A New Mutual Information Based Measure for Feature Selection

Ahmed Al-Ani, Mohamed Deriche and Jalel Chebil
Signal Processing Research Centre
Queensland University of Technology
GPO Box 2434, Brisbane, Q 4001, Australia
a.alani@qut.edu.au, m.deriche@qut.edu.au, j.chebil@qut.edu.au

Abstract

In this paper, we discuss the problem of feature selection and the importance of using mutual information in evaluating the discrimination ability of feature subsets between class labels. Because of the difficulties associated with estimating the exact value of mutual information, we propose a new evaluation measure that is based on the information gain and takes into consideration the interaction between features. The proposed measure is integrated into a robust feature selection scheme and compared with the well-known mutual information feature selection (MIFS) algorithm using the problems of texture classification, speech segment classification and speaker identification.

Keywords – Feature selection, filter evaluation function, information measure.

1 Introduction

Feature selection is a very important step in classification since the inclusion of irrelevant and redundant features often degrade the performance of classification algorithms both in speed and prediction accuracy. The objective of feature selection is to find the smallest subset of features that minimizes the probability of error. Ideally, this can be achieved by examining all possible subsets and finding the one that satisfies the above criterion. This approach is known as exhaustive feature selection. Even with a moderate number of features, the exhaustive selection is impractical because of its high computational requirements. Other feature selection methods were developed to reduce computational complexity by compromising performance.

All feature selection methods need to use some sort of evaluation function together with a search procedure to find the optimal feature set. The evaluation functions measure how good a specific subset can be in discriminating between classes, and can be divided into two main groups: filters and wrappers. Filters measure the relevance of feature subsets independently of any classifier, whereas wrappers use the classifier's performance as the evaluation measure. Search procedures, on the other hand, are methods that only consider small portions of all possible subsets, e.g. the branch and bound [13], stepwise [12], genetic algorithms [16], etc. In this paper, our objective is to develop an evaluation function that can be used with any search procedure. We will consider filter evaluation measures because they are much faster than wrappers and can handle large datasets [4]. A variety of filter-based measures have already been proposed in the literature. The most popular fall under the following three categories: distance measures, consistency measures, and information measures.

Distance measures for a subset of features are based on inter-class separation, or distance between classes. Logically, the larger the separation between classes, the easier it will be to define a decision boundary, and to achieve a lower error rate [5, 6]. The main problem with using distance measures as evaluation functions is that some feature subsets may have large inter-class distances, but subsequently produce poor classification results [10]. This is because of the curse of dimensionality, where as the number of features of a dataset increases, the resulting of estimates of inter-class distance become less reliable.

Consistency measures on the other hand rely heavily on the training dataset in order to check the consistency of a given subset. A subset is said to be inconsistent if there are at least two patterns in the dataset having equal values for all features under examination with different class labels [1]. Features that have many different values make problem for this kind of measure. An extreme example is a feature that has as many values as there are patterns. Such feature has little power to generalize beyond the training data, and yet according to this measure is the most consistent [9].

The information measure is based on the concept of mutual information (MI). MI measures arbitrary dependencies between random variables and thus is suitable for assessing the "information content". Fano [7] has shown that maximizing the MI between transformed data and the desired target achieves the lowest probability of error. This idea has inspired Battiti in developing his feature selection method, which he named mutual information feature selection (MIFS) [2]. The method evaluates MI between individual features and class labels, and selects those features that have maximum MI with class labels and are less redundant. The drawback of the MIFS algorithm is that it does not take into consideration the interaction between features. In [14], it has been proved that choosing features individually does not lead to an optimal solution.

Another way to reduce the dimensionality of the original feature set is through feature transformation. Unlike feature selection, feature transformation (or subspace mapping) reduces the dimensionality of features by transforming all of them, either linearly or nonlinearly, to a lower dimension set. If the original feature set $\mathcal{F}$ consists of N features, then a new feature set $\mathcal{S}$ that consists of M features, $M < N$, can be generated through the following transformation: $\mathcal{S} = W\mathcal{F}$, where W is the transformation matrix. A well-known feature transformation method is the principal component analysis (PCA), which is a classical statistical method which is usually used to create an ordered orthogonal basis with the first eigenvector corresponding to the largest variance in the input set [5]. In other words, PCA is a general-purpose redundancy reduction method that does not take into consideration relationships with the class labels. In this study, we will only be concerned with feature selection, and will propose a new information-based evaluation function that overcomes the drawbacks of the MIFS algorithm. The performance of our proposed method will be compared to that of the MIFS and PCA.

The paper is organized as follows: The measurement of MI and its importance in performing feature selection is explained in section two. Section three presents our proposed evaluation function. Experimental results and comparison in terms of classification accuracy between our proposed method, Battiti's method and the PCA is presented in section four, while section five concludes the paper.

2 Mutual Information

2.1 Definition and Measurement

The MI between random variables X and Y, $I(X;Y)$, measures the amount of information in X that can be predicted when Y is known. If X and Y are continuous,

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= \int P_{XY}(x,y) \log[P_{XY}(x,y)/P_X(x)P_Y(y)]dx\,dy \end{aligned} \tag{1}$$

where $H(X)$ is the entropy of X, which is a measure of its uncertainty, $H(X|Y)$ is the conditional entropy, which represents the uncertainty in X after knowing Y. When we deal with real data, the main problem for evaluating $I(X;Y)$ is the estimation of probabilities $P_X(x)$,$P_Y(y)$ and $P_{XY}(x,y)$. One possible solution is to subdivide the XY plane into boxes of size $\Delta x \Delta y$. By doing so, we are able to estimate the *discrete* values of P_X, P_Y and P_{XY}. Fraser and Swinney [8] proposed an alternative approach which uses variable box size over the XY plane, while Darbellay and Vajda [3] developed a method that estimates the MI by an adaptive partitioning of the observation space. However, for simplicity, we used here a fixed box size implementation.

Given the assumption above, the MI can be rewritten as:

$$I(X;Y) = \sum_{r_x} \sum_{r_y} P_{XY}(r_x, r_y) \log[P_{XY}(r_x, r_y)/P_X(r_x)P_Y(r_y)] \tag{2}$$

where, r_x and r_y are the discrete levels for X and Y respectively. If $r_x = r_y = R$, we would need R^2 boxes to estimate P_{XY}. In the case of three variables, we would need R^3 boxes to estimate P_{XYZ}, and so on. This number becomes very large as the number of variables increases. Note that Eq. 2 is also applicable to random variables which are originally discrete, and hence, it is applicable to categorical features, where r_x in this case represent the categories.

2.2 Importance of Using MI for Feature Selection

An evaluation function is defined as a measure of the ability of feature subsets in distinguishing between class labels. If a number of feature subsets have different levels of knowledge about the class labels, then an evaluation function is said to be optimal if it can rank the subsets according to their discrimination abilities.

To show the importance of using MI in feature selection, we consider here a simple case of five input variables, $\mathcal{F} = \{f_1, f_2, f_3, f_4, f_5\}$, and three class labels, $\mathcal{C} = \{c_1, c_2, c_3\}$, which are pre-specified using linear combinations of input variables. An Artificial neural network (ANN) trained with a backpropagation algorithm was used as a classifier. We first measured the MI between class labels and all possible subsets of 1, 2, 3 and 4 variables, and sorted them accordingly, then a comparison was performed between these sorted subsets with respect to their classification accuracy. The results, displayed in Fig. 1, show that for the four cases of 1, 2, 3 and 4 variables, and for almost all subsets, the classification accuracy increases as the MI between variable subsets and class label increases. This monotonic characteristic between these two quantities can provide useful information about the ability of feature subsets in distinguishing between class labels (*i.e.*, the highest the MI, the better the classification accuracy). Therefore, the importance of any subset, $\mathcal{L}$, can be measured by $I(\mathcal{C};\mathcal{L})$.

Insert Fig. 1 here

However, because of the computational load involved in estimating $I(\mathcal{C};\mathcal{L})$, especially when $\mathcal{L}$ contains a large number of features, (as described in section 2.1), we need to find an alternative evaluation function that is computationally feasible and yet reflects the goodness of each subset.

The MIFS algorithm reduces the computational load using a "greedy" procedure that only computes $I(\mathcal{C};f)$ and $I(f;f')$, where f and f' are individual features, according to the following evaluation function:

$$g_{_{MIFS}}(f) = I(\mathcal{C};f) - \frac{\beta}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} I(f;s) \tag{3}$$

where $\mathcal{S}$ is the subset of already selected features and the parameter β regulates the relative importance of MI between the non-selected feature f and the already selected feature s with respect to MI between f and $\mathcal{C}$.

Insert Fig. 2 here

Fig. 2 shows the classification accuracy of the different feature subsets and their corresponding MIFS evaluation measures. It is clear from both the number and amount of fluctuations for the three cases that the MIFS is far from achieving the objective of optimal evaluation function. This is mainly because this measure evaluates features individually and it does not perfectly regulates the importance of each feature with respect to its dependency with the already selected features. The next section describes a new proposed evaluation function that overcomes the drawbacks of the MIFS algorithm by considering the interaction between features and appropriately evaluating the redundancy with the already selected features.

3 The Proposed MI-based Evaluation Function

The new proposed evaluation function for a given subset $\mathcal{L}$, denoted as $g(\mathcal{L})$, will be based on the MI but with less computational complexity than $I(\mathcal{C};\mathcal{L})$. Accordingly, $g(.)$ should have similar properties to $I(\mathcal{C};.)$, and thus should satisfy the following conditions for any two subsets $\mathcal{K}$ and $\mathcal{L}$[1]:

1. Lower bound: if $\mathcal{K} \subset \mathcal{L}$, then $g(\mathcal{K} \cup \mathcal{L}) = g(\mathcal{L})$.
2. Upper bound: if $\mathcal{K}$ and $\mathcal{L}$ are independent, then $g(\mathcal{K} \cup \mathcal{L}) = g(\mathcal{K}) + g(\mathcal{L})$.
3. Monotonicity: if $\mathcal{K} \subset \mathcal{L}$, then $g(\mathcal{K}) \leq g(\mathcal{L})$.

The first and second properties highlight the dependency between features. If the features in $\mathcal{K}$ are either irrelevant or totally dependent upon those in $\mathcal{L}$, then $\mathcal{K}$ does not provide any new information compared to that of $\mathcal{L}$. However, if $\mathcal{K}$ and $\mathcal{L}$ are independent, then $\mathcal{K}$ provides new information that is not given by $\mathcal{L}$. The relevance of each feature, f_i, can be indicated by its MI with class labels, $I(\mathcal{C}; f_i)$. The degree of dependency between two features f_i and f_j is measured using $I(f_i; f_j)$. The monotonic property guarantees that any feature added to $\mathcal{K}$ will not reduce the new value of the evaluation function. Another important property that should be considered is the interaction between features. f_i and f_j are said to be working well together if their joint MI with class labels, $I(\mathcal{C}; \{f_i, f_j\})$, is high.

From the above discussion, the terms $I(\mathcal{C}; f_i)$, $I(\mathcal{C}; \{f_i, f_j\})$, $I(f_i; f_j)$, and $H(f_i)$ will be considered in order to determine an expression for the evaluation function $g(.)$. The proposed algorithm for determining $g(.)$ can be summarized in four steps, as described below:

Step 1: choose the feature $f_j \in \mathcal{L}$, that produces maximum value of $I(\mathcal{C}; f_j)$; set $\mathcal{K} \leftarrow \{f_j\}$; $g(\mathcal{K}) = I(\mathcal{C}; f_j)$

Step 2: For each feature $f_i \in \mathcal{L}, f_i \notin \mathcal{K}$

- compute: $m(f_i) = g(\mathcal{K}) + \lambda I(\mathcal{C}; f_i)$, where λ represents the information gain
- choose the feature f_i that maximizes m
- set $\mathcal{K} \leftarrow \mathcal{K} \cup \{f_i\}$; $g(\mathcal{K}) = m(f_i)$

Step 3: if $|\mathcal{K}| < |\mathcal{L}|$ go to step 2

Step 4: $g(\mathcal{L}) = g(\mathcal{K})$

[1] The reader may refer to appendix A for proofs that MI between $\mathcal{C}$ and the two subsets $\mathcal{K}$ and $\mathcal{L}$, satisfies these conditions

In the first step, $g(.)$ is initialized to the maximum MI between a single feature and the class labels, $I(\mathcal{C}; f_j)$, and the subset of the chosen features, $\mathcal{K}$, is initialized to $\{f_j\}$. Step 2 defines the intermediate function m of feature f_i, which is the summation of the latest value of $g(.)$ and its MI with class labels multiplied by λ. This is an approximation of the amount of information that can be added to $g(.)$ when choosing f_i. The value of λ ranges between $[0, 1]$. If no information is gained from f_i, $\lambda = 0$. On the other hand, $\lambda \to 1$ if f_i is independent from all the features chosen so far. If f_i is partially dependent upon any of the features in $\mathcal{K}$, then λ will vary between 0 and 1. Using a concept drawn by analogy with ANNs, we found that a suitable expression for λ can be formulated as follows:

$$\lambda = \frac{2}{1 + \exp(-\alpha D)} - 1 \tag{4}$$

where

$$D = \min_{f_j \in \mathcal{K}} \left[\frac{H(f_i) - I(f_i; f_j)}{H(f_i)} \right] \times \frac{1}{|\mathcal{K}|} \sum_{f_j \in \mathcal{K}} \exp \left[\beta \left(\frac{I(\mathcal{C}; \{f_i, f_j\})}{I(\mathcal{C}; f_i) + I(\mathcal{C}; f_j)} \right)^{\gamma} \right] \tag{5}$$

the parameters α, β and γ are constants, and $|\mathcal{K}|$ is the cardinal of $\mathcal{K}$.

The first term of D reflects the degree of dependency between f_i and the already chosen features in $\mathcal{K}$, f_j. If f_i is highly dependent upon any of the chosen features, then $D \to 0$, and hence $\lambda \to 0$. To achieve this, we subtract the value of MI between f_i and f_j from the entropy of f_i. If the outcome approaches 0, then f_i and f_j are highly dependent.

When f_i is independent from all the features chosen so far, then D should be large enough to make $\lambda \to 1$. Note that the first term of D will be 1 because $I(f_i; f_j) = 0$. On the other hand, the second term will be $\exp(\beta)$, because $I(\mathcal{C}; \{f_i, f_j\}) = I(\mathcal{C}; f_i) + I(\mathcal{C}; f_j)$ (see appendix A.2). After intensive experiments, we found that $\beta = 2.5$ and $\alpha = 0.3$ are appropriate for many classification tasks.

For the case when f_i is partially dependent upon all the features considered so far, which is the common case, it is very important that we take into consideration how f_i and f_j interact together in order to calculate D properly. The second expression of D evaluates the ratio of how f_i and f_j work together to that when they are considered individually. A reasonable choice for γ was found to be 3.

After computing m for all non-chosen features in the subset $\mathcal{L}$, we add the feature that maximizes m to $\mathcal{K}$, and substitute its value into $g(.)$ (step 2). This procedure is repeated until we consider all the features in the subset $\mathcal{L}$. When tested on the simulated data of section 2.2, its performance was found to be comparable to that of $I(\mathcal{C}; \mathcal{L})$, as shown in Fig. 3, where the relationship between the proposed subset measure and classification accuracy tends to be monotonic. The figure also shows that both the number and amount of fluctuations are less than those of MIFS. The proposed evaluation function will be called mutual information evaluation function (MIEF), and in this paper it will be used with the stepwise search procedure.

Insert Fig. 3 here

Given the MI and entropy arguments of Eq. 5, the MIEF can be implemented quite fast using an ordinary PC. In other words, since both MIFS and MIEF require the computation of MI between pairs of features, the computational time required by MIEF, *i.e.*, the implementation of Eqs. 4 and 5, is very much similar to that of the MIFS, which requires the execution of Eq. 3.

4 Experimental Results

4.1 Texture Classification

A first set of experiments were carried in texture classification. The textures considered here were: bark, brick, bubbles, leather, raffia, water, weave, wood and wool [15]. In the first experiment, the classification of the first two textures was carried out. The classification of the first five, then the classification of all the nine textures was considered in the second and third experiments respectively. Gaussian noise, with different signal-to-noise ratio, has been added to (1024 × 1024 pixels) images of each texture class to form the training and testing sets. 961 patterns were obtained from each image using (64 × 64) windows with an overlap of 32 pixels. Figs. 4 and 5 show the clean and noisy texture images used.

Insert Figs. 4 and 5 here

Four 9 dimensional feature vectors were calculated using statistics of sum and difference histogram (SDH) of the co-occurrence matrix with different directions: vertical, horizontal, and the two diagonals (SDH_1, SDH_2, SDH_3 and SDH_4). For each direction, the features used were: mean, variance, energy, correlation, entropy, contrast, homogeneity, cluster shade, and cluster prominence. The fractal dimension (FD) has also been used to form the tenth feature of each vectors. The energy contents of texture images (E) has been used to form another feature vector using 9 different masks, and its tenth feature was FD.

Each one of these five feature vectors was used as input to an ANN. The numbers of training and testing patterns depend upon the number of classes considered, *i.e.*, for the case of two classes, 15376 patterns were used to train the networks and 5766 to test them. The classification accuracies obtained are shown in Table 1. Note that as the number of classes increases the overall accuracy decreases. It is worth mentioning that the first four feature vectors were found to exhibit a high degree of correlation. Fig. 6 shows the classification accuracy of the selected and transformed features obtained by applying the MIFS, MIEF and PCA methods for the three experiments.

Insert Table 1 here

It is clear that the PCA has the lowest classification accuracy, compared to MIFS and MIEF, for the three experiments. This is expected because it does not take into consideration relationships between input features and class labels. Therefore, for the rest of the paper, we will only analyze the performance of MIFS and MIEF. As shown in Table 1, the E vector performed extremely well compared to the other four vectors. For example, in the first experiment, the ratio between the error rate of the second best vector and that of E is 5.64. A good feature selection method must achieve a similar performance to that of the E vector with less number of features. This is the case for MIEF, where it achieves similar classification accuracy using 9, 7 and 6 features for the first, second and third experiments respectively, as shown in Fig. 6. A further improvement in performance is achieved when the number of features increases. On the other hand, the MIFS is not able to achieve the same performance of the E vector for the first experiment even when considering up to 16 features. For the second and third experiments, it reaches similar results when considering 10 and 7 features respectively. Therefore, the MIFS does not achieve the goal of feature selection in the first two cases, but it is successful in the last case. In addition, Fig. 6 shows that the MIEF is better than the MIFS by at least 2% when selecting more than 5 features for the three experiments. The difference between their performance is especially clear when the number of selected features ranges between 6 and 9 for the second experiment, as it exceeds 6%. It is quite evident that the MIEF outperforms the MIFS in all cases. The standard deviation of the classification accuracy of the two methods is measured and found to be less than 1 for all cases. As such, the difference in classification accuracy between the two methods suggests that the MIEF can achieve better results, for almost all the cases, compared to the MIFS.

The performance of the whole feature set, which contains 50 feature, is indicated by the horizontal dotted lines in the figure. We find that both MIEF and MIFS could achieve more than 96% of the performance of the whole feature set, with better results achieved by the MIEF.

The above three experiments clearly show the superiority of the MIEF, since it achieves a similar or better performance to the best original feature vector with a lower number of features, and outperforms the MIFS in all considered cases.

Insert Fig. 6 here

4.2 Classification of Speech Segments

The purpose of this experiment is to classify speech segments according to their manner of articulation. The classification results can later be incorporated in speech therapy, learning second language or speech recognition systems. Six classes were considered: vowel, nasal, fricative, stop, glide, and silence. Speech signals have been divided into segments (e.g. S_{n-1}, S_n, S_{n+1}, etc.) and frames (e.g. f_1, f_2, f_3, etc.) as shown in Fig. 7. From each frame, three different set of features were extracted: 16 log mel-filter bank (MFB), 12 linear predictive reflection coefficients (LPR), and 10 wavelet energy bands (WVT). A context dependent approach was adopted to perform the classification. So, for each speech segment (e.g. S_n), the baseline features were the average frame features over the first and second half of segment S_n and the average frame features of the previous and following segments (S_{n-1} and S_{n+1} respectively). Thus, based on MFB, the baseline features of S_n were 64, since there were 32 features extracted from the average frame features of the first and second halves of S_n, and 16 features extracted from each of the two segments S_{n-1} and S_{n+1}. The baseline features based on LPR and WVT were calculated in a similar way with total number of 48 and 40 respectively. Each of these three baseline features was used as input to an ANN, hence, the number of input units were 64, 48 and 40 for MFB, LPR and WVT respectively. For this experiment, speech was obtained from the TIMIT database [11]. Segments from 152 speakers (56456 segments) were used to train the ANNs, and from 52 speakers (19228 segments) to test them. The classification accuracy for MFB, LPR and WVT were 85.50%, 74.06% and 84.33% respectively. It is clear that the MFB is superior to both LPR and WVT, but it used more features. The performance of LPR, on the other hand, was not good as it used more features than WVT.

Insert Fig. 7 here

It is worth mentioning that because of the large number of input units used by the three ANNs compared to that of the texture experiments, more weights were needed to be adjusted in the training face, and hence more computational time. Therefore, it will be very useful to select the most important features such that similar or better performance is achieved with reduced computational time. The three baseline features are concatenated to form 152 features for each segment. Both MIFS and MIEF are then used to select from these features. The classification accuracy of the selected MIFS and MIEF features are shown in Fig. 8. The figure shows that both MIFS and MIEF meet the objective of feature selection by achieving similar performance but with a reduced number of features compared to the MFB, LPR and WVT baseline features. The MIEF and the MIFS achieve similar performance to the MFB baseline features using 45 and 50 features respectively. When comparing the performance of MIFS to that of WVT, no improvement is achieved when selecting up to 45 features. In contrast, the MIEF achieves better performance when selecting 30 and more features. The figure also shows that both MIFS and MIEF are able to achieve more than 97% of the performance of the whole feature set using 75 features, which is less than half the number of original features. In summary, the results show that the MIEF outperforms the MIFS most cases, especially for the first 50 features, where an improvement of up to 3% is achieved. It also outperforms the original feature vectors using same number of features.

Insert Fig. 8 here

4.3 Speaker Identification

In this experiment, speech signals of four different speakers (two females and two males) are used to identify the speakers. Similar to the speech segment experiment, the speech data were obtained from the TIMIT database [11], and the three set of features used to represent each frame were MFB, LPR and WVT, which use 16, 12 and 11 features respectively. The baseline features of a given pattern consisted of the features of three successive frames, which are 48, 36, and 30 features for MFB, LPR and WVT respectively. 5931 patterns were used for training and 2542 for testing. Each set of baseline features was used to train an ANN to identify between the four speakers. The classification accuracy for MFB, LPR, and WVT were: 85.1%, 74.8% and 71.6% respectively. It is worth mentioning that unlike the speech segment experiment, the set of LPR features outperform that of the WVT features. General purpose dimensionality reduction techniques that only look for specific properties in the features without considering their relationship with class labels, like the principal component analysis, will certainly fail to provide optimal feature subsets.

Insert Fig. 9 here

Both MIFS and MIEF are used to select features from the concatenated three baseline sets (a total of 114 features), and the results obtained are shown in Fig. 9. The MIFS meets the objective of feature selection by achieving similar performance, compared to WVT and LPR, by using 12 and 16 features. However, it fails with the MFB, where it needs at least 52 features to achieve similar performance. On the other hand, the MIEF achieves similar performance to WVT, LPR and MFB using 6, 8, and 18 features only, and its 48 selected features achieve an accuracy of 90%. In addition, the performance of MIEF is found to be superior to that of MIFS in all cases, where an improvement of up to 10% is achieved. The MIFS needs to use 60 features to achieve the performance of 27 features selected by the MIEF. Moreover, when selecting 57 features, which represent half the number of all features, the MIEF achieved 99% of the performance of the whole feature set compared to 95% for the MIFS.

Even thought for all our experiments we used ANNs for classification, the findings of this paper can be extended to other classifiers, such as Bayes classifiers, since the algorithm focuses on finding the best set of features independently of the classifier used.

The above experiments show the strength of our proposed measure, and suggests that even if the domain expert, who can choose the most relevant features, is not involved, we can still achieve very good results. This is implied by the very close performance of the selected features to that of the whole feature set.

5 Conclusion

A new evaluation function based on the concept of mutual information has been presented. The function takes into consideration the interaction between features and measures the ability of feature subsets in distinguishing between class labels. When the function was used with the stepwise selection procedure, it improves classification accuracy with a lesser number of features compared to the conventional MIFS method and the best original feature vector for all three problems of texture classification, speech segment classification and speaker identification. The major advantage of the proposed measure is that it takes into account different features interaction without noticeable increase in the computational complexity compared to the MIFS.

A Appendix: Evaluation Function Conditions for $I(\mathcal{C};\mathcal{L})$

A.1 Lower Bound

If $\mathcal{K} \subset \mathcal{L}$ or $\mathcal{K}$ is fully dependent upon $\mathcal{L}$, then $P(\mathcal{K},\mathcal{L}) = P(\mathcal{L})$, and $P(\mathcal{C},\mathcal{K},\mathcal{L}) = P(\mathcal{C},\mathcal{L})$.

$$\begin{aligned} I(\mathcal{C};\{\mathcal{K},\mathcal{L}\}) &= \sum_{c,k,l} P(\mathcal{C},\mathcal{K},\mathcal{L}) \log \left[\frac{P(\mathcal{C},\mathcal{K},\mathcal{L})}{P(\mathcal{C})P(\mathcal{K},\mathcal{L})}\right] \\ &= \sum_{c,l} P(\mathcal{C},\mathcal{L}) \log \left[\frac{P(\mathcal{C},\mathcal{L})}{P(\mathcal{C})P(\mathcal{L})}\right] \\ \therefore \quad I(\mathcal{C};\{\mathcal{K},\mathcal{L}\}) &= I(\mathcal{C};\mathcal{L}) \end{aligned} \tag{6}$$

A.2 Upper Bound

If $\mathcal{K}$ and $\mathcal{L}$ are independent, then $P(\mathcal{K},\mathcal{L}) = P(\mathcal{K})P(\mathcal{L})$, and:

$$\begin{aligned}
& P((\mathcal{K},\mathcal{L})|\mathcal{C}) = P(\mathcal{K}|\mathcal{C})P(\mathcal{L}|\mathcal{C}) \\
\text{on the other hand} \quad & P((\mathcal{K},\mathcal{L})|\mathcal{C}) = P(\mathcal{C},\mathcal{K},\mathcal{L})/P(\mathcal{C}) \\
\therefore \quad & P(\mathcal{C},\mathcal{K},\mathcal{L}) = P(\mathcal{K}|\mathcal{C})P(\mathcal{L}|\mathcal{C})P(\mathcal{C}) \\
& = P(\mathcal{C},\mathcal{K})P(\mathcal{C},\mathcal{L})/P(\mathcal{C})
\end{aligned}$$

Using these formulas in calculating the MI will lead to:

$$\begin{aligned}
& I(\mathcal{C};\{\mathcal{K},\mathcal{L}\}) = \sum_{c,k,l} P(\mathcal{C},\mathcal{K},\mathcal{L}) \log\left[\frac{P(\mathcal{C},\mathcal{K},\mathcal{L})}{P(\mathcal{C})P(\mathcal{K},\mathcal{L})}\right] \\
& = \sum_{c,k,l} \frac{P(\mathcal{C},\mathcal{K})P(\mathcal{C},\mathcal{L})}{P(\mathcal{C})} \log\left[\frac{P(\mathcal{C},\mathcal{K})P(\mathcal{C},\mathcal{L})}{P(\mathcal{C})^2 P(\mathcal{K})P(\mathcal{L})}\right] \\
& = \sum_{c,k,l} \frac{P(\mathcal{C},\mathcal{K})P(\mathcal{C},\mathcal{L})}{P(\mathcal{C})} \left[\log\left[\frac{P(\mathcal{C},\mathcal{K})}{P(\mathcal{C})P(\mathcal{K})}\right] + \log\left[\frac{P(\mathcal{C},\mathcal{L})}{P(\mathcal{C})P(\mathcal{L})}\right]\right] \\
& = \sum_{c,k} P(\mathcal{C},\mathcal{K}) \log\left[\frac{P(\mathcal{C},\mathcal{K})}{P(\mathcal{C})P(\mathcal{K})}\right] + \sum_{c,l} P(\mathcal{C},\mathcal{L}) \log\left[\frac{P(\mathcal{C},\mathcal{L})}{P(\mathcal{C})P(\mathcal{L})}\right] \\
& \therefore \quad I(\mathcal{C};\{\mathcal{K},\mathcal{L}\}) = I(\mathcal{C};\mathcal{K}) + I(\mathcal{C};\mathcal{L}) \qquad (7)
\end{aligned}$$

A.3 Monotonicity

If $\mathcal{K} \subset \mathcal{L}$, then let $\mathcal{J}$ be a subset such that $\mathcal{K} \cup \mathcal{J} = \mathcal{L}$. According to Eqs. 6 and 7, the lower and upper bounds of $I(\mathcal{C};\mathcal{L})$ are $I(\mathcal{C};\mathcal{K})$ and $I(\mathcal{C};\mathcal{K}) + I(\mathcal{C};\mathcal{J})$ respectively. Therefore,

$$I(\mathcal{C};\mathcal{K}) \leq I(\mathcal{C};\mathcal{L}) \qquad (8)$$

References

[1] H. Almuallim and T.G. Dietterich. Learning boolean concepts in the presence of many irrelevant features. Artificial Intelligence, 69 (1994), 279–305.

[2] R. Battiti. Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on Neural Networks, 5 (1994), 537–550.

[3] G.A. Darbellay and I. Vajda. Estimation of the mutual information by an adaptive partitioning of the observation space. IEEE Transactions on Information Theory, 45 (1999), 1315–1321.

[4] M. Dash and H. Liu. Feature selection for classification. Intelligent Data Analysis, 1 (1997), 1–27.

[5] P.A. Devijver and J. Kittler. Pattern recognition: A statistical approach. Prentice–Hall, 1982.

[6] R.H. Duda and P.H. Hart. Pattern classification and scene analysis. Wiley, 1973.

[7] R. Fano. Transmission of information: A statistical theory of communications. Wiley, 1961.

[8] A.M. Fraser and H.L. Swinney. Independent coordinates for strange attractors from mutual information. Physical Review A, 33 (1986), 1134–1140.

[9] M.A. Hall. Correlation-based feature selection for machine learning. PhD thesis, The University of Waikato, 1999.

[10] A. Jain and D. Zongker. Feature selection: evaluation, application, and small sample performance. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19 (1997), 153–158.

[11] MIT, SRI, and TI. DARPA TIMIT acoustic-phonetic continuous speech corpus, 1990. http://www.ldc.upenn.edu/doc/TIMIT.html.

[12] A.N. Mucciardi and E.E. Gose. A comparison of seven techniques for choosing subsets of pattern recognition properties. IEEE Transactions on Computers, C–20 (1971), 1023–1031.

[13] P.M. Narendra and K. Fukunaga. A branh and bound algorithm for feature subset selection. IEEE Transactions on Computers, C–26 (1977), 917–922.

[14] B.D. Ripley. Pattern recognition and neural networks. Cambridge university press, 1996.

[15] Signal and Image Processing Institute, USC. USC-SIPI image database, 1981. http://sipi.usc.edu/services/database/.

[16] H. Vafie and I.F. Imam. Feature selection methods: genetic algorithms vs. greedy-like search. In Proc. 9th Intl. Conf. Fuzzy and Intelligent Control Systems, 1994.

Table 1: Texture classification accuracy of the five original classifiers for different number of output classes

No. of classes	SDH_1	SDH_2	SDH_3	SDH_4	E
2	88.45	89.65	89.85	88.81	98.20
5	79.64	79.79	79.06	78.96	90.49
9	75.07	71.89	71.72	71.79	89.01

List of figures

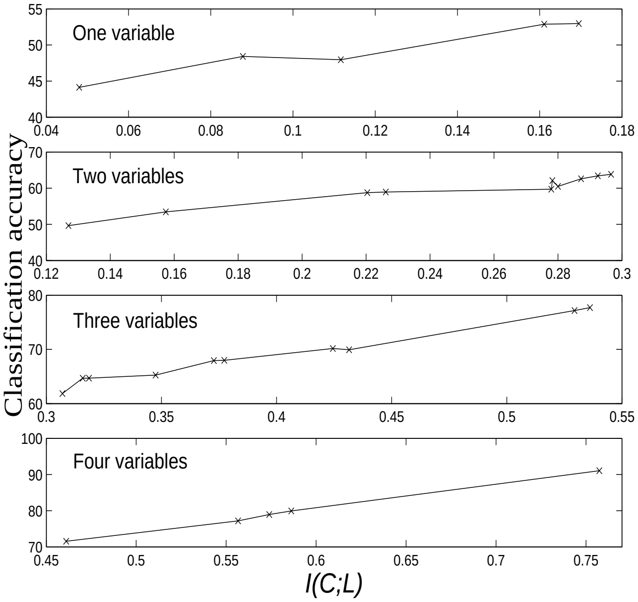
One variable
Two variables
Three variables
Four variables
Classification accuracy
I(C;L)
55
50
45
40
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
70
60
50
40
0.12
0.14
0.16
0.18
0.2
0.22
0.24
0.26
0.28
0.3
80
70
60
0.3
0.35
0.4
0.45
0.5
0.55
100
90
80
70
0.45
0.5
0.55
0.6
0.65
0.7
0.75

Figure 1:

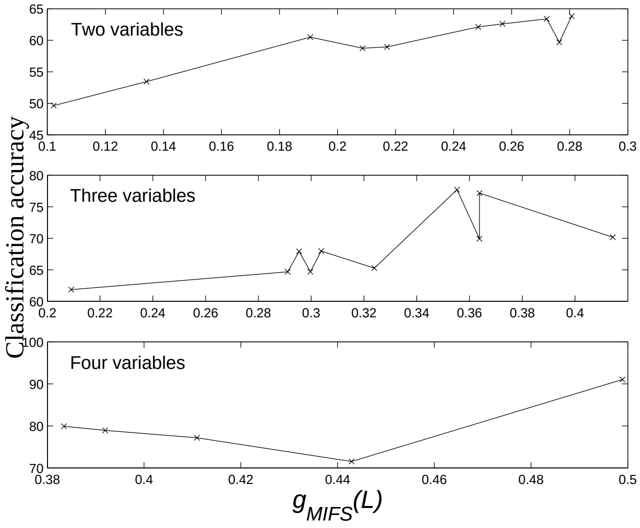
Two variables
Three variables
Four variables
Classification accuracy
65
60
55
50
45
0.1
0.12
0.14
0.16
0.18
0.2
0.22
0.24
0.26
0.28
0.3
80
75
70
65
60
0.2
0.22
0.24
0.26
0.28
0.3
0.32
0.34
0.36
0.38
0.4
100
90
80
70
0.38
0.4
0.42
0.44
0.46
0.48
0.5
$g_{MIFS}(L)$

Figure 2:

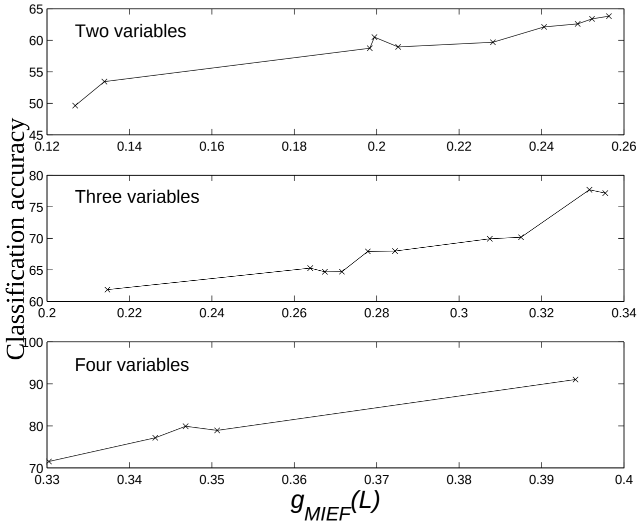
Classification accuracy
Two variables
65
60
55
50
45
0.12
0.14
0.16
0.18
0.2
0.22
0.24
0.26
Three variables
80
75
70
65
60
0.2
0.22
0.24
0.26
0.28
0.3
0.32
0.34
Four variables
100
90
80
70
0.33
0.34
0.35
0.36
0.37
0.38
0.39
0.4
$g_{MIEF}(L)$

Figure 3:

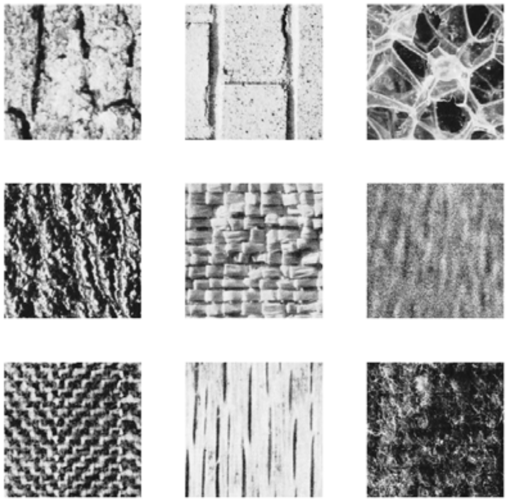

Figure 4:

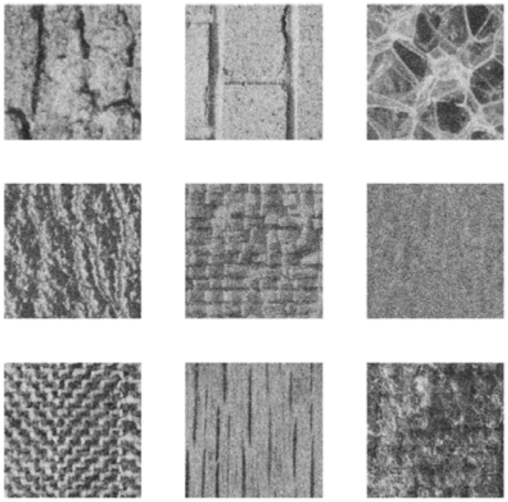

Figure 5:

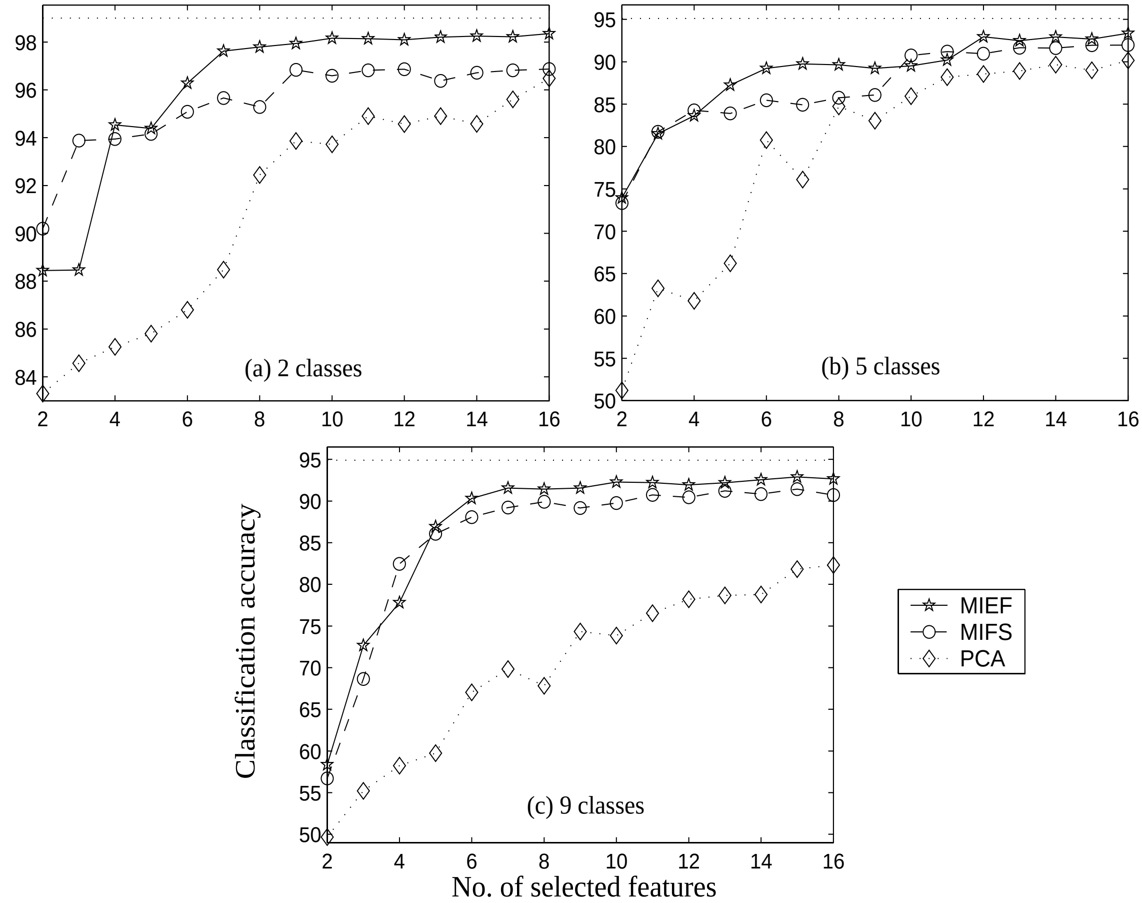

Figure 6:

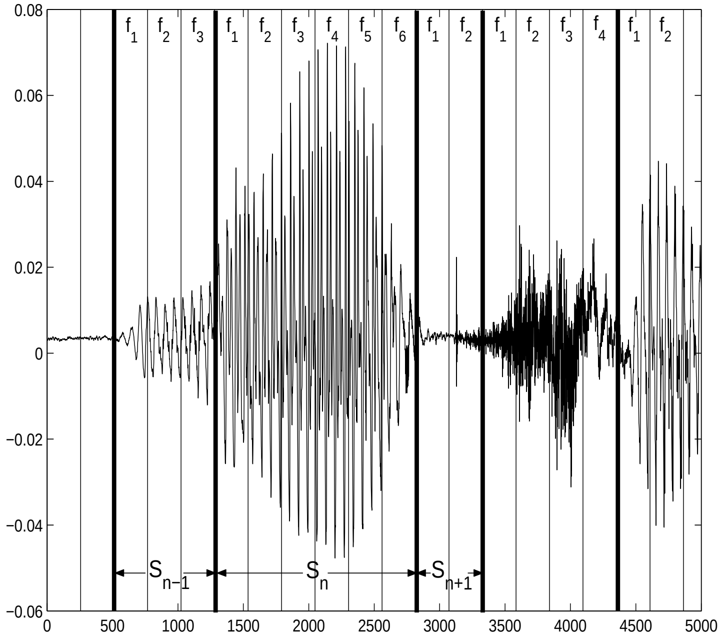

Figure 7:

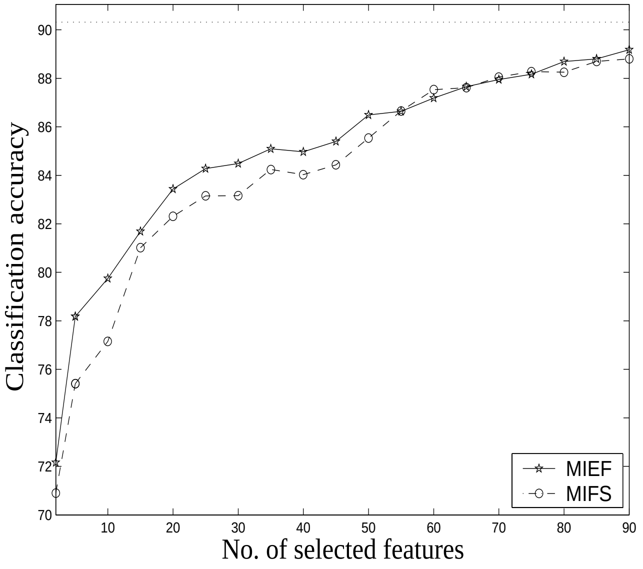

Figure 8:

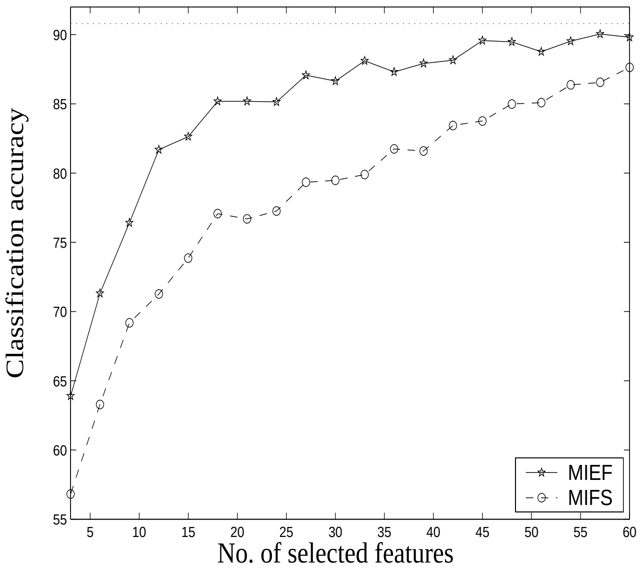
Classification accuracy
No. of selected features
MIEF
MIFS
55
60
65
70
75
80
85
90
5
10
15
20
25
30
35
40
45
50
55
60

Figure 9: