Pattern Recognition 46 (2013) 3315–3327

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Mutual information-based method for selecting informative feature sets

Gunawan Herman [a,b], Bang Zhang [a,b], Yang Wang [a,b,*], Getian Ye [c], Fang Chen [a,b]

[a] National ICT Australia, Level 5, 13 Garden Street, Eveleigh NSW 2015, Australia
[b] The University of New South Wales, NSW 2052, Australia
[c] Canon Information System Research Australia, Level 7, 3 Thomas Holt Drive, North Ryde NSW 2113, Australia

ARTICLE INFO

Article history:
Received 23 April 2012
Received in revised form
14 April 2013
Accepted 27 April 2013
Available online 23 May 2013

Keywords:
Feature selection
Mutual information

ABSTRACT

Feature selection is one of the fundamental problems in pattern recognition and data mining. A popular and effective approach to feature selection is based on information theory, namely the mutual information of features and class variable. In this paper we compare eight different mutual information-based feature selection methods. Based on the analysis of the comparison results, we propose a new mutual information-based feature selection method. By taking into account both the class-dependent and class-independent correlation among features, the proposed method selects a less redundant and more informative set of features. The advantage of the proposed method over other methods is demonstrated by the results of experiments on UCI datasets (Asuncion and Newman, 2010 [1]) and object recognition.

1. Introduction

Feature selection is one of the fundamental problems in pattern recognition and data mining. In recent years, the data that we process have become increasingly larger, both in the number of examples and the number of features. Processing such data is often a problem due to the associated long computation time and the required computing resources. To make it worse, those data often contain irrelevant and redundant features. Irrelevant and redundant features not only add to the computational burden, but also have negative effect on the accuracy of trained classifiers. A straightforward solution to this problem is to apply feature selection to the datasets as a preprocessing step.

Feature selection is a process for selecting a subset of the original features, such that as much information as possible about the data is retained in the selected feature set. The data will then be represented by the selected feature set, which is usually much smaller compared to the original feature set. Feature selection is useful for increasing the efficiency in both learning and classification. Also, the selected feature set ideally contains only the most informative features so as to improve the generalization performance of trained classifiers.

In this paper we focus on first-order incremental feature selection methods, that select one feature after another based on a certain objective function. Specifically, we study mutual information-based feature selection methods where the objective function is based on the mutual information of features and class variable. There are a number of such methods known in the literature, and prior to this, there are few studies that compare them.

Based on the maximal statistical dependency criterion, Peng et al. [2] proposed a minimal-redundancy-maximal-relevance (mRMR) criterion for first-order incremental feature selection. Following this work, Balagani and Phoha [3] derived a feature selection criterion based on multidimensional mutual information between features and class. Liu et al. [4] treated feature selection as a feature clustering procedure, and mutual information was used as measurement for the between-cluster distance. Kerroum et al. [5] applied Gaussian mixture model on the mutual information calculation between input features and output classes to achieve maximal relevance-based feature selection. Fleuret [6] proposed a conditional mutual information-based feature selection method. It selects features which can maximize their mutual information with the class label conditional to the selected features. Lin and Tang [7] introduced a new concept called class-relevant redundancy, and a novel algorithm called conditional informative feature extraction is proposed by maximizing the joint class-relevant information. Ding and Peng [8] proposed a minimum redundancy maximum relevance (MRMR) feature selection framework for microarray gene expression data.

We study mutual information-based feature selection methods as they are simple and some of them have been shown to perform

[*] Corresponding author at: National ICT Australia, Level 5, 13 Garden Street, Eveleigh NSW 2015, Australia. Tel.: +61 2 9376 2200.
E-mail addresses: Gherman.email@gmail.com (G. Herman), bang.zhang@nicta.com.au (B. Zhang), yang.wang@nicta.com.au (Y. Wang), getian.ye@gmail.com (G. Ye), fang.chen@nicta.com.au (F. Chen).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.04.021

similarly or better than other (more sophisticated) methods [6,9]. There are three main contributions in this paper:

1. We present a general formulation of mutual information-based feature selection and show how different methods fit in this formulation.
2. We compare the performance of eight popular mutual information-based feature selection methods (maxRel, CMIM [6,10], CIFE (Infomax) [7], Avg-CMIM, MID [2], MIQ [8], DDC [11] and DIC [11]).
3. Based on the analysis of the comparison results, we propose a feature selection method that effectively selects a less redundant and more informative set of features.

The rest of this paper is organized as follows. In Section 2, we present a general formulation of mutual information-based feature selection and show how different methods fit into this formulation. Section 3 compares these methods using ten UCI datasets and two popular machine learning algorithms (Support Vector Machines and naive Bayes). We then present the proposed feature selection method and show how its parameters are tuned in Section 4. The effectiveness of the proposed feature selection method is demonstrated in Section 5 with experiments on UCI datasets and object recognition. Section 6 presents the conclusions and future work.

2. Mutual information-based feature selection

Given a dataset, feature selection incrementally selects a subset of original features. At iteration m, a feature selection method selects a feature x_m and adds it to a set of already selected features S_{m-1}, resulting in the set S_m which contains m features,

$$S_m = S_{m-1} \cup \{x_m\} = \{x_1, \ldots, x_m\},$$

where $m \in \{1, 2, \ldots\}$ and $S_0 = \{\}$.

In mutual information-based feature selection, a selected feature set S_m is optimal if the mutual information of the class y and the feature set S_m is maximal. This criterion is called *maximal dependency* [2]. The dependency of class y on feature set S_m is defined as

$$\begin{aligned} I(y;S_m) &= \iint p(S_m, y) \log \frac{p(S_m, y)}{p(S_m)p(y)} dS_m\, dy \\ &= \iiint p(S_{m-1}, x_m, y) \log \frac{p(S_{m-1}, x_m, y)}{p(S_{m-1}, x_m)p(y)} dS_{m-1}\, dx_m\, dy \\ &= \int \ldots \int p(x_1 \ldots, x_m, y) \log \frac{p(x_1, \ldots, x_m, y)}{p(x_1, \ldots, x_m)p(y)} dx_1 \ldots dx_m\, dy. \end{aligned} \quad (1)$$

$I(y;S_m)$ is the amount of information about y that can be gained from the features in S_m. By using the *chain rule for information* [12] it can be written as

$$I(y;S_m) = I(y;S_{m-1}) + I(y;x_m|S_{m-1}), \quad (2)$$

where $I(y;x_m|S_{m-1})$ is the mutual information of y and x_m conditional on S_{m-1}, which tells us the amount of new information about y that can be gained from x_m, and has not already been gained from S_{m-1}.

When many features have been selected, it is inevitable that x_m offers some redundant information about y that has already been offered by some other features in S_{m-1}. Similar to [11], we adopt a Venn diagram, Fig. 1, to illustrate this. In the rightmost diagram, the shaded area shows the redundancy between two features a and b with respect to class y, *i.e.* $R_y(a,b) = I(a,b) - I(a,b|y)$, which can also be represented as $I(y,a) - I(y,a|b)$ or $I(y,b) - I(y,b|a)$. The conditional mutual information $I(a,b|y)$ is represented by the area under the shaded area.

Here both x and x_i denote features. Without losing generality, x represents the feature which is going to be selected, and x_i represents the feature which has already been selected. The subscript i of x_i is used to index different selected features. The redundancy between features x and x_i with respect to class y can be formulated as

$$\begin{aligned} R_y(x, x_i) &= I(y;x) + I(y;x_i) - I(y;x,x_i) \\ &= I(y;x) - I(y;x|x_i). \end{aligned} \quad (3)$$

Following the above equation, the redundancy between feature x and feature set S with respect to class y is

$$\begin{aligned} R_y(x, S) &= I(y;x) + I(y;S) - I(y;x,S) \\ &= I(y;x) - I(y;x|S), \end{aligned} \quad (4)$$

and so

$$I(y;x|S) = I(y;x) - R_y(x,S). \quad (5)$$

By expanding the conditional mutual information in Eq. (2) according to Eq. (5), we can write the dependency of class y on feature set S_m as

$$I(y;S_m) = I(y;S_{m-1}) + \{I(y;x_m) - R_y(x_m, S_{m-1})\}, \quad (6)$$

and because $I(y;S_{m-1})$ is a constant with respect to feature x_m, the dependency is maximized when we select a feature x_m such that

$$x_m = \underset{x}{\operatorname{argmax}} \{I(y;x) - R_y(x, S_{m-1})\}. \quad (7)$$

The above equation is a *general formulation of mutual information-based feature selection*, which as we can see consists of two terms. The first term, $I(y;x)$, measures the relevance of feature x to class y, and the second term, $R_y(x, S_{m-1})$, measures the redundancy between feature x and a set of already selected features S_{m-1} with respect to class y. Therefore, the dependency is a combination of two factors: *relevance* and *redundancy*.

Despite the general formulation, calculating redundancy factor $R_y(x, S_{m-1})$ is as difficult as calculating the dependency (Eq. (1)) for the following reasons [2]:

- There are often not enough examples available.
- Accurate estimation for multivariate density $p(x_1, \ldots, x_m, y)$ and $p(x_1, \ldots, x_m)$ is difficult.

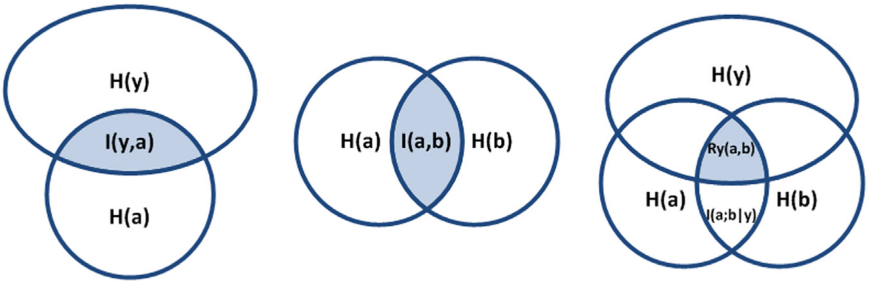

Fig. 1. Illustrations of the correlations between features (shown as circles) and class variable (shown as oval). $H(a)$ denotes the information entropy of variable a. Left: the shaded area is the information-gain from feature a about class y (i.e., $I(y;a)$). Middle: the shaded area is the class-independent correlation between features a and b (i.e., $I(a;b)$). Right: the shaded area is the class-dependent correlation between features a and b with respect to class y (i.e., $R_y(a,b)$). See text for details.

- The computational complexity is high.

Here we show how the redundancy factor is approximated by different mutual information-based feature selection methods.

The simplest approach obviously is to ignore the redundancy factor:

$$x_m = \underset{x}{\operatorname{argmax}}\{I(y;x)\}. \tag{8}$$

This is called maximal relevance (maxRel) method [2].

We can also approximate the redundancy between feature x_m and feature set S_{m-1} by using pairwise redundancy between x_m and any feature in S_{m-1}. For instance, if we approximate $R_y(x,S_{m-1})$ by the maximum pairwise redundancy between x_m and any feature in S_{m-1} then Eq. (7) becomes

$$\begin{aligned} x_m &= \underset{x}{\operatorname{argmax}}\left\{I(y;x) - \max_{1\le i\le m-1} R_y(x,x_i)\right\} \\ &= \underset{x}{\operatorname{argmax}}\left\{I(y;x) - \max_{1\le i\le m-1}\{I(y;x)-I(y;x|x_i)\}\right\} \\ &= \underset{x}{\operatorname{argmax}}\left\{I(y;x) - \left\{I(y;x) - \min_{1\le i\le m-1} I(y;x|x_i)\right\}\right\} \\ &= \underset{x}{\operatorname{argmax}}\left\{\min_{1\le i\le m-1} I(y;x|x_i)\right\}, \end{aligned} \tag{9}$$

which is the formulation of CMIM [6,10] method.

The formulation of CIFE (Infomax) [7] method differs from CMIM in that it approximates $R_y(x,S_{m-1})$ by the sum of redundancy between x_m and every feature in S_{m-1},

$$\begin{aligned} x_m &= \underset{x}{\operatorname{argmax}}\left\{I(y;x) - \sum_{i=1}^{m-1} R_y(x,x_i)\right\} \\ &= \underset{x}{\operatorname{argmax}}\left\{I(y;x) - \sum_{i=1}^{m-1}\{I(y;x)-I(y;x|x_i)\}\right\} \\ &= \underset{x}{\operatorname{argmax}}\left\{I(y;x) - (m-1)I(y;x) + \sum_{i=1}^{m-1} I(y;x|x_i)\right\} \\ &= \underset{x}{\operatorname{argmax}}\left\{(2-m)I(y;x) + \sum_{i=1}^{m-1} I(y;x|x_i)\right\}. \end{aligned} \tag{10}$$

Alternatively, we can approximate $R_y(x,S_{m-1})$ by the average redundancy between x_m and every feature in S_{m-1},

$$\begin{aligned} x_m &= \underset{x}{\operatorname{argmax}}\left\{I(y;x) - \frac{1}{m-1}\sum_{i=1}^{m-1} R_y(x,x_i)\right\} \\ &= \underset{x}{\operatorname{argmax}}\left\{I(y;x) - \frac{1}{m-1}\sum_{i=1}^{m-1}\{I(y;x)-I(y;x|x_i)\}\right\} \\ &= \underset{x}{\operatorname{argmax}}\left\{I(y;x) - I(y;x) + \frac{1}{m-1}\sum_{i=1}^{m-1} I(y;x|x_i)\right\} \\ &= \underset{x}{\operatorname{argmax}}\left\{\frac{1}{m-1}\sum_{i=1}^{m-1} I(y;x|x_i)\right\}, \end{aligned} \tag{11}$$

which is the formulation of what we call Avg-CMIM method.

So far we have shown how CMIM, CIFE, and Avg-CMIM methods are derived from the general formulation (Eq. (7)) by approximating redundancy factor using pairwise redundancy $R_y(x,x_i)$ in different ways. Qu et al. [11] further differentiate the redundancy between two features into two types, namely *class-dependent correlation* and *class-independent correlation*. And accordingly, a feature extraction algorithm is proposed based on the former in [11]. For thorough comparison, we also consider the latter in the comparative experiments. According to the definitions in [11], the *class-dependent correlation* between x and x_i could be regarded as normalized $R_y(x,x_i)$. Redundancy between x and x_i also exists regardless of the class, which means that there is some mutual information between x and x_i (i.e., $I(x;x_i) \ge 0$). This type of redundancy is *class-independent correlation* and is illustrated by the middle diagram in Fig. 1, where the shaded area shows the class-independent correlation between two features.

Table 1
Summaries of various mutual information-based feature selection methods. The second column shows the objective function of each method. The third column shows the type of correlation used by different methods to model redundancy factor. CIC stands for class-independent correlation. CDC stands for class-dependent correlation.

Methods	Objective function	Redundancy
maxRel	$I(y;x)$	–
CMIM [6]	$I(y;x) - \max_{x_i} R_y(x,x_i)$	CDC
CIFE [7]	$I(y;x) - \sum_{i}^{m-1} R_y(x,x_i)$	CDC
Avg-CMIM	$I(y;x) - \frac{1}{m-1}\sum_{i}^{m-1} R_y(x,x_i)$	CDC
MID [2]	$I(y;x) - \frac{1}{m-1}\sum_{i}^{m-1} I(x;x_i)$	CIC
MIQ [8]	$I(y;x) \Big/ \frac{1}{m-1}\sum_{i}^{m-1} I(x;x_i)$	CIC
DDC [11]	$a)I(y;x)\ b) - \frac{R_y(x,x_i)}{H(y)}\ i=1,\ldots,m-1$ [a]	CDC
DIC [11]	$a)I(y;x)\ b) - \frac{I(x,x_i)}{H(x_i)}\ i=1,\ldots,m-1$	CIC
Proposed	$\lambda_m I(y;x) + \frac{1}{m-1}\sum_{i}^{m-1}\{\eta I(y;x|x_i) - (1-\eta)I(x;x_i)\}$	CDC+CIC

[a] See steps 8 and 9 of Algorithm 1 in [11].

From Fig. 1 we can relate the two types of redundancy as follows:

$$I(x;x_i) = R_y(x,x_i) + I(x;x_i|y). \tag{12}$$

Because $I(x;x_i|y) \ge 0$, two features that are little or uncorrelated with respect to the class (i.e., low $R_y(x,x_i)$) may be highly correlated with each other (i.e., high $I(x;x_i)$). Qu et al. [11] suggest that only class-dependent correlation is significant for feature selection. One may see class-independent correlation as an over-estimation of the redundancy of features as far as selecting discriminative features is concerned.

Nevertheless, class-independent correlation is used in the formulation of the following two methods. MID [2] approximates $R_y(x,S_{m-1})$ by the average class-independent correlation between x and all the features in S_{m-1},

$$x_m = \underset{x}{\operatorname{argmax}}\left\{I(y;x) - \frac{1}{m-1}\sum_{i=1}^{m-1} I(x;x_i)\right\}. \tag{13}$$

We can think of MID as a variant of Avg-CMIM where the redundancy $R_y(x,x_i)$ is overestimated by $I(x;x_i)$, thus MID still follows from the general formulation (Eq. (7)).

MIQ [8] is slightly different from MID in that it takes the ratio between relevance and redundancy factors,

$$x_m = \underset{x}{\operatorname{argmax}}\left\{I(y;x) \Big/ \frac{1}{m-1}\sum_{i=1}^{m-1} I(x;x_i)\right\}. \tag{14}$$

Unlike the aforementioned methods, MIQ does not follow from Eq. (7), therefore we argue that MIQ does not maximize the dependency measure.

Table 1 summarizes the objective functions and the type of feature correlation used in different feature selection methods.

3. Comparative experiments

In this section, we compare the performance of the methods described in the previous section. We use ten UCI datasets [1] which are summarized in Table 2. The ten datasets cover a wide range of characteristics, in terms of the number of features, the

Table 2
Summaries of the ten UCI datasets [1] used in Section 3. The last two columns are the average accuracy ± the standard deviation (%) of five repetitions of 10-fold cross-validation using SVM and NB with all the features, respectively.

Datasets	#Features	#Examples	#Classes	SVM	NB
Promoters	57	106	2	79.06 ± 3.09	90.15 ± 0.89
Spambase	57	4601	2	91.77 ± 0.03	79.49 ± 0.13
Splice	60	3175	3	84.69 ± 0.11	95.48 ± 0.12
OptDigits	64	3823	10	98.16 ± 0.08	92.60 ± 0.17
Musk1	166	476	2	85.59 ± 0.61	74.11 ± 0.79
Musk2	166	6598	2	95.23 ± 0.03	83.90 ± 0.09
Arrhythmia	279	452	16	70.27 ± 0.64	62.43 ± 0.75
Madelon	500	2000	2	53.45 ± 0.80	58.43 ± 0.19
Multi-features	649	2000	10	98.39 ± 0.14	95.22 ± 0.10
Advertisements	1558	3279	2	96.71 ± 0.12	96.68 ± 0.08
Average	–	–	–	85.33 ± 14.48	82.85 ± 13.96

number of examples, and the number of classes. These datasets contain both continuous and nominal features. Because mutual information-based feature selection expects nominal features, we binarize each continuous feature using a threshold that is chosen such that the mutual information of the class and the binarized feature is maximized. The same binarization method is also used in [13,10]. It is worth noting that the binarization is only used for feature selection. The classification process still uses the original feature values, so that the reported classification accuracy would be objective. As an approximation for computing the mutual information of continuous features, feature discretization has been widely used in different machine learning algorithms [14–16]. An alternative is Parzen window approach [17,18]. However, its performance depends on the selections of window function and window width parameter. It is often nontrivial to obtain the optimal window function and window width parameter. As mentioned in [2], Parzen window method is usually used when it is unclear how to properly discretize the continuous data. Comparing with Parzen window method, the method used in this work is relatively simple and suitable for automatic parameter determination.

For classification we use two popular machine learning algorithms: Support Vector Machines with linear kernel (SVM) and naive Bayes (NB). We use LIBSVM [19] for the implementation of SVM,[1] and WEKA [20] for NB. We use classification accuracy (of SVM or NB), when used together with a feature selection method, as an indicator of the performance of the feature selection method, where higher classification accuracy indicates a better performance of the feature selection method.

The overall performance of a feature selection method is measured according to Algorithm 1. Firstly, a feature selection method selects a number (m) of features from a dataset and returns a dataset which contains only the selected features (line 3 in Algorithm 1). We then run 10-fold cross-validation on the dataset five times (lines 4–6) and take the average accuracy (line 7). Because the performance of a feature selection method might vary greatly depending on the number of features used, we repeat the above procedure using from 1 up to n features (i.e., we repeat the above procedure n times, line 1) and take the average accuracy (line 9). In this paper, we set $n=20$ for those datasets with less than 100 features, and $n=50$ for other datasets. These settings are chosen so that each dataset is represented by a sufficient number of features.

Algorithm 1. Measuring the overall performance of a feature selection method

```
Inputs: dataset $D$, feature selection method $F$, machine learning algorithm $L$, maximum number of features used $n$
Outputs: The overall performance $perf$
1:  for $m=1$ to $n$ do
2:      $acc[m]=0$;
3:      $D'=\text{select\_features}(D,F,m)$;
4:      for $i=1$ to 5 do
5:          $acc[m]=acc[m]+\text{10fold\_cross\_val}(D',L)$;
6:      end for
7:      $acc[m]=acc[m]/5$;
8:  end for
9:  $perf=\text{average}(acc)$;
```

There are two issues considered in Algorithm 1 for fair comparison of different mutual information based feature selection methods. (1) Each time evaluating the performances of different approaches, the same number of features (k features) should be selected for comparing classification accuracies. (2) For different number of features k, the approaches' performances may vary. Therefore, for fair comparison, we conduct n times' comparisons with k increasing from 1 to n. Then the final comparison is based on the averaged performance over n times' comparisons.

Table 3 shows the performance of different feature selection methods when using SVM as a classifier. Similarly, Table 4 shows the performance when using NB as a classifier.[2] The last row of Tables 3 and 4 shows the average performance over all the datasets. For comparison, the last two columns in Table 2 show the average accuracy of SVM and NB when using all the features (i.e., without using any feature selection methods). The experimental results show that mutual information-based feature selection is able to select discriminative features, judging by the accuracy of the classifiers trained using the selected features, which for some datasets (e.g., *Promoters*, *Madelon*, *Arrhythmia*) are comparable or better than the classifiers trained using all features.

Fig. 2 shows the margins between the performance of different methods and that of the baseline (maxRel) method when using SVM as a classifier. Using NB displays similar patterns and therefore it is not shown. From Fig. 2, we make the following observations:

- maxRel is inferior to the other methods. Unlike maxRel, the other methods also take into consideration the redundancy factor. This shows that *both relevance and redundancy factors are important*.
- There is no single method that performs best on all the datasets, although CMIM and MID tend to perform better than the other methods.
- There is no clear winner between CMIM and MID. This disproves the belief that only class-dependent correlation matters for feature selection. Therefore, *both class-independent and class-dependent correlation are significant for feature selection*. The comparison between DIC and DDC also justifies this.
- Avg-CMIM performs equally or better than CMIM and CIFE. Avg-CMIM differs from CMIM in that it uses pairwise redundancy $R_y(x_m,x_i)$ over all the previously selected features $\{x_1,\ldots,x_{m-1}\}$, and it differs from CIFE in that it takes the average, instead of the sum, of all the pairwise redundancy. This means that *average pairwise redundancy over all the previously selected features is suitable for modelling redundancy factor*.

[1] With parameter $C=1$.

[2] The last column in Tables 3 and 4 shows the performance of the proposed method, which will be discussed in Section 4.

Table 3
Performance (mean accuracy ± std. dev.) of different feature selection methods when using SVM as a classifier, measured according to Algorithm 1. The number next to each dataset is the maximum number of features used (i.e., variable n in Algorithm 1).

Datasets		maxRel	CMIM [6]	CIFE [7]	Avg-CMIM	MID [2]	MIQ [8]	DDC [11]	DIC [11]	Proposed
Promot	(20)	81.70 ± 0.53	81.18 ± 0.64	78.08 ± 1.14	81.20 ± 0.71	81.62 ± 0.79	81.45 ± 0.72	83.96 ± 0.03	83.37 ± 0.02	83.31 ± 1.12
Spambs	(20)	85.46 ± 0.43	86.29 ± 0.04	84.03 ± 0.04	85.96 ± 0.04	85.84 ± 0.05	82.49 ± 0.03	82.90 ± 0.02	80.18 ± 0.02	86.61 ± 0.01
Splice	(20)	81.39 ± 0.06	80.78 ± 0.07	79.31 ± 0.06	81.23 ± 0.05	81.23 ± 0.03	78.07 ± 0.05	81.74 ± 0.06	81.55 ± 0.06	81.25 ± 0.04
OptDgts	(20)	79.73 ± 0.02	80.24 ± 0.05	81.56 ± 0.04	80.94 ± 0.05	80.56 ± 0.05	82.10 ± 0.03	80.64 ± 0.20	79.89 ± 0.19	81.44 ± 0.05
Musk1	(50)	74.73 ± 0.13	75.72 ± 0.38	73.55 ± 0.49	75.73 ± 1.13	77.89 ± 0.30	74.64 ± 0.32	74.93 ± 0.04	76.57 ± 0.04	78.03 ± 0.24
Musk2	(50)	90.28 ± 0.01	90.47 ± 0.01	90.77 ± 0.01	90.79 ± 0.03	90.41 ± 0.02	89.82 ± 0.03	91.15 ± 0.02	88.67 ± 0.03	90.63 ± 0.01
Arrhyth	(50)	68.62 ± 0.16	67.96 ± 0.26	68.03 ± 0.16	69.79 ± 0.92	69.10 ± 0.24	66.31 ± 0.16	66.10 ± 0.02	66.82 ± 0.02	69.78 ± 0.14
Madelon	(50)	61.30 ± 0.06	61.27 ± 0.10	61.70 ± 0.23	61.44 ± 0.10	61.26 ± 0.08	60.79 ± 0.18	61.37 ± 0.01	60.96 ± 0.01	61.56 ± 0.13
Multi-fts	(50)	91.78 ± 0.07	93.94 ± 0.08	93.30 ± 0.09	93.39 ± 0.52	93.68 ± 0.08	90.30 ± 0.06	93.95 ± 0.13	93.29 ± 0.12	94.51 ± 0.05
Advertis	(50)	94.99 ± 0.02	95.74 ± 0.04	95.84 ± 0.03	95.46 ± 0.05	95.58 ± 0.02	95.25 ± 0.01	93.94 ± 0.01	95.10 ± 0.01	95.80 ± 0.02
Average		81.00 ± 10.55	81.36 ± 11.00	80.62 ± 10.97	81.59 ± 10.62	81.72 ± 10.65	80.12 ± 10.73	81.07 ± 11.03	80.64 ± 10.74	82.29 ± 10.68

Table 4
Performance (mean accuracy ± std. dev.) of different feature selection methods when using NB as a classifier, measured according to Algorithm 1. The number next to each dataset is the maximum number of features used (i.e., variable n in Algorithm 1).

Datasets		maxRel	CMIM [6]	CIFE [7]	Avg-CMIM	MID [2]	MIQ [8]	DDC [11]	DIC [11]	Proposed
Promot	(20)	92.80 ± 0.32	93.10 ± 0.57	91.29 ± 0.52	93.11 ± 0.38	93.48 ± 0.34	93.42 ± 0.50	91.72 ± 0.05	92.91 ± 0.06	94.92 ± 0.34
Spambs	(20)	82.99 ± 0.06	84.24 ± 0.04	70.59 ± 0.11	84.60 ± 0.05	85.23 ± 0.06	73.19 ± 0.02	84.81 ± 0.02	83.92 ± 0.02	80.47 ± 0.02
Splice	(20)	91.10 ± 0.01	91.17 ± 0.01	90.67 ± 0.01	91.20 ± 0.01	91.20 ± 0.01	89.15 ± 0.02	89.72 ± 0.08	90.05 ± 0.08	91.23 ± 0.01
OptDgts	(20)	76.09 ± 0.10	76.94 ± 0.10	76.58 ± 0.10	77.03 ± 0.05	78.21 ± 0.09	80.15 ± 0.04	77.95 ± 0.18	78.40 ± 0.17	78.72 ± 0.07
Musk1	(50)	73.77 ± 0.20	75.82 ± 0.46	69.08 ± 0.33	74.80 ± 0.22	76.91 ± 0.29	68.14 ± 0.44	72.40 ± 0.02	72.71 ± 0.02	76.57 ± 0.17
Musk2	(50)	83.48 ± 0.02	79.64 ± 0.01	65.28 ± 0.07	79.91 ± 0.01	