Input Feature Selection by Mutual Information Based on Parzen Window

Nojun Kwak and Chong-Ho Choi

{triplea|chchoi}@csl.snu.ac.kr

Phone:(+82-2)880-7310, Fax:(+82-2)885-4459

School of Electrical Engineering and Computer Science,

Seoul National University

San 56-1, Shinlim-dong, Kwanak-ku, Seoul 151-742 KOREA

Nojun Kwak is a Ph.D student in the School of Electrical Engineering and Computer Science, Seoul National Univ., Seoul,Korea.

Chong-Ho Choi is with the School of Electrical Engineering and Computer Science, and the Automation and Systems Research Institute, Seoul National Univ., Seoul, Korea.

The corresponding author is Nojun Kwak and his e-mail address is underlined.

This work is partly supported by the Brain Neuroinformatics Research Program from Korean government.

June 26, 2002

Abstract

Mutual information is a good indicator of relevance between variables, and have been used as a measure in several feature selection algorithms. However, calculating the mutual information is difficult, and the performance of a feature selection algorithm depends on the accuracy of the mutual information. In this paper, we propose a new method of calculating mutual information between input and class variables based on the Parzen window, and we apply this to a feature selection algorithm for classification problems.

Keywords

Feature selection, mutual information, Parzen window.

I. Introduction

Mutual information is considered as a good indicator of relevance between two random variables [1]. Recently, efforts to adopt mutual information in feature selection problem resulted a series of researches [2]-[4]. Because the computation of mutual information between continuous variables is a very difficult job requiring probability density functions (pdf) and involving integration of those functions, mutual information feature selector (MIFS) [2] and its variants [3] [4] used histograms in approximating the pdfs to avoid these complexities. Thus, the performance can be degraded as a result of large errors in estimating the mutual information. In addition, MIFS methods have another limitation in that these methods do not provide a direct measure to judge whether to add additional features or not. More direct calculation of mutual information is attempted using the quadratic mutual information in the feature transformation field [5]-[7], but the relationship between Shannon's mutual information and the quadratic mutual information is not clear so far.

In this paper, a new feature selection method with the mutual information maximization scheme is proposed for classification problems. In calculating the mutual information between the input features and the output class, instead of dividing the input space into several partitions, we use the Parzen window method to estimate the input distribution. With this method, more accurate mutual information is calculated giving better performance than other methods.

In the following section, the basics of information theory and the Parzen window method are briefly presented. In Section III, we propose a new feature selection method and in Section IV, the proposed algorithms are applied to several classification problems to show their effectiveness. And finally, conclusions follow in Section V.

II. Preliminaries

A. Entropy and Mutual Information

The entropy is a measure of uncertainty of random variables. If a discrete random variable X has $\mathcal{X}$ alphabets and the *pdf* is $p(x) = \Pr\{X = x\}$, $x \in \mathcal{X}$, the entropy of X is defined as

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x). \tag{1}$$

Here the base of log is 2 and the unit of entropy is the bit.

When certain variables are known and others are not, the remaining uncertainty is measured by the conditional entropy:

$$H(Y|X) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log p(y|x). \tag{2}$$

The information found commonly in two random variables is of importance in our work, and this is defined as the mutual information between two variables:

$$I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}. \tag{3}$$

If the mutual information between two random variables is large (small), it means two variables are closely (not closely) related. The mutual information and the entropy have the following relation:

$$I(X;Y) = H(Y) - H(Y|X). \tag{4}$$

For continuous random variables, though the differential entropy and mutual information are defined as

$$\begin{aligned} H(X) &= -\int p(x) \log p(x) dx \\ I(X;Y) &= \int p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy, \end{aligned} \tag{5}$$

it is very difficult to find *pdf*s $(p(x), p(y), p(x,y))$ and to perform the integrations. Therefore we usually divide the continuous input feature space into several discrete partitions and calculate the entropy and mutual information using the definitions for discrete cases. The inherent error that exists in this process is of concern in the computation of entropy and mutual information of continuous variables.

