Neural Comput & Applic (2004) 13: 175–184
DOI 10.1007/s00521-004-0400-9

ORIGINAL ARTICLE

Shuang Cang · Derek Partridge

# Feature ranking and best feature subset using mutual information

Received: 22 May 2002 / Accepted: 10 February 2004 / Published online: 1 April 2004

**Abstract** A new algorithm for ranking the input features and obtaining the best feature subset is developed and illustrated in this paper. The asymptotic formula for mutual information and the expectation maximisation (EM) algorithm are used to developing the feature selection algorithm in this paper. We not only consider the dependence between the features and the class, but also measure the dependence among the features. Even for noisy data, this algorithm still works well. An empirical study is carried out in order to compare the proposed algorithm with the current existing algorithms. The proposed algorithm is illustrated by application to a variety of problems.

**Keywords** EM algorithm · Feature ranking · Feature selection · Feature space · Mixture model · Mutual information

## 1 Introduction

A central problem in pattern recognition is to identify a best feature subset from an initial feature space (feature set) containing many possible features. If we use all features of the initial feature space as the input, the classification performance may suffer due to irrelevant or redundant input information. It may also cause excessive complexity. In real applications, it is not economic to collect input features that are irrelevant or redundant respecting to the classification task. Determining the best feature subset for the inputs is the critical stage in the classification task. The best feature subset is the one that contains as few as possible features and yet is highly predictive of the class.

Two well-established sub-optimal techniques for feature selection are sequential forward selection (SFS) and sequential backward selection (SBS) methods. Both methods accept features or reject them, one at a time, in order to construct an optimal subset (see, for example, [12]). The proposed method may be classed as an SBS method that successively rejects features on the basis of minimum contribution to mutual information.

A number of different algorithms [1, 2, 3, 4, 5] have been proposed for the feature ranking and selecting optimal feature subset. The clamping method in [1] uses trained multi-layer perceptrons (MLPs) to rank the relative significances of input features for predicting network output. The advantage of the method is that it delivers a reliable result efficiently even on noisy data. The disadvantage is that it lacks a theoretical analysis, and does not expose any detail of feature interaction. The technique developed in [2] uses repeated partial retraining each time a new subset of input features is assessed. Both papers survey a variety of alternative feature-ranking techniques. The method studied in [3, 4] uses mutual information theory; the advantage of the algorithm is simplicity, but it only considers the mutual information between the individual feature and the class, and the mutual information between each pair of individual features. It is effectively limited to consider the cases with just one or two dimensions for the density functions. A further problem with this algorithm is that it is difficult to select the tuning parameters. It may lead to unreliable results.

For a feature space with $n$ features, the algorithm proposed in this paper is based on the mutual information between the feature space containing $n-1$ features after deleting a feature in turn and the class. The idea is that if a feature for this feature space is truly the least important, then the feature space excluding this feature is the more accurately predictive of the class than

S. Cang (✉)
Department of Computer Science,
University of Wales, Aberystwyth,
SY23 3DB, UK
E-mail: sdc@aber.ac.uk

D. Partridge
Department of Computer Science,
University of Exeter,
Exeter, EX4 4QF, UK

any other feature spaces by deleting other individual feature.

## 2 Formal basics

### 2.1 Mutual information

Mutual information is a measure of the dependence between random variables. It is always symmetric and non-negative. It is zero if and only if the variables are statistically independent.

The mutual information between two discrete random variables $U=(u_1,u_2,...,u_d)$, $V=(v_1,v_2,...,v_d)$ is defined as [11]

$$I(U,V)=\sum_u\sum_v p(u,v)\log\frac{p(u,v)}{p(u)p(v)} \tag{1}$$

where $p(u,v)$ is a joint density function, and $p(u)$ and $p(v)$ are the marginal density functions, respectively. The mutual information between two continuous random variables $X=(x_1,x_2,...,x_d)$, $Y=(y_1,y_2,...,y_d)$ is defined as [11]

$$I(X,Y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}dxdy \tag{2}$$

The mutual information between one continuous $X=(x_1,x_2,...,x_d)$ and discrete $U=(u_1,u_2,...,u_d)$ random variables is defined as

$$I(X,U)=\sum_u\int_{-\infty}^{\infty} p(x,u)\log\frac{p(x,u)}{p(x)p(u)}dx \tag{3}$$

The mutual information between class $C$ and discrete features $U=(u_1,u_2,...,u_d)$ can be calculated from Eq. 1.

From Eq. 3, we can measure the mutual information between class $C$ and continuous feature space $X=(x_1,x_2,...,x_d)$

$$\begin{aligned} I(X,C)&=\sum_c p(c)\int_{-\infty}^{\infty} p(X|c)\log\frac{p(X|c)}{p(X)}dX \\ &=\sum_c p(c)\int_{-\infty}^{\infty} p(X|c)\log p(X|c)dX \\ &\quad-\int_{-\infty}^{\infty} p(X)\log p(X)dX \end{aligned} \tag{4}$$

### 2.2 The mixture model

The expectation maximisation (EM) algorithm is widely used to construct Gaussian mixture models [10]. For $K$ components, where $K$ can be determined by using algorithms in [8, 9], the mixture distribution can be written as a linear combination of component density functions $p(X|j)$ in the form

$$p(X)=\sum_{j=1}^{K} p(X|j)P(j) \tag{5}$$

where $P(j)$ are the parameters in the mixture model and satisfy the following conditions

$$\sum_{j=1}^{K} P(j)=1,\quad 0\leqslant P(j)\leqslant 1 \tag{6}$$

The component density functions $p(X|j)$ satisfy

$$\int_{-\infty}^{\infty} p(X|j)dX=1 \tag{7}$$

The most widely used distribution function for each component density is the Gaussian distribution function. The form of the Gaussian distribution function for each component is

$$p(X|j)=\frac{1}{(2\pi)^{d/2}\left|\Sigma_j\right|^{1/2}}e^{-\frac{1}{2}\left(X-\mu_j\right)^T\Sigma_j^{-1}\left(X-\mu_j\right)} \tag{8}$$

where the parameters $\mu_j$ and $\Sigma_j$ are the means of a $d$-dimensional vector and a d×d covariance matrix, respectively. Values for the parameters P(j), $\mu_j$ and $\Sigma_j$ are determined in each component using the EM algorithm [10] as follows. Firstly, a K-means clustering method is used for a fixed number of the components $K$ in Eq. 5 to determine parameters $P(j)$, $\mu_j$ and $\Sigma_j$ for each component $p(X|j)$ in Eq. 8. Clearly the condition in Eq. 7 must be satisfied. Then, using the following recursive formulas, the parameters $P(j)$, $\mu_j$ and $\Sigma_j$ are obtained for each component.

$$\begin{aligned} &P(j)=\frac{1}{N}\sum\nolimits_{n=1}^{N} w_j^n,\quad \mu_j=\frac{\sum_{n=1}^{N} w_j^n X^n}{\sum_{n=1}^{N} w_j^n}, \\ &\sum\nolimits_j=\frac{\sum_{n=1}^{N} w_j^n\left(X^n-\mu_j\right)\left(X^n-\mu\right)'}{\sum_{n=1}^{N} w_j^n} \end{aligned} \tag{9}$$

where $N$ is the size of the data set and the weight is

$$w_j^n=\frac{p(X^n|j)P(j)}{\sum_{j=1}^{M} p(X^n|j)P(j)} \tag{10}$$

where $p(X^n|j)$ is defined in Eq. 8.

## 3 The clamping technique

The clamping technique has been described in [1]. The idea is that if a feature is truly redundant with respect to the classification, then clamping the corresponding network's input to a fixed value will have no adverse effect on the generalisation performance of a trained neural network. The importance or salience of the input feature $x_i$ on the output is defined as

$$\xi(x_i) = 1 - \frac{g(X_{x_i=\bar{x}_i})}{g(X)} \tag{11}$$

where $g(x)$ is the generalisation performance of the neural net when all $x_i$ have their natural values, and $g(X_{x_i=\bar{x}_i})$ is the clamped generalisation performance of the net when input $x_i$ is clamped to its mean $\bar{x}_i$.

## 4 An asymptotic formula

If we want to calculate the mutual information between class $C$ and continuous feature space $X=(x_1,x_2,\ldots,x_d)$ defined in Eq. 4, the main difficulty is to calculate the following term

$$\Phi = -\int_{-\infty}^{\infty} f(X)\log f(X)dX \tag{12}$$

$\Phi$ is the entropy of $f(X)$ and plays an important role in the mutual information. We can't obtain the exact analytic solution of Eq. 12 if $f(X)$ has more than one component in a mixture model (Eq. 5), but we can use the asymptotic formula to approximate Eq. 12.

For sufficient data, we proposed that $\Phi$ in Eq. 12 is approximated by using the following form

$$\Phi \approx -\frac{1}{N}\sum_{n=1}^{N}\log f(X_n) \tag{13}$$

where $N$ is the size of the data $\{X_n\}$.

Next, we will demonstrate the accuracy of the asymptotic formula (Eq. 13) compared with the true value (Eq. 12) for one component in a mixture model (Eq. 5).

We suppose that the true probability density function is a standard Gaussian normal distribution with dimension $d$.

$$f(X) = \frac{1}{(2\pi)^{d/2}}e^{-\frac{1}{2}X^T X} \tag{14}$$

The true entropy of $f(X)$ is

$$\begin{aligned}\Phi &= -\int_{-\infty}^{\infty} f(X)\log f(X)dX \\ &= \frac{d}{2}\log(2\pi) + \pi^{-d/2}\Gamma\left(\frac{d}{2}+1\right)\lambda\end{aligned} \tag{15}$$

where

$$\lambda = \begin{cases} 1, & d=1 \\ \pi, & d=2\,, \\ B\left(\frac{d-1}{2},\frac{1}{2}\right)B\left(\frac{d-2}{2},\frac{1}{2}\right)\cdots B\left(\frac{2}{2},\frac{1}{2}\right)\pi, & d>2 \end{cases}$$

$\Gamma$ is a Gamma function and $B$ is a Beta function.

In our experimental study, we randomly draw 1000 samples from the distribution of Eq. 14 as the data set $\{X_n\}$, each input generated as a standard normal distribution at random. The density function can be obtained using the EM algorithm with one component. Next, we present the true and the approximate values $\Phi$ using Eq. 13 for different dimensions in Table 1. The experiment is repeated 5 times with different seeds to generate the random samples.

From Table 1, we can see that asymptotic formula (Eq. 13) produces accurate approximations of the true value (Eq. 12).

## 5 The ranking algorithms

In this section, we will present the proposed algorithm for feature selection. The algorithm will not only generates the ranking feature, but also gives an optimal best feature subset.

### 5.1 A: feature ranking algorithms

Suppose that a feature space $F^{(n)}$ contains $n$ features ($n\geq 2$), and with initially set $m=n$. The feature ranking algorithm is operated by the following steps:

**Step 1**. Computing mutual information between the subset feature space and the class:

**Table 1** Values for $\Phi$ using the approximation formula for different dimensions

| | | | | | |
|---|---|---|---|---|---|
| d = 1; True value: Φ = 1.419; Mean: 1.415; STD: 0.013 | | | | | |
| Approx: | 1.409 | 1.420 | 1.423 | 1.429 | 1.395 |
| d = 2; True value: Φ = 2.838; Mean: 2.824; STD: 0.015 | | | | | |
| Approx: | 2.798 | 2.836 | 2.833 | 2.823 | 2.829 |
| d = 3; True value: Φ = 4.257; Mean: 4.267; STD: 0.031 | | | | | |
| Approx: | 4.287 | 4.273 | 4.220 | 4.253 | 4.300 |
| d = 10;True value: Φ = 14.189; Mean: 14.173; STD: 0.038 | | | | | |
| Approx: | 14.172 | 14.206 | 14.159 | 14.210 | 14.118 |
| d = 20; True value: Φ = 28.379; Mean: 28.309; STD: 0.070 | | | | | |
| Approx: | 28.206 | 28.337 | 28.273 | 28.345 | 28.385 |
| d = 40; True value: Φ = 56.758; Mean: 56.446; STD: 0.155 | | | | | |
| Approx: | 56.268 | 56.366 | 56.441 | 56.687 | 56.469 |

Delete each feature $f_i$ ($i=1,2,...,m$) from the feature space $F^{(m)}$ to get $F_{-f_i}^{(m)}$,]] >where $F_{-f_i}^{(m)}$ indicates the feature space $F^{(m)}$ with the feature $f_i$ removed, and calculate the mutual information between the feature space $F_{-f_i}^{(m)}$ and the class using Eq. 4 and Eq. 13. Carry on this step for each feature in turn in the feature space $F^{(m)}$.

**Step 2**. Finding the least important feature in the feature space $F^{(m)}$:

The feature $f_j = \max\left\{I\left(F_{-f_j}^{(m)}, C\right)\right\}$ is the least important and the ranking of the feature $f_j$ is $m$.

**Step 3**. Deleting feature $f_j$ from the feature space $F^{(m)}$:

Delete the feature $f_j$ (which is found from **Step 2**) from the feature space $F^{(m)}$ and set $m=m-1$.

**Step 4**. If $m>1$ go to **Step 1**, otherwise stop and the last feature is ranked 1.

### 5.2 B: optimal best feature subset

After the feature ranking for all features, choose the feature space

$$F^{(m)} = \max_m I\left(F^{(m)}, C\right)$$

which reaches the maximum mutual information with respect to the class. This is the optimal best feature subset which will be used in the classification task as inputs.

## 6 The experiment studies results

To demonstrate the feature selection algorithm proposed in the previous section, a number of experimental tests are presented in this section. The first two experimental tests are well defined. The third example re-examines previously published results. The last one is an application to a practical open problem.

### 6.1 Example A: LIC1 with an irrelevant feature

The LIC1 problem which has been studied in [1], and is defined as

$$LIC1 = \begin{cases} 1 & \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} > length \\ 0 & \qquad otherwise \end{cases} \tag{16}$$

The input feature space $F^{(5)}=[x_1, x_2, y_1, y_2, length]$. With the addition of an extra input feature $in_6$ to the feature space, $F^{(6)}=[x_1, x_2, y_1, y_2, length, in_6]$. The input feature $in_6$ will be first set as a dummy or irrelevant feature, and then reset as a middle ranked significant feature that is a combination of $x_1$ and $x_2$ ($in_6=x_1-x_2$). A set of 1000 data values was generated randomly, with each input feature uniformly distributed on [0,1].

Firstly, the input feature $in_6$ is a dummy or irrelevant feature. It is not obvious to decide how many components to be used in the density functions because the features are uniformly distributed on [0,1]. However, we can obtain the general trend which is around 4 components in the density functions by using the method in [8].

**Table 2** $F^{(6)}=(x_1, x_2, y_1, y_2, length, in_6)$, $F^{(5)}=(x_1, y_1, x_2, y_2, length)$, $F^{(4)}=(x_1, y_1, y_2, length)$, $F^{(3)}=(y_1, y_2, length)$, $F^{(2)}=(y_1, length)$

| Feature | $I\left(F_{-f_i}^{(6)}, C\right)$ | $I\left(F_{-f_i}^{(5)}, C\right)$ | $I\left(F_{-f_i}^{(4)}, C\right)$ | $I\left(F_{-f_i}^{(3)}, C\right)$ | $I\left(F_{-f_i}^{(2)}, C\right)$ |
|---|---|---|---|---|---|
| $f_1=x_1$ | 0.490 | 0.491 | 0.468 | | |
| $f_2=y_1$ | 0.456 | 0.459 | 0.329 | 0.334 | 0.304 |
| $f_3=x_2$ | 0.497 | 0.494 | | | |
| $f_4=y_2$ | 0.439 | 0.443 | 0.354 | 0.366 | |
| $f_5=length$ | 0.217 | 0.194 | 0.129 | 0.116 | 0.014 |
| $f_6=in_6$ | 0.622 | | | | |

The mutual information between the initial feature space and the class is $I(F^{(6)},C)=0.569$. Remove a feature $f_i$ ($i=1,2,...,6$) from the feature space $F^{(m)}$ ($m=5,...,2$) in turn. The mutual information results between $F_{-f_i}^{(m)}$ ($i=1,2,...,m$) and the class are show in Table 2, where $F_{-f_i}^{(m)}$ indicates the feature space $F^{(m)}$ with the feature $f_i$ removed. In each table the maximum mutual information value, i.e. the one to be deleted, is indicated in bold.

The above mutual information values are shown in Fig. 1, where $I\left(F^{(6)}, C\right) =$ 'ΔΔΔ'; $I\left(F_{-f_i}^{(6)}, C\right) =$ 'xxx'; $I\left(F_{-f_i}^{(5)}, C\right) =$ 'ooo'; $I\left(F_{-f_i}^{(4)}, C\right) =$ '+++'; $I\left(F_{-f_i}^{(3)}, C\right) =$ '□□□'; $I\left(F_{-f_i}^{(2)}, C\right) =$ '◇◇◇'.

Notice that for each feature space $I\left(F_{-f_i}^{(6)}, C\right)$, ($i=1,2,...,6$), which contains five features, the mutual information is represented as 'x' in Fig. 1. The maximum mutual information is

$$I\left(F_{-f_6}^{(6)}, C\right) = 0.622$$

Thus, the feature $in_6$ is the least important feature. Also, the feature $in_6$ appears to be irrelevant because the mutual information

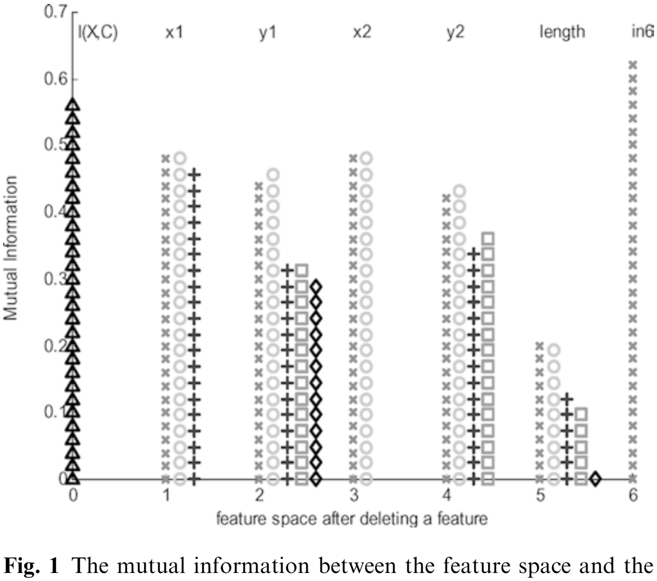

**Fig. 1** The mutual information between the feature space and the class

$$I\left(F^{(6)}_{-f_6}, C\right)$$

is considerably larger than the rest

$$I\left(F^{(6)}_{-f_i}, C\right)$$

(i ≠ 6). The feature length is the most important feature because the mutual information

$$I\left(F^{(6)}_{-f_5}, C\right)$$

is smaller than any of

$$I\left(F^{(6)}_{-f_i}, C\right)$$

(i ≠ 5). For the remaining four features, $x_1$, $y_1$, $x_2$, $y_2$, the mutual information values

$$I\left(F^{(6)}_{-f_i}, C\right)$$

($i = 1, 2, 3, 4$) are almost the same. Thus there are only two possibilities: either these four features are equally significant, or there are redundant features among these four features. We further analysed these four features by deleting one feature in turn and repeating the same process until all the features were ranked. The ranking of the feature from the most important to least important is *length*, $y_1$, $y_2$, $x_1$, $x_2$ and $in_6$. From Fig. 1, we can see that the best feature space is ($x_1$, $y_1$, $x_2$, $y_2$, *length*), because the mutual information between this feature space and the class reaches the maximum value which is 0.622. We use this feature space as inputs to a neural network classifier. We also can see that the four features, $x_1$, $y_1$, $x_2$, $y_2$ are equally significant, because if any of the four had been redundant, the mutual information between the feature space ($x_1$, $y_1$, $x_2$, $y_2$, *length*) and the class would not have been achieve the maximum value.

By repeating the experiment five times with different seeds, we obtain the feature rankings shown in Table 3. From Table 3, the *length* is always the most important, the feature $in_6$ is always the most unimportant and the rest of the features are equally important.

## 6.2 Example B: LIC1 with redundant features

As a variation on the LIC1 experiment, a set of 1000 data was generated randomly, with each input feature uniformly distributed on [0,1], but with the input feature $in_6$ as $x_1 - x_2$. This change should make $in_6$ more important than any single co-ordinate feature, and it should render $x_1$ and $x_2$ redundant. We will demonstrate this fact using the proposed method. The mutual information between the initial feature space and the class is

$$I\left(F^{(6)}, C\right) = 0.042$$

**Table 3** Salience ranking for the inputs' features ($in_6$ = dummy)

| $x_1$ | $y_1$ | $x_2$ | $y_2$ | Length | $in_6$ |
|---|---|---|---|---|---|
| 4 | 2 | 5 | 3 | 1 | 6 |
| 3 | 4 | 2 | 5 | 1 | 6 |
| 4 | 3 | 5 | 2 | 1 | 6 |
| 2 | 4 | 3 | 5 | 1 | 6 |
| 3 | 5 | 2 | 4 | 1 | 6 |

**Table 4** $F^{(6)} = (x_1, x_2, y_1, y_2, \text{length}, in_6)$, $F^{(5)} = (x_1, y_1, y_2, \text{length}, in_6)$, $F^{(4)} = (y_1, y_2, \text{length}, in_6)$, $F^{(3)} = (y_2, \text{length}, in_6)$, $F^{(2)} = (\text{length}, in_6)$

| Feature | $I\left(F^{(6)}_{-f_i}, C\right)$ | $I\left(F^{(5)}_{-f_i}, C\right)$ | $I\left(F^{(4)}_{-f_i}, C\right)$ | $I\left(F^{(3)}_{-f_i}, C\right)$ | $I\left(F^{(2)}_{-f_i}, C\right)$ |
|---|---|---|---|---|---|
| $f_1 = x_1$ | 0.563 | 0.638 | | | |
| $f_2 = y_1$ | 0.037 | 0.480 | 0.446 | | |
| $f_3 = x_2$ | 0.591 | | | | |
| $f_4 = y_2$ | 0.038 | 0.437 | 0.433 | 0.432 | |
| $f_5 = \text{length}$ | 0.010 | 0.162 | 0.175 | 0.141 | 0.083 |
| $f_6 = in_6$ | 0.575 | 0.500 | 0.437 | 0.350 | 0.287 |

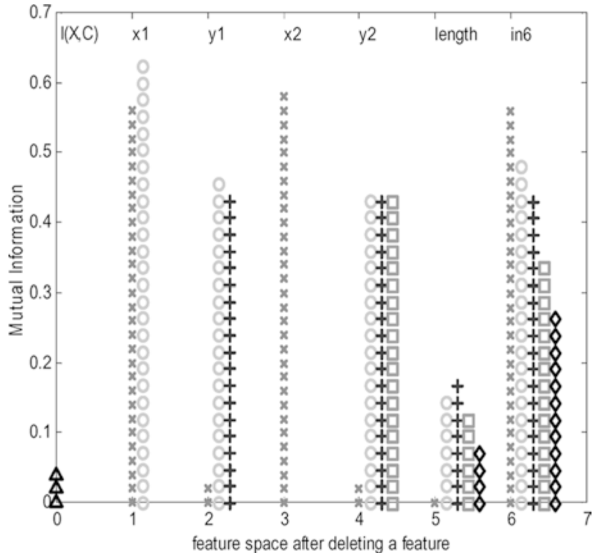

**Fig. 2** The mutual information between the feature space and the class

Applying the algorithm presented earlier, we obtain the results shown in Table 4.

The above mutual information values are summarised in Fig. 2, where $I\left(F^{(6)}, C\right) = $'ΔΔΔ'; $I\left(F^{(6)}_{-f_i}, C\right) = $'xxx'; $I\left(F^{(5)}_{-f_i}, C\right) = $ 'ooo'; $I\left(F^{(4)}_{-f_i}, C\right) = $ ' + + + '; $I\left(F^{(3)}_{-f_i}, C\right) = $'□□□'; $I\left(F^{(2)}_{-f_i}, C\right) = $'◇◇◇'.

From the mutual information between the feature space $F^{(6)}_{-f_i}$ ($i = 1,...,6$) and the class in the Fig. 2, we can see that *length* is the most important, and $y_1$ and $y_2$ are also important, but not as important as *length*. The rest of the features, $x_1$, $x_2$ and $in_6$, have almost the same mutual information, which indicates that these three features are either irrelevant, or that there is redundant information among these three features at this stage. However, in Table 4, we find that the ranking of the

**Table 5** Salience ranking for the inputs features ($in_6 = x_1 - x_2$)

| $x_1$ | $y_1$ | $x_2$ | $y_2$ | Length | $In_6$ |
|---|---|---|---|---|---|
| 6 | 3 | 5 | 2 | 1 | 4 |
| 5 | 4 | 6 | 3 | 1 | 2 |
| 5 | 4 | 6 | 3 | 1 | 2 |
| 6 | 3 | 5 | 4 | 1 | 2 |
| 5 | 3 | 6 | 2 | 1 | 4 |

**Table 6** $F^{(4)} = (SL, SW, PL, PW)$, $F^{(3)} = (SW, PL, PW)$, $F^{(2)} = (PL, PW)$

| Feature | $I\left(F^{(4)}_{-f_i}, C\right)$ | $I\left(F^{(3)}_{-f_i}, C\right)$ | $I\left(F^{(2)}_{-f_i}, c\right)$ |
|---|---|---|---|
| $f_1 = SL$ | 0.963 | | |
| $f_2 = SW$ | 0.961 | 0.970 | |
| $f_3 = PL$ | 0.957 | 0.953 | 0.961 |
| $f_4 = PW$ | 0.913 | 0.895 | 0.913 |

features from the most important to the least important is *length*, $in_6$, $y_2$, $y_1$, $x_1$, $x_2$. After deleting the features $x_1$ and $x_2$ from the initial feature space, the mutual information reaches the maximum which is 0.638, which indicates redundancy among $x_1$, $x_2$ and $in_6$. The optimal best feature subset is $F^{(4)} = (y_1, y_2, length, in_6)$.

Repeating the experiment five times with different seeds, the ranking of features is presented in Table 5. The mean and standard division (STD) of mutual information $I\left(F^{(6)}_{-f_i}, C\right)$ ($i = 1,\ldots,6$) for five experimental tests are presented in Table 5.

From Table 5, we can see that *length* is still the most important, $x_1$ and $x_2$ are the least important, and $y_1$, $y_2$ and $in_6$ are approximately equally important with $in_6$ edging ahead of the other two in Table 5.

## 6.3 Example C: (Fisher's Iris data example)

Fisher's Iris data [3] set contains 150 data items which are obtained from three species (Setosa, Versicolor and Verginica). The data set has four dimensions of measurements, which are sepal length (SL), sepal width (SW), petal length (PL) and petal width (PW).

We use one component in the density function for each class, and three components in the unconditional density function. The mutual information between initial feature space and the class is

$$I\left(F^{(4)}, C\right) = 0.972$$

Using the algorithm presented earlier, the mutual information results between the feature space and the class are presented in Table 6.

A summary of Table 6 is given in Fig. 3, where

$$I\left(F^{(4)}, C\right) = \text{`}\Delta\Delta\Delta\text{'}; I\left(F^{(4)}_{-f_i}, C\right) = \text{`}xxx\text{'};$$

$$I\left(F^{(3)}_{-f_i}, C\right) = \text{`ooo'}; I\left(F^{(2)}_{-f_i}, c\right) = \text{`}+++\text{'}.$$

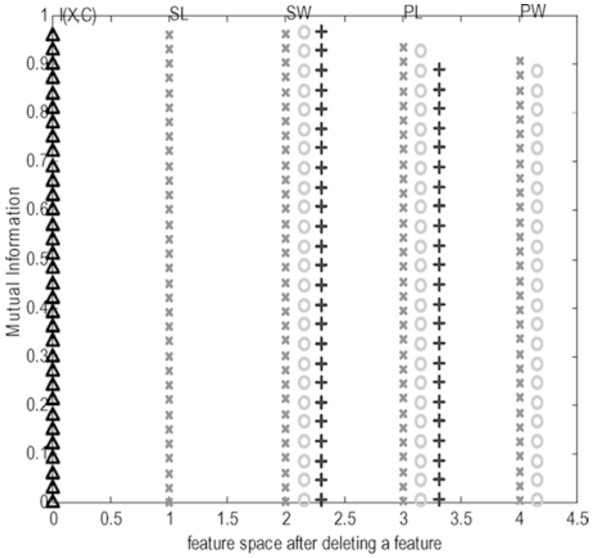

**Fig. 3** The mutual information between the feature space and the class

**Table 7** The mutual information for all subsets of two features for Fisher's Iris data coded in order SL, SW, PL and PW

| All subsets of 2 features | Mutual Information | Ranking |
|---|---|---|
| SLSW | 0.695 | 6 |
| SLPL | 0.924 | 5 |
| SLPW | 0.962 | 4 |
| SWPL | 0.896 | 3 |
| SWPW | 0.953 | 2 |
| PLPW | 0.970 | 1 |

From Fig. 3, the ranking from the least important to the most important is *SL*, *SW*, *PL* and *PW*. The best feature space is the initial feature space $F^{(4)} = (SL, SW, PL, PW)$.

The mutual information is presented for all subsets of two features in Table 7, using one component in the density function for each class and three components [8] in the unconditional density function.

From Table 7, we can see that the worst case is SLSW. (Table 8 shows the feature set for the linear prediction filter of the STCA system). The best cases are PLPW and SLPW. The different between the results in Table 9 and the results in [3] is that we find the best case to be PLPW instead of the best case SLPL in [3]. The results for the cases PLPW and SLPL are show in Figs. 4 and 5; there are three classes which are presented as 'Δ', 'o' and '...' respectively.

From Figs. 4 and 5, it can be seen that our selection of the best feature pair (Fig. 5) is better than that in Fig. 4, the best feature pair in [3], because the projection of our selection allows partitioning with less incorrect classifications.

**Table 8** The feature set for the linear prediction filter of the STCA system

| Name | Type | Description |
|---|---|---|
| ΔX | R | $\Delta X = X_2 - X_1$(X relative distance of pair track) |
| ΔY | R | $\Delta Y = Y_2 - Y_1$(Y relative distance of pair track) |
| $\Delta V_x$ | R | $\Delta V_x = V_{x_2} - V_{x_1}$ (X relative velocity of pair track) |
| $\Delta V_y$ | R | $\Delta V_y = V_{y_2} - Y_{y_1}$ (Y relative velocity of pair track) |
| ΔZ | R | $\Delta Z = Z_2 - Z_1$(Z relative distance of pair track) |
| $\Delta V_z$ | R | $\Delta V_z = V_{z_2} - V_{z_1}$ (Z relative velocity of pair.) |
| L | R | $L = \sqrt{(\Delta X)^2 + (\Delta Y)^2}$ (Lateral distance of pair track.) |
| $V_{clos}$ | R | $V_{clos} = \frac{\Delta V_x \Delta X + \Delta V_y \Delta Y}{\sqrt{(\Delta X)^2 + (\Delta Y)^2}}$ (Lateral closing speed.) |
| $V_r$ | R | $V_r = \sqrt{(\Delta V_x)^2 + (\Delta V_y)^2}$ (Lateral velocity of pair track.) |
| LMD | R | $LMD = \left\lvert \frac{\Delta X \Delta V_y - \Delta Y \Delta V_x}{\sqrt{(\Delta V_x)^2 + (\Delta V_y)^2}} \right\rvert$ (Lateral misses distance.) |
| TLMA | R | $TLMA = \frac{\Delta V_x \Delta X + \Delta V_y \Delta Y}{(\Delta V_x)^2 + (\Delta V_y)^2}$ (Time of minimum lateral approach.) |
| Alert | N | Classification (0 indicates non-alert, 1 indicates alert) |

**Table 9** Salience ranking for the inputs features using mutual information

| | ΔX | ΔY | $\Delta V_x$ | $\Delta V_y$ | ΔZ | $\Delta V_z$ | L | $V_{clos}$ | $V_r$ | LMD | TLMA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 2 | 5 | 10 | 11 | 1 | 9 | 3 | 6 | 8 | 7 | 4 |
| | 8 | 3 | 11 | 9 | 4 | 7 | 5 | 2 | 10 | 1 | 6 |
| | 6 | 7 | 10 | 9 | 3 | 5 | 8 | 4 | 11 | 2 | 1 |
| | 3 | 6 | 9 | 11 | 4 | 5 | 1 | 2 | 10 | 8 | 7 |
| | 4 | 7 | 11 | 9 | 3 | 6 | 1 | 2 | 10 | 8 | 5 |
| | 4 | 7 | 11 | 10 | 9 | 8 | 3 | 1 | 2 | 5 | 6 |
| | 6 | 5 | 11 | 8 | 2 | 7 | 4 | 1 | 10 | 3 | 9 |
| | 4 | 7 | 11 | 9 | 3 | 6 | 1 | 2 | 10 | 8 | 5 |
| | 3 | 6 | 9 | 11 | 4 | 5 | 1 | 2 | 10 | 8 | 7 |
| | 7 | 4 | 10 | 11 | 9 | 2 | 1 | 3 | 8 | 5 | 6 |
| | 4 | 7 | 11 | 9 | 3 | 6 | 1 | 2 | 10 | 8 | 6 |
| | 3 | 6 | 9 | 11 | 4 | 5 | 1 | 2 | 10 | 8 | 7 |
| Ranking | | | 11 | 10 | | | 1 | 2 | 9 | | |

## 6.4 Example D: STCA data set

Short term conflict alert (STCA) is an air traffic control system designed to give air-traffic controllers an alert of potential conflicts with sufficient warning time. The purpose of the STCA system is to predict the likelihood of a pair of aircraft breaching proximity restrictions. The current STCA system contains three fine filter modules, i.e. a linear prediction filter, a current proximity filter and a manoeuvre hazard filter, each of which attempts to predict likely breaches of proximity restrictions in different flight situations. In this paper, we concentrate on the linear prediction filter that accounts for 90% of all alerts in practice. 11 continuous features that may be extracted from the raw flight path data are described in Table 8.

A large quantity of flight-path radar data from London's Heathrow airport has been concentrated to provide a reasonable balance between "Alert" and "Non-Alert" cases. In a data set from the years 1988 to 1998 there are 12522 patterns, in total 4647 alerts and 7875 non-alerts generated by the linear prediction filter.

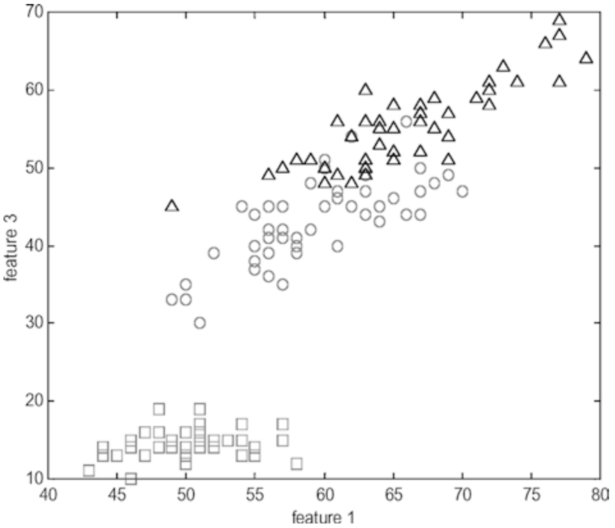

**Fig. 4** Feature 1 and feature 3 (SLPL)

We randomly selected 4000 patterns from the data set, and set five components in the density function for each class and data by ignoring the class label. The initial feature space is

$F^{(11)} = (\Delta X, \Delta Y, \Delta V_x, \Delta V_y, \Delta Z, \Delta V_z, L, V_{clos}, V_r, LMD, TLMA)$. By repeatedly randomly selecting 4000 patterns from the data set, the ranking was repeated 12 times for the 11 input features using the method presented in this paper and the clamping method [1]. The results are presented in Tables 9 and 10, respectively.

The results are not totally stabled due to the similarity of importance between some features and the noise in the data. But from Tables 9–10, we can see that the most important features are $L$ and $V_{clos}$, and the least important features are $\Delta V_x$, $\Delta V_y$ and $V_r$ using the new

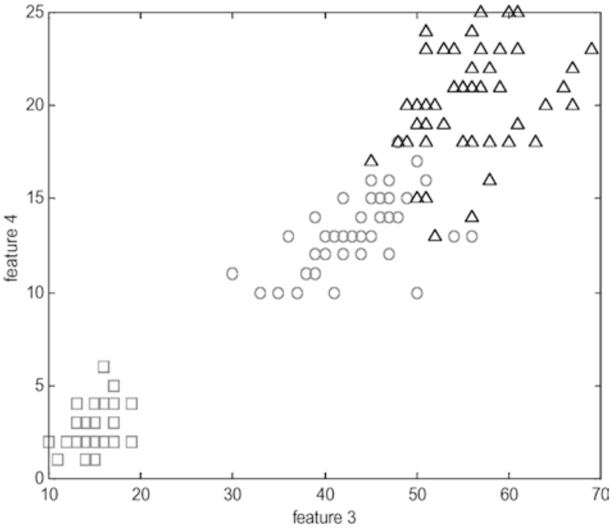

**Fig. 5** Feature 3 and feature 4 (PLPW)

**Table 10** Salience ranking for the inputs features using the clamping technique

| | ΔX | ΔY | $\Delta V_x$ | $\Delta V_y$ | ΔZ | $\Delta V_z$ | L | $V_{clos}$ | $V_r$ | LMD | TLMA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 5 | 11 | 6 | 4 | 9 | 8 | 1 | 2 | 3 | 7 | 10 |
| | 5 | 10 | 4 | 6 | 7 | 9 | 1 | 2 | 3 | 8 | 11 |
| | 4 | 9 | 6 | 5 | 7 | 8 | 1 | 2 | 11 | 3 | 10 |
| | 3 | 9 | 7 | 6 | 5 | 8 | 1 | 2 | 11 | 4 | 10 |
| | 5 | 11 | 8 | 7 | 9 | 6 | 1 | 2 | 4 | 3 | 10 |
| | 5 | 9 | 8 | 7 | 10 | 6 | 1 | 2 | 3 | 4 | 11 |
| | 6 | 9 | 7 | 10 | 4 | 11 | 2 | 1 | 3 | 5 | 8 |
| | 3 | 7 | 11 | 9 | 4 | 6 | 1 | 2 | 8 | 5 | 10 |
| | 8 | 10 | 3 | 7 | 5 | 9 | 1 | 2 | 4 | 6 | 11 |
| | 3 | 11 | 4 | 8 | 7 | 6 | 1 | 2 | 9 | 5 | 10 |
| | 8 | 10 | 4 | 7 | 5 | 9 | 1 | 2 | 3 | 6 | 11 |
| | 5 | 10 | 6 | 4 | 9 | 7 | 1 | 2 | 8 | 3 | 11 |
| Ranking | | 10 | | | | | 1 | 2 | | | 11 |

method. If we use the clamping method for the feature ranking [1], the two most important features are selected as *L* and $V_{clos}$, while *TLMA* and $\Delta Y$ are the least important. The general trend for both methods is the same. Deleting the least important features, which are $\Delta V_x$, $\Delta V_y$ and $V_r$ using the new method presented in this paper, the rest 8 features constitute feature set **A**. Deleting the least important features *TLMA* and $\Delta Y$ according to the clamping method, the rest of the nine features constitute feature set **B**. We randomly select 4000 patterns from the data set as the training data, and select another 2000 patterns from the data set as the test data. The generalisation performance of the test data using feature set **A** is better than using feature set **B** for the MLP method.

We then randomly select 4000 patterns from the data set as the training data, and select another 4000 patterns from the data set as the test data set. The generalisation performance for the test data set is about 83% using the feature set **A** as inputs and about 80% using the feature set **B** as inputs. The ROC curves are presented in Fig. 6 using the Bayes' rule, where '*' and 'o' indicate the training and the test data sets, respectively, for feature set **A**, and '...' and 'ΔΔΔ' indicate the training and the test data sets, respectively, for feature set **B**. From Fig. 6, we can see that the performance of feature set **A** is better than that of feature set **B**. This suggests that the new method proposed in this paper may be superior to the clamping method on the complex, noisy data. This result is particularly interesting as feature set **A** contains one less feature than feature set **B**.

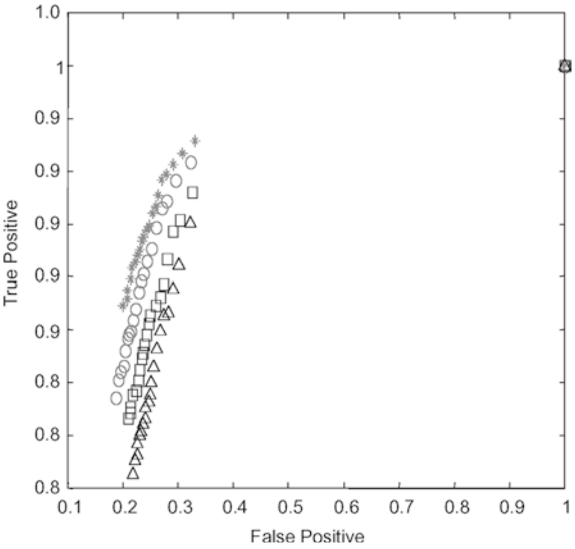

**Fig. 6** ROC curves for feature sets **A** and **B**

## 7 Discussion and conclusions

Results on the LIC1 problem, for which the feature rankings were known and manipulated, are exactly as expected; irrelevant, redundant and approximately equally ranked features are all identified correctly by the proposed method in this paper. Using the Iris data, we provide a more comprehensive analysis than was previously published, and we identity a two-feature combination that is better than the previously published one.

This paper has presented a new approach to the important pattern recognition task of optimal best feature subset finding. The approach is based on the formula for the mutual information, and the results are encouraging on the range of problems sampled. There are a number of advantages when compared with [1] and [3].

1. [3] only considers the features individually with the class, and the individual features with each other. For the most significant feature this strategy is sufficient. However, it is difficult to identify non-significant features such as $in_6$ when it is a dummy feature in LIC1. It may be selected as the same importance as the other features $y_2$, $y_1$, $x_1$, $x_2$. Since [3] only considers the dependencies between each individual feature and the class, and between each individual

feature and *length*, they do not consider the individual features in relation to all other features together.

2. The algorithms in [3] need to compute $s{\times}n$ times for the selection of the next feature, where $n$ is the total number of the features for selection and $s$ is the total number of the features that are already in the selected feature set. The algorithm presented in this paper is robust compared with the algorithm in [3], and it is more efficient to run $n$-$k$ rather than $\binom{n}{k}$ times to find the subset containing $k$ features from the initial set of $n$ features. It is also more general and more informative.
3. We can analyse the nature of the feature, whether it is irrelevant or redundant. In LIC1 example, if we set $in_6$ as an irrelevant feature, only one feature space, $F^{(5)} = (x_1, x_2, y_1, y_2, length)$, records a maximum mutual information. This tells us that the feature $in_6$ is an irrelevant feature. If we set $in_6$ as the linear combination of features $x_1$ and $x_2$, the mutual information between the feature space and the class

   $$I\left(F^{(6)}_{-f_i}, c\right)$$

   is small if $f_i$ is $y_1$, $y_2$ or *length*, thus $y_1$, $y_2$ and *length* are neither irrelevant nor redundant features. But the mutual information between the feature space and the classis more than an order of magnitude larger if $f_i$ is $x_1$, $x_2$ or $in_6$. So the features $x_1$, $x_2$ and $in_6$ are either irrelevant or redundant features, but the mutual information between the feature space

   $$I\left(F^{5}_{-x_1}, C\right) = F^{(4)} = (y_1,\ y_2,\ length,\ in_6)$$

   and the class suddenly become larger than any alternative. In this case, we can see that $in_6$ has a relationship with $x_1$ and $x_2$, and that $x_1$ and $x_2$ are redundant features. In addition, we can analyse the relationship between the features during the process, but we can't obtain any of this information from [1]. The complexity of the method in [1] is linear in the total number of features, $n$, with the addition that a network must first be trained using all $n$ features. Our method is more complex, but it is quite tractable and yields significantly more useful information.
4. The density functions in [3] are approximated by histograms, i.e., by counting the number of cases with values of the variables belonging to a set of intervals. It is difficult to count in more than a one-dimensional feature space and the values depend on the width of the bins. It might be supposed that we can select the most important feature $f_{j_1}$ from the initial feature space first, i.e., one which maximises the mutual information $I\left(f_{j_1}, C\right)$, and then select another feature $f_{j_2}$ $(j_2 \neq j_1)$ that maximises

   $$I\left(\{f_{j_1}, f_{j_2}\}, C\right)$$

   from the feature space ignoring the features that are already selected. From this process we can obtain the feature ranking and "best" feature space selection. However, we found this algorithm is not as accurate as the algorithm specified earlier. Using the process which selects an important feature first for the above example, *length* is the most important, because it is significant feature compare with the other features. But then it selects the second important feature from $x_1$, $y_1$, $x_2$, $y_2$, $in_6$ at random. It sometimes selects $in_6$ as the second important feature, which is wrong.

A limitation of our proposed method is that the algorithm only applies to continuous feature spaces, because the density functions are constructed using the EM algorithm. We can expand the formula in Eq. 4 to the feature spaces that are constructed using both continuous and discrete sub-feature spaces. The density function in Eq. 4 for the feature space

$$X \cup V$$

where $X = (x_1, x_2, \ldots, x_p)$ indicates the continuous sub-feature space and $V = (v_1, v_2, \ldots, v_q)$ indicates that the discrete sub-feature space, can be written as

$$\begin{aligned} p(X \cup V) &= p(X|V)p(V) \\ &= \sum_{v_1} \cdots \sum_{v_q} p(X|V)p\left(v_1|v_2 \cdots v_q\right) \\ &\quad \times p\left(v_2|v_3 \cdots v_q\right) \cdots p\left(v_{q-1}|v_q\right)p\left(v_q\right). \end{aligned}$$

From the abovementioned formula, we can see that the computation is extensive if $q$ and the choice of nominal values $v_i$ are large.

**Acknowledgements** We wish to thank Julia Sonander and Harri Howells of National Air-Traffic Services for the STCA data, and the Engineering and Physical Science Research Council of the UK for supporting this work (grant no. GR/M75143).

## References

1. Wang WJ, Jones P, Partridge D (1999) Assessing the impact of input features in a feedforward network. Neural Computing and Applications 9:101–112
2. van de Laar P, Heskes TM, Gielen CCAM (1999) Partial retraining: a new approach to input relevance determination. Int J Neur Sys 9:75–85
3. Battiti R (1994) Using mutual information for selecting features in supervised neutral net learning. IEEE Trans Neur Netwks 5:537–550
4. Kwak N, Choi C-H (2002) Input feature selection for classification problems. IEEE Trans Neur Netwks 13:143–159
5. Tchaban T, Taylor MJ, Griffin J (1998) Establishing impacts of the inputs in a feedforward network. Neural Computing and Applications 7:309–317
6. Young TY, Coraluppi G (1970) Stochastic estimation of a mixture of normal density functions using an information criterion. IEEE Trans Info Theor 16:258–263
7. Carreira-Perpinan MA (2000) Mode-finding for mixtures of Gaussian distributions. IEEE Trans Patt Anal Mach Intell 22(11):1318–1323
8. Cang S, Partridge D (2001) Determining the number of components in mixture models using Williams' statistical test. In: Proceedings of the 8th International Conference on Neural Information Processing, Shanghai, China, November 2001

9. Richardson S, Green PJ (1997) On Bayesian analysis of mixtures with an unknown number of components. J Roy Stat Soc B59:731–792
10. Bishop C (1995) Neural networks for pattern recognition. Oxford University Press, Oxford, UK
11. Haykin S (1999) Neural networks: a comprehensive foundation. Prentice-Hall, Englewood Cliffs, NJ
12. Theodoridis S, Koutroumbas K (1999) Pattern recognition. Academic Press, San Diego, CA