Journal of Zhejiang University SCIENCE
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

Feature selection based on mutual information and redundancy-synergy coefficient*

YANG Sheng (杨 胜)†[1], GU Jun (顾 钧)[2]

([1]*Institute of Image Processing & Pattern Recognition, Shanghai Jiaotong University, Shanghai 200030, China*)

([2]*Department of Computer Science, Hongkong University of Science and Technology, Hongkong, China*)

†E-mail: yangsheng@sjtu.edu.cn

Received May 7, 2003; revision accepted Dec. 14, 2003

Abstract: Mutual information is an important information measure for feature subset. In this paper, a hashing mechanism is proposed to calculate the mutual information on the feature subset. Redundancy-synergy coefficient, a novel redundancy and synergy measure of features to express the class feature, is defined by mutual information. The information maximization rule was applied to derive the heuristic feature subset selection method based on mutual information and redundancy-synergy coefficient. Our experiment results showed the good performance of the new feature selection method.

Key words: Mutual information, Feature selection, Machine learning, Data mining
doi:10.1631/jzus.2004.1382 **Document code:** A **CLC number:** TP391

INTRODUCTION

Feature subset selection (FSS) is a data mining fundamental problem to select out relevant features and cast away irrelevant and redundant features from an original feature set (Liu and Motoda, 1998). If a feature subset satisfies the FSS measure and has the minimal size, it is regarded as the optimal feature subset. Complete search strategy is the way to obtain an optimal feature subset. Branch and Bound (Narendra and Fukunaga, 1977), Focus (Almuallim and Dietterich, 1991), ABB (Liu *et al.*, 1998) use the complete search strategy. However, the complete search strategy is NP-hard (Blum and Rivest, 1992; Chen *et al.*, 1997). When dataset is high-dimensioned, it becomes time-consuming. Heuristic search and stochastic search are employed to find the sub-optimal feature subset quickly. However, these methods often yield unsatisfying FSS results. A good FSS method should achieve high-quality feature subset selection quickly.

Mutual information is an important information measure for feature subset. It has been taken as an FSS measure, where the high-valued features are selected and the low-valued features are simply discarded. That often reserves redundant features and deletes relevant features. This paper presents a novel FSS method to solve these problems; and is organized as follows. In Section 2, the mutual information is calculated by a hashing mechanism. A novel FSS measure, redundancy-synergy coefficient, is defined. Then, the information maximization rule is introduced. In Section 3, a new FSS method based on information maximization rule is presented, which is called Maintaining Mutual

* Project supported by the National Natural Science Foundation of China (No. 60075007) and the National Basic Research Program (973) of China (No. G1998030401)

Information and Minimizing Redundancy-Synergy Coefficient (MMIMRSC). The basic idea of MMIMRSC is introduced first, and then the naive version of MMIMRSC (Naive MMIMRSC) is given. After discussing the shortcoming of Naive MMIMRSC, the final and better version of MMIMRSC is presented. In Section 4, MMIMRSC is tested by thirteen benchmark UCI datasets. Section 5 gives conclusions.

THEORETICAL FRAMEWORK

Basic definitions and theorems

A sample set S can be denoted by $\{(F,P)_i \mid i=1,\ldots,m\}$, where m is the total number of instances, F is the original feature set of S, $\{f_1,\ldots,f_p\}$, and P is the class feature of S. The mutual information of F and P can be calculated by (Cover, 1991),

$$I(F;P) = I(P)-E(P \mid F) \tag{1}$$

where $I(F;P)$ is the mutual information between F and P, $I(P)$ is the entropy of the class feature P, and $E(P|F)$ is the conditional entropy of the class feature P. They can be calculated by,

$$I(P) = -\sum_{P} p(P)\log_2 p(P) \tag{2}$$

$$E(P \mid F) = -\sum_{F}\sum_{P} p(F,P)\log_2 p(P \mid F) \tag{3}$$

where $p(\cdot)$ is the probability density function (*pdf*). However, the calculation of $E(P|F)$ will be very hard when there are continuous features in F. It is difficult to obtain the *pdf*s of the continuous features. Moreover, since the feature set F consists of multiple features, the integration of those continuous *pdf*s is also very difficult. Thus, all continuous features are discretized first in this paper.

