Pattern Anal Applic (2006) 9:127–137
DOI 10.1007/s10044-006-0030-1

THEORETICAL ADVANCES

Classifier-independent feature selection on the basis of divergence criterion

Naoto Abe · Mineichi Kudo · Jun Toyama · Masaru Shimbo

Received: 4 May 2005 / Accepted: 22 January 2006 / Published online: 22 August 2006

Abstract Feature selection aims to choose a feature subset that has the most discriminative information from the original feature set. In practical cases, it is preferable to select a feature subset that is universally effective for any kind of classifier because there is no underlying information about a given dataset. Such a trial is called classifier-independent feature selection. We took notice of Novovičová et al.'s study as a classifier-independent feature selection method. However, the number of features have to be selected beforehand in their method. It is more desirable to determine a feature subset size automatically so as to remove only garbage features. In this study, we propose a divergence criterion on the basis of Novovičová et al.'s method.

Keywords Classifier-independent feature selection · Bayes classifier · Gaussian mixture · Garbage feature · J-divergence · Two-stage feature selection

1 Introduction

In pattern recognition, the goal of feature selection is to find a feature subset that has the most discriminative information from a given set of candidate features. The main benefits of feature selection are (i) to reduce of the measurement cost and storage requirements, (ii) to cope with the degradation of classification performance due to the finiteness of training sample sets, (iii) to reduce training and utilization time and (iv) to facilitate data visualization and data understanding.

Algorithms of feature selection can be divided into two groups. One group is called *classifier-specific feature selection* (CSFS) or *wrapper approach*. CSFS selects a feature subset that maximizes the value of a given criterion function such as the estimated recognition rate for a specified classifier. A large number of CSFS methods have been proposed in the field of pattern recognition and machine learning [1–8], and there have been some comparative studies on CSFS algorithms for large-scale feature selection problems [9–11]. The *Filter approach* [12–16] does not assume any classifier but needs a measure for evaluating a feature subset such as contextual merit [12], correlation [13, 14] and mutual information [15, 16]. There have also been some studies in which those measures have been compared in terms of the performance of practical classifiers [17–20]. However, once a specific measure is assumed, it implies the existence of a hypothetical classifier to which the metric works best. Hence, we regard such filter approaches as belonging to the CSFS group.

The other group is called *classifier-independent feature selection* (CIFS). In CIFS, we seek a feature subset that is effective for any classifier. To do this, we find a feature subset that maximizes the separation measure between class-conditional probability densities. This means that we have to estimate the class-conditional probability densities as precisely as possible and implicitly consider the Bayes classifier that achieves the minimal expected error rate. In other words, CIFS is

N. Abe (✉) · M. Kudo · J. Toyama
Division of Computer Science,
Graduate School of Information Science and Technology,
Hokkaido University, Sapporo 060-0814, Japan
e-mail: chokujin@main.ist.hokudai.ac.jp

M. Shimbo
Faculty of Information Media,
Hokkaido Information University,
Ebetsu 069-8585, Japan

equivalent to CSFS for the Bayes classifier, and the goal of CIFS is to remove *garbage features* that have no discriminative information for any kind of classifier. To perform CIFS, we can choose one of two approaches on the basis of training samples: (1) try to estimate the class-conditional density of each class so that we can have a quasi Bayes classifier or (2) try to estimate the Bayes classification boundary. The first approach was used in the Novovičová et al.'s work [21] and the second approach was used in the subclass method [22], RFI [23] and PRISM [17]. The subclass method selects a feature subset on the basis of hyper-rectangles that approximate the class boundary. RFI measures the degree of separation between class-conditional joint feature distributions, and a feature subset is selected by ranking features by their RFI values. PRISM evaluates a feature subset on the basis of the class separability called *neighborhood separability* which approximates the decision boundaries. Among these CIFS approaches, we noticed that the work by Novovičová et al. is the most straightforward, and we therefore decided to use it as the basis. Hereafter, we denote their method by ''DIV'' because it uses a Kullback-Leibler J-divergence criterion. In DIV, the class-conditional probability densities are firstly estimated by axis-parallel Gaussian mixtures, and then a feature subset is selected according to the J-divergence values [24] between the estimated densities. The problem when we use DIV as a CIFS method is that we have to determine the number of features to be selected beforehand. This is not satisfactory, because it is desirable to remove the garbage features regardless of the feature subset size.

In Sect. 2, the algorithm of DIV is explained first. In Sect. 3, we propose a divergence criterion to determine the number of features automatically so as to remove only garbage features on the basis of DIV. In Sect. 4, the proposed criterion is applied to one synthetic and nine real-world datasets to investigate the effectiveness of the proposed criterion. Moreover, we try two-stage feature selection, in which CIFS is carried out first and then CSFS is executed, to reveal the effectiveness of such a trial. The effectiveness of the proposed criterion is discussed in Sect. 5, and conclusions are given in Sect. 6.

2 Novovičová et al.'s work

2.1 Estimation of class-conditional probability densities

In many parametric approaches for density estimation, overly simple assumptions about the given data tend to be used. For more flexibility, a mixture model is used in DIV [21]. The following Gaussian mixture model is postulated for the class-conditional probability density function of class ω:

$$\begin{aligned} p(\boldsymbol{x}|\omega) &= \sum_{m=1}^{M_\omega} \alpha_m^\omega p(\boldsymbol{x}|m,\omega) \\ &= \sum_{m=1}^{M_\omega} \alpha_m^\omega \prod_{i=1}^{D} \left\{ f(x_i|\boldsymbol{b}_{mi}^\omega)^{\phi_i} f_0(x_i|\boldsymbol{b}_{0i})^{1-\phi_i} \right\}, \\ \boldsymbol{\Phi}_d &= (\phi_1, \phi_2, \ldots, \phi_D) \in \{0,1\}^D. \end{aligned} \tag{1}$$

Here, $\alpha_m^\omega \left(\sum_{m=1}^{M_\omega} \alpha_m^\omega = 1 \right)$ is a mixing weight, M_ω is the number of components, and D is the number of features of given data. In addition, $\boldsymbol{\Phi}_d$ is a parameter to indicate a feature subset that consists of d 1's and $(D - d)$ 0's, where $\phi_i = 1$ (i = 1,2,...,D) means that the ith feature is used and $\phi_i = 0$ means that the ith feature is not used. Also, f is a Gaussian with mean μ_{mi}^ω and variance σ_{mi}^ω, which are combined into $\boldsymbol{b}_{mi}^\omega = (\mu_{mi}^\omega, \sigma_{mi}^\omega)$, and f_0 is the background density common to all classes with mean μ_{0i} and variance σ_{0i} which is specified by $\boldsymbol{b}_{0i} = (\mu_{0i}, \sigma_{0i})$. In Eq. (1), the limitation due to the assumption of feature independence can be absorbed by mixing the necessary number of components. That is, we can estimate any density as precisely as possible by mixing many components. The parameters $\boldsymbol{b}_{0i}$, $\boldsymbol{b}_{mi}^\omega$ and α_m^ω are calculated by the EM algorithm [25]. In Ref. [21], determination of the number of components M_ω is not described. Therefore, we rely on MDL criterion [26] to determine the number of components M_ω in each class. The background density f_0 is only used in the construction of a sub-optimal Bayes classifier and thus can be ignored for the feature selection procedure. Therefore, we estimate the class-conditional probability density function without f_0.

2.2 Evaluation of a feature subset

In this section, we will explain how we evaluate the importance of features on the basis of the DIV method. A feature subset $\boldsymbol{\Phi}_d$ is evaluated using J-divergence between densities of two classes defined by

$$J(\boldsymbol{\Phi}_d) = \sum_{\omega \in \Omega} P(\omega) E_\omega \left\{ \log \frac{p(\boldsymbol{x}|\omega)}{p(\boldsymbol{x}|\Omega - \omega)} \right\}, \tag{2}$$

where $\boldsymbol{\Phi}_d$ is embedded into $p(\cdot)$. We consider only two classes $\Omega = \{\omega_1, \omega_2\}$ in the following, but it is naturally extended to multi-class problems in the formula (2) by taking into account the two hypothetical classes of ω and $\Omega - \omega$. Therefore, the abbreviation $\Omega - \omega$ means the class opposite to ω. To calculate this value, we need the log likelihood $\sum_{\boldsymbol{x} \in \omega} \log p(\boldsymbol{x}|\omega)$ for each ω. Thus, let

us consider only one class by ignoring ω. By Bayes theorem, for any component m, we have

$$\log p(\boldsymbol{x}) = \log \frac{p(\boldsymbol{x}|m)p(m)}{p(m|\boldsymbol{x})} = \log \frac{p(m)}{p(m|\boldsymbol{x})} + \log p(\boldsymbol{x}|m).$$

Here, we notice that only the second term is directly related to the density form of $p(\boldsymbol{x}|m)$. Next, we consider the membership value $p(m|\boldsymbol{x})$ of sample $\boldsymbol{x}$ to component m and the expected probability mass $v(\boldsymbol{x}|m) = p(m|\boldsymbol{x})/\sum_{\boldsymbol{y}} p(m|\boldsymbol{y})$ of $\boldsymbol{x}$ given m, as we do in the EM algorithm. Also, we assume that each sample $\boldsymbol{x}$ arises from a certain component m. Then according to the expectation step of the EM algorithm and the feature-independence assumption, we obtain the expected log likelihood with N samples by

$$\begin{aligned}
E\{\log p(\boldsymbol{x})\} &\simeq \frac{1}{N}\sum_{\boldsymbol{x}} \log p(\boldsymbol{x}) \\
&= \frac{1}{N}\sum_{\boldsymbol{x}}\sum_{m} p(m|\boldsymbol{x}) \log p(\boldsymbol{x}) \\
&= \frac{1}{N}\sum_{\boldsymbol{x}}\sum_{m} N\hat{\alpha}_m v(\boldsymbol{x}|m) \log p(\boldsymbol{x}) \\
&\quad \left(\because N\hat{\alpha}_m \triangleq \sum_{\boldsymbol{y}} p(m|\boldsymbol{y})\right) \\
&= \sum_{\boldsymbol{x}}\sum_{m} \hat{\alpha}_m v(\boldsymbol{x}|m) \\
&\quad \left\{\log \frac{p(m)}{p(m|\boldsymbol{x})} + \log p(\boldsymbol{x}|m)\right\} \\
&= \sum_{\boldsymbol{x}}\sum_{m} \hat{\alpha}_m v(\boldsymbol{x}|m) \\
&\quad \left\{\log \frac{p(m)}{p(m|\boldsymbol{x})} + \sum_{i} \phi_i \log f(x_i|\hat{\boldsymbol{b}}_{mi})\right\}. \qquad (3)
\end{aligned}$$

Therefore, by (3) and by adding class symbol ω, the value of J-divergence in (2) is estimated as

$$\begin{aligned}
J(\boldsymbol{\Phi}_d) &= \sum_{\omega\in\Omega} P(\omega) E_\omega\{\log p(\boldsymbol{x}|\omega) - \log p(\boldsymbol{x}|\Omega-\omega)\} \\
&= \sum_{\omega\in\Omega} P(\omega) \sum_{\boldsymbol{x}\in\omega} \left\{\sum_{m=1}^{M_\omega} \hat{\alpha}_m^\omega v(\boldsymbol{x}|m,\omega) \right. \\
&\quad \times \left(\sum_{i=1}^{D} \phi_i \log f\left(x_i|\hat{\boldsymbol{b}}_{mi}^{\omega}\right)\right) \\
&\quad - \sum_{m\prime=1}^{M_{\Omega-\omega}} \hat{\alpha}_{m\prime}^{\Omega-\omega} v(\boldsymbol{x}|m\prime,\Omega-\omega) \\
&\quad \left. \times \left(\sum_{i=1}^{D} \phi_i \log f\left(x_i|\hat{\boldsymbol{b}}_{m\prime i}^{\Omega-\omega}\right)\right)\right\} + C \\
&= \sum_{i=1}^{D} \phi_i J_i + C, \qquad (4)
\end{aligned}$$

where

$$\begin{aligned}
J_i = \sum_{\omega\in\Omega} P(\omega) \sum_{\boldsymbol{x}\in\omega} \left\{\sum_{m=1}^{M_\omega} \hat{\alpha}_m^\omega v(\boldsymbol{x}|m,\omega) \log f(x_i|\hat{\boldsymbol{b}}_{mi}^{\omega}) \right. \\
\left. - \sum_{m\prime=1}^{M_{\Omega-\omega}} \hat{\alpha}_{m\prime}^{\Omega-\omega} v(\boldsymbol{x}|m\prime,\Omega-\omega) \log f\left(x_i|\hat{\boldsymbol{b}}_{m\prime i}^{\Omega-\omega}\right)\right\}. \qquad (5)
\end{aligned}$$

In (4), we put all terms that work for every feature in common into a constant C. For example, the probability of a component, $p(m) \approx \hat{\alpha}_m^\omega$, affects evenly to every feature. After all, we can measure the importance of the ith feature by J_i, which is a weighted version of the original estimated divergence $1/N_\omega \sum_{\boldsymbol{x}} \log\left(f(x_i|\boldsymbol{b}_{mi}^{\omega})/f(x_i|\boldsymbol{b}_{m\prime i}^{\Omega-\omega})\right)$ over any pair of m (for class ω) and m' (for class $\Omega - \omega$). The parameters $(\alpha_m^\omega, \boldsymbol{b}_{mi}^\omega)$ are estimated by

$$\hat{\alpha}_m^\omega = \frac{1}{N_\omega}\sum_{\boldsymbol{x}\in\omega} p(m|\boldsymbol{x},\omega),$$

$$\hat{\boldsymbol{b}}_{mi}^{\omega} = \arg\max_{\boldsymbol{b}} \left\{\sum_{\boldsymbol{x}\in\omega} v(\boldsymbol{x}|m,\omega) \log f(x_i|\boldsymbol{b})\right\}.$$

Here, N_ω is the number of samples in ω. Once we have in the decreasing order $J(\boldsymbol{\Phi}_D) = J_{i_1} + J_{i_2} + \cdots + J_{i_D} (J_{i_j} \geq J_{i_{j+1}})$ by dropping a constant C, it suffices to choose the indices of the first d terms for a given d for obtaining $\boldsymbol{\Phi}_d$. In DIV, starting from $\boldsymbol{\Phi}_D = (1, 1, \ldots, 1)$ of D 1's, we repeat EM steps until the parameters $(\alpha_m^\omega, \boldsymbol{b}_m^\omega)$ converge and $\boldsymbol{\Phi}_d$ becomes unique for a given d. If $\boldsymbol{\Phi}_d$ does not converge, then repeat the EM steps with the current $\boldsymbol{\Phi}_d$. This makes sense because $J(\boldsymbol{\Phi}_d)$ can be affected by C other than the current $\boldsymbol{\Phi}_d$ in (4). However, in practice, it is sufficient to obtain $\boldsymbol{\Phi}_d$ after one convergence of EM steps.

3 Feature selection with divergence criterion

In DIV, the size d of a feature subset to be selected has to be given in advance by the user. However, in practice, it is difficult to determine the correct size d without knowing the background information of the given data. In the light of CIFS, it is preferable to find the size automatically as a result of removing all the garbage features. In this section, we propose a divergence criterion so as to achieve this goal. Hereafter, DIV incorporated with such an automatic selection mechanism of the number of features is abbreviated as "MDIV".

3.1 Divergence-based backward feature selection

The most important point of CIFS is to leave only informative features, in other words, to remove all garbage features. To do this, we obey Kudo and Sklansky's suggestion [27] that a criterion curve is expected to simulate the performance of a given feature subset in each size. Then we choose the smallest feature subset that keeps almost the same criterion value as that of the full feature set.

To achieve this, we take a sequential backward procedure. Since the best feature subset of size d is obtained by $\{i_1,i_2,...,i_d\}$ for $J_{i_1} \geq J_{i_2} \geq \cdots \geq J_{i_d}$, we remove the worst feature i_D, then $i_{D\text{-}1}$, and so on. For a specific degradation parameter θ $(0 < \theta < 1)$, we stop the procedure when $\sum_{j=1}^{d-1} J_{i_j} < (1-\theta)\sum_{j=1}^{D} J_{i_j}$ is satisfied for the first time (Fig. 3b). It is preferable to reconstruct the axis-parallel Gaussian mixture models after the removal of one feature. However, as described in [21], there are often cases in which the ranking of features does not change after the first evaluation of J_i $(i = 1,2,...,D)$. Therefore, in MDIV, we first construct a mixture model of some components in (1) without the function f_0 in each class through the EM algorithm using all features. Then we evaluate the importance of individual features in the J-divergence criterion so that we can have a criterion curve $J(\mathbf{\Phi}_d)$ by $J(\mathbf{\Phi}_d) = \sum_{j=1}^{d} J_{i_j} (J_{i_j} \geq J_{i_{j+1}})$. Moreover, in MDIV, we do not need initialization and updating of $\mathbf{\Phi}_d$ owing to this algorithm. To make this greedy algorithm work, we have to confirm that ''monotonicity'' is satisfied in our divergence-based criterion curve. Indeed, it is well known that the divergence holds this property in the set inclusion relation of feature subsets.

To quantify the degree of degradation of performance, we introduce a degradation parameter, θ. For a given degradation value θ, we find a feature subset of size d by removing the worst-evaluated $D - d$ features in such a way that the remaining feature subset still keeps $J(\mathbf{\Phi}_d) \geq (1-\theta)J(\mathbf{\Phi}_D)$. As for the value of θ, it is ideally desired to let it be zero. This means that we want to remove only non-informative (garbage) features. However, as a practical requirement, we are often requested to choose a smaller subset at the expense of a small degree of performance degradation. In our approach, such a demand suggests to take a small but positive value of θ. Therefore, in this paper, we use two typical values: $\theta = 1\%$ for removal of garbage features and $\theta = 10\%$ for a practical demand.