B. The Parzen Window Density Estimate

The Parzen window density estimate can be used to approximate the probability density $p(\boldsymbol{x})$ of a vector of continuous random variables $\boldsymbol{X}$ [8]. (From now on, the boldfaced letters represent vectors.) It involves the superposition of a normalized window function centered on a set of random samples. Given a set of n d-dimensional training vectors $D = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_n}\}$, the *pdf* estimate of the Parzen window is given by

$$\hat{p}(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} \phi(\boldsymbol{x} - \boldsymbol{x}_i, h), \tag{6}$$

where $\phi(\cdot)$ is the window function and h is the window width parameter. Parzen showed that $\hat{p}(\boldsymbol{x})$ converges to the true density if $\phi(\cdot)$ and h are selected properly [8]. The window function is required to be a finite-valued non-negative density function where

$$\int \phi(\boldsymbol{y}, h) d\boldsymbol{y} = 1, \tag{7}$$

and the width parameter is required to be a function of n such that

$$\lim_{n \to \infty} h(n) = 0, \tag{8}$$

and

$$\lim_{n \to \infty} n h^d(n) = \infty. \tag{9}$$

For window functions, the rectangular and the Gaussian window functions are commonly used. The Gaussian window function is given by

$$\phi(\boldsymbol{z}, h) = \frac{1}{(2\pi)^{d/2} h^d |\Sigma|^{1/2}} \exp(-\frac{\boldsymbol{z}^T \Sigma^{-1} \boldsymbol{z}}{2h^2}), \tag{10}$$

where Σ is a covariance matrix of a d-dimensional vector of random variables $\boldsymbol{z}$.

III. Maximizing mutual information with Parzen window

A. Problem Formulation

The success of a feature selection algorithm depends critically on how much information about the output class is contained in the selected features. Using Fano's inequality [1], the minimal probability of incorrect estimation P_E of class C using inputs $\boldsymbol{X}$ is lower bounded by

$$P_E \geq \frac{H(C|\boldsymbol{X}) - 1}{\log N} = \frac{H(C) - I(\boldsymbol{X}; C) - 1}{\log N}. \tag{11}$$

Because the entropy of class $H(C)$ and the number of classes N is fixed, the lower bound of P_E is minimized when $I(\boldsymbol{X}; C)$ becomes the maximum. Thus it is necessary for good feature selection methods to maximize the mutual information $I(\boldsymbol{X}; C)$.

Battiti [2] formalized this concept of selecting the most relevant k features from a set of n features in the following *FRn-k* problem and adopted a greedy selection scheme to solve this problem.

[*FRn-k*]: Given an initial set F with n features and an output class C, find the subset $S \subset F$ with k features that minimizes $H(C|\boldsymbol{S})$, i.e., that maximizes the mutual information $I(\boldsymbol{S}; C)$, where $\boldsymbol{S}$ is a k-dimensional random vector whose components are the elements of S.

In this scheme, starting from the empty set of selected features, we add the best available input feature to the selected feature set one by one until the size of the set reaches k.

The ideal greedy selection algorithm using mutual information (MI) is realized as follows:

1. (Initialization) set $F \longleftarrow$ "initial set of n features," $S \longleftarrow$ "empty set."
2. (Computation of the MI with the output class) $\forall f_i \in F$, compute $I(f_i; C)$.
3. (Selection of the first feature) find the feature that maximizes $I(f_i; C)$, set $F \longleftarrow F \backslash \{f_i\}$, $S \longleftarrow \{f_i\}$.
4. (Greedy selection) repeat until desired number of features are selected.
   (a) (Computation of the joint MI between variables) $\forall f_i \in F$, compute $I(f_i, \boldsymbol{S}; C)$.
   (b) (Selection of the next feature) choose the feature $f_i \in F$ that maximizes $I(f_i, \boldsymbol{S}; C)$, and set $F \longleftarrow F \backslash \{f_i\}$, $S \longleftarrow \{f_i\}$.
5. Output the set S containing the selected features.