In a sample set S where all features are discrete, instances that match their features in the feature subset A ($A\subseteq F$) constitute a subset of S, so S can be divided into different subsets. This process is called a partition. Assume that the class feature P partitions S into smaller subsets $\{P_i|i=1,\ldots,u\}$, and u is the number of classes. The size of a subset P_i is p_i. The feature set F partitions S into smaller subsets $\{S_j| j=1,\ldots,v\}$. The size of a subset S_j is s_j. The class feature P partitions S_j into smaller subsets $\{S_{ij}|i=1,\ldots,u, j=1,\ldots,v\}$. The size of the subset S_{ij} is s_{ij}. Now, $I(P)$ and $E(P|F)$ can be expressed in detail under the condition of dataset partition, by

$$I(P) = -\sum_{i=1}^{u}\frac{p_i}{m}\log_2\frac{p_i}{m} \tag{4}$$

$$E(P \mid F) = -\sum_{j=1}^{v}\sum_{i=1}^{u}\frac{s_{ij}}{m}\log_2\frac{s_{ij}}{s_j} \tag{5}$$

In calculating the mutual information of feature subset, the partition of sample set is essential. Normally, the Cartesian product of feature values is used for the partition, which needs to consider every possible feature value combination (a feature value combination forms a sample subset in partition). Therefore, much time and memory are needed to partition the sample set. The equation 0lg0=0 is stated in the definition of entropy (Cover, 1991). Thus, only those sample subsets that are produced by partitioning S, are needed to calculate the value of mutual information. Solving the above problem, hashing mechanism is proposed to partition the sample set (Liu and Setiono, 1996). The hashing function used in this paper is stated as in Eq.(6), where MOD is the function to obtain the remainder of D dividing $\sum_{i=1}^{p} i * vf_i$, vf_i is the value of feature f_i ($i=1,\ldots,p$) (feature value is denoted by integer), and D is the length of the hashing table.

$$Hash(F)=\mathrm{MOD}\left(\sum_{i=1}^{p} i * vf_i, D\right) \tag{6}$$

The hashing mechanism saves a lot of time and memory for partition, and it makes the size of any partition smaller than m. Assuming that an addition operation is a basic operation, the mutual information can be calculated with the approximate time complexity $O(m)$.

Monotonicity If $A\subset B\subseteq F$, then $I(A;P)\leq I(B;P)$

(Cover, 1991).

Definition 1 (equivalent feature set and equivalent feature subset)

$A,B\subseteq F$. If $I(A;P)=I(B;P)$, then A is called an equivalent feature set of B, or A and B are equivalent. Additionally, if $A\subseteq B$, then A is called an equivalent feature subset of B.

Definition 2 (reduced equivalent feature set and reduced equivalent feature subset)

$A,B\subseteq F$. If A and B are equivalent, and any feature subset of A except A is not the equivalent feature set of B, then A is called a reduced equivalent feature set of B. Additionally, if $A\subseteq B$, then A is called a reduced equivalent feature subset of B.

In terms of the above two definitions, all equivalent feature subsets of F partition S into the same sample subsets. They contain the same mutual information as F.

Theorem 1 $A,B\subseteq F$. If A is an equivalent feature set (subset) of B, there is one reduced equivalent feature set (subset) of B contained by A at least. This theorem can be proved easily by Definition 1 and Definition 2.

Theorem 2 $A\subseteq B\subseteq F$. If A is an equivalent feature subset of F, then B is an equivalent feature subset of F. This theorem can be proved easily by the monotonicity of mutual information and Definition 1.

Theorem 3 $A\subseteq B\subseteq F$, $f\in A,B$, and A is an equivalent feature subset of B. If $B-\{f\}$ is not an equivalent feature subset of B, then $A-\{f\}$ is not an equivalent feature subset of A.

Proof Since $A\subseteq B$, $A-\{f\}\subseteq B-\{f\}$. By the monotonicity of mutual information, $I(A-\{f\};P)\leq I(B-\{f\};P)$, and $I(B-\{f\};P)<I(B;P)$, so $I(A-\{f\};P)<I(A;P)$. Therefore, $A-\{f\}$ is not an equivalent feature subset of A. Theorem 3 is proved.

In terms of Theorem 3, if $I(B-\{f\};P)<I(B;P)$, then f must be in all equivalent feature subsets of B.

Redundancy-synergy coefficient of feature subset

Brenner defined redundancy-synergy index (Eq.(7)) taken as a measure of the synergistic ability and redundancy for a pair of neurons (f_1, f_2) conveying information about the stimulus P (Brenner *et al.*, 2000).

$$RS_{\text{pairs}}(f_1,f_2)=I(f_1,f_2;P)-[I(f_1;p)+I(f_2;P)] \quad (7)$$

In the extreme case where $f_1=f_2$, the two neurons provide the same information about the stimulus P, yielding $RS_{\text{pairs}}(f_1,f_2)=I(f_1,f_2;P)-[I(f_1;P)+I(f_1;P)]=-I(f_1;P)$. That means one of the two neurons is completely redundant. On the other hand, a bigger RS_{pairs} value shows a bigger synergistic interaction between f_1 and f_2. Practically, f_1 and f_2 can be regarded as two random variables or features. An extended redundancy-synergy measure of RS_{pairs} is defined by Definition 3. Unlike RS_{pairs} defined by the difference of mutual information, redundancy-synergy coefficient is defined by the quotient of mutual information as follows:

Definition 3 Redundancy-synergy coefficient of F ($RSC(F)$) is determined by Eq.(8),

$$RSC(F)=\frac{I(F;P)}{\sum_{i=1}^{p}I(f_i;P)} \quad (8)$$

Redundancy-synergy coefficient describes the synergistic ability of features to contain the class feature information. It ranges from 0 to ∞ (Yaglom and Yaglom, 1983). The smaller the redundancy-synergy coefficient, the weaker the synergistic capability. On the other hand, redundancy-synergy coefficient also describes the redundancy between the features. The more the redundancy between features is, the smaller the redundancy-synergy coefficient is.

