Improved Mutual Information Feature Selector for Neural Networks in Supervised Learning

Nojun Kwak and Chong-Ho Choi
{triplea|chchoi}@csl.snu.ac.kr
School of Electrical Eng., ERC-ACI, ASRI, Seoul National University, Seoul, Korea

Abstract

In classification problems, we use a set of attributes which are relevant, irrelevant or redundant. By selecting only the relevant attributes of the data as input features of a classifying system and excluding redundant ones, higher performance is expected with smaller computational effort. In this paper, we propose an algorithm of feature selection that makes more careful use of the mutual informations between input attributes and others than MIFS [3]. The proposed algorithm is applied in several feature selection problems and compared with MIFS. Experimental results show that the proposed algorithm can be well used in feature selection problems.

1. Introduction

Input feature selection plays an important role in classification problems. Irrelevant and redundant attributes in input features not only complicate the network structure, but also degrade the performance of the networks. By selecting only the relevant attributes systematically, higher performance can be achieved with smaller number of input features.

This feature selection problem has been tackled by some researchers, and one of the most popular methods for this problem is the PCA (principal component analysis) which transforms the existing attributes into new ones thought to be crucial for classification [1]. But this method does not fit in the aspect of the maintenance of data as it needs to process the whole data when a new data is added. And the main drawback of this method is that it is not invariant under transformation. Merely scaling the attributes changes results. Recently this problem has been dealt with intensely and some solutions have been proposed. One of most important contributions is to find relevant attributes one by one using the decision trees. But these methods have some problems in memory management or consuming time. Battiti's MIFS (mutual information feature selector) uses mutual information between inputs and outputs [3]. In his paper he showed that mutual information can be very useful in feature selection problems and the MIFS can be used for any classifying system without regard to learning algorithms for its simplicity. In this paper we propose an algorithm which improves the performance of MIFS.

In the following section the shortcoming of MIFS is analyzed and an improved version of MIFS is proposed. In section 3, the proposed algorithm is applied to several problems to show the effectiveness of the proposed methods. And finally conclusions and future works follows in section 4.

2. Improved Feature Selector using Mutual Information

In this section a new algorithm for input feature selection using mutual information is presented.

FRn-k Problem In the process of selecting input features, we would like to reduce the number of input features by excluding irrelevant or redundant features among the features that can be extracted from raw data. This concept is formalized as selecting most relevant k features from a set of n features, and Battiti named it as a "feature reduction" problem [3]:

FRn-k Given an initial set of n features, find the subset with $k < n$ features that is "maximally informative" about the class.

From the information theory, mutual information between two random variables measures the amount of common information contained in these variables [4]. The problem of selecting input features which contain much of the information of output can be solved by computing the mutual information between input features and output classes. If the mutual information between input features and output classes could be exactly obtained, the *FRn-k* problem could be reformulated as follows:

FRn-k Given an initial set F with n features, find the subset $S \subset F$ with k features that minimizes $H(C|S)$, i.e., that maximizes the mutual information $I(C;S)$.

Here $H(C|S)$ is the conditional entropy of C for given S, and $I(C;S)$ is the mutual information between C and S.

0-7803-5529-6/99/$10.00 ©1999 IEEE

We can think of three strategies for solving this *FRn-k* problem. The first one is 'generate and test' strategy. Namely, all the feature subsets S are generated and their $H(C|S)$ are compared. As a matter of course, this is optimal but it is almost impossible for too many number of combinations. Secondly, we can think 'backward elimination' strategy. In this strategy, from the full feature set F that contains n elements we eliminate the worst feature one-by-one till k elements remain. This method also has quite a lot of difficulties in computing $H(C|S)$ [1]. The final strategy is 'greedy selection'. In this method, from the empty set of selected features, we add the best feature of the current state one by one. This ideal greedy selection algorithm using mutual information can be 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) $\forall f \in F$, compute $I(C;f)$.
3. (Selection of the first feature) find the feature that maximizes $I(C;f)$, set $F \longleftarrow \setminus \{f\}$, $S \longleftarrow \{f\}$.
4. (Greedy selection) repeat until desired number of features are selected.
   (a) (Computation of the joint MI between variables) $\forall f \in F$, compute $I(C;f,S)$.
   (b) (Selection of the next feature) choose the feature $f \in F$ that maximizes $I(C;f,S)$, and set $F \longleftarrow F \setminus \{f\}$, $S \longleftarrow \{f\}$.
5. Output the set S containing the selected features.

To compute mutual information we must know the exact pdf of variables, but in practice, it is hard to know and the best we can do is to use the histogram of the data.

If the output classes are composed of K_c classes and we divide the input space of each input feature i into P_i partitions to get the histogram, there must be $K_c \times \Pi_{i=1}^{k} P_i$ cells to compute $I(C;f,S)$. In this case, even a simple problem of selecting 10 important features, we need $K_c \times 10^{10}$ memories if each feature space is divided into 10 partitions. So realization of the ideal greedy selection algorithm is practically impossible. To overcome this practical obstacle an alternative method of computing $I(C;f,S)$ has to be devised.

MIFS and Its Limitation Battiti proposed an algorithm for solving *FRn-k* problem using mutual information as mentioned before [3]. Instead of calculating $I(C;f,S)$, the mutual information between a *candidate for newly selected feature* f plus *already selected feature vector* S and the class variable c, he used only $I(C;f)$ and $I(f;f')$ where f and f' are individual features. Then the 'greedy' selection algorithm proceeds.

The MIFS is the same as the ideal greedy selection algorithm except the step 4. It is replaced as follows [3] :

4 (Greedy selection) repeat until desired number of features are selected.
   (a) (Computation of the MI between variables) for all couples of variables (f,s) with $f \in F, s \in S$ compute $I(f;s)$, if it is not already available.
   (b) (Selection of the next feature) choose feature $f \in F$ as the one that maximizes $I(C;f) - \beta \sum_{s \in S} I(f;s)$; set $F \longleftarrow F \setminus \{f\}$, $S \longleftarrow \{f\}$.

Here β is a redundancy variable which is used to consider the redundancy between input features. If $\beta = 0$, mutual information between input features is not taken into consideration and the algorithm selects the features in the order of the mutual information between input features and output classes, so the redundancy between input features is never reflected. As β grows, the mutual informations between input features influence the selection procedure much and the redundancy is reduced. But in the case of too large β, the algorithm only considers the relation between inputs and cannot reflect the input-output relation.

The relation between input features and output classes can be represented as in Fig. 1 where f_i is the feature to be selected, f_s is the already selected feature, C represents the output class. The ideal greedy feature selection algorithm using mutual information chooses the feature f_i that maximizes joint mutual information $I(C;f_i,f_s)$ which is the area 2,3, and 4 in Fig. 1. This is shown as dashed area in Fig. 2. Because $I(C;f_s)$ region (area 2 and 4) is common for all the unselected features f_i in computing the joint mutual information $I(C;f_i,f_s)$, the ideal greedy algorithm selects the feature f_i that maximizes the area 3 in Fig. 1. On the other hand, MIFS selects the feature that maximizes $I(C;f_i) - \beta I(f_i;f_s)$. For $\beta = 1$, MIFS maximizes (area 3 - area 1) as shown in Fig. 3, and this is different from ideal one. The MIFS maximizes the region subtracting the area 1 from the ideal one in the figure.

So if a feature is closely related to the already selected feature f_s, the area 1 in Fig. 3 is large and this degrades the performance of MIFS.

For this reason, MIFS does not work well in nonlinear problems as the following example.

Example 1 *Each of the random variables X and Y is uniformly distributed on [-0.5,0.5], and assume that there are 3 input features $X, X-Y$ and Y^2 The output class Z are given as*

$$Z = \begin{cases} 0 & \textit{if } X + 0.2Y < 0 \\ 1 & \textit{if } X + 0.2Y \geq 0 \end{cases}$$

[1] The number of memory cells needed to compute $H(C|S)$ is $K_c \times \Pi_{i=1}^{m} P_i$, where P_i is the number of partitions for ith input feature space, and m is the size of S or the number or elements in S

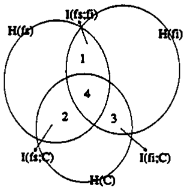


Figure 1: Relation between Input Features and Output Class

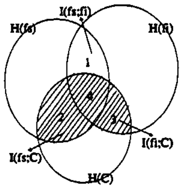


Figure 2: Ideal Algorithm

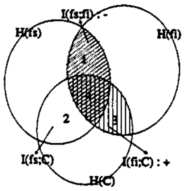


Figure 3: MIFS Algorithm

Table 1: MIFS Results for *Example 1*

(a) MI between output class ($I(f_i; Z)$)

X	$X-Y$	Y^2
0.8459	0.2621	0.0170

(b) MI between input feature and output class ($I(f_i; f_j)$)

	X	$X-Y$	Y^2
X	–	0.6168	0.0610
$X-Y$	0.6168	–	0.5624
Y^2	0.0610	0.5624	–

(c) $I(f_i; Z) - I(f_i; f_s)$

$X-Y$	$I(X-Y; Z) - I(X-Y; X) = -0.3537$
Y^2	$I(Y^2; Z) - I(Y^2; X) = -0.0439$

(d) Selection Order

	X	$X-Y$	Y^2
Selection Order (Ideal)	1	2	3
Selection Order (MIFS ($\beta = 1$))	1	3	2

When we take 1,000 samples and partition each input feature space into 10, the mutual informations between each input feature and the output classes and those between input features are shown in TABLE 1. The order of selection using MIFS($\beta = 1$) is also shown and is $X, Y^2, X-Y$ in that order.

As shown in TABLE 1 (c) the MIFS selects Y^2 rather than more important feature $X - Y$ as the second one [2].

It is due to the relatively large β, and this is an example where the relations between the inputs are weighted too much. This is due to the difference of the algorithm from the ideal one described in the above.

Proposed Algorithm (Improved MIFS) Now we propose a feature selection algorithm based on information theory that is closer to the ideal one than MIFS. The ideal algorithm tries to maximize $I(C; f_i, f_s)$ (area 2,3,4 in Fig. 2) and this can be rewritten as

$$I(C; f_i, f_s) = I(C; f_s) + I(C; f_i | f_s). \qquad (1)$$

Here $I(C; f_i|f_s)$ represents the remaining mutual information between class C and feature f_i for given f_s. This is shown as area 3 in Fig. 2, whereas the area 2 plus 4 represents $I(C; f_s)$. For all the candidate features to be selected in the ideal feature selection algorithm, $I(C; f_s)$ is common and there is no need to compute this. So the ideal greedy algorithm now tries to find the feature that maximizes $I(C; f_i|f_s)$ (area 3) in (1). But in general to calculate $I(C; f_i|f_s)$, we need to divide the input feature space into lots of partitions and this is practically impossible [3].

So we will approximate $I(C; f_i|f_s)$ with $I(f_s; f_i)$ and $I(C; f_i)$, which are relatively easy to calculate. The conditional mutual information $I(C; f_i|f_s)$ can be represented as

$$I(C; f_i|f_s) = I(C; f_i) - \{I(f_s; f_i) - I(f_s; f_i|C)\}. \quad (2)$$

Here $I(f_s; f_i)$ is area 1 and 4 and $I(f_s; f_i|C)$ represents area 1 respectively. So the braced term $I(f_s; f_i) - I(f_s; f_i|C)$ represents area 4 in Fig. 1. The term $I(f_s; f_i|C)$ means the mutual information between already selected feature f_s and the candidate feature f_i for given class C. If conditioning by the class C does not change the ratio of the entropy of f_s and the mutual information between f_s and f_i, or the following relation holds,

$$\frac{H(f_s|C)}{H(f_s)} = \frac{I(f_s; f_i|C)}{I(f_s; f_i)}, \qquad (3)$$

$I(f_s; f_i|C)$ can be represented as

$$I(f_s; f_i|C) = \frac{H(f_s|C)}{H(f_s)} I(f_s; f_i). \qquad (4)$$

Using the equation above and (2) together we obtain

$$\begin{aligned} I(f_i; C|f_s) &= I(f_i; C) - (1 - \frac{H(f_s|C)}{H(f_s)}) I(f_s; f_i) \\ &= I(f_i; C) - \frac{I(f_s; C)}{H(f_s)} I(f_s; f_i). \qquad (5) \end{aligned}$$

The condition (3) holds when information is distributed evenly throughout the $H(f_s)$ region in Fig. 1. When we compute 3 for *Example 1*, the result is as in TABLE 2. It shows that our assumption holds with less than 10% error. With this formula, we revise the step 4 of the ideal greedy selection algorithm as follows:

[2] Y can be exactly calculated by the linear combination of X and $X-Y$. Because the output class Z can be computed exactly by X and $X-Y$, we can say $X-Y$ rather than Y^2 is much informative about Z for given X.

[3] The number of partitions needed to calculate $I(C; f_i|f_s)$ is the same as in calculating the joint entropy $H(f_i, f_s, C)$, which is very difficult.

Table 2: Validation of (3) for *Example 1*

$H(f_s|C)/H(f_s)$

$H(X)$	3.3181
$H(X\|Z)$	2.4723
$H(X\|Z)/H(X)$	0.745

$I(f_s;f_i|C)/I(f_s;f_i)$

$I(X-Y;X)$	0.6168	$I(Y^2;X)$	0.0610
$I(X-Y;X\|Z)$	0.4379	$I(Y^2;X\|Z)$	0.0491
$I(X-Y;X\|Z)/I(X-Y;X)$	0.709	$I(Y^2;X)/I(Y^2;X\|Z)$	0.805

4 (Greedy selection) repeat until desired number of features are selected.

  (a) (Computation of entropy) $\forall s \in S$, compute $H(s)$ if it is not already available.

  (b) (Computation of the MI between variables) for all couples of variables (f, s) with $f \in F, s \in S$ compute $I(f;s)$, if it is not already available.

  (c) (Selection of the next feature) choose feature $f \in F$ as the one that maximizes $I(C;f) - \beta \sum_{s \in S} \frac{I(C;s)}{H(s)} I(f;s)$; set $F \longleftarrow F \backslash \{f\}, S \longleftarrow \{f\}$.

Here the entropy $H(s)$ can be computed in the process of computing the mutual information with output class C, so there is little change in computational load with respect to MIFS.

The variable β gives flexibility to the algorithm as in MIFS. If we set β zero the proposed algorithm chooses features in the order of the mutual information with the output. As β grows, it deselects the redundant features more efficiently. In general we can set $\beta = 1$ in compliance with (5). For all the experiments to be discussed later we set it 1 when there is no comment.

3. Experimental Results

In this section we will represent some experimental results using the algorithm proposed in section 2 for several datasets.

Improved MIFS vs. MIFS for *Example 1*

For the *Example 1* we compared our algorithm with MIFS for some different β, and the results are shown in Table 3. The data consist of 1,000 patterns and all the 3 input features are normalized as the value on [0,1]. The entropy and mutual information were calculated by partitioning each input feature space into 10 partitions.

The result shows that when using MIFS if β are chosen $0.5 \sim 1$ as Battiti suggested [3], the selection is not as expected. The reason is from the characteristic of the problem which is: after the first selection of the feature X that has the greatest mutual information with the output, the mutual information with the X has too much influence on the procedure of the second feature selection as shown in Table 1 (c). Our improved MIFS performs well for all values of β including for the suggested value 1.

Table 3: Comparison of MIFS and Improved MIFS for *Example 1* (F1 = X, F2 = $X - Y$, F3 = Y^2)

(a) Results with MIFS

β	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4
First Selection	F1	F1	F1	F1	F1	F1	F1	F1
Second Selection	F2	F2	F2	F3	F3	F3	F3	F3
Third Selection	F3	F3	F3	F2	F2	F2	F2	F2

(b) Results with Improved MIFS

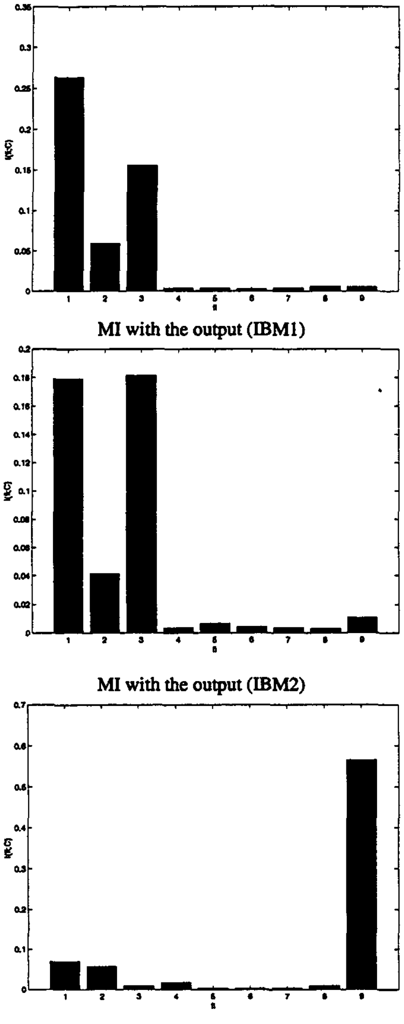
β	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4
First Selection	F1	F1	F1	F1	F1	F1	F1	F1
Second Selection	F2	F2	F2	F2	F2	F2	F2	F2
Third Selection	F3	F3	F3	F3	F3	F3	F3	F3

IBM datasets These datasets had been generated by Agrawal *et al.* to test their data mining algorithm CDP, and Setiono *et al.* also used for testing the performance of their feature selector [2]. All the patterns of the datasets consist of nine attributes and the three classification functions are as in [2].

In this paper we generated 1,000 input-output patterns and all the nine attributes were normalized in the value on [0,1]. Each input space was divided into ten partitions to compute the entropies and the mutual informations. For convenience, we will refer three datasets generated by using each function as IBM1, IBM2, IBM3 and nine input features as $F1, F2, \cdots, F9$ respectively. Fig. 4 shows the mutual information between each input feature and the output class for IBM1, IBM2, and IBM3 datasets. In Table 4 we compared the feature selection results of our improved MIFS and conventional MIFS for the three datasets. We also showed the selection results for $\beta = 0$ case. The selection order is exactly the same as the order of the mutual information with the output shown in Fig. 4 for $\beta = 0$. The important features used in classification functions are bold faced in the table.

As we can see in Table 4 both MIFS and Improved MIFS selected correct features as desired for IBM1 and IBM2 datasets. Note that for IBM2 when $\beta = 0$ the important feature $F4$ is chosen as eighth important one, while when we set $\beta = 1$, it is selected third for both MIFS and improved MIFS. This shows that both MIFS and improved MIFS can exclude the redundant features effectively.

For IBM3 dataset the classification is determined by four features, i.e., salary($F1$), commission($F2$), elevel($F4$), and loan($F9$). So these four features must chosen as important ones in good feature selectors. Table 4 shows that MIFS selects $F1, F4, F9$ in the first three selection, but $F2$ is classified as the worst feature among the nine. This is one example of too much consideration of the mutual information



MI with the output (IBM1)

MI with the output (IBM2)

MI with the output (IBM3)

Figure 4: Mutual information between each feature and the output class for IBM datasets

between features. As noted in section III the MIFS's ability of excluding redundant features may result in bad performance. For IBM3 we can see that our improved MIFS selected the four features in order. In Table 5 we showed the selection results of IBM3 for various βs. Note the change of the selection order of the feature $F2$ over β in MIFS case. Even for the relatively small β MIFS regarded $F2$ as bad features, while Improved MIFS does not. This shows that improved MIFS performs better than MIFS for most values of β in this case.

Table 4: Feature Selection Result for IBM datasets. The bold faced features are the crucial ones in the classification.

IBM 1

	F1	F2	**F3**	F4	F5	F6	F7	F8	F9
MIFS/Improved MIFS ($\beta = 0$)	**1**	3	**2**	8	7	9	6	4	5
MIFS ($\beta = 1$)	**1**	9	**2**	3	5	4	8	6	7
Improved MIFS ($\beta = 1$)	**1**	9	**2**	3	6	8	7	4	5

IBM 2

	F1	F2	**F3**	**F4**	F5	F6	F7	F8	F9
MIFS/Improved MIFS ($\beta = 0$)	**2**	3	**1**	**8**	5	6	7	9	4
MIFS ($\beta = 1$)	**2**	9	**1**	**3**	5	4	8	6	7
Improved MIFS ($\beta = 1$)	**2**	9	**1**	**3**	5	6	7	8	4

IBM 3

	F1	**F2**	F3	**F4**	F5	F6	F7	F8	**F9**
MIFS/Improved MIFS ($\beta = 0$)	**2**	**3**	6	**4**	8	9	7	5	**1**
MIFS ($\beta = 1$)	**2**	**9**	7	**3**	5	4	8	6	**1**
Improved MIFS ($\beta = 1$)	**2**	**3**	5	**4**	8	7	9	6	**1**

Table 5: Feature Selection Order of IBM3 for Various βs

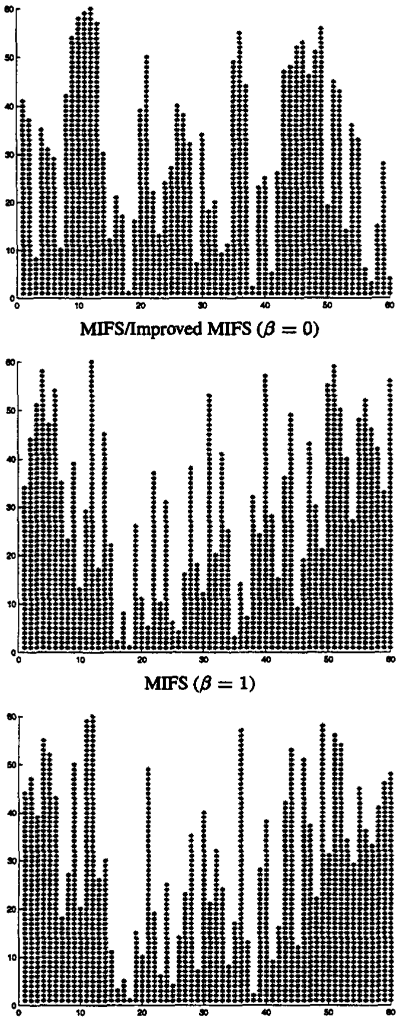
MIFS

β	Selection Order								
0.0	F9	F1	**F2**	F4	F8	F3	F7	F5	F6
0.1	F9	F1	F4	F8	F3	F6	F5	**F2**	F7
0.2	F9	F1	F4	F6	F8	F5	F3	F7	**F2**
0.5	F9	F1	F4	F6	F5	F8	F3	F7	**F2**
0.7	F9	F1	F4	F6	F5	F8	F3	F7	**F2**
1.0	F9	F1	F4	F6	F5	F8	F3	F7	**F2**
1.2	F9	F1	F4	F6	F5	F8	F3	F7	**F2**
1.5	F9	F1	F4	F6	F5	F8	F3	F7	**F2**

Improved MIFS

β	Selection Order								
0.0	F9	F1	**F2**	F4	F8	F3	F7	F5	F6
0.1	F9	F1	**F2**	F4	F8	F3	F7	F5	F6
0.2	F9	F1	**F2**	F4	F8	F3	F7	F5	F6
0.5	F9	F1	**F2**	F4	F8	F3	F6	F5	F7
0.7	F9	F1	**F2**	F4	F8	F3	F6	F5	F7
1.0	F9	F1	**F2**	F4	F3	F8	F6	F5	F7
1.2	F9	F1	**F2**	F4	F3	F8	F6	F5	F7
1.5	F9	F1	**F2**	F4	F3	F6	F8	F5	F7

Sonar target dataset This dataset was constructed to discriminate between the sonar returns bounced off a metal cylinder and those of a rock for identification of a submarine, and it was used by Battiti to test the MIFS's performance [3]. This dataset is consist of 208 patterns that has 60 input features and two output classes, *metal/rock*. As in [3], 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 informations. Unlike the IBM datasets, for this dataset as we cannot know which feature is important *a priori*, we selected 3 ~ 12 features namely top 5% ~ 20% of the 60 features, and trained the neural network using these input features. We compared the classification rates of MIFS and



MIFS/Improved MIFS ($\beta = 0$)

MIFS ($\beta = 1$)

Improved MIFS ($\beta = 1$)

Figure 5: Selection Order for the Sonar dataset - MIFS and Improved MIFS

that of Improved MIFS. Neural networks with three layers (input/hidden/output) were used and the hidden layer has three nodes as in [3]. The conventional back-propagation learning algorithm was used with the momentum of 0.9 and 0.01 learning rate. We trained the network 30,000 iteration for all cases.

Fig. 5 shows the selection results MIFS and Improved MIFS for sonar target dataset. In the figure the x-axis denotes 60 features and the y-axis is the selection order corresponding to the feature, the larger the magnitude of the feature the sooner the feature selected.

In Table 6 we compared the performance of MIFS and Improved MIFS. The classification rates for various number of selected features are compared. The resultant classification rates are the averages of three experiments each.

Table 6: Classification Rate for Various Number of Selected Features for Sonar Dataset (%)

Number of features	MIFS	Improved MIFS
3	60.10	71.15
4	63.46	73.56
6	67.30	81.73
9	71.15	87.02
12	91.83	94.23
All (60)	100	

The results shows that Improved MIFS performs better than MIFS over 10% of classification rate for the 3 ~ 9 selection of features. In 12 case it works better, too.

4. Conclusion

The feature selection in neural networks is very important part, regardless of the learning algorithm which is used to train the network. Due to the existence of irrelevant and redundant attributes, by selecting only the relevant attributes of the data, higher predictive accuracy can be acquired. In this paper, the problem of feature selection, which is to select only the important features in classification procedure is dealt with.

Among many solutions of this problem, the algorithms based on the information theory are preferable for they do not need much time, while the others tend to have problems of taking too much time in training the network. The existing MIFS, one of these algorithms, has very good aspect of excluding redundant features effectively, but it may fail when redundant features have much information about the output.

To resolve this problem we proposed improved MIFS that has both abilities of excluding the redundant features and considering the amount of output informations contained in the input features. The improved MIFS performed better than the conventional one for most of problems tested.

The improved MIFS can be used for many situations that needs selecting features among various candidates.

References

[1] Joliffe, I.T., *Principal Component Analysis*, New York: Springer-Verlag, 1986.

[2] R. Setiono and H. Liu, "Neural network feature selector," *IEEE Trans. Neural Networks*, vol. 8, no. 3, May 1997.

[3] Roberto Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Networks*, vol. 5, no. 4, July 1994.

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