To compute the mutual information, we must know the *pdf*s of input and output variables, but this is difficult in practice, so the histogram method has been used in estimating the *pdf*s. But the histogram method needs extremely large memory space in calculating mutual information. For example, in selecting k features problem, if the output classes are composed of K_c classes and we divide the jth input feature space into P_j partitions to get the histogram, there must be $K_c \times \Pi_{j=1}^{k} P_j$ cells to compute $I(f_i, \boldsymbol{S}; C)$. In this case, even for a simple problem of selecting 10 important features, $K_c \times 10^{10}$ memories are needed if each feature space is divided into 10 partitions. Therefore realization of the ideal greedy selection algorithm is practically impossible by estimating the *pdf*s with histogram. To avoid this practical obstacle, alternative methods [2]–[4] use only joint *pdf*s of two variables in calculating mutual informations. Although these methods report good results on some problems, these are prone to errors because they do not use direct mutual information. To overcome these problems, we propose a new method for computing the mutual information in the following subsection.

B. Calculation of Mutual Information with Parzen Window

In classification problems, the class has discrete values while the input features are usually continuous variables. In this case, rewriting the relation of (4), the mutual information between the input features $\boldsymbol{X}$ and the class C can be represented as follows:

$$I(\boldsymbol{X}; C) = H(C) - H(C|\boldsymbol{X}).$$

In this equation, because the class is a discrete variable, the entropy of the class variable $H(C)$ can be easily calculated as in (1). But the conditional entropy

$$H(C|\boldsymbol{X}) = -\int_{\boldsymbol{X}} p(\boldsymbol{x}) \sum_{c=1}^{N} p(c|\boldsymbol{x}) \log p(c|\boldsymbol{x}) d\boldsymbol{x}, \tag{12}$$

where N is the number of classes, is hard to get because it is not easy to estimate $p(c|\boldsymbol{x})$.

Now, we present a new method to estimate the conditional entropy and the mutual information by the Parzen window method. By the Bayesian rule, the conditional probability $p(c|\boldsymbol{x})$ can be written as

$$p(c|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|c)p(c)}{p(\boldsymbol{x})}. \tag{13}$$

If the class has N values, say $1, 2, \cdots, N$, we get the estimate of the conditional *pdf* $\hat{p}(\boldsymbol{x}|c)$ of each class using the Parzen window method as

$$\hat{p}(\boldsymbol{x}|c) = \frac{1}{n_c} \sum_{i \in I_c} \phi(\boldsymbol{x} - \boldsymbol{x}_i, h), \tag{14}$$

where $c = 1, \cdots, N$; n_c is the number of the training examples belonging to class c; and I_c is the set of indices of the training examples belonging to class c. Because the summation of the conditional probability equals one, i.e.,

$$\sum_{k=1}^{N} p(k|\boldsymbol{x}) = 1,$$

the conditional probability $p(c|\boldsymbol{x})$ is

$$p(c|\boldsymbol{x}) = \frac{p(c|\boldsymbol{x})}{\sum_{k=1}^{N} p(k|\boldsymbol{x})} = \frac{p(c)p(\boldsymbol{x}|c)}{\sum_{k=1}^{N} p(k)p(\boldsymbol{x}|k)}.$$

The second equality is by the Bayesian rule (13). Using (14), the estimate of the conditional probability becomes

$$\hat{p}(c|\boldsymbol{x}) = \frac{\sum_{i \in I_c} \phi(\boldsymbol{x} - \boldsymbol{x}_i, h_c)}{\sum_{k=1}^{N} \sum_{i \in I_k} \phi(\boldsymbol{x} - \boldsymbol{x}_i, h_k)}, \tag{15}$$

where h_c and h_k are class specific window width parameters. Here we used $\hat{p}(k) = n_k/n$ instead of true density $p(k)$. If we use the Gaussian window function (10) with the same window width parameter and the same covariance matrix for each class [1], (15) becomes

$$\hat{p}(c|\boldsymbol{x}) = \frac{\sum_{i \in I_c} \exp(-\frac{(\boldsymbol{x}-\boldsymbol{x}_i)^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{x}_i)}{2h^2})}{\sum_{k=1}^{N} \sum_{i \in I_k} \exp(-\frac{(\boldsymbol{x}-\boldsymbol{x}_i)^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{x}_i)}{2h^2})}. \tag{16}$$

Now in the calculation of the conditional entropy (12) with n training samples, if we replace the integration with a summation of the sample points and suppose each sample has the same probability, we get

$$\hat{H}(C|\boldsymbol{X}) = -\sum_{j=1}^{n} \frac{1}{n} \sum_{c=1}^{N} \hat{p}(c|\boldsymbol{x}_j) \log \hat{p}(c|\boldsymbol{x}_j), \tag{17}$$

[1] For multiclass classification problems, there may not be enough samples such that the error for the estimate of class specific covariance matrix can be large. Thus, we use the same covariance matrix for each class throughout this paper.

where $\boldsymbol{x}_j$ is the jth sample of the training data. With (16) and (17), we get the estimate of the mutual information.

The computational complexity for (17) is propotional to $n^2 \times d$. When there is a computational problem because of large n, we may use the clustering method [9] or the sample selection method [10] to speed up the calculation. The methods based on histograms require computational complexity and memory proportional to q^d, where q represents number of quantization levels. Note that the proposed method does not require excessive memory, unlike the histogram based methods.

With this estimation, the *FRn-k* problem can be solved by the greedy selection algorithm represented in the previous subsection. Note that the dimension of a input feature vector $\boldsymbol{x}$ starts from one at the beginning and increases one by one as a new feature is added to selected feature set S.

C. Properties of the Proposed Method

In the proposed mutual information estimation, the selection of the window function and the window width parameter is very important. As mentioned in Section II, the rectangular window and the Gaussian window is normally used for the Parzen window function. In our simulation, we used the Gaussian window rather than the rectangular window because it does not contain any discontinuity. For the window width parameter h, we used $k/\log n$ as in [11], where k is a positive constant and n is the number of the samples. This choice of h satisfies the conditions (8) and (9).

To see the properties of the proposed algorithm, let us consider the typical four points XOR problem. Let $\boldsymbol{x} = (x_1, x_2)$ be a continuous input feature vector and the samples for $\boldsymbol{x}$ are given (0,0), (0,1), (1,0), (1,1). The term c is the discrete output class which takes a value in $\{-1, 1\}$. In the Parzen window method, each sample point influences the conditional probability throughout the entire feature space. The influence $\phi(\boldsymbol{x} - \boldsymbol{x}_i, h)$ of a sample point $\boldsymbol{x}_i$ is drawn according to the polarity of its corresponding class. We call it a class specific influence field, which is similar to an electric field produced by a charged particle. The influence fields generated by given four sample points in the XOR problem are shown in Fig. 1. In the figure, the slope and the range of the influence field is determined by the window width parameter h. The smaller h is, the sharper the slope and

the narrower the range of influence becomes. Figure 1 was drawn with $h = \frac{1}{2\log n}$ where n is the number of sample points which is four in this case. With this h, the higher (lower) estimate for the conditional probability of class c being -1 or 1 for each sample point is 0.90 (0.10) by (16). With (17), the conditional entropy estimate $\hat{H}(c|x_1, x_2)$ becomes 0.465, and the entropy $H(c)$ is 1 by (1). Thus, the estimate of the mutual information between two input features and the output class $\hat{I}(c; x_1, x_2)$ $(= H(c) - \hat{H}(c|x_1, x_2))$ is 0.535. The significance of $\hat{I}(c; x_1, x_2)$ being greater than zero will become clear later.