Theorem 4 $f_1,f_2\in A$. If $A-\{f_1\}$ and $A-\{f_2\}$ are two equivalent feature subsets of A, and $I(f_1;P)>I(f_2;P)$, then $RSC(A-\{f_1\})>RSC(A-\{f_2\})$. This theorem can be proved easily by Definition 1 and Definition 3.

Theorem 5 If A is an equivalent feature subset of B, then $RSC(A)\geq RSC(B)$. This theorem can be proved easily by Definition 1 and Definition 3.

Redundancy-synergy coefficient of feature subset can be understood better by two cases. Given two features f_1 and f_2 ($A=\{f_1, f_2\}$), we show the

factors of $RSC(A)$ (Fig.1). The shadow in Fig.1 is the redundant information between f_1 and f_2. A big shadow produces a small $RSC(A)$, since $I(A;P)$ is small and $I(f_1;P)+I(f_2;P)$ does not vary. Assume that A and B are two different equal-sized subsets of F, and $I(A;P)=I(B;P)$. The inequality, $RSC(A)>RSC(B)$, means there are more redundant class information in B. Therefore, more redundant features can be cast away from B without decreasing the mutual information.

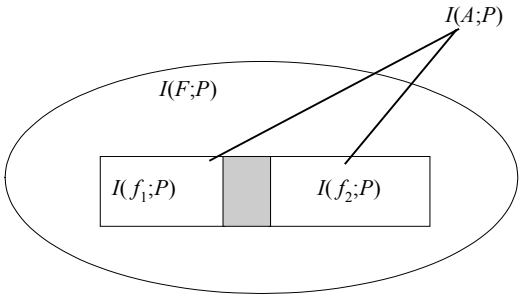

Fig.1 Factors of *RSC*(*A*)

Information maximization rule

Fano's inequality is expressed in Eq.(9) (Fano, 1961), where $\hat{P}$ is the estimate of P after observing F, and $P_e(P \neq \hat{P})$ is the probability of $P \neq \hat{P}$ (i.e. probability of error). Fano's inequality determines the lower bound of the probability of error for a classifier, $(E(P|F)-1)/\log u$. Of course, the classifier determines whether the lower bound can be reached. In terms of Fano's inequality, the lower bound of probability of error is minimized when the mutual information of feature subset is maximal. For a certain sample set S, any feature subset containing the maximal class information obtains the minimal lower bound of probability of error.

$$P_e(P \neq \hat{P}) \geq \frac{E(P \mid F)-1}{\log u} = \frac{I(P)-I(F;P)-1}{\log u} \quad (9)$$

Theorem 6 For a sample set S, its original feature set F and the equivalent feature subsets of F obtain the minimal lower bound of probability of error. This theorem can be proved easily by Definition 1 and Fano's inequality.

FEATURE SUBSET SELECTION METHOD

According to Theorem 6, the minimal reduced equivalent feature subset of F obtains the minimal lower bound of probability of error, so it has the same classifiability as F. It is the object of FSS to find the minimal reduced equivalent feature subset of F. However, it is time-consuming to find the minimal reduced equivalent feature subset of F by a complete search, especially for high-dimensioned datasets. In this paper, a heuristic FSS method is presented, which aims to find a small reduced equivalent feature subset of F. This method is a process of maintaining the mutual information of F (i.e., the mutual information of F is a bound for FSS) and minimizing redundancy-synergy coefficient, which is called Maintaining Mutual Information and Minimizing Redundancy-Synergy Coefficient (MMIMRSC).

The method starts from F as a root, finds a new root according to two rules from the child feature subsets of a current root (a child feature subset of the current root is produced by deleting a feature from the current root), and so forth, then stops if no new root is found. The two rules are: (1) the new root is the child equivalent feature subset of the current root; (2) the new root has the smaller redundancy-synergy coefficient than other child equivalent feature subsets of the current root. That is the basic idea of MMIMRSC.

There is often more than one child equivalent feature subset of a current root. Which child equivalent feature subset of the current root should be selected as the new root? Smaller redundancy-synergy coefficient means that more possible redundant features can be cast away without decreasing mutual information, so the child equivalent feature subset of the current root with minimal redundancy-synergy coefficient is selected as the new root. Obviously, the FSS result of MMIMRSC is a reduced equivalent feature subset of F.