3.2 Apparent J-divergence

If the number of training samples is sufficiently large, the estimated distribution is expected to be close to the true distribution. In that case, the estimated J-divergence is also reliable. However, it is not so reliable when the number of available samples is limited. For example, in Fig. 1a, we can recognize separability to some extent, although these points are generated randomly according to the same distribution regardless of classes. Indeed, if we estimate Gaussians for the two classes separately, the means and the covariance matrices differ (Fig. 1b). In such a case, J-divergence also shows some amount of the difference against the fact that the true J-divergence is zero. This phenomena is generally observed in almost all modelings. Such an apparency is also justified by the concept of VC dimension [28], which tells us that up to a certain number of samples we can recognize such an apparent separability for any labelling of classes, though the degree depends on the flexibility of the family of

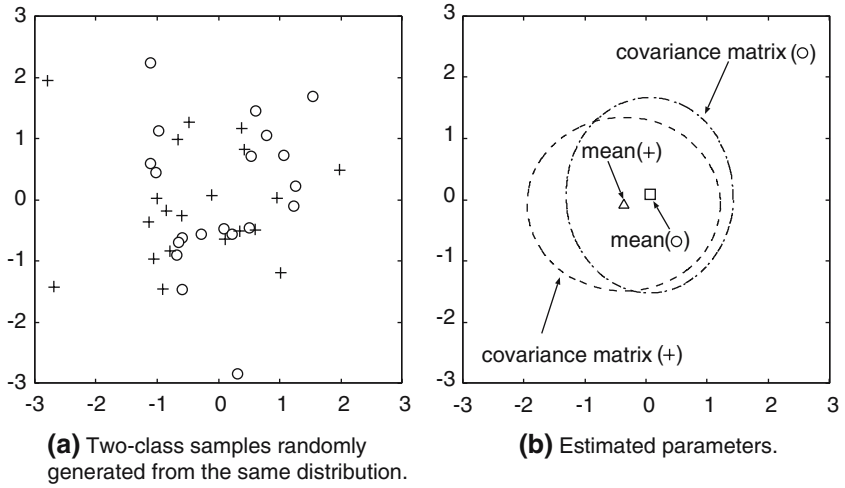

Fig. 1 Apparent separability

(a) Two-class samples randomly generated from the same distribution.

(b) Estimated parameters.

classifiers. In this study, we call this kind of apparency in J-divergence *apparent J-divergence*.

To confirm the existence of apparent J-divergence, we conducted a simple experiment. In this experiment, we considered two classes that shared the same uniform distribution in $[0, 1]^d$ of d-dimensional space. In this case, no feature has discriminative information and it is expected that $J(\mathbf{\Phi}_d)$ takes a value of almost zero. We investigated whether this was the case or not. The number of training samples was the same for two classes. A Gaussian mixture of (1) was used for the estimation of this uniform distribution in each class. We examined how the value of $J(\mathbf{\Phi}_d)(\mathbf{\Phi}_d = (1, 1, \ldots, 1))$ changes as the number of features d increases for several sizes of training samples. The number of features was set to d = 2, 4, 8, 16, 32, 64, 128 and the total number of training samples was varied as N = 50, 100, 200, 400, 800, 1,600. The results are shown in Fig. 2.

In Fig. 2, a large degree of apparent J-divergence is seen in a small number of training samples, and it is also seen that the degree of apparent J-divergence is almost proportional to the dimensionality d. From these points, the apparent J-divergence can be approximated by the following linear regression form:

$$\bar{J}(\mathbf{\Phi}_d) = \frac{a}{N} \times d. \tag{6}$$

To estimate a, we carried out regression analysis using all of the obtained 42 (= 7 × 6) points of $(d/N, J(\mathbf{\Phi}_d))$ at the same time. As a result, we obtained a = 3.186. In Fig. 2, the regression lines of (6) using a = 3.186 for N = 50, 100, 200, 400, 800, 1,600 are shown.

It is possible to use the score 3.186/N as the degree of apparent J-divergence of a single garbage feature for a given N. However, we do not think that 3.186/N is universally invariant for all kinds of distribution. Therefore, we take a more straightforward approach instead of using 3.186/N.

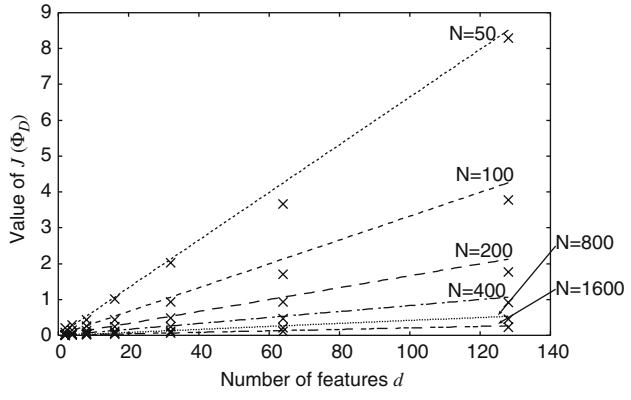

Fig. 2 Experimental results of apparent J-divergence. The *lines* are the regression lines of 3.186/N × d

We assume that there exists at least one garbage feature and that it can be found as the worst-evaluated feature in the ranking of J_i (i = 1,2,...,D), namely, J_{i_D} of the sequence $J_{i_1} \geq J_{i_2} \geq \cdots \geq J_{i_D}$. Since it can be assumed that every feature includes this degree of apparent J-divergence, we estimate its true divergence by subtracting J_{i_D}. Therefore, in the criterion curve, we correct the criterion curve by

$$J(\mathbf{\Phi}_d) \leftarrow \sum_{j=1}^{d} J_{i_j} - d \times J_{i_D} \quad (d = 1, 2, \ldots, D). \tag{7}$$

Such an estimation of apparent J-divergence and a modified criterion curve is shown in Fig. 3a. Here, we notice that the estimated apparent divergence J_{i_D} includes implicitly the sample size N in itself. In this study, the feature selection with the modification of J-divergence is denoted as "MDIV". In MDIV, after modification of the J-divergence value, we find size d_θ on the basis of the modified J-divergence curve with the θ-degradation criterion (Fig. 3b). The algorithm of MDIV is shown in Fig. 4.