In Fig. 2, we provide the conditional probability of class 1 calculated by (16) on the input feature space. Note that we can get a Baye's classifier if we classify a given input to class 1 when $p(c = 1|\boldsymbol{x}) > 0.5$ and to class -1 when $p(c = 1|\boldsymbol{x}) < 0.5$. This classifier system is a type of Parzen classifier [9], [12], [10], [13]. Since the classifier system is not our concern, we do not go further with this issue.

In the process of the greedy selection scheme, the mutual informations $I(x_1; c)$, $I(x_2; c)$ between the variables x_1, x_2 and the class c is zero, while the estimate of the mutual information $\hat{I}(c; x_1, x_2)$ between the output class and both input features is far greater than zero. Thus, we know that using both features gives more information about the output class than using only one of the variables in the greedy selection scheme with the Parzen window. But, in the conventional feature selection methods such as MIFS [2] and MIFS-U [3], we do not get this knowledge because these methods do not use the mutual information of multiple variables. Instead, to avoid using too many memory cells in calculating mutual information with the histogram method, they make use of some measure on redundancy between variables which can be obtained by calculating the mutual information between two input features. These methods report good performances in several problems, but they are prone to errors in highly nonlinear problems like XOR problem and have to resort to some other methods like Taguchi method [4].

One more advantage of the proposed method is that it provides a measure that indicates whether to use additional features or not. Though it is quite difficult to estimate how much the performance will increase with one more feature by the increase of the mutual information, we can at least get a lower bound of error probability by the Fano's inequality and can compare the increments of mutual information or the error probability which will

aid the decision whether to add more features or not.

IV. Experimental Results

In this section, we applied greedy selection algorithm with Parzen window to some of the classification problems and show the effectiveness of the proposed method.

In all the following experiments, we set $h = \frac{1}{\log n}$ where n is the sample size of a particular data set as in [11]. Because the off diagonal terms in the covariance matrix can be prone to large errors and need great computational efforts, we used only diagonal terms in the covariance matrix for simplicity if not otherwise stated.

In addition, to expedite the computation, we restricted the influence range of a sample point to $2\sigma \cdot h$ for each dimension, i.e., made the influence to zero in the outer domain of $2\sigma \cdot h$ from the sample point, where σ is a standard deviation of the corresponding feature. This can greatly reduce the computational effort, especially when there are already enough selected features. For convenience, we will refer to the proposed method as PWFS (Parzen window feature selector) from now on.

A. Sonar dataset [14]

This dataset was constructed to discriminate between the sonar returns bounced off a metal cylinder and those bounced off a rock, and it was used in [2] and [4] to test the performances of their feature selection methods. It consists of 208 patterns including 104 training and testing patterns each. It has 60 input features and two output classes: *metal* and *rock*. As in [2], we normalized the input features to have the values in [0,1] and allotted one node per each output class for the classification. We divided each input feature space into ten partitions to calculate the entropies and mutual information. We do not know which features are important *a priori*, so we selected $3 \sim 12$ features (top $5\% \sim 20\%$) among the 60 features, and trained the neural network with the set of training patterns using these input features. Multilayer perceptrons (MLP) with one hidden layer were used and the hidden layer had three nodes as in [2]. The conventional back-propagation (BP) learning algorithm was used with the momentum of 0.0 and learning rate of 0.2. We trained the network for 300 epochs in all cases as Battiti did [2].

For comparison, we used two types of PWFS for this dataset; first one only uses diagonal

terms in the covariance matrix (Type I), and the other uses full covariance matrix (Type II). We present the selection order and the mutual information estimate $\hat{I}(\boldsymbol{S}; C)$ for PWFS in Fig. 3. In the figure, the left bars show the results of Type I and the right bars show those of Type II. Here, C and $\boldsymbol{S}$ are as defined in Section III-A. In the figure, the number on top of each bar represents the index of selected feature. We can see the estimate of the mutual information is saturated after 10 (9) features were selected with Type I (Type II); thus, we used 10 (9) features and did not use more features in PWFS. Note that the selected features of Type I and Type II give nearly the same $\hat{I}(\boldsymbol{S}; C)$ and are the same when the number of selected features is small.

In Table I, we compare the performance of PWFS with those of the conventional MIFS and MIFS-U. In addition, we also report the result of stepwise regression [15]. The results of MIFS, MIFS-U and stepwise regression are from [4]. In the table, all the resulting classification rates are the average values of 10 experiments and the corresponding standard deviations are shown in the parentheses.

From the table, we can see that PWFS produced better performances than the others and the performances of Type I and Type II do not differ much.

B. Vehicle dataset [16]

The purpose of the dataset is to classify a given silhouette as one of four types of vehicle, "Opel," "Saab," "bus," and "van," using a set of features extracted from the silhouette. The vehicle may be viewed from one of many different angles. There are 18 numeric features that were extracted from the silhouettes. Total number of examples are 946, which includes 240 Opel, 240 Saab, 240 bus, and 226 van. Among these we used 200 data as a training set and the other 646 data as a test set.

We compared PWFS with MIFS and MIFS-U. The stepwise regression cannot be used, because this is a classification problem with more than two classes. The classification was performed using MLP with the standard BP algorithm. Three hidden nodes were used with learning rate of 0.2 and zero momentum. We trained the MLP for 300 iterations, 10 times for each experiment. Table II is the classification rates of various numbers of selected features. The numbers in the parentheses are the standard deviations of 10 experiments. The result show that PWFS is better than the other algorithms for vehicle dataset.

C. Other UCI datasets

We tested PWFS for various datasets in the UC-Irvine repository [14] and compared the performances with those of MIFS and MIFS-U. Table III is the brief information of the datasets used in this paper. For these datasets, we have selected several features, and the results are shown in Tables IV $\sim$ VII. As classifier systems, we used the decision tree classifier C4.5 [16] for "letter" and "breast cancer" datasets and the nearest neighborhood classifier with neighborhood size of three for "waveform" and "glass" datasets. In the experiments, we used 75% as the training set and the other 25% as the test set for "letter" data, 50% as the training set and the other 50% as the test set for "breast cancer", 30% as the training set and 70% as the test set for "waveform". Since the number of instances is relatively small in "glass" dataset, we used the 10-fold cross-validation for this dataset. In most experiments, we can see that PWFS exhibits better performances than MIFS and MIFS-U.

V. Conclusions

In this paper, we have proposed a method for calculating mutual information between continuous input features and discrete output class and applied this to a greedy input feature selection algorithm for classification problems. Although the mutual information is a very good indicator of the relevance between variables, the reasons why it is not widely used is its computational difficulties, especially for continuous multiple variables. The proposed method make use of the Parzen window in getting the conditional density in a feature space. With this method, we can compute the mutual information between output class and multiple input features without requiring a large amount of memory.

The computational complexity of the proposed method is proportional to the square of the given sample size. This might be a limiting factor for huge data sets, but with a simple modification that confines each influence field in a finite area, we can greatly reduce the computational efforts. Furthermore, it is expected that a clustering or sample selection method can be used to overcome this limitation.

We applied the method for several classification problems and obtained better performances than those of the conventional methods such as MIFS, MIFS-U, and the stepwise

regression.

References

[1] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, John Wiley & Sons, 1991.

[2] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Networks*, vol. 5, no. 4, pp. 537 – 550, July 1994.

[3] N. Kwak and C.-H. Choi, "Improved mutual information feature selector for neural networks in supervised learning," in *Proc. 1999 Int'l Joint Conf. on Neural Networks*, Washington D.C., July 1999.

[4] N. Kwak and C.-H. Choi, "Input feature selection for classification problems," *IEEE Trans. Neural Networks*, vol. 13, no. 1, pp. 143 – 159, Jan. 2002.

[5] D. Xu and J.C. Principe, "Learning from examples with quadratic mutual information," in *Proc. of the 1998 IEEE Signal Processing Society Workshop*, 1998, pp. 155–164.

[6] K. Torkkola and W.M. Campbell, "Mutual information in learning feature transformations," in *Proc. Int'l Conf. Machine Learning*, Stanford, CA, 2000.

[7] K. Torkkola, "Nonlinear feature transforms using maximum mutual information," in *Proc. 2001 Int'l Joint Conf. on Neural Networks*, Washington D.C., July 2001, pp. 2756 –2761.

[8] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statistics*, vol. 33, pp. 1065–1076, Sep. 1962.

[9] G.A. Babich and O.I. Camps, "Weighted parzen window for pattern classification," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 18, no. 5, pp. 567–570, May 1996.

[10] K. Fukunaga and R.R. Hayes, "The reduced parzen classifier," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 11, no. 4, pp. 423–425, April 1989.

[11] C. Meilhac and C. Nastar, "Relevance feedback and category search in image databases," in *Proc. IEEE Int'l Conf. on Content-based Access of Video and Image databases*, Florence, Italy, June 1999, pp. 512 –517.

[12] Y. Muto, H. Nagase, and Y. Hamamoto, "Evaluation of a modified parzen classifier in high-dimensional spaces," in *Proc. 15th Int'l Conf. Pattern Recognition*, 2000, vol. 2, pp. 67–70.

[13] Y. Hamamoto, S. Uchimura, and S. Tomita, "On the behavior of artificial neural network classifiers in high-dimensional spaces," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 18, no. 5, pp. 571–574, May 1996.

[14] P. M. Murphy and D. W. Aha, "Uci repository of machine learning databases," 1994, For more information contact ml-repository@ics.uci.edu or http://www.cs.toronto.edu/∼delve/.

[15] N.R. Draper and H. Smith, *Applied Regression Analysis*, John Wiley & Sons, New York, 2nd edition, 1981.

[16] R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA., 1993.

[17] J.P. Siebert, "Vehicle recognition using rule based methods," Tech. Rep., Turing Institute, March 1987, Research Memorandum TIRM-87-018.

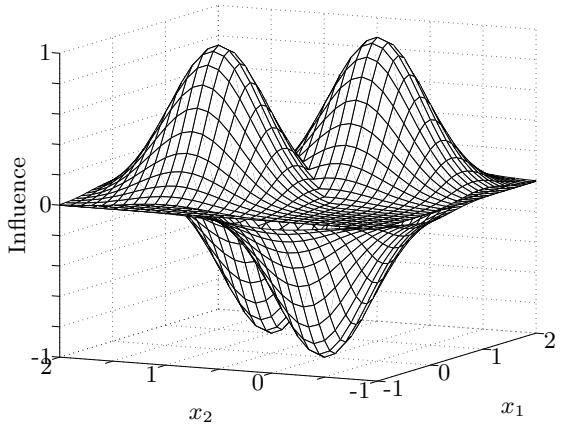

Fig. 1. Influence fields generated by four sample points in the XOR problem

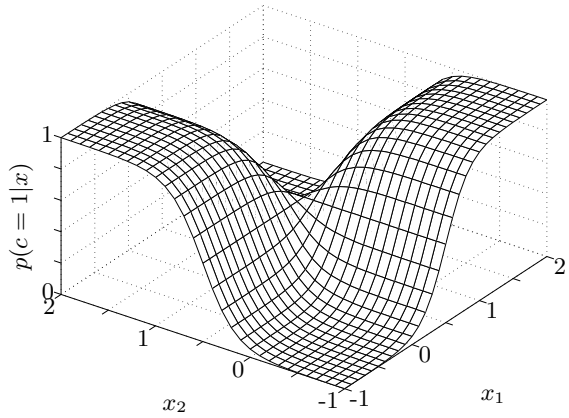

Fig. 2. Conditional probability of class 1 $p(c = 1|\boldsymbol{x})$ in XOR problem

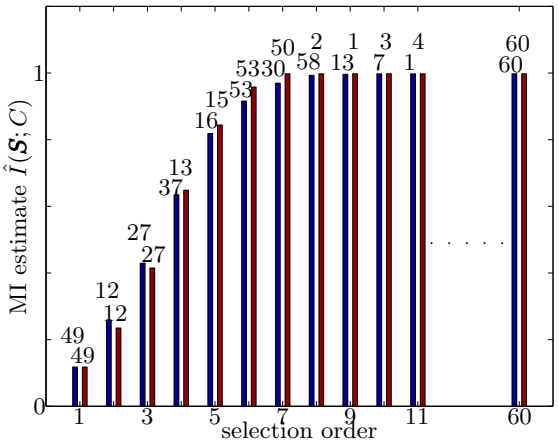

Fig. 3. Selection order and mutual information estimate of PWFS for sonar dataset (Left bar: Type I, Right bar: Type II. The number on top of each bar is the selected feature index.)

TABLE I

CLASSIFICATION RATES WITH DIFFERENT NUMBERS OF FEATURES FOR SONAR DATASET (%) (THE NUMBERS IN THE PARENTHESES ARE THE STANDARD DEVIATIONS OF 10 EXPERIMENTS)

Number of features	PWFS (Type I)	PWFS (Type II)	MIFS	MIFS-U	Stepwise regression
3	70.23 (1.2)	70.23 (1.2)	51.71 (2.1)	65.23 (1.6)	68.19 (1.1)
6	79.80 (0.8)	77.82 (0.6)	74.81 (1.4)	77.03 (0.4)	76.12 (0.3)
9	80.01 (0.9)	80.44 (1.1)	76.45 (2.4)	78.98 (0.7)	–
10	81.42 (1.4)	–	77.12 (3.1)	78.94 (0.8)	–
12	–	–	78.12 (1.8)	81.51 (0.4)	–
All (60)			87.92 (0.2)		

TABLE II

CLASSIFICATION RATES WITH DIFFERENT NUMBERS OF FEATURES FOR VEHICLE DATASET (%) (THE NUMBERS IN THE PARENTHESES ARE THE STANDARD DEVIATIONS OF 10 EXPERIMENTS)

Number of features	PWFS	MIFS	MIFS-U
2	58.77 (0.5)	40.23 (0.6)	57.53 (2.5)
4	62.50 (0.5)	57.32 (0.7)	59.97 (2.2)
6	68.89 (1.3)	65.50 (1.7)	63.94 (1.1)
8	71.59 (1.5)	70.04 (1.2)	70.35 (2.5)
10	73.20 (0.8)	71.57 (1.5)	72.70 (1.9)
All (18)		76.45 (1.0)	

TABLE III

BRIEF INFORMATION OF THE DATASETS USED

Name	# features	# instances	# classes
Letter	16	20,000	26
Breast Cancer	9	699	2
Waveform	21	1,000	3
Glass	9	214	6

TABLE IV

Classification Rates for Letter Dataset

Number of features	PWFS	MIFS	MIFS-U
2	36.36	35.44	35.44
4	67.58	62.46	68.56
6	82.86	81.00	80.50
8	84.72	84.94	83.18
All (16)		87.68	

TABLE V

Classification Rates for Breast Cancer Dataset

Number of features	PWFS	MIFS	MIFS-U
1	92.28	92.28	92.28
2	95.71	93.42	95.71
3	96.00	93.42	95.00
4	96.57	93.71	94.28
All (9)		96.28	

TABLE VI

Classification Rates for Waveform Dataset

Number of features	PWFS	MIFS	MIFS-U
2	67.71	65.85	58.85
4	75.42	67.57	73.85
6	75.42	67.14	71.57
8	78.85	66.28	77.24
10	79.10	67.71	79.57
All (21)		76.57	

TABLE VII

Classification Rates for Glass Dataset

Number of features	PWFS	MIFS	MIFS-U
1	48.13	48.13	48.13
2	62.61	57.94	57.94
3	68.22	64.95	65.42
4	71.49	66.35	66.35
All (9)		70.56	