Naive Maintaining Mutual Information and Minimizing Redundancy-Synergy Coefficient method

The Naive MMIMRSC shown below is a straightforward and naive MMIMRSC method. For each root, all child equivalent feature subsets are found first through evaluating each child feature subset of the root, and then the child equivalent feature subset with minimal redundancy-synergy coefficient is taken as the new root.

Method: Naive MMIMRSC

Input: S – a sample set

F – the original feature set of S

R – an empty feature subset

Q – a set of feature subsets

Output: a small reduced equivalent feature subset of F

Step 1 Calculate $I(F;P)$;

Step 2 $R \leftarrow F$;

Step 3 Find all child equivalent feature subsets of R, and put them into Q;

Step 4 If Q is empty, go to Step 7; else go to Step 5;

Step 5 Find the feature subset with the smallest redundancy-synergy coefficient from Q as R;

//find a new root

Step 6 Empty(Q); Go to Step 3;

//make Q be empty, and loop from Step 3

Step 7 Return R;

The running time of FSS method is related to two factors: (1) the search space size (the number of evaluated feature subsets); (2) the time of evaluating a feature subset. Naive MMIMRSC runs a backward deleting search. Considering the original feature subset F is also evaluated, the number of evaluated feature subsets is smaller than $0.5*p*(p+1)+1$. Since the time complexity of evaluating a feature subset is $O(m)$ approximately, the time complexity of Naive MMIMRSC is $O(mp^2)$ approximately.

Maintaining Mutual Information and Minimizing Redundancy-Synergy Coefficient method

Though Naive MMIMRSC is a heuristic method, there are many redundant feature subsets evaluated. In one case, assume that R is a root and $R-\{f_1\}$ is the new root found from R. If $R-\{f_2\}$ is not an equivalent feature subset of R (i.e., $I(R-\{f_2\};P)<I(R;P)$), then $R-\{f_1\}-\{f_2\}$ is not an equivalent feature subset of R according to Theorem 3. That means $R-\{f_1\}-\{f_2\}$ should not be evaluated any more after $R-\{f_2\}$ is evaluated. However, $R-\{f_1\}-\{f_2\}$ is still evaluated in Naive MMIMRSC.

In another case, in order to find the new root, Naive MMIMRSC needs to find all child equivalent feature subsets of R, which results in evaluating all child feature subsets of R. However, assume that the features in R are sorted ascendingly by their mutual information values. By visiting in order the feature in the sorted features, the first found child equivalent feature subset of R is the new root since it has the minimal redundancy-synergy coefficient according to Theorem 4. Therefore, the other child equivalent feature subsets of R need not be evaluated anymore.

In order to avoid the above two cases, two additional rules are proposed as follows: (1) for a root R, if $I(R-\{f \mid f\in R\};P)<I(R;P)$, f is reserved in any evaluated feature subset in future; (2) sort features ascendingly by their mutual information values, which makes the first found child equivalent feature subset of R be the new root. The final version of MMIMRSC method is presented as follows. For simplicity, it is called MMIMRSC.

Method: MMIMRSC

Input: S – a sample set

F – the original feature set of S

R – an empty feature subset

Ascending_Sequence – an ascending sequence of features by the mutual information values

Output: a small reduced equivalent feature subset of F

Step 1 Calculate each $I(f_i;P)$;

Step 2 Ascending_Sequence$\leftarrow$Sort($I(f_i;P)$);

Step 3 Calculate $I(F;P)$;

Step 4 $R \leftarrow F$;

Step 5 Get feature f from Ascending_Sequence in order, if $I(R-\{f\};P)=I(F;P)$ $R \leftarrow R-\{f\}$; //find a

new root, else f is reserved in R;

Step 6 Return R;

The MMIMRSC method sorts features ascendingly by their mutual information values, starts first from the original feature subset F as a root, takes the first found child equivalent feature subset of the current root as a new root by visiting in order the features in Ascending_Sequence, and so forth. This is stopped after each feature in Ascending_Sequence is visited in order.

In MMIMRSC, when a feature $\{f \mid f \in R\}$ in Ascending_Sequence is visited, the feature subset $R-\{f\}$ is evaluated. Two cases may be presented. In one case, if $I(R-\{f\};P)<I(R;P)$, the feature f is reserved in future evaluated feature subset according to Theorem 3. In the other case, if $I(R-\{f\};P)=I(R;P)$, $R-\{f\}$ is just the new root since it has the minimal redundancy-synergy coefficient according to Theorem 4. MMIMRSC begins to search for a new root from $R-\{f\}$. In MMIMRSC, when the feature f in Ascending_Sequence is visited, the features ahead of f in Ascending_Sequence have been deleted or reserved. If $R-\{f\}$ is the new root found from R, MMIMRSC only needs to visit the rest of the features in Ascending_Sequence to find the new root from $R-\{f\}$. MMIMRSC satisfies the two rules for a new root and the two additional rules. Therefore, MMIMRSC finds the same roots as Naive MMIMRSC.

In MMIMRSC, each feature in Ascending_Sequence is visited only once. Considering the original feature subsets F being evaluated, the number of evaluated feature subsets is $p+1$, so the time complexity of MMIMRSC is $O(mp)$ approximately. With the same FSS result, MMIMRSC is a simpler method than Naive MMIMRSC.

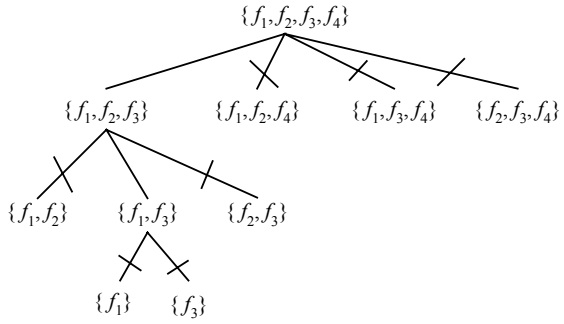

Fig.2 FSS process of Naive MMIMRSC

An example

For a better understanding of MMIMRSC, an example is given. Assume that $F=\{f_1, f_2, f_3, f_4\}$, $I(f_1;P)>I(f_2;P)>I(f_3;P)>I(f_4;P)$, and $\{f_1, f_3\}$ is the only reduced equivalent feature subset of F. The FSS processes of Naive MMIMRSC and MMIMRSC are shown in Fig.2 and Fig.3 respectively, where boldface denotes equivalent feature subset of F, and broken arrowhead line denotes the order of evaluating feature subsets in MMIMRSC. Any equivalent feature subset of F contains f_1 and f_3 according to Theorem 1, and any feature subset containing f_1 and f_3 is an equivalent feature subset of F according to Theorem 2. Thus, the search process can be understood easily.

The number of evaluated feature subsets is 4+3+2+1=10 in Naive MMIMRSC, but 4+1=5 in MMIMRSC. However, the roots are the same in Naive MMIMRSC and MMIMRSC. MMIMRSC is indeed a fast and simplified version of Naive MMIMRSC.

EXPERIMENTS

Our experiments will verify three objectives: (1) MMIMRSC can find a small equivalent feature subset of F; (2) MMIMRSC has a small search spa-

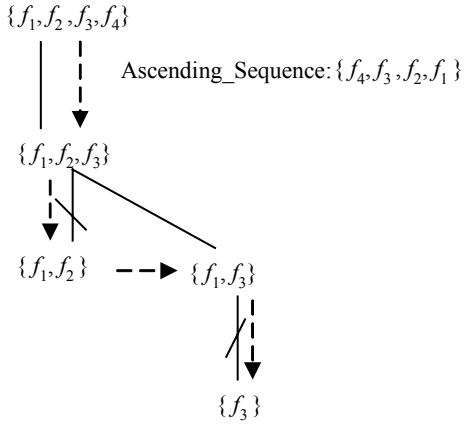

Fig.3 FSS process of MMISMRSC

ce and spends short running time; and (3) the feature subset selected by MMIMRSC can be fit for various machine learning methods.

Experiment design

In order to verify the three objectives, thirteen UCI datasets (www.ics.uci.edu/~mlearn/MLRepository.html) were employed in our experiments. The datasets were Corral, Monk1, Monk3, Parity5+5, Parity5+2, LED7, Vote, Lenses, Zoo, Solar, Mushroom, Sonar and Mfeat (Multiple Features Database). Parity5+2 is a modified version of Parity5+5 by replacing its 6th and 7th features with its 1st and 2nd features respectively. Obviously, there are redundant features in Parity5+2.

All continuous features were discretized by 10 equal width intervals. The thirteen datasets were divided into two groups: the Small Feature Set Datasets (SFSD, $p\leq 20$) and the Big Feature Set Datasets (BFSD, $p>20$). The first ten datasets belonged to SFSD, and the next three datasets belonged to BFSD. Two groups of experiments were designed for SFSD and BFSD respectively as follows.

SFSD: For the objective (1), MMIMRSC was employed to run FSS for above ten UCI datasets. For comparison, FOCUS and ABB were also employed to run FSS for above ten UCI datasets. For the objective (2), the numbers of evaluated feature subsets in MMIMRSC, FOCUS and ABB were recorded when they ran FSS for each dataset. The number of feature subsets in the whole search space ($2^p-1\approx 2^p$) was given. Also, the running time of each method was tested indirectly. In order to avoid disturbance by the instability of the computer running in this experiment, we ran 100 times of an FSS continuously (called a continuum of the FSS). The continuum was run 10 times independently, and the mean running time of the FSS in each continuum (denoted by t_i, i=1,…,10) was calculated by 100 dividing the running time of the continuum. The running time of the FSS t was calculated by 10 dividing $\sum_{i=1}^{10} t_i$. For the objective (3), C4.5 and Naive-Bayes were employed to evaluate the feature subsets selected by MMIMRSC. For comparison, the original feature sets were also evaluated by the above two machine learning methods.

BFSD: For the objective (1), MMIMRSC was employed to run FSS for the three UCI datasets. We did not obtain the FSS results by using complete search after many hours of FSS running. For comparison, LVF (Liu and Setiono, 1996) was employed to run FSS for BFSD, since the complete search method was time-consuming and even inapplicable. Unlike ABB running a complete search, LVF runs a stochastic search. In this paper, the parameters of LVF, maxTries and allowed inconsistency rate were fixed as p^2 and 0 respectively. For the objective (2), the search space size and running time of MMIMRSC were recorded when MMIMRSC was running FSS for each dataset by the method mentioned previously. The running time and search space of LVF were ignored since they vary with different runnings. For the objective (3), C4.5 and Naive-Bayes were employed to evaluate the feature subsets selected by MMIMRSC and LVF. For comparison, the original feature sets were also evaluated by the above two machine learning methods.

The default parameters (-m2 -c25) of C4.5 were fixed as the parameters for C4.5 running. All the experiments were done on Dell PowerEdge2500 server (Pentium IV 1 GHz, 512 M memory) running Microsoft Windows 2000 Server Edition.

Group I

In this group of experiments, MMIMRSC was tested by SFSD. The FSS results are shown in Table 1. MMIMRSC had the same FSS solutions as FOCUS and ABB. That showed MMIMRSC almost found an optimal feature subset, though MMIMRSC was not a complete search method.

For Parity5+2, the FSS results of MMIMRSC, FOCUS and ABB were the same since $f_1=f_6$ and $f_2=f_7$. The found feature subsets of LED7 and Lenses were the same as their original feature sets respectively, which showed that each feature of them was relevant.

The search space sizes and running time of MMIMRSC, FOCUS and ABB are shown in Table 2. The search space sizes and running time of MM-

IMRSC were much less than that of FOCUS and ABB. FOCUS, a sequential forward exhaustive search had a smaller search space for Zoo than ABB since the optimal feature subset of Zoo had a small size. For LED7 and Lenses, MMIMRSC and ABB had equal search space sizes and almost equal running time, since the search processes stopped after the original feature sets were visited. Notably, the bigger the whole search space is, the smaller the Ratio of MMIMRSC is.

10-fold cross validation of C4.5 method was employed as a machine learning method to test the feature subsets selected by MMIMRSC. The results of 10-fold cross validation of C4.5 method are shown in Table 3. Monk3 had the same tree sizes and error rates before and after FSS. Monk1, Parity5+5 and Vote had smaller tree sizes and error rates after FSS. Corral and Zoo had bigger tree sizes and smaller error rates after FSS. Parity5+2 and Solar had smaller tree sizes and bigger error rates after FSS. As in (Murphy and Pazzani, 1994), a smaller tree size did not always lead a smaller error rate. LED7 and Lenses had the same tree sizes and error rates respectively before and after FSS since all features were relevant.

The error rates of 10-fold cross validation of Naive-Bayes before and after FSS are shown in Table 4. Monk1, Monk3, Parity5+5, Parity5+2, Vote and Zoo had smaller error rates after FSS. Corral and Solar had bigger error rates after FSS. LED7 and Lenses had equal error rates.

Group II

In this group of experiments, MMIMRSC was te-

Table 1 FSS results for SFSD

Dataset	m	p	u	r	MMIMRSC	FOCUS	ABB
Corral	128	6	2	4	$\{f_1–f_4\}$	$\{f_1–f_4\}$	$\{f_1–f_4\}$
Monk1	432	6	2	3	$\{f_1, f_2, f_5\}$	$\{f_1, f_2, f_5\}$	$\{f_1, f_2, f_5\}$
Monk3	432	6	2	3	$\{f_2, f_4, f_5\}$	$\{f_2, f_4, f_5\}$	$\{f_2, f_4, f_5\}$
Parity5+5	1024	10	2	5	$\{f_2–f_4, f_6, f_8\}$	$\{f_2–f_4, f_6, f_8\}$	$\{f_2–f_4, f_6, f_8\}$
Parity5+2	1024	10	2	5	$\{f_3–f_7\}$	$\{f_1–f_5\}$	$\{f_1–f_5\}$
LED7	3200	7	10	7	$\{f_1–f_7\}$	$\{f_1–f_7\}$	$\{f_1–f_7\}$
Vote	435	16	2	9	$\{f_1–f_4, f_9, f_{11}, f_{13}, f_{15}, f_{16}\}$	$\{f_1–f_4, f_9, f_{11}, f_{13}, f_{15}, f_{16}\}$	$\{f_1–f_4, f_9, f_{11}, f_{13}, f_{15}, f_{16}\}$
Lenses	24	4	3	4	$\{f_1–f_4\}$	$\{f_1–f_4\}$	$\{f_1–f_4\}$
Zoo	101	16	7	5	$\{f_3, f_4, f_6, f_8, f_{13}\}$	$\{f_3, f_4, f_6, f_8, f_{13}\}$	$\{f_3, f_4, f_6, f_8, f_{13}\}$
Solar	323	12	6	10	$\{f_1–f_6, f_9–f_{12}\}$	$\{f_1–f_6, f_9–f_{12}\}$	$\{f_1–f_6, f_9–f_{12}\}$

Table 2 Search space sizes and running time of FSS for SFSD

Dataset	#All	MMIMRSC			FOCUS			ABB		
		#Ev	Ratio	t (ms)	#Ev	Ratio	t (ms)	#Ev	Ratio	t (ms)
Corral	2^6	7	0.109	2±0	42	0.656	6±0	12	0.188	3±0
Monk1	2^6	7	0.109	4±0	24	0.374	10±0	20	0.313	19±0
Monk3	2^6	7	0.109	4±0	35	0.547	15±0	20	0.313	19±0
Parity5+5	2^{10}	11	0.011	50±0	518	0.506	636±1	112	0.109	406±2
Parity5+2	2^{10}	11	0.011	49±0	386	0.377	419±0	228	0.223	650±2
LED7	2^7	8	0.063	50±0	127	0.992	492±1	8	0.063	51±0
Vote	2^{16}	17	0.000	42±1	39967	0.610	48595±51	908	0.014	2697±4
Lenses	2^4	5	0.313	1±0	15	0.938	1±1	5	0.313	1±0
Zoo	2^{16}	17	0.000	5±0	4951	0.076	884±13	25344	0.387	132456±377
Solar	2^{12}	13	0.003	10±0	4031	0.984	2823±1	24	0.006	32±0

#All is the whole search space size, #Ev is the number of feature subsets evaluated, Ratio is #Ev divided by #All, and t is denoted by mean±standard deviation

Table 3 Results of 10-fold cross validation of C4.5 for SFSD

Dataset	Before		After	
	Tree size	Err rate (%)	Tree size	Err rate (%)
Corral	12.56±1.33	0.86±2.57	13.00±0.00	0.00±0.00
Monk1	41.44±0.88	2.33±5.46	41.00±0.00	0.00±0.00
Monk3	19.00±0.00	0.00±0.00	19.00±0.00	0.00±0.00
Parity5+5	67.67±10.05	7.19±3.87	61.60±8.38	4.49±6.87
Parity5+2	63.00±0.00	0.00±0.00	62.40±1.90	0.98±3.10
LED7	73.20±3.82	26.66±1.50	73.20±3.82	26.66±1.50
Vote	15.70±4.99	5.32±4.26	15.40±5.44	4.89±3.44
Lenses	6.50±1.08	16.66±32.39	6.50±1.08	16.66±32.39
Zoo	18.60±3.50	8.00±6.32	19.40±2.45	6.32±6.84
Solar	34.10±4.04	28.84±4.30	32.50±2.36	30.06±5.96

'Before' denotes C4.5 running before FSS, and 'After' denotes C4.5 running after FSS. All results are denoted by mean±standard deviation

Table 4 Error rates of 10-fold cross validation of Naive-Bayes for SFSD

Dataset	Before (%)	After (%)
Corral	10.96±9.33	12.50±6.52
Monk1	25.00±4.72	24.49±5.17
Monk3	2.78±2.63	2.37±1.82
Parity5+5	63.25±4.55	63.14±4.57
Parity5+2	60.36±3.61	57.49±7.23
LED7	26.50±1.87	26.50±1.87
Vote	9.67±6.02	6.70±3.15
Lenses	34.99±41.90	34.99±41.90
Zoo	9.00±11.01	4.82±6.65
Solar	31.28±4.95	33.75±8.70

'Before' denotes Naive-Bayes running before FSS, and 'After' denotes Naive-Bayes running after FSS. All results are denoted by mean±standard deviation

sted BFSD. The FSS results, search space sizes and running time of FSS are shown in Table 5. Table 5 shows that MMIMRSC found the equivalent feature subset of F with a quite small size. LVF found much bigger feature subsets than MMIMRSC. Especially, there were still 279 features in the FSS result for Mfeat dataset. In this case, LVF almost did not complete a role of dimension reduction. The search space sizes and running time of MMIMRSC were also quite small. The search space sizes and running time of LVF were ignored since it was a stochastic search method.

The results of 10-fold cross validation of C4.5 are shown in Table 6. For Mushroom, the error rate after MMIMRSC running was the same as that of F. For Sonar, the tree size (from 55 to 30) and error rate (from 33.14% to 28.83%) were both decreased largely after MMIMRSC running. For Mfeat, the tree size (from 495 to 391) was decreased largely. Table 6 shows that C4.5 had smaller error rates after MMIMRSC running than after LVF running.

The error rates of 10-fold cross validation of Naive-Bayes are given in Table 7. Table 7 shows that all the error rates were decreased after MMIMRSC running. Especially for Mfeat, the error rate was decreased from 52.60% to 9.10% after MMIMRSC running, but the error rate was still 48.40% after LVF running. That means the MMIMRSC upgraded obviously the learning result of Naive-Bayes.

CONCLUSION

A hashing mechanism was used in this paper to calculate mutual information of feature subset. By mutual information, redundancy-synergy coefficient was defined as a measure of synergistic ability and redundancy of features. MMIMRSC was presented in terms of the information maximization rule.

Thirteen benchmark datasets were employed

Table 5 Results, search space sizes and running time of FSS for BFSD

Dataset	m	p	u	MMIMRSC				LVF
				FS	#Ev	r	t (ms)	r_{LVF}
Mushroom	8124	22	2	$\{f_5, f_8, f_{12}, f_{19}, f_{20}\}$	23	5	2389±3	6
Sonar	208	60	2	$\{f_{10}, f_{11}, f_{12}, f_{36}, f_{49}\}$	61	5	104±1	16
Mfeat	2000	649	10	$\{f_{73}, f_{77}, f_{131}, f_{185}, f_{257}, f_{644}, f_{645}, f_{647}, f_{649}\}$	650	9	83107±51	279

FS denotes the feature subset selected by MMIMRSC, t is denoted by mean±standard deviation, r_{LVF} is the size of feature subset selected by LVF

Table 6 Results of 10-fold cross validation of C4.5 for BFSD

Dataset	Before		After MMIMRSC		After LVF	
	Tree Size	Err Rate (%)	Tree Size	Err Rate (%)	Tree Size	Err Rate (%)
Mushroom	31.20±1.52	0.00±0.00	33.00±0.00	0.00±0.00	31.70±1.49	0.00±0.00
Sonar	55.00±12.65	33.14±12.16	30.00±11.97	28.83±7.11	41.00±12.64	31.62±8.74
Mfeat	495.00±45.75	11.00±1.65	391.00±47.48	12.05±2.30	433.00±43.71	13.25±2.75

Table 7 Error rates of 10-fold cross validation of Naive-Bayes for BFSD

Dataset	Before (%)	After MMIMRSC (%)	After LVF (%)
Mushroom	0.42±0.21	0.17±0.13	0.32±0.26
Sonar	29.79±17.97	24.50±4.62	29.62±12.38
Mfeat	52.60±2.53	9.10±1.54	48.40±2.58

to test MMIMRSC. The feature set size ranged from 4 to 649, and the sample set size ranged from 24 to 8124. Experiments showed that MMIMRSC gives good feature selection results quickly, especially for high-dimensioned datasets. Importantly, the feature subsets selected by MMIMRSC improve the learning performance of the machine learning method.

References

Almuallim, H., Dietterich, T.G., 1991. Learning with Many Irrelevant Features. Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91), Anaheim, California, p.547-552.

Blum, A.L., Rivest, R.L., 1992. Training a 3-node neural network is NP-complete. *Neural Networks*, **5**:117-127.

Brenner, N., Strong, S.P., Koberle, R., Bialek, W., De Ruyter van Steveninck, R., 2000. Synergy in a neural code. *Neural Computation*, **13**(7):1531-1552.

Chen, B., Hong, J.R., Wang, Y.D., 1997. Minimum feature subset selection problem. *Journal of Computer Science and Technology*, **12**:145-153.

Cover, T.M., 1991. Elements of Information Theory. Wiley, New York.

Fano, R., 1961. Tranmission of Information: A Statistical Theory of Communications. Wiley, New York.

Liu, H., Motoda, H., 1998. Feature Selection for Knowledge Discovery and Data Mining. Kluwer Acadcemic Press, Boston.

Liu, H., Motoda, H., Dash, M., 1998. A Monotonic Measure for Optimal Feature Selection. Proceedings of ECML-98, p.101-106.

Liu, H., Setiono, R., 1996. A Probabilistic Approach to Feature Selection – A Filter Solution. *In*: ICML-96. Morgan Kaufmann Publishers, p.319-327.

Murphy, P.M., Pazzani, M.J., 1994. Exploring the decision forest: An empirical investigation of Occam's razor in decision tree induction. *Journal of Art. Intel.*, **1**: 257-319.

Narendra, P., Fukunaga, K., 1977. A branch and bound method for feature subset selection. *IEEE Trans. on Computer*, **26** (9):917-922.

Yaglom, A.M., Yaglom, I.M., 1983. Probability and Information. D. Reidel Publishing Company.