. This suggests that WPKID retains PKID's desirable performance on larger datasets.

  - With respect to the win/lose/tie records, WPKID achieves lower classification error than FKID, FD and FID with frequency significant at 0.05 level.
  - WPKID achieves higher classification error than PKID for only one dataset more than the reverse.
3. For All Datasets
  - The win/lose/tie records across all datasets favor WPKID over FKID, FD and FID with frequency significant at 0.05 level. WPKID also achieves lower error more often than not compared with PKID.
  - It seems possible to attribute WPKID's improvement upon PKID primarily to variance reduction. WPKID has lower variance than PKID for 19 datasets but higher variance for only 8. This win/lose record is significant at 0.05 level (sign test = 0.03). In contrast, WPKID has lower bias than PKID for 13 datasets while higher bias for 15.

7 Conclusion

We have previously argued that discretization for naive-Bayes classifiers can tune classification bias and variance by adjusting the interval size and number proportional to the number of training instances, an approach called proportional k-interval discretization (PKID). However, PKID allocates equal weight to bias reduction and variance reduction. We argue that this is inappropriate for learning from small data and propose weighted proportional k-interval discretization (WPKID), which more elegantly manages discretization bias and variance by assigning a minimum interval size and then adjusting the interval size and number as more training data allow. This strategy is expected to improve on PKID by preventing high discretization variance. Our experiments demonstrate that compared to previous discretization techniques FKID, FD and FID, WPKID reduces the naive-Bayes classification error with significant frequency. Our experiments also demonstrate that when learning from small data, WPKID significantly improves on PKID by achieving lower variance, as predicted.

References

1. Bay, S. D. The UCI KDD archive [http://kdd.ics.uci.edu], 1999. Department of Information and Computer Science, University of California, Irvine.
2. Blake, C. L., and Merz, C. J. UCI repository of machine learning databases [http://www.ics.uci.edu/∼mlearn/mlrepository.html], 1998. Department of Information and Computer Science, University of California, Irvine.
3. Cestnik, B. Estimating probabilities: A crucial task in machine learning. In *Proc. of the European Conf. on Artificial Intelligence* (1990), pp. 147–149.
4. Domingos, P., and Pazzani, M. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning 29* (1997), 103–130.
5. Dougherty, J., Kohavi, R., and Sahami, M. Supervised and unsupervised discretization of continuous features. In *Proc. of the Twelfth International Conf. on Machine Learning* (1995), pp. 194–202.

6. Fayyad, U. M., and Irani, K. B. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. of the Thirteenth International Joint Conf. on Artificial Intelligence* (1993), pp. 1022–1027.
7. Friedman, J. H. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery 1*, 1 (1997), 55–77.
8. Gama, J., Torgo, L., and Soares, C. Dynamic discretization of continuous attributes. In *Proc. of the Sixth Ibero-American Conf. on AI* (1998), pp. 160–169.
9. Hsu, C. N., Huang, H. J., and Wong, T. T. Why discretization works for naive Bayesian classifiers. In *Proc. of the Seventeenth International Conf. on Machine Learning* (2000), pp. 309–406.
10. Hussain, F., Liu, H., Tan, C. L., and Dash, M. Discretization: An enabling technique, 1999. Technical Report, TRC6/99, School of Computing, National University of Singapore.
11. Johnson, R., and Bhattacharyya, G. *Statistics: Principles and Methods.* John Wiley & Sons Publisher, 1985.
12. Kohavi, R., and Wolpert, D. Bias plus variance decomposition for zero-one loss functions. In *Proc. of the Thirteenth International Conf. on Machine Learning* (1996), pp. 275–283.
13. Kong, E. B., and Dietterich, T. G. Error-correcting output coding corrects bias and variance. In *Proc. of the Twelfth International Conf. on Machine Learning* (1995), pp. 313–321.
14. Kononenko, I. Naive Bayesian classifier and continuous attributes. *Informatica 16*, 1 (1992), 1–8.
15. Kononenko, I. Inductive and Bayesian learning in medical diagnosis. *Applied Artificial Intelligence 7* (1993), 317–337.
16. Mora, L., Fortes, I., Morales, R., and Triguero, F. Dynamic discretization of continuous values from time series. In *Proc. of the Eleventh European Conf. on Machine Learning* (2000), pp. 280–291.
17. Pazzani, M. J. An iterative improvement approach for the discretization of numeric attributes in Bayesian classifiers. In *Proc. of the First International Conf. on Knowledge Discovery and Data Mining* (1995).
18. Quinlan, J. R. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, 1993.
19. Scheaffer, R. L., and McClave, J. T. *Probability and Statistics for Engineers*, fourth ed. Duxbury Press, 1995.
20. Silverman, B. *Density Estimation for Statistics and Data Analysis.* Chapman and Hall Ltd., 1986.
21. Torgo, L., and Gama, J. Search-based class discretization. In *Proc. of the Ninth European Conf. on Machine Learning* (1997), pp. 266–273.
22. Webb, G. I. Multiboosting: A technique for combining boosting and wagging. *Machine Learning 40*, 2 (2000), 159–196.
23. Weiss, N. A. *Introductory Statistics*, vol. Sixth Edition. Greg Tobin, 2002.
24. Yang, Y., and Webb, G. I. Proportional k-interval discretization for naive-Bayes classifiers. In *Proc. of the Twelfth European Conf. on Machine Learning* (2001), pp. 564–575.