4 Experiments

4.1 Effectiveness of considering apparent separability

We used a synthetic *friedman* dataset [29] to investigate the effectiveness of subtracting the amount of apparent J-divergence. In the *friedman* dataset, there are two classes, ω_1 and ω_2. The samples of ω_1 are generated according to a Gaussian with a unit covariance matrix and zero mean. The samples of ω_2 surround those of ω_1 in the first four features that are distributed uniformly within a four-dimensional spherical slab centered at the origin with an inner radius of 3.5 and an outer radius of 4.0. The last six features of ω_2 are distributed as a Gaussian with a unit covariance matrix and zero mean. Therefore, the best discriminative feature subset is the first four features.

The number of total samples N was set to N = 10, 50, 100, 500, 1,000 and 5,000. We compared the selected feature subsets between the criterion curve subtracting apparent J-divergence (with correction) and the criterion curve using the original accumulation

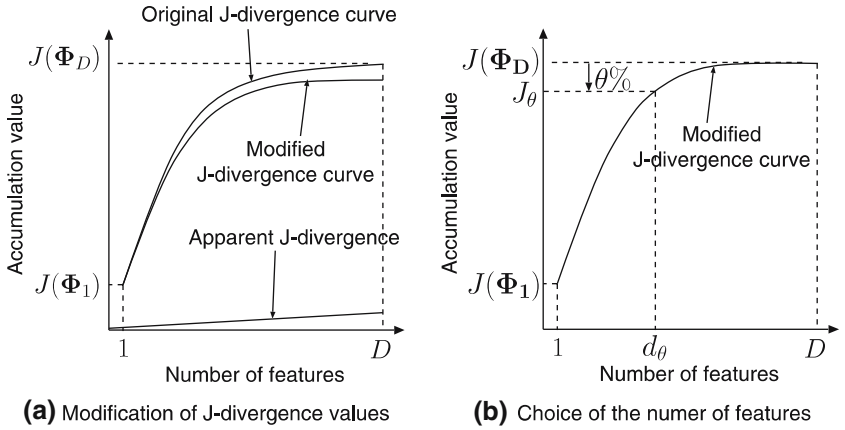

Fig. 3 Modification of J-divergence values and choice of the number of features with a degradation parameter θ in MDIV

Algorithm of MDIV

1. (Initialize) Determine the value of θ.
2. (Construction of Mixture Model) Estimate a Gaussian mixture model of Eq. (1) without the function f_0 in each class by the EM algorithm.
3. (Evaluation of features) Calculate J-divergence values J_i $(i = 1, 2, \ldots, D)$.
4. (Ranking) Rank the features in the descending order of J_i's as $J_{i_1} \geq J_{i_2} \geq \cdots \geq J_{i_D}$.
5. (Apparent divergence) Determine the value of apparent divergence by J_{i_D}.
6. (Subtraction of apparent divergence) Subtract the estimated apparent divergence value from each value to obtain the criterion curve $J(\Phi_d) = \sum_{j=1}^{d} J_{i_j} - d \times J_{i_D}$ $(d = 1, 2, \ldots, D)$.
7. (Output) Find the smallest feature subset of size d such that $J(\Phi_d) \geq (1 - \theta)J(\Phi_D)$.

Fig. 4 Algorithm of the proposed method

value (without correction). The number of components in (1) was determined using the way described in Sect. 2.1. The threshold θ should be set to a small value in order to remove only garbage features. Therefore, in this study, the threshold θ was taken by 1.0%. The selected feature subsets are shown in Table 1. In Table 1, it was confirmed that the divergence criterion subtracted apparent J-divergence is better than that of the case without subtraction.

4.2 Evaluation by recognition rates of classifiers

In this experiment, we compared the recognition rates of many classifiers before and after the removal of garbage features using the proposed divergence criterion. If garbage features are removed properly, the performance of classifiers is expected to be maintained or improved. We examined whether this was actually the case or not using seven practical classifiers in nine real-world datasets taken from UCI machine learning repository [30]. The nine real-world datasets are shown in Table 2. The problems were divided into small-scaled problems and large-scaled problems according to the product of the sample size and the dimensionality $D \times N$.

The seven classifiers used were the plug-in linear classifier (LDF), the plug-in quadratic classifier (QDF), the nearest neighbor classifier (1NN), a decision tree classifier (C4.5), a hyper-rectangle classifier [31] (SUB), a neural network classifier (NNC) and a support vector machine [32] with the linear kernel (SVM). The software *c4.5* [33] was used for a decision tree classifier, and the program *SVMTorch* [34] was used for the support vector machine classifier. In C4.5 and SVM, we tuned the parameters for each dataset so as to obtain the best classification performance. In C4.5, such a parameter tuning did not show a significant difference compared with the case using default values

Table 1 Effectiveness of the correction by the apparent J-divergence in MDIV with θ = 1.0% on *friedman* dataset

#Samples	Without correction		With correction	
	#Feature	Selected	#Feature	Selected
10	10	1 2 3 4 5 6 7 8 9 10	8	1 2 3 4 5 6 7 8
50	7	1 2 3 4 7 9 10	7	1 2 3 4 7 9 10
100	8	1 2 3 4 5 7 9 10	7	1 2 3 4 5 7 9
500	5	1 2 3 4 6	4	1 2 3 4
1,000	4	1 2 3 4	4	1 2 3 4
5,000	4	1 2 3 4	4	1 2 3 4

Table 2 Real-world datasets used in the experiments

Problem scale ($D \times N$)	Name	#Class	#Feature (D)	#Sample (1st class,2nd class,...)	#Total (N)
Small (<10K)	Spect	2	44	40,40	80
	Wpbc	2	30	47,151	198
	Bupa	2	6	145,200	345
Large (≥ 10K)	Wdbc	2	30	212,357	569
	Tic-tac-toe	2	27	626,332	958
	Sonar	2	60	97,111	208
	Musk	2	166	269,207	476
	Waveform	3	40	327,348,325	1,000
	Mushroom	2	125	3488,2156	5,644

for all of the datasets. Therefore, we used the default parameters equipped with *c4.5*. In SVM, the margin parameter C was chosen to attain the best performance from 1,000.0, 100.0, 10.0, 1.0, 0.1, 0.01, and 0.001 (Table 3). The iteration number was set to 1,000 in SUB. The number of layers was three and the number of units in the hidden layer was set to (number of features + number of classes)/2 in NNC. The features were normalized, and the number of components for each class was determined by the method described in Sect. 2.1. The threshold θ was taken as 1.0 and 10.0%. The recognition rates of seven classifiers were calculated by tenfold cross validation.

The number of improved classifiers and the size of the selected feature subset are shown in Table 4. The curves of recognition rates of classifiers and the criterion curve in the *waveform* dataset are shown in Fig. 5.

Table 3 The margin parameter C of SVM using the linear kernel

Margin(C)	Dataset
1,000.0	Mushroom
100.0	Tic-tac-toe, bupa
1.0	Musk
0.1	Spect, wdbc, sonar, waveform
0.01	Wpbc

In the *waveform* dataset, the last 19 features are known to be garbage features [30], and the best size of the feature subset is 21. The selected feature subset on *waveform* dataset using $\theta = 1.0$ and 10.0% are shown in Table 5. From Fig. 5, we see that the curves of recognition rates in the *waveform* dataset showed a flatness or a gently downward-slope after the selected size using $\theta = 1.0\%$. As shown in Table 4, the performance of seven classifiers was improved compared with the case in which all features were used with $\theta = 1.0\%$. On the other hand, for $\theta = 10.0\%$, a little smaller feature subsets were chosen at the expense of the loss of discriminative information.

4.3 Evaluation in two-stage feature selection

In this experiment, the effectiveness of the proposed method was evaluated in a two-stage feature selection scheme [27]. In two-stage feature selection, we use a CIFS method in the first step and then a CSFS method. We expect that a better feature subset can be found compared with the case in which CSFS is directly applied to the whole feature set as long as the same computation time is allowed. This is because CSFS takes longer than CIFS. We used the sequential backward floating search (SBFS) [4] for CSFS. The criterion in SBFS was taken as the recognition rate of the nearest neighbor classifier with tenfold cross vali-

Table 4 Number of improved or maintained classifiers and selected feature subset sizes

Problem scale	Dataset	$\theta = 1.0\%$		$\theta = 10.0\%$	
		c/c_0[a]	d_θ/D	c/c_0[a]	d_θ/D
Small	Spect	4/6	38/44	4/6	27/44
	Wpbc	5/7	23/30	3/7	11/30
	Bupa	4/7	5/6	4/7	4/6
Large	Wdbc	7/7	27/30	7/7	21/30
	Tic-tac-toe	4/7	22/27	2/7	15/27
	Sonar	6/7	53/60	6/7	37/60
	Musk	6/7	108/166	3/7	25/166
	Waveform	7/7	18/40	7/7	14/40
	Mushroom	6/7	27/125	2/7	14/125

[a] c Is the number of improved or maintained classifiers, and c_0 is the number of classifiers used

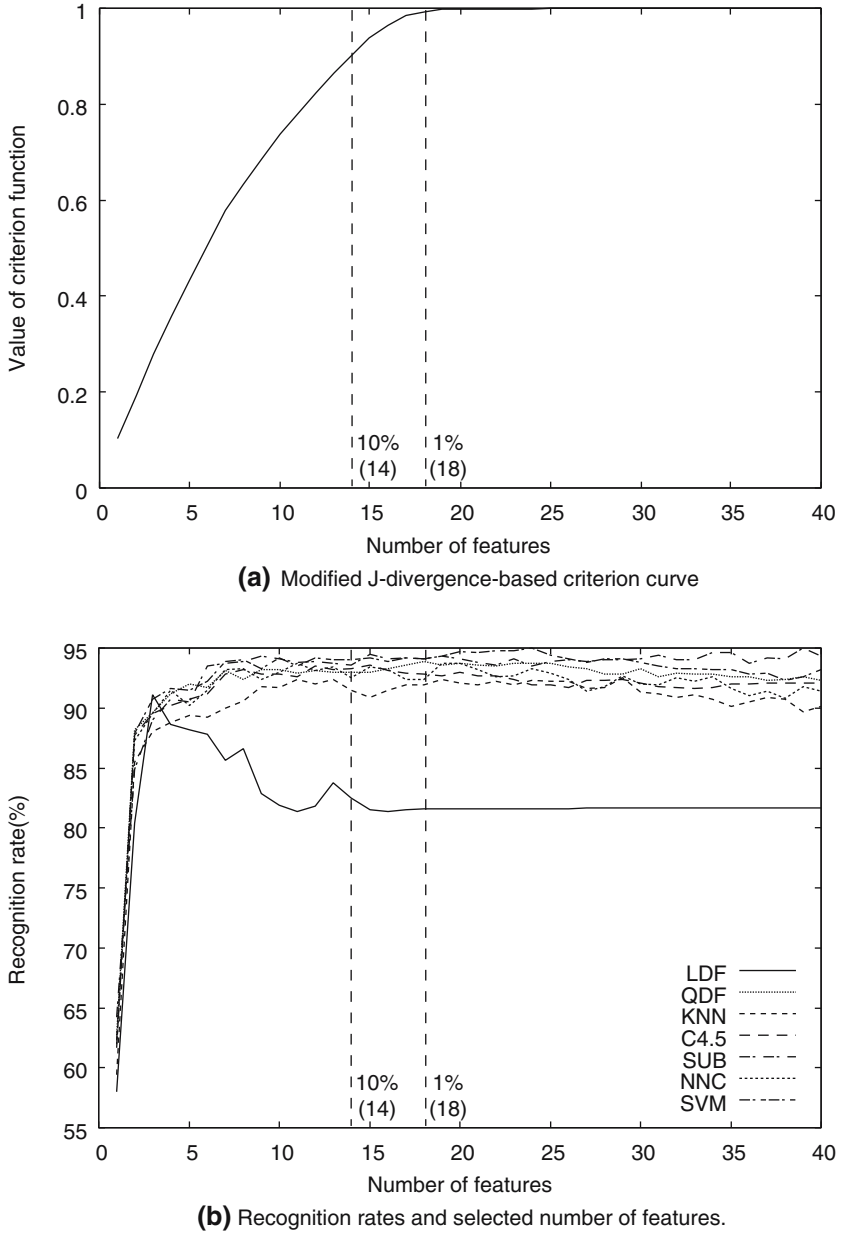

Fig. 5 Experimental results on *waveform* dataset

Table 5 Feature subset selected by the proposed method on the *waveform* dataset

θ	d_θ	Selected features
1%	18	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
10%	14	4 5 6 7 9 10 11 12 13 14 15 16 17 18

Nos. 22–40 are garbage features

dation, and we tried to find the best feature subset that attained the maximum recognition rate during the feature subset search. In the first stage, we removed garbage features by the proposed method with $\theta = 1.0\%$, and we applied SBFS to the selected feature subsets in the second stage. This combined method is denoted as MDIV(1%) + SBFS. For comparison, a single SBFS was performed directly on the original feature set. The results are shown in Table 6.

From Table 6, we can see that MDIV(1%) + SBFS found a feature subset with comparable performance in a shorter time than a single SBFS for large-scale datasets and that it succeeded to find a feature subset with better performance in small-scale datasets. From this point, the effectiveness of the garbage feature removal by the proposed method is confirmed. It is also noted that the time consumed by SBFS in the two-stage feature selection was less than that of the single-stage SBFS even though SBFS consumed much than MDIV in time. We can use any CSFS method instead

Table 6 Comparison of two-stage feature selection (CIFS + CSFS) and single CSFS

Problem scale	Dataset		SBFS	MDIV(1%)	MDIV(1%) + SBFS
Small	Spect	Time (s)	4.5	8.4	12.5(8.4+4.1)
		Rate(%)	91.2	67.5	93.8
		#Feature	14	38	9
	Wpbc	Time(s)	18.0	11.8	20.1(11.8+8.3)
		Rate(%)	73.7	63.1	73.7
		#Feature	2	23	2
	Bupa	Time(s)	0.2	6.6	6.7(6.6+0.1)
		Rate(%)	67.5	61.4	67.5
		#Feature	4	5	4
Large	Wdbc	Time(s)	43.3	28.0	56.0(28.0+28.0)
		Rate(%)	94.2	91.2	94.2
		#Feature	4	27	4
	Tic-tac-toe	Time(s)	138.2	32.0	97.8(32.0+65.8)
		Rate(%)	100.0	77.2	90.9
		#Feature	18	22	10
	Sonar	Time(s)	269.4	19.4	128.9(19.4+109.5)
		Rate(%)	94.2	82.2	92.8
		#Feature	14	53	16
	Musk	Time(s)	27444.0	113.8	6221.8(113.8+6108.0)
		Rate(%)	97.9	84.8	97.1
		#Feature	22	108	46
	Waveform	Time(s)	623.8	54.6	96.6(54.6+42.0)
		Rate(%)	94.6	91.9	94.7
		#Feature	22	18	11
	mushroom	Time(s)	354044.0	711.0	4263.5(711.0+3552.5)
		Rate(%)	100.0	100.0	100.0
		#Feature	5	27	5

of SBFS after CIFS. This means that we may use more time-consuming but more sophisticated algorithms for CSFS in the second stage to find a better feature subset.

5 Discussion

Our study was a trial to distinguish so-called *garbage features* from informative features. It was carried out on the basis of an approximation of class densities in terms of a mixture model of the axis-parallel Gaussian components. If this approximation is precise enough, we can measure the importance of features by their J-divergence values. To do this, we have done two things as follows. First, we subtracted the apparent J-divergence value from the estimated J-divergence value so that we could compensate the insufficient number of training samples to enable more accurate evaluation. Next, we introduced a threshold to distinguish useful and useless features. It should be taken to zero ideally; however, we have to keep a small margin for it due to the insufficiency of training information.

We succeeded in having fairly good results on nine real-world datasets in feature selection, especially in classifier-independent feature selection by these trials. However, there are some problems when dealing with more datasets. The main problem is that the mixture model is sometimes not sufficient to approximate class densities precisely. This comes from two factors: the incorrect number of components and the limited type for components. We determined the number of components by means of an MDL criterion. MDL criteria are known as being not good for small-sample cases. In addition, our mixture model uses Gaussian components of a diagonal covariance matrix. However, we need this simplicity for evaluating feature importance independently. A sufficient number of components guarantees a good approximation for a general density. Moreover, Gaussian mixture with diagonal covariance matrices brings us a stability in estimation of densities for small-sample cases. Therefore, in this study, this simplicity worked in such two ways.

We have to pay attention to the precision of the approximation of the densities. Some confirmations are required for general usage. Data partitioning techniques such as cross validation would be useful for this goal. It is also noted that this way may not work for highly nonlinear densities. The diagonal covariance assumption is not suitable for such cases, even if a sufficient number of components is used. This is because a larger number of components requires more training samples for obtaining better approximation.

6 Conclusion

We proposed a divergence criterion to determine the size of a feature subset automatically on the basis of Novovičová et al.'s method. This was done by removing garbage features that were totally useless for any classifier. We also introduced a measure of apparent J-divergence to compensate the insufficient number of training samples. We investigated the effectiveness of the proposed criterion by comparing recognition rates of many classifiers before and after garbage feature removal. With nine real-world datasets, we confirmed the basic effectiveness of the proposed criterion. In addition, we demonstrated that the two-stage feature selection scheme is effective in practical feature subset selection. For a further study, we will try to find a criterion to determine the optimal value of θ for a given dataset and to estimate apparent J-divergence without the assumption that we have to remove at least one feature.

Acknowledgments The authors would like to thank the anonymous reviewers who gave us helpful comments to improve the manuscript.

References

1. Sebban M, Nock R (2002) A hybrid filter/wrapper approach of feature selection using information theory. Pattern Recognit 35(4):835–846
2. Somol P, Pudil P, Kittler J (2004) Fast branch & bound algorithms for optimal feature selection. IEEE Trans Pattern Anal Mach Intell 26(7):900–912
3. Oh IS, Lee JS, Moon BR (2004) Hybrid genetic algorithms for feature selection. IEEE Trans Pattern Anal Mach Intell 26(11):1424–1437
4. Pudil P, Novovičová J, Kittler J (1994) Floating search methods in feature selection. Pattern Recognit Lett 15(11):1119–1125
5. Kira K, Rendell L (1992) A practical approach to feature selection. In: Sleeman D, Edwards P (eds) Proceedings of the 9th International Conference on Machine Learning, pp 249–256
6. Kononenko I (1994) Estimating attributes: analysis and extensions of RELIEF. In: Bergadano F, De Raedt L (eds) Proceedings of the European Conference on Machine Learning, pp 171–182
7. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97(1-2):273–324
8. Almuallim H, Dietterich TG (1991) Learning with many irrelevant features. In: Maybury MT (ed) Proceedings of the 9th National Conference on Artificial Intelligence, pp 547–552
9. Ferri FJ, Pudil P, Hatef M, Kittler J (1994) Comparative study of techniques for large-scale feature selection. In: Gelsema ES, Kanal LN (eds) Pattern Recognition in Practice, vol IV, Elsevier, pp 403–413
10. Blum AL, Langley P (1997) Selection of relevant features and examples in machine learning. Artif Intell 97(1-2):245–271
11. Kudo M, Sklansky J (2000) Comparison of algorithms that select features for pattern classifiers. Pattern Recognit 33(1):25–41
12. Hong S (1997) Use of contextual information for feature ranking and discretization. IEEE Trans Knowl Data Engineering 9(5):718–730
13. Hall M (2000) Correlation-based feature selection for discrete and numeric class machine learning. In: Langley P (ed) Proceedings of the 17th International Conference on Machine Learning, pp 259–266
14. Yu L, Liu H (2004) Efficient feature selection via analysis of relevance and redundancy. J Mach Learn Res 5:1205–1224
15. Koller D, Sahami M (1996) Toward optimal feature selection. In: Saitta L (ed) Proceedings of the 13th International Conference on Machine Learning, pp 284–292
16. Kwak N, Choi CH (2002) Input feature selection by mutual information based on Parzen window. IEEE Trans Pattern Anal Mach Intell 24(12):1667–1671
17. Singh S (2003) PRISM—a novel framework for pattern recognition. Pattern Anal Appl 6(2):131–149
18. Singh S (2003) Multiresolution estimates of classification complexity. IEEE Trans Pattern Recognit Mach Intell 25(12):1534–1539
19. Ho TK, Basu M (2000) Measuring the complexity of classification problems. In: Sanfeliu A, Villanueva JJ et al. (eds) Proceedings of the 15th International Conference on Pattern Recognition, vol 2, pp 43–47
20. Ho TK, Basu M (2002) Complexity measures of supervised classification problems. IEEE Trans Pattern Anal Mach Intell 24(3):289–300
21. Novovičová J, Pudil P, Kittler J (1996) Divergence based feature selection for multimodal class densities. IEEE Trans Pattern Anal Mach Intell 18(2):218–223
22. Kudo M, Shimbo M (1993) Feature selection based on the structural indices of categories. Pattern Recognit 26(6):891–901
23. Holz HJ, Loew MH (1994) Relative feature importance: a classifier-independent approach to feature selection. In: Gelsema ES, Kanal LN (eds) Pattern Recognition in Practice, vol IV, Elsevier, pp 473–487
24. Boekee DE, Van der Lubbe JCA (1979) Some aspects of error bounds in feature selection. Pattern Recognit 11(5-6):353–360
25. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc 39(1):1–38
26. Ichimura N (1995) Robust clustering based on a maximum likelihood method for estimation of the suitable number of clusters. Transactions of the Institute of Electronics Information and Communication Engineers J78-D-II(8):1184–1195 (in Japanese)
27. Kudo M, Sklansky J (1998) Classifier-independent feature selection for two-stage feature selection. In: Amin A, Dori D, Pudil P, Freeman H (eds) Proceedings of the Joint IAPR International Workshops on SSPR'98 and SPR'98, pp 548–554
28. Vapnik VN, Chervonenkis AY (1971) On the uniform convergence of relative frequencies of events to their probabilities. Theory Probab Appl 16(2):264–280
29. Friedman JH (1977) A recursive partitioning decision rule for nonparametric classification. IEEE Trans Comput 26:404–408
30. Newman DJ, Hettich S, Blake CL, Merz CJ (1998) UCI Repository of Machine Learning Databases, Department of Information and Computer Science. Irvine, University of California. http://www.ics.uci.edu/ mlearn/MLRepository.html

31. Kudo M, Yanagi S, Shimbo M (1996) Construction of class regions by a randomized algorithm: a randomized subclass method. Pattern Recognit 29(4):581–588
32. Vapnik V (1995) The nature of statistical learning theory. Springer, Berlin Heidelberg New York
33. Quinlan JR (1993) C4.5:programs for machine learning, Morgan Kaufmann
34. Collobert R, Bengio S (2001) SVMTorch: support vector machines for large-scale regression problems. J Mach Learn Res 1:143–160

Author Biographies

Naoto Abe received the B.E. and M.E. degrees from Hokkaido University, Sapporo, Japan, in 1999 and 2001, respectively. He is currently pursuing the Ph.D degree at the Information Engineering Department of Hokkaido University. His research interests include statistical pattern recognition and image processing.

Mineichi Kudo received the B.E. degree in 1983, the M.E. degree in 1985, and the D.E. degree in 1988, all from Hokkaido University, Sapporo, Japan. In 1996, he was visiting associate professor of the University of California, Irvine. At Hokkaido University, he is professor of Division of Systems and Information Engineering in the Graduate School of Engineering. His research interests include the design of pattern classifiers, machine learning, and image processing.

Jun Toyama received the B.S. degree in 1982 and the M.E. degree in 1984 from Hokkaido University, Sapporo, Japan. Since 1986, he has been instructor of information engineering at Hokkaido University. His research interests include speech recognition and speech perception. He is a member of IPSJ, IEEE, ASJ, and ASA.

Masaru Shimbo received his D.E. degree in Mathematical Engineering from the University of Tokyo, Japan, in 1976. From 1979 to 2001, he had been professor of Systems and Information Engineering at Hokkaido University, Sapporo, Japan. Since 2001, he has been professor of Information Media at Hokkaido Information University, Ebetsu, Japan. From 1980 to 1981, he had been visiting professor of Cornell University, Ithaca, NY, and in 1981 senior visiting fellow of the Science Research Council in the United Kingdom. His current research interests include speech recognition, visual perception and other human information processing.