To appear in the 17th International Conference on Machine Learning (ICML-2000)

Why Discretization Works for Naïve Bayesian Classifiers

Chun-Nan Hsu CHUNNAN@IIS.SINICA.EDU.TW
Institute of Information Science, Academia Sinica, Nankang 115, Taipei City, Taiwan

Hung-Ju Huang GIS85563@CIS.NCTU.EDU.TW
Department of Computer and Information Science, National Chiao-Tung University, Hsinchu City 300, Taiwan

Tzu-Tsung Wong TZU-TSUN@IIS.SINICA.EDU.TW
Lunghwa Institute of Technology, Kueishan 333, Taoyuan County, Taiwan

Abstract

This paper explains why well-known discretization methods, such as entropy-based and ten-bin, work well for naive Bayesian classifiers with continuous variables, regardless of their complexities. These methods usually assume that discretized variables have Dirichlet priors. Since *perfect aggregation* holds for Dirichlets, we can show that, generally, a wide variety of discretization methods can perform well with insignificant difference. We identify situations where discretization may cause performance degradation and show that they are unlikely to happen for well-known methods. We empirically test our explanation with synthesized and real data sets and obtain confirming results. Our analysis leads to a *lazy* discretization method that can simplify the training for naive Bayes. This new method can perform as well as well-known methods in our experiment.

1. Introduction

Learning a naive Bayesian classifier (a.k.a. naive Bayes) (Langley et al., 1992) from data is an important technique for data classification. In spite of its simplicity, naive Bayes constantly outperforms competing algorithms in the experiments reported in the literature. Remarkably, in KDD-CUP-97, two of the top three contestants are based on naive Bayes (Parsa, 1997). Meanwhile, Domingos and Pazzani (1997) provided an interesting analysis of the reason why naive Bayes can perform well even though the independence assumption is violated. They reported an experiment that compared naive Bayes with several classical learning algorithms such as C4.5 with a large ensemble of real data sets. The result again showed that naive Bayes can significantly outperform those algorithms. But a more interesting respect of their result is that it was achieved by a version of naive Bayes that uses a very "naive" discretization method to handle continuous data. That method, usually referred to as "ten-bin," divides the domain of each continuous variable into ten equal-width bins. Such a simple method was considered to be inferior and many complex methods have been proposed, but surprisingly, simple methods perform well while not much improvement is found when complex methods are used. Preciously, Dougherty et al. (1995) conducted an empirical study comparing the performance of four well-known discretization methods and concluded that all of the discretization methods can significantly outperform a version of naive Bayes that assumes normal for all continuous variables. Moreover, these discretization methods performed approximately the same, though an entropy-based method (Fayyad & Irani, 1993) performed slightly better for some data sets. Kohavi and Sahami (1996) compared the entropy-based method with an error-based method, which is not considered in (Dougherty et al., 1995). Again, both methods outperformed the normal version whereas no significant difference between their performance was found.

Previous work provided some informal discussion but none explained why these discretization methods can work well for naive Bayes regardless of their complexities. In this paper, we present an explanation on this phenomenon. The key of our explanation is the "*perfect aggregation*" property of the Dirichlet distributions. A discretized continuous variable is usually assumed to have a Dirichlet prior. Perfect aggregation of Dirichlets implies that we can estimate the class-conditional probabilities of discretized intervals with an arbitrary accuracy. This explains why naive Bayes with discretization can be effective in approximating a broad range of probability distributions of continuous data. We also found that to achieve optimal performance, the probability corresponding to an interval must simulate the role of the true density in the classification. This allows us to identify situations where discretization may cause performance degradation. We show that these situations are unlikely to happen under zero-one loss for a wide variety of discretization methods including the well-known methods. Therefore, these methods will produce similar performance for many data sets regardless of their complexities. We also empirically test our explanation on synthesized and real data sets and obtain confirming results.

Our analysis leads to a "lazy" discretization method. Experimental results show that on average this new method performs equally well as well-known methods. The remainder of this parer presents our explanation and the lazy discretization method.

2. Dirichlet Distributions

In a naive Bayesian classifier, discrete variables as well as discretized continuous variables are usually assumed to have a Dirichlet prior. This section reviews the important properties of Dirichlet distributions. A Dirichlet distribution is formally defined as follows:

Definition 1 *Random vector* $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_k)$ *has a* $\boldsymbol{k}$**-variate Dirichlet distribution** *with parameters* $\alpha_j > 0$ *for* $j = 1, 2, \dots, k+1$ *if it has density*

$$f(\boldsymbol{\theta}) = \frac{\Gamma(\sum_{j=1}^{k+1} \alpha_j)}{\prod_{j=1}^{k+1} \Gamma(\alpha_j)} \prod_{j=1}^{k} \theta_j^{\alpha_j - 1} (1 - \theta_1 - \cdots - \theta_k)^{\alpha_{k+1} - 1}$$

for $\theta_1 + \theta_2 + \cdots + \theta_k \leq 1$ *and* $\theta_j \geq 0$ *for* $j = 1, 2, \dots, k$. *This distribution will be denoted* $D_k(\alpha_1, \alpha_2, \dots \alpha_k; \alpha_{k+1})$. *A* **Beta distribution** *is a univariate Dirichlet distributions and usually denoted* $Beta(\alpha_1, \alpha_2)$. □

The following "closure properties" of Dirichlet distributions are critical to our analysis. These properties greatly simplify the computation of the moments of the Dirichlet distribution in Bayesian analysis. Suppose random vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_k)$ has a Dirichlet distribution $D_k(\alpha_1, \alpha_2, \dots, \alpha_k; \alpha_{k+1})$, then by Wilks (1962) any subvector $(\theta_{n1}, \theta_{n2}, \dots, \theta_{nm})$ of $\boldsymbol{\theta}$ has an m-variate Dirichlet distribution

$$D_m(\alpha_{n1}, \alpha_{n2}, \dots, \alpha_{nm}; \alpha - \textstyle\sum_{j=1}^{m} \alpha_{nj}).$$

We call this *subvector lemma*. Also by Wilks (1962), the sum of any subset $\Phi \subseteq \{\theta_1, \theta_2, \dots, \theta_k\}$ has a Beta distribution with parameters $\sum_{j\in\Phi} \alpha_j$ and $\alpha - \sum_{j\in\Phi} \alpha_j)$, where $\alpha = \alpha_1 + \alpha_2 + \cdots + \alpha_{k+1}$. This is called *sum-of-subset lemma*.

Another important property of Dirichlets is that a Dirichlet distribution is *conjugate* to the multinomial sampling (Heckerman, 1998). This property basically states that the posterior distribution of a Dirichlet given our observation is also a Dirichlet. Formally, let $D = \{y_1, y_2, \dots, y_{k+1}\}$ be a data set for the outcomes in n trials, where y_j denotes the number of trials turning to be outcome j. When the prior distribution $p(\boldsymbol{\theta})$ is a Dirichlet distribution with parameters α_j for $j = 1, 2, \dots, k+1$, and the likelihood function $L(D|\boldsymbol{\theta})$ follows a multinomial distribution, then the posterior distribution $p(\boldsymbol{\theta}|D)$ is also a Dirichlet distribution with parameters $\alpha'_j = \alpha_j + y_j$ for $j = 1, 2, \dots, k+1$. Similarly, the Beta distribution is conjugate to the binomial sampling. The expected value for θ_j given D is $E[\theta_j|D] = \frac{\alpha_j + y_j}{\alpha + n}$, for $j = 1, 2, \dots, k+1$.

3. Perfect Aggregation

Let Φ be a subset of $\{\theta_1, \theta_2, \dots, \theta_k\}$, and let the probability of interest q be the sum of the variables in Φ; i.e. , $q = \sum_{j\in\Phi} \theta_j$. Suppose the prior distribution of $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_k)$ is a Dirichlet distribution $D_k(\alpha_1, \alpha_2, \dots, \alpha_k; \alpha_{k+1})$. A straightforward application of the Bayesian approach is to use the training data D to update the prior distribution to obtain the posterior distribution $f(\boldsymbol{\theta}|D)$, which is a Dirichlet distribution $D_k(\alpha_1 + y_1, \alpha_2 + y_2, \dots, \alpha_k + y_k; \alpha_{k+1} + y_{k+1})$. Then the posterior distribution $f(q|D)$ is derived from $f(\boldsymbol{\theta}|D)$. By subvector lemma and sum-of-subset lemma, $f(q|D)$ is a Beta distribution:

$$Beta(\textstyle\sum_{j\in\Phi}(\alpha_j + y_j), \alpha + n - \sum_{j\in\Phi}(\alpha_j + y_j)) \quad (1)$$

However, by the properties of the Dirichlet distribution, there exists a simpler alternative to compute $f(q|D)$. We can first derive the prior distribution for the probability of interest $f(q)$ from the prior distribution of $f(\boldsymbol{\theta})$. Then we can convert the training data D into a new set of training data D' in terms of q by computing the sum of the observations of our interest in D. Now, we can use D' to update the prior distribution of $f(q)$ to obtain the posterior distribution $f(q|D')$.

We can show that in general it is always the case that $f(q|D) = f(q|D')$. Since the multinomial likelihood function $L(D|\boldsymbol{\theta})$ implies that the trials for obtaining data set D are all independent, the likelihood function $L(D'|q)$ will follow a binomial distribution. By sum-of-subset lemma, the prior distribution of $f(q)$ has a beta distribution $Beta(\sum_{j\in\Phi} \alpha_j, \alpha - \sum_{j\in\Phi} \alpha_j)$ when $\boldsymbol{\theta}$ has a Dirichlet distribution. Since the beta distribution is conjugate to the binomial sampling, the posterior distribution $f(q|D')$ will have a Beta distribution with parameters $\sum_{j\in\Phi}(\alpha_j + y_j)$ and $\alpha + n - \sum_{j\in\Phi}(\alpha_j + y_j)$. This is exactly the same as Equation (1). This property is always true for the Dirichlet distribution and is first derived by Azaiez (1993), called *"perfect aggregation"* (Iwasa et al., 1987; Wong, 1998).

For example, suppose that we are interested in the probability of showing odd number in throwing a die. Let θ_j be the probability that the die shows number j in a trial, and let y_j be the number of trials that the die shows j in n trials. Then the probability of interest can be represented as $q = \theta_1 + \theta_3 + \theta_5$. In the straightforward approach, we derive the distribution $f(q|D)$ from the data $\{y_1, y_2, \ldots, y_6\}$, while in the alternative approach, we can use $D' = \{n, y_1 + y_3 + y_5\}$ instead and will obtain the same result. An implication of perfect aggregation is that with Dirichlet distributions, to estimate the posterior probability of a union of disjoint events, there is no need to estimate the probabilities of individual events.

4. Discretization for Naïve Bayes

Naive Bayes classifies a feature vector $\mathbf{x}$ by selecting class c that maximizes the posterior probability

$$p(c|\mathbf{x}) \propto p(c) \prod_{x\in\mathbf{x}} p(x|c), \tag{2}$$

where x is a variable in $\mathbf{x}$. $p(x|c)$ is the *class-conditional density* of x given class c. Let $\boldsymbol{\theta}$ denote the vector whose elements are the parameters of the density of $p(x|c)$. In a Bayesian learning framework, we assume that $\boldsymbol{\theta}$ is an uncertain variable (Heckerman, 1998) and can be learned from a training data set. This estimation is at the heart of training in a naive Bayes.

Suppose x is a discrete variable with $k+1$ possible values. In principle the class label c of the data vector $\mathbf{x}$ dictates the probability of the value of x. Thus the appropriate p.d.f. is a multinomial distribution and its parameters are a set of probabilities $\{\theta_1, \theta_2, \ldots \theta_{k+1}\}$ such that for each possible value X_j, $p(x = X_j|c) = \theta_j$ and $\sum_j \theta_j = 1$. Now, let $\boldsymbol{\theta} \equiv (\theta_1, \theta_2, \ldots \theta_k)$. We choose a Dirichlet distribution with parameters $\alpha_1, \ldots, \alpha_{k+1}$ as the prior for $\boldsymbol{\theta}$. Given a train data set, we can update $p(x = X_j|c)$ by its expected value:

$$\hat{p}(x = X_j|c) = \frac{\alpha_j + y_{cj}}{\alpha + n_c}, \tag{3}$$

where n_c is the number of the training examples belonging to class c and y_{cj} is the number of class c examples whose $x = X_j$. Since a Dirichlet distribution is conjugate to multinomial sampling, after the training, the posterior distribution of $\boldsymbol{\theta}$ is still a Dirichlet, but with the updated parameters $\alpha_j + y_{cj}$ for all j. This property allows us to incrementally train a naive Bayes.

In practice, we usually choose the Jaynes prior (Almond, 1995) $\alpha_j = \alpha = 0$ for all j and have $\hat{p}(x|c) = \frac{y_{cj}}{n_c}$. However when the training data set is too small, this often yields $\hat{p}(x|c) = 0$ and impedes the classification. To avoid this problem, another popular choice is $\alpha_j = 1$ for all j. This is known as *smoothing* or *Laplace's estimate* (Cestnik & Bratko, 1991).

If x is a continuous variable, a conventional approach is to assume that $p(x|c) = N(x; \mu_c, \sigma_c^2)$, where $N(x; \mu_c, \sigma_c^2)$ is the p.d.f. of a normal distribution. In this case, training involves learning the parameters μ_c and σ_c from the training data. This approach has been shown to be less effective than discretization when $p(x|c)$ is not normal (John & Langley, 1995), and discretization is often used. Generally, discretization involves partitioning the domain of x into $k+1$ intervals as a pre-processing step. Then we can treat x as a discrete variable with $k+1$ possible values and conduct the training and classification.

More precisely, let I_j be the j-th discretized interval. Training and classification in a naive Bayes with discretization is to use $\hat{p}(x \in I_j|c)$ as an estimate of $\hat{p}(x|c)$ as in Equation (3) for each continuous variable. This is equivalent to assuming that after discretization, the class-conditional density of x has a Dirichlet prior. We call this assumption *"Dirichlet discretization assumption"*. Apparently, this assumption holds for all well-known discretization methods, including ten-bin, entropy-based, etc. See (Dougherty et al., 1995) for a comprehensive survey.

Dirichlet discretization assumption is reasonable because of another implicit assumption described below. Let $f(x|c)$ be the "true" probability density function of $p(x|c)$. Assuming that $f(x|c)$ is integrable everywhere. Then for any discretized interval I_j, the "true" probability of $p(x \in I_j|c) = \int_{I_j} f(x|c)dx$. By choosing equivalent sample size α, the Dirichlet parameter corresponding to random variable "$x \in I_j|c$" is $\alpha \int_{I_j} f(x|c)dx$. We call this assumption *"partition independence assumption."* By partition independence assumption, any discretization of x can have a Dirichlet prior. Partition independence assumption implies that for any interval, the Dirichlet parameter corresponding to this interval depends only on the area below the curve of the p.d.f. $f(x|c)$, but is independent of the shape of the curve in the interval.

5. Performance of Discretization Methods

In this section, we discuss the factors that affect the performance of different discretization methods. When the density of a continuous variable x is known, the point density $p(x|c)$ will be used in Equation (2) for classification. But when we discretize the variable, the probability $\hat{p}(x \in I|c)$ of the discretized interval I will be used instead. This probability is assumed to have a Dirichlet prior due to Dirichlet discretization assumption. Since perfect aggregation holds for Dirichlet distributions, $\hat{p}(x \in I|c)$ can be estimated arbitrarily close to real $p(x \in I|c)$.

This can be proved as follows. Let $f(x|c)$ be the "true" density of $p(x|c)$, then

$$p(x \in I|c) = \int_{x \in I} f(x|c)dx \approx \sum_{\Delta \subseteq I} \hat{p}(x \in \Delta|c)$$

where Δ is a disjoint sub-interval in I. Arbitrary accuracy can be achieved by increasing the number of sub-intervals in I. However, by perfect aggregation, there is no need to estimate $\hat{p}(x \in \Delta|c)$ for each Δ. Instead, we can estimate $\hat{p}(x \in I|c)$ directly and obtain the same result. By partition independence assumption, this result is independent of the shape of the curve of $f(x|c)$. Therefore, discretization methods can more accurately approximate a distribution than a naive Bayes that assumes normal for continuous variables. This shows why discretization can be effective.

Suppose we have the "true" densities of $p(x|c)$, it can be shown that Equation (2) (i.e., the Bayes rate) is the optimal classification accuracy achievable by any classifier (Duda & Hart, 1973). With discretization, to achieve the optimal classification accuracy, $\hat{p}(x \in I|c)$ must simulate the role of $p(x|c)$ by distinguishing the class that gives x high density from the class that gives x low density. More precisely, if $p(x = X_j|c_i)$ is the largest among all classes c_i, then at least we should require that $\hat{p}(x \in I|c_i)$ is the largest among all classes, too. Otherwise, performance degradation may still occur.

Consider a simple case with only one continuous variable x with normally distributed classes. We can plot the curves of the conditional densities as the one shown in Figure 1. The domain of x can be divided into regions by *decision boundaries*, which correspond to the intersection points of the curves where ties occur among the largest conditional densities. The optimal classification is to pick the class c such that $p(x|c)$ is the largest, and the pick of the class is different when x is on different sides of a decision boundary.

Suppose a discretization method creates an interval I that contain a decision boundary. For x values in the interval but at the different sides of the decision boundary, the classifier should pick a different class for each value. But we will have the same $\hat{p}(x \in I|c)$ for these x values. That is, x values at one of the two sides of the decision boundary will be misclassified. Therefore, to minimize the distortion, the boundaries of intervals that contain decision boundaries should be close to the decision boundaries, or these intervals should be as narrow as possible.

On the other hand, the widths of intervals that contain no decision boundary can be as large as possible. Their widths do not affect too much on the performance but when these intervals are too narrow, the number of examples in the training data set in the intervals will be too small to allow accurate estimation of $\hat{p}(x \in I|c)$. In this case, discretization will cause performance degradation. A loose bound can be derived as follows. Let n be the size of the training data set. The number of examples that will fall in an interval I has a binomial distribution with parameters n and $p(x \in I|c)$ and has expected value $n * p(x \in I|c)$. If we want the estimate to be accurate to two digits, we will need $n > 100/\hat{p}(x \in I|c)$. This bound is loose, but is sufficient to illustrate our idea. The entropy-based method and 1RD (see (Dougherty et al., 1995)) are designed to avoid narrow intervals, and thus their performance can be slightly better.

In summary, to avoid performance degradation, a discretization method should partition the domain of a continuous variable into intervals such that their cut points are close to decision boundaries to minimize the distortion due to the discretization, and their width should be large enough to cover sufficient examples.

The above analysis can be extended to multivariate cases and the "right" choice of cut points can be widely spread. Since a naive Bayes takes all variables into account simultaneously, the impact of a "wrong" discretization for one variable can be absorbed by other variables under zero-one loss performance measure. Also, due to the way that a naive Bayes learns, correct classification of a given example depends only on the interval containing the values of that example, and is completely independent of how other regions are discretized. As a result, a wide variety of discretization methods can have similar performance in general.

6. Empirical Evidence

In this section, we report the empirical verification of our analysis on synthesized and real data sets.

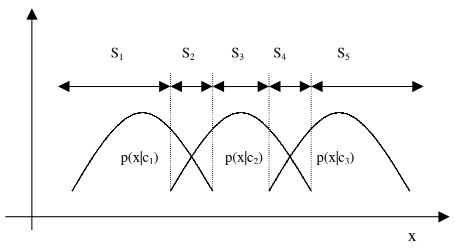

Figure 1. Conditional distributions of the synthesized data

6.1 Controlled Experiments

We synthesized an artificial data set composed of 1500 examples with three different classes. Each class has the same number of examples. Each example has two continuous attributes A_1, A_2 and a class label. The values of A_1 and A_2 are generated by normal distributions $\{N(10, 2), N(15, 2)\}$ for class 1, $\{N(15, 2), N(20, 2)\}$ for class 2, and $\{N(20, 2), N(10, 2)\}$ for class 3. For each attribute, we divided its domain into five intervals as shown in Figure 1 such that S_2 and S_4 contain the decision boundaries of each variable. That is, the intersection points that are also the peak values of all classes.

In the first experiment, we investigated how the distance of the cut points of S_2 and S_4 to the intersection points affect the performance. S_2 and S_4 are initialized with a small width and then their boundaries are extended with a constant percentage of the initial width of S_3. The intersection points within S_2 and S_4 were kept in the middle such that the regions at both sides of the intersection points within the intervals always have the same width. For each resulting discretization, we ran a five-fold cross-validation to obtain an average accuracy. Figure 2 plots the results. As we predicted, when S_2 and S_4 become wide, the accuracies drop rapidly.

The second experiment is to see if there are other factors that may result in significant changes of the accuracies in addition to the width of S_2 and S_4. We first kept S_2 and S_4 unchanged and randomly partitioned S_1, S_3, and S_5 into at most fifty intervals. We repeated the random partition thirty times and ran a five-fold cross-validation for each repetition. The variance of the resulting accuracies is 0.27. The average accuracy is 96.25% with the maximum and minimum accuracies being 96.67% and 95.47%, respectively. We repeated the same experiment, but this time, the entire domains are randomly partitioned. By contrast, the variance of the resulting accuracies jumps to 3.84. The average accuracy is 93.79% with the maximum and minimum accuracies being 96.80% and 82.73%, respectively. The

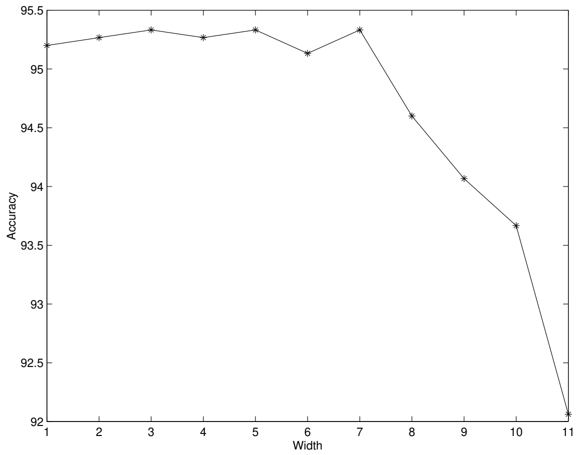

Figure 2. Accuracies drop when we increase the widths of the intervals that contain the intersection points.

results show that when the widths of S_2 and S_4 are fixed, the partitions of the other intervals have only tiny impact on the classification accuracy.

The third experiment is to show that "equal-width bins" such as ten-bin can be effective in many cases. In fact, if the number of bins is not too "extreme," the performance will not be significantly different. Our analysis predicts that when the number of intervals is two, it is likely that these intervals will contain intersection points and the performance will be generally poor. As the number of bins increases, the accuracies will improve and reach a plateau. When the number of bins is extremely large, the widths of the bins will be too small for all intervals and the accuracies will drop gradually. That is, the curve of the accuracy as a function of the number of bins will appear like an up-side-down "J". Moreover, how fast the curve drops will depend on the size of the training data set. A small data set will cause the accuracy to drop faster.

We conducted the experiment on two data sets: our synthesized data set (with 1500 examples) and its subset (with 300 examples). The resulting curves are given in Figure 3, which confirms our prediction.

6.2 Experiments on Real Data sets

We select ten real data sets from UCI ML repository (Blake & Merz, 1998) for our experiments. All attributes of those data sets are continuous. The first experiment is similar to the third experiment described in Section 6.1. In this experiment,we partitioned each continuous variable into two equal-width bins at first, and then increased the number of the equal-width bins by one until 50. Figure 4 shows the results.

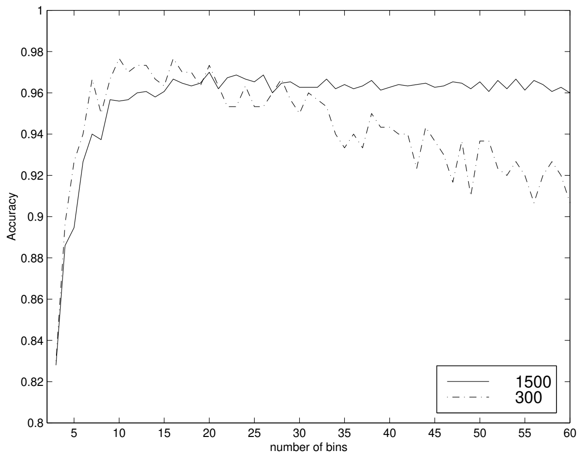

Figure 3. Experimental results match our prediction; also, with fewer data, accuracies drop faster as the number of bins increases.

Just like our expectation, the curves for most data sets match the pattern given in Figure 3, especially for large data sets. All accuracy figures are obtained by using five-fold cross-validation. The variances of the accuracies for most data sets are found to be surprisingly small. As long as the numbers of equal-width bins are not too large or too small, the accuracies are approximately the same. The curves of some data sets (`glass`, `iris`, `liver`, `wine`) are rougher than others, since the variances of the accuracy appear to depend on the size of the data sets. In general, small data sets (with less than 300 examples) have relatively large variances. As discussed earlier, this is because the examples are not sufficient for accurate probability estimation. Data set `glass` has a particularly zigzag curve because the number of examples in this data set is small. In addition, this data set has seven classes, which produce too many intersection points and the accuracies become sensitive to discretization.

In the above experiment, we focus on domains that are entirely continuous. We also performed the same experiment for some data sets with mixed domains. The curves of those mix-domain data sets again generally match the pattern. We examined these data sets that do not follow the pattern and found that the continuous variables in those exceptional data sets have almost no impact in classifying an example. We remove continuous variable from exceptional data sets, and their accuracies for classification are almost as good as before. When a continuous variable in a data set is nearly dormant in classification, the distributions of different classes of that variable will be similar everywhere. Therefore, a naive Bayes with only two bins can still achieve a high accuracy.

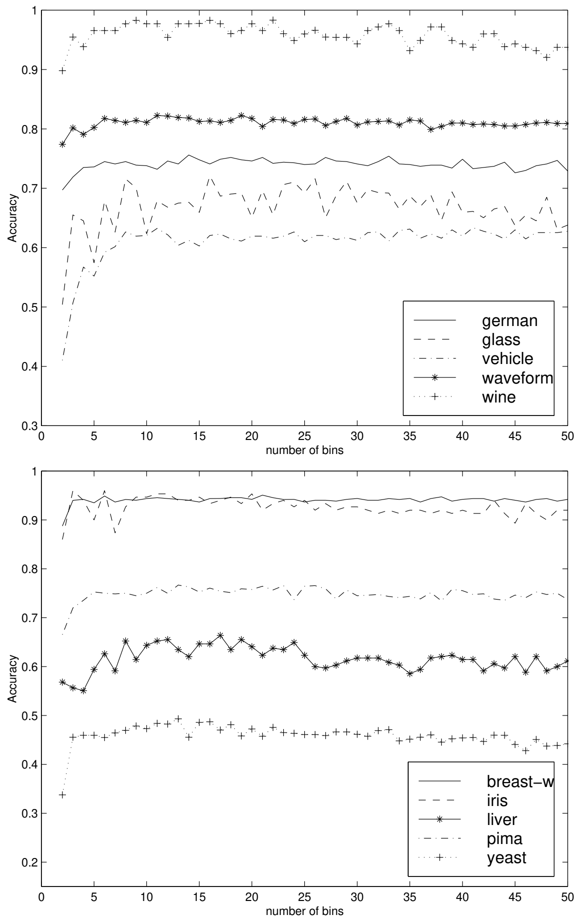

Figure 4. Accuracy results by increasing the number of equal-width bins.

However, the results presented above does not exclude the possibility that by carefully determining the cut points, the performance will be improved drastically. Therefore, we conducted a similar experiment with the entropy-based method. Since the entropy-based method always selects a cut point from training data at the boundary where the classes switch (Fayyad & Irani, 1993), it is more likely for this method to select a cut point near an intersection point.

For each data set, we recursively called the entropy heuristic to determine the cut points and reported the five-fold cross-validation accuracy. In each recursion, cut points are selected in the resulting intervals of the previous calls without applying MDL stopping criterion. Figure 5 shows the resulting curves. The horizontal axis indicates how many levels of the recursion have been invoked. At the l^{th} level of the recursion, a continuous variable will have at most 2^l cut points. However, when all of the data containing in an interval have the same class, no cut point will be selected.

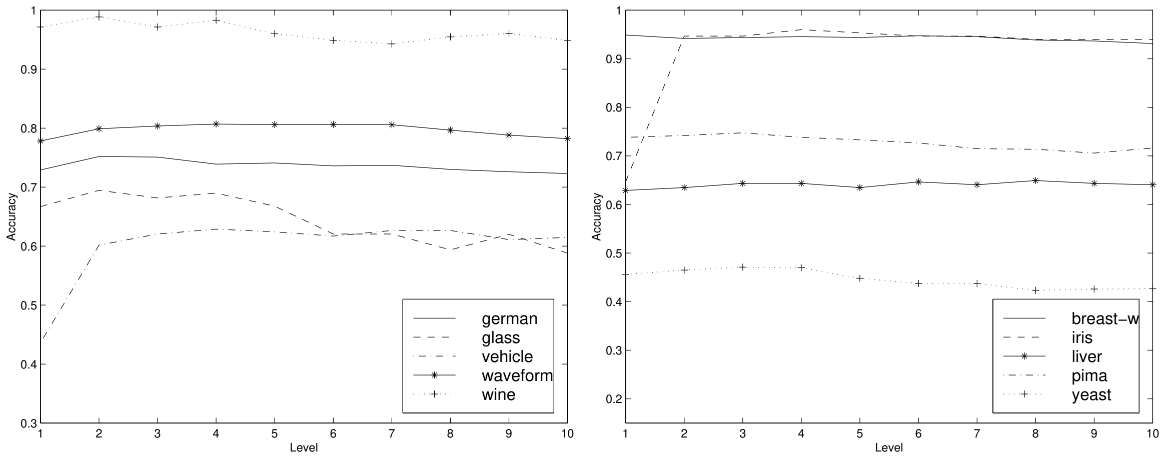

Figure 5. Accuracy results by using entropy to determine cut points.

Therefore, the number of the levels of recursions does not correspond to the number of the intervals created, and each variable may be partitioned into a different number of intervals.

The resulting curves show that the accuracies with different levels of recursions are surprisingly similar for all data sets. Their shapes match the pattern of the curves in Figure 3. This again confirms that discretization methods can perform similarly even though they select radically different sets of cut points.

7. Lazy Discretization

The next question is whether any possible improvement can be made from this analysis. This section describes our preliminary investigation on this question. We propose a lazy discretization method based on perfect aggregation. This method waits until one or more test data are given to determine the cut points for each continuous variable. Moreover, this method produces only a pair of cut points surrounding the value of each test datum for each variable. That is, it creates only one interval (denoted as I) and leaves the other region untouched. From the training data, we can estimate $\hat{p}(x \in I|c)$ by Equation (3) and use this estimate to classify the test datum. This method can invoke different instantiations to determine the cut points. For example, we can select a pair of cut points such that the value of x is in the middle and the width of the interval is the same as the width of the intervals created by ten-bin. We will call this instantiation "lazy ten-bin." Similarly, we can have "lazy entropy," "lazy bin-log," etc.

This discretization method is derived from the perfect aggregation and other properties of the Dirichlet distribution. Suppose that the partition independence assumption holds. Then by the sum-of-subset lemma of the Dirichlet distribution, "$x|c$" will have a Beta prior with parameters $\alpha \int_I p(x|c)dx$ and $\alpha(1 - \int_I p(x|c)dx)$, where c is a class and α is an equivalent sample size. By perfect aggregation, we can estimate $\hat{p}(x \in I|c)$ by counting how many c examples with $x \in I$ and how many are not. In other words, there is no need to check the exact value of x for those examples whose value of x is not in I and we may simplify the training effort.

We empirically compared lazy ten-bin, ten-bin, entropy-based, and a normal version of naive Bayes on the real data sets. We ran a five-fold cross-validation for each data set and reported the average and the standard deviation of the accuracies. Table 1 gives the results, which reveal that the lazy method can perform as well as the well-known discretization methods.

8. Conclusions and Future Work

In this paper, we presented an explanation of why discretization works for naive Bayes based on the perfect aggregation property of the Dirichlet distributions, and why well-known discretization methods perform about the same regardless of their complexities. The explanation is based on perfect aggregation of Dirichlets, which ensures that naive Bayes with discretization can effectively approximate the distribution of a continuous variable. We also presented a new lazy discretization method and showed that it works equally well. We note that our result may not be applicable to other learning algorithms such as decision trees.

In our future work, we plan to investigate whether our analysis can shed some light on handling continuous variables in general Bayesian networks.

Table 1. Accuracies of naive Bayes with different discretization methods.

DATA SET	LAZY TEN-BIN	TEN BINS	ENTROPY	CONTINUOUS
BREAST-W	96.00±3.08	94.37±2.16	94.37±1.93	92.79±2.61
GERMAN	74.00±2.98	73.80±2.77	74.70±2.84	70.00±2.49
GLASS	64.43±5.29	62.34±6.03	67.12±3.60	47.58±4.96
IRIS	96.00±2.49	94.67±3.40	95.33±1.63	96.00±3.89
LIVER	64.64±4.26	64.35±5.84	65.22±2.25	56.81±5.09
PIMA	76.18±4.84	75.01±4.64	74.47±2.92	74.34±2.57
VEHICLE	61.23±1.75	62.06±1.39	62.29±2.15	43.26±3.82
WAVEFORM	81.75±3.22	81.08±2.87	82.17±2.87	82.00±2.92
WINE	97.19±1.81	97.71±2.14	97.14±1.81	97.14±2.52
YEAST	48.44±2.77	47.30±2.85	44.82±3.76	39.39±5.31
AVERAGE	75.99	75.27	75.76	69.93
WIN:LOSS	–	8:2	5:5	9:1

Acknowledgements

The research reported here was supported in part by the National Science Council in Taiwan under Grant No. NSC 89-2213-E-001-031.

References

Almond, R. (1995). *Graphical belief modelling.* New York: Chapman and Hall.

Azaiez, M. N. (1993). *Perfect aggregation in reliability models with Bayesian updating.* Doctoral dissertation, Department of Industrial Engineering, University of Wisconsin-Madison, Madison, WI.

Blake, C., & Merz, C. (1998). UCI repository of machine learning databases.

Cestnik, B., & Bratko, I. (1991). On estimating probabilities in tree pruning. In *Machine learning – EWSL-91, European working session on learning*, 138–150. Berlin, Germany: Springer-Verlag.

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

Dougherty, J., Kohavi, R., & Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. *Machine Learning: Proceedings of the Twelfth International Conference.* San Francisco, CA: Morgan Kaufmann.

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

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

Heckerman, D. (1998). A tutorial on learning with Bayesian networks. In M. I. Jordan (Ed.), *Learning in graphical models*, 301–354. Boston: Kluwer Academic Publishers.

Iwasa, Y., Levin, S., & Andreasen, V. (1987). Aggregation in model ecosystem: Perfect aggregation. *Ecological Modeling*, *37*, 287–302.

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

Kohavi, R., & Sahami, M. (1996). Error-based and entropy-based discretization of continuous features. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (pp. 114–119). Portland, OR.

Langley, P., Iba, W., & Thompson, K. (1992). An analysis of bayesian classifiers. *Proceedings of the tenth national conference on Artificial Intelligence* (pp. pp. 223–228). San Jose, CA: AAAI Press.

Parsa, I. (1997). KDD-CUP 1997 presentation.

Wilks, S. S. (1962). *Mathematical statistics.* New York: Wiley and Sons.

Wong, T.-T. (1998). *Perfect aggregation in dependent Bernoulli systems with Bayesian updating.* Doctoral dissertation, Department of Industrial Engineering, University of Wisconsin-Madison, Madison, WI.