Weighted Naïve Bayes Classifiers by Renyi Entropy

Tomomi Endo and Mineichi Kudo

Division of Computer Science
Graduate School of Information Science and Technology
Hokkaido University, Sapporo, Japan
{tomomi,mine}@main.ist.hokudai.ac.jp

Abstract. A weighted naïve Bayes classifier using Renyi entropy is proposed. Such a weighted naïve Bayes classifier has been studied so far, aiming at improving the prediction performance or at reducing the number of features. Among those studies, weighting with Shannon entropy has succeeded in improving the performance. However, the reasons of the success was not well revealed. In this paper, the original classifier is extended using Renyi entropy with parameter α. The classifier includes the regular naïve Bayes classifier in one end ($\alpha = 0.0$) and naïve Bayes classifier weighted by the marginal Bayes errors in the other end ($\alpha = \infty$). The optimal setting of α has been discussed analytically and experimentally.

1 Introduction and Related Studies

In the field of pattern recognition, there are various kinds of large-scale datasets. The features expressing an object may be continuous, discrete or categorical, and sometimes all these kinds of features appear at the same time. Such a feature set containing more than one kind of features is called a *mixed feature set*. In a large-scale dataset with mixed features, we have to solve two problems: 1) how we deal with mixed features consistently and effectively and 2) how we suppress the bad effect due to many features useless for classification (feature selection). Especially, feature selection for large-scale datasets is desired to have a low computational complexity, e.g., linear or less in the number of features.

The authors had proposed a weighted naïve Bayes classifiers in which every continuous/ordered feature was converted into categorical one, coping with the first issue, and feature weights were introduced to reduce the effective number of features, coping with the second issue. In the weighting, the degree of importance of each feature was measured by a Shannon entropy of data and the computation cost was linear in the number of features. The proposed classifier, indeed, succeeded to reduce a large number of features without a large degradation of performance [1]. However, in the viewpoint of performance improvement, it was not satisfactory. It was better in only a few cases compared with the naïve Bayes classifiers. This is probably because the way of using Shannon entropy was not optimal. Therefore, we examine the validity to use a more general Renyi entropy and analyze the property in the same framework.

J. Ruiz-Shulcloper and G. Sanniti di Baja (Eds.): CIARP 2013, Part I, LNCS 8258, pp. 149–156, 2013.

The studies related to weighted naïve Bayes classifiers are mainly divided into two groups. One group aims at choosing a small number of features or at shrinking weights of useless features [2,3]. Another group aims at improving the performance by controlling the weights appropriately [4,5]. For example, by regarding each term as a new feature and the weight as a coefficient, we can control these coefficients as in the same way of linear classifiers or of linear support vector machines.

Some of these studies, however, suffer from the large computation cost [3,4]. The others could not improved the performance as expected or could not deal with mixed features appropriately. Therefore, we proposed another way [1] in which all features were converted into categorical features and the weights were derived from the Shannon entropy or mutual information of each feature.

After we proposed the method, we noticed a very similar study by Chang-Hwan Lee *et al.* [6]. In their method, the weights are derived from a different formulation via Kullback-Leibler divergence, but the resultant weights coincide with ours. They showed the effectiveness of their approach as well as ours. The difference is that they normalized the weights within a finite range while we left them as they were, preferring two extreme values of zero and infinity.

2 Weighted Naïve Bayes and Proposed Methods

The naïve Bayes classifier is a Bayes classifier simplified by the assumption of independence between features in each class. It assigns a class label $c^* \in \{1, 2, \ldots, K\}$ for a class-unknown sample $x = (x_1, x_2, \ldots, x_D)$ by the rule:

$$\begin{aligned}
c^* &= \arg\max_c P(c \mid x) \quad \text{(maximum posterior method)} \\
&= \arg\max_c P(x \mid c)P(c) \quad \text{(Bayes rule)} \\
&= \arg\max_c P(c) \prod_{d=1}^{D} P(x_d \mid c) \quad \text{(Independence assumption)} \\
&= \arg\max_c \left\{ \log P(c) + \sum_{d=1}^{D} \log P(x_d \mid c) \right\} \\
&= \arg\max_c \left\{ \log P(c) + \sum_{d=1}^{D} \log \frac{P(c \mid x_d)P(x_d)}{P(c)} \right\} \\
&= \arg\max_c \left\{ \log P(c) + \sum_{d=1}^{D} \log \frac{P(c \mid x_d)}{P(c)} \right\}.
\end{aligned}$$

We have derived the last two formulae because it is more natural to use $P(c \mid x_d)$ than $P(x_d \mid c)$ when x_d is one value of a categorical feature. Indeed, we can estimate the probability by counting the number of samples taking the value x_d class-wisely without a special assumption of distribution necessary for estimation of $P(x_d \mid c)$. Here, we use the base 2 for log through the paper.

Referring to this assignment rule, we consider the following discriminant functions to be maximized by c^*:

$$\delta^c(x) = w_0 \log \frac{P(c)}{P_0(c)} + \sum_{d=1}^{D} w_d \log \frac{P(c|x_d)}{P(c)}, \quad P_0(c) = 1/K. \tag{1}$$

This is different from many studies including our previous study in the point that is includes w_0 and $P_0(c)$. These modifications are made for two reasons: 1) to control the degree of affection by the prior probability by w_0, e.g., make $w_0 = 0$ if $P(c) = 1/K$ and 2) to bring consistency and interpretability to every term as pieces of evidence to support class c, e.g., a positive log-odds $\log \frac{P(c|x_d)}{P(c)}$, i.e., $P(c|x_d) > P(c)$, gives a positive piece of evidence to support class c by knowing the value of x_d and a negative odds gives a negative piece of evidence.

In this study, we convert continuous and discrete features into categorical ones to enable a unified treatment of mixed features. This is because it is hard to give a reasonable metric between categorical values, e.g. people's names. Conversely, discretizing a continuous value does not always mean a loss of significant information. Discretization sometimes even improves the performance of classifiers [7]. In this paper, we simply use, equally-spaced bins within the minimum and maximum values of training samples because our objective is to investigate the effect of the proposed weighting method. The number of bins is in common set to $\log N$ where N is the number of samples.

3 Extension by Renyi Entropies

When we introduce a weight on each term in the discrimination function, we have to determine two things: 1) how we measure the importance of each feature and 2) how we connect the degree of importance to the corresponding weight.

For the second issue, we use a monotonically decreasing function $w(h)$ of entropy h such that $w(h) \to 0$ as $h \to \log K$ and $w(h) \to \infty$ as $h \to 0$. This property is required to achive 1) features having no or less information for classification should be given zero or a smaller value of weight and 2) a feature having a "perfect" information for classification, if any, should govern the other features by taking a very large value of weight. In this study, we use a class of functions:

$$w(h) = \frac{e^{-ah} - e^{-ac_0}}{1 - e^{-ah}}, \quad \text{where } h = H(\mathsf{C}|\mathsf{X}),\ c_0 = H(C),\ a \geq 0, \tag{2}$$

where $H(\mathsf{C}|\mathsf{X})$ is the conditioned entropy given a feature X. Since $w(h) \to (c_0 - h)/h$ as $a \to 0$, this weight function includes our previous one [1]. For avoiding many parameters included, we use $a = 1$ in this paper.

As a measure of importance on the first issue, we use Renyi entropy that is a general entropy taking a value in $[0, \log K]$, where K is the number of possible events. The formal definition with a random variable C is given by

$$H_\alpha(\mathsf{C}) = \frac{1}{1-\alpha} \log \sum_{i=1}^{K} p_i^\alpha \quad \alpha \geq 0,\ \alpha \neq 1. \tag{3}$$

This includes Shannon entropy as a special case of when $\alpha \to 0$. The other two special cases are $H_0(\mathsf{C}) = \log \#\{p_i > 0\}$ $(= \log K$ in most cases), and $H_\infty(\mathsf{C}) = -\log \max_i p_i = -\max_i \log p_i$. The following monotonicity holds: $\log K = H_0 \geq H_1 = H \geq H_2 \geq \cdots \geq H_\infty$.

3.1 Bounds by the Prediction Error

Next, let us make clear the relationship between a conditional Renyi entropy and a Bayes error. In the following, we will first show lower and upper bounds of Renyi entropy $H_\alpha(\mathsf{C}|\mathsf{X} = x)$ by a prediction error $\epsilon(x) = 1 - \max_c P(\mathsf{C} = c|\mathsf{X} = x)$ at a point x. Here C is a random variable for class and X is a random variable for one feature.

For $H_\alpha = \frac{1}{1-\alpha} \log \sum_{i=1}^{K} p_i^\alpha$ and $\epsilon = 1 - \max_i p_i$ on a probability distribution $\mathbf{p} = (p_1, p_2, \ldots, p_K)$, we have derived lower bounds $\phi_\alpha(\epsilon)$ and upper bounds $\Phi_\alpha(\epsilon)$ of H_α defined by $\phi_\alpha(\epsilon) = \frac{1}{1-\alpha} \log \{i(1-\epsilon)^\alpha - (K-1)\,(i\epsilon - i + 1)^\alpha\}$, $\frac{i-1}{i} \leq \epsilon < \frac{i}{i+1}$, $i = 1, 2, \ldots, K$ and $\Phi_\alpha(\epsilon) = \frac{1}{1-\alpha} \log \left\{(1-\epsilon)^\alpha - (K-1)\left(\frac{\epsilon}{K-1}\right)^\alpha\right\}$, In particular, $\phi_\infty(\epsilon) = \Phi_\infty(\epsilon) = -\log(1-\epsilon)$. These bounds are shown in Fig. 1. These bounds are extensions of a known result for Shannon entropy [8]. The derivation is easily understood by considering two extreme distributions attaining the minimum and maximum. Since both $\phi_\alpha(\epsilon)$ and $\Phi_\alpha(\epsilon)$ are monotonically increasing in ϵ, we may regard Renyi entropy as a measure of importance of knowing the value x.

Now let us show the relationship between the Bayes error $\epsilon_{\text{Bayes}} = \sum_x P(x)\epsilon(x)$ and the *conditional* Renyi entropy $H_\alpha(\mathsf{C}|\mathsf{X}) = \sum_x P(x) H_\alpha(\mathsf{C}|\mathsf{X} = x)$. Since the point $(\epsilon_{\text{Bayes}}, H_\alpha(\mathsf{C}|\mathsf{X}))$ is given by averaging similar points at $\mathsf{X} = x$ as $\sum_x P(x)(\epsilon(x), H_\alpha(\mathsf{C}|\mathsf{X} = x))$, the point is included in the convex region bounded by $\phi_\alpha(\epsilon)$, $\Phi_\alpha(\epsilon)$ and two special functions: $\phi^*(\epsilon) =$ lines connecting $(0,0)$, $(1/2, \log 2)$, …, $((K-1)/K, \log K)$ and $\Phi_*(\epsilon) = \frac{K \log K}{K-1}\epsilon$ (Fig. 1). Then we have the lower and upperbound as:

$$\{\phi^*(\epsilon_{\text{Bayes}}), \phi_\infty(\epsilon_{\text{Bayes}}), \phi_\infty(\epsilon_{\text{Bayes}})\} \leq H_\alpha(\mathsf{C}|\mathsf{X}) \leq \{\Phi_\alpha(\epsilon_{\text{Bayes}}), \Phi_1(\epsilon_{\text{Bayes}}), \Phi_*(\epsilon_{\text{Bayes}})\},$$

where $\{A, B, C\}$ corresponds to three cases $\{\alpha \leq 1, 1 < \alpha < \infty, \alpha = \infty\}$. We inverted relationships with Φ_α^{-1} or ϕ_α^{-1} from their strict monotonicity. Note in Fig. 1 that $H_\infty(\mathsf{C}|\mathsf{X})$ is close to $-\log(1 - \epsilon_{\text{Bayes}})$ for small values of ϵ_{Bayes}.

3.2 Weights with H_0, H_∞ and H_α

We will show that 1) a weighted naïve Bayes classifier with H_0 simulates the original naïve Bayes classifier without weights and 2) weights with H_∞ reflect the marginal Bayes errors.

When we concentrate on a feature represented by X, for example, dth feature, our weight w_d with Renyi entropy becomes

$$w_d = w_\alpha(h) = \frac{e^{-h} - e^{-c_0}}{1 - e^{-h}}, \quad \text{where } h = H_\alpha(\mathsf{C}|\mathsf{X}),\ c_0 = H_\alpha(\mathsf{C}). \tag{4}$$

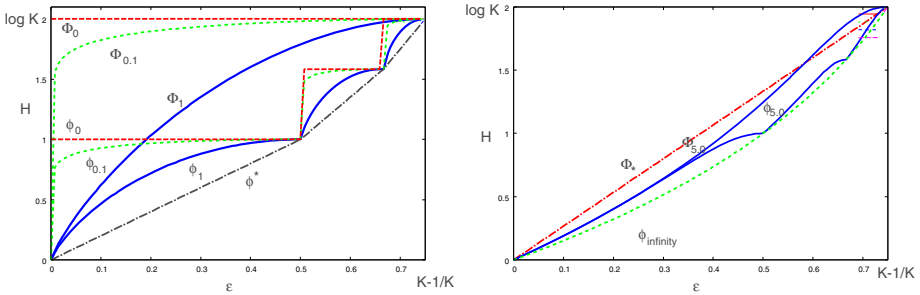


Fig. 1. Lower bound ϕ_α and upper bound Φ_α of Renyi entropy H_α by the prediction error ϵ for $K = 4$. Left: those for $\alpha = 1.0, 0.1, 0.0$; Right: those for $\alpha = 1.0, 5.0, +\infty$. The definition of ϕ^*, Φ_* and ϕ_∞ are given in the text.

Hereafter, we will not mention to w_0 for simplicity, but a similar process is applied to w_0, e.g., above values are replaced with $h = H_\alpha(\mathsf{C})$ and $c_0 = H_\alpha(\mathsf{C}_0)$.

When $\alpha \to 0$, e^{-h} and e^{-c_0} approach to $e^{-\log K}$ in most cases, but $w_\alpha(h)/w_\alpha(h')$ (h' is for a different feature) converges to a different value other than one depending on the underlying distributions. Thus, we introduce a small value δ in the weight function so as to converge into a common value:

$$w_\alpha(h) = \frac{e^{-h} - e^{-c_0} + \delta}{1 - e^{-h}} \to \frac{\delta}{1 + \log K} \quad \text{as} \quad \alpha \to 0. \tag{5}$$

With this modification, we can guarantee that $\alpha = 0$ achieves the original naïve Bayes classifier, because a constant multiplication does not change the ranking of discriminant functions.

When $\alpha = \infty$, our weighted naïve Bayes classifier is connected to the marginal Bayes errors. Here, we call the error of Bayes classifier constructed on a subset of features a *marginal Bayes error*. From the monotonicity of Bayes error on feature subsets, the Bayes error is less than or equal to marginal Bayes errors. When we denote by $\epsilon^d_{\text{Bayes}}$ the marginal Bayes error on dth feature only, it holds that $\epsilon_{\text{Bayes}} \le \epsilon^d_{\text{Bayes}}, \forall d \in \{1, 2, \ldots, D\}$. Since, as described, for $\alpha = \infty$ and small values of ϵ_{Bayes}, we may assume that $\epsilon^d_{\text{Bayes}} = 1 - \exp(-H_\infty(\mathsf{C}|\mathsf{X}_d))$. Therefore we have

$$w_d = w_\infty(h) \simeq \frac{1 - \epsilon^d_{\text{Bayes}} - e^{-c_0} + \delta}{\epsilon^d_{\text{Bayes}}} \simeq \frac{\epsilon^{\text{prior}}_{\text{Bayes}} - \epsilon^d_{\text{Bayes}}}{\epsilon^d_{\text{Bayes}}} \le \frac{\epsilon^{\text{prior}}_{\text{Bayes}} - \epsilon_{\text{Bayes}}}{\epsilon_{\text{Bayes}}},$$

where $\epsilon^{\text{prior}} = 1 - \max_c P(c)$, that is, the error on the basis of prior probabilities. The second approximation holds because $c_0 = H_\infty(\mathsf{C}) = -\log \max_c P(c)$. This relationship means that weight w_d is inversely proportional to the marginal Bayes error $\epsilon^d_{\text{Bayes}}$ and upper-bounded by Bayes error ϵ_{Bayes} in the same function.

When a middle value of α is taken, the corresponding weights and classifier have a neutral nature between two extreme cases.

4 Experiment

We conducted experiments on 17 real-life datasets taken from the UCI machine learning repository [9]. Missing values were removed beforehand. We converted a numerical value to a discrete value by equally-spaced intervals. We used 10-fold cross validation for accuracy calculation. We compared the original naïve Bayes with our weighted naïve Bayes with some values of α and $\delta = 0.01$. We also compared with a support vector machine (SVM) with a radial basis function of default values and a decision tree (C4.5) for reference.

To compensate the lack of samples, we used Laplace estimator to estimate the probability $P(c|x)$: $\hat{P}(c|x) = \frac{n_{x,c}+1}{n_x+K}$, where n_x is the number of samples taking value x and $n_{x,c}$ is the number of samples belonging to class c among them. This works especially for when $n_{x,c}$ is close to zero. We used this Laplace modification for the other probabilities as well.

4.1 Result

The result is shown in Table 1. It includes also the number of selected features whose weights are more than 10% of the maximum weight.

The optimal value of α varies over datasets. In summary, 1) a larger value of α accelerates feature selection at the expense of a small amount of performance degradation (about a half of features is removed at $\alpha = 10.0$) , 2) a smaller value of α is useful for improving the performance while keeping almost all features (in 8 cases of 17 cases, $\alpha = 0.1$ achieved the best performance), and 3) a small value of α is effective for problems with many classes (e.g., `dermatology`). In these senses, changing the value of α from the Shannon settling ($\alpha = 1$) is worth considering. It is also noted that the weighted naïve Bayes outperforms C4.5 and SVM in datasets whose features are almost all categorical (e.g., `splice` and `dermatology` have only categorical features).

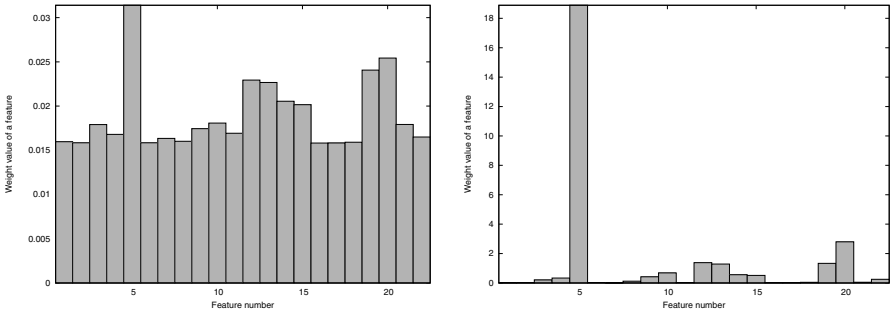
A better performance seems to be obtained for a smaller value of α. Indeed, we confirmed that a better rate of 99.20% can be obtained for $\alpha = 0.01$ in `mushroom`. However, it does not mean that the smaller value of α, the better. This is obvious from the recognition rate of 97.39% attained by the original naïve Bayes classifier for which $\alpha = 0.0$. The weights in `mushroom` are shown in Fig. 2 at $\alpha = 0.01$ and $\alpha = 10$. Although no feature selection was made at $\alpha = 0.01$, the difference of weights contributed to the increase of performance.

5 Discussion

We have analyzed the weighted naïve Bayes classifier using Renyi entropy H_α. It inherits the merit and demerit from the original naïve Bayes classifier without weights. It cannot show a satisfactory performance if the classification problem needs combinations of features in essence. However, it can often beat the curse of dimensionality by the virtue of the simplicity. Weighting on features suppresses useless features and improves the performance.

Table 1. Recognition rate estimated by 10-fold CV on 17 datasets taken from UCI. Here, D is the number of features, N is the number of samples and K is the number of classes. The proposed method (Weighted NB :$\alpha = 0.1, 1.0, 10.0$) are compared with Naïve Bayes (NB), C4.5 and linear SVM. A recognition rate is represented at percent and a number in parentheses is the number of selected features. In the last two rows, #wins is the number of victories in the naïve Bayes family and "Reduction" is the average of reduction rate of features. The best classifier is underlined.

				NB	Weighted NB			C4.5	SVM
Dataset	D	N	K		$\alpha = 0.1$	$\alpha = 1$	$\alpha = 10$		
haberman	3	306	2	**74.83**	72.89(3)	73.22(3)	73.22(1)	72.22	71.90
breast-c	9	277	2	74.39	73.94(9)	**74.83**(9)	71.87(8)	75.09	71.84
tic	9	958	2	69.93	**70.42**(9)	68.98(9)	65.62(1)	84.55	98.33
heart-c	13	296	2	83.46	83.34(13)	84.71(13)	**84.72**(11)	77.70	84.12
credit-a	15	653	2	85.60	**86.85**(15)	86.39(4)	86.39(5)	85.14	85.60
hepatitis	19	80	2	82.50	82.50(19)	**87.50**(9)	83.75(1)	86.25	82.50
credit-g	20	1000	2	74.80	**74.90**(20)	74.20(20)	70.00(2)	71.10	74.40
mushroom	22	5644	2	97.39	**98.47**(15)	98.44(4)	98.44(2)	100.0	100.0
leukemia	7129	72	2	73.49	80.31(7129)	82.85(2240)	**84.28**(1210)	83.33	98.61
iris	4	150	3	93.33	**94.66**(4)	**94.66**(4)	**94.66**(3)	95.33	96.00
tae	5	151	3	55.54	**56.25**(5)	54.29(5)	53.00(5)	51.65	52.32
cmc	9	1473	3	49.29	48.95(9)	49.83(9)	**52.47**(9)	51.53	47.72
splice	60	3190	3	**95.36**	95.23(60)	94.54(30)	94.20(7)	94.26	92.98
car	6	1728	4	**85.64**	85.02(6)	70.02(4)	70.02(1)	92.47	93.23
lymph	18	148	4	56.62	**80.06**(18)	78.89(10)	79.35(11)	77.70	84.46
glass	9	214	6	49.39	49.39(9)	**52.25**(9)	47.01(9)	66.35	57.47
dermatology	34	366	6	**97.62**	**97.62**(34)	**97.62**(34)	95.95(32)	93.71	96.72
#wins				4	8	5	4		
Reduction(%)				(0%)	(1.87%)	(23.77%)	(49.93%)		



(a) $\alpha = 0.01$ (b) $\alpha = 10$

Fig. 2. The weights of features in mushroom for two extreme values of α. Here, feature #5 is "odor", #20 is "spore-print-color." The task is to predict the edibility.

It is not easy to find an optimal value of parameter α in the proposed method, because it depends on the problems, e.g., the number of features and the number of classes. Our analysis gave just a simple guideline: choose a smaller value, say $\alpha = 0.1$, if you put a priority on improving the performance; choose a larger value, say $\alpha = 10.0$, if you need to reduce the feature set size; otherwise $\alpha = 1$ as an acceptable compromise.

As predicted from our analysis, a very small value of α made the classifier be close to the the original Bayes classifier. A large value of α emphasizes on the marginal Bayes errors. Note that Bayes error does not tell anything about classes except for the class with the largest probability. This may explain why larger values of α did not bring better results for problems with many classes.

6 Conclusion

We have extended a weighted naïve Bayes classifier using Renyi entropy H_α from the Shannon entropy version and analyzed its property. It becomes the regular naïve Bayes classify in one end with $\alpha = 0$ and a naïve Bayes classifier of which wights are inversely proportional to the marginal Bayes errors using individual features in the other end with $\alpha = \infty$. It is not so easy to find the optimal value of α depending on dataset at hand, but we gave a rough guideline for selection.

References

1. Omura, K., Kudo, M., Endo, T., Murai, T.: Weighted naïve Bayes classifier on categorical features. In: Proc. of 12th International Conference on Intelligent Systems Design and Applications, pp. 865–870 (2012)
2. Chen, L., Wang, S.: Automated feature weighting in naive bayes for high-dimensional data classification. In: Proc. of the 21st ACM International Conference on Information and Knowledge Management, pp. 1243–1252 (2012)
3. Zhang, H., Sheng, S.: Learning weighted naive Bayes with accurate ranking. In: Proc. of Fourth IEEE International Conference on Data Mining, pp. 567–570 (2004)
4. Grtner, T., Flach, P.A.: WBCSVM: Weighted Bayesian Classification based on Support Vector Machines. In: Proc. of the Eighteenth International Conference on Machine Learning, pp. 207–209 (2001)
5. Frank, E., Hall, M., Pfahringer, B.: Locally weighted naive bayes. In: Proc. of the Nineteenth Conference on Uncertainty in Artificial Intelligence, pp. 249–256 (2002)
6. Lee, C.H., Gutierrez, F., Dou, D.: Calculating Feature Weights in Naive Bayes with Kullback-Leibler Measure. In: Proc. of 11th IEEE International Conference on Data Mining, pp. 1146–1151 (2011)
7. Yang, Y., Webb, G.I.: On why discretization works for naive-bayes classifiers. In: Proc. of 16th Australian Conference on AI, pp. 440–452 (2003)
8. Feder, M., Merhav, N.: Relations Between Entropy and Error Probability. IEEE Transactions on Information Theory 40, 259–266 (1994)
9. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. University of California, Department of Infromation and Computer Science, Irvine, `http://www.ics.uci.edu/~mlearn/MLRepository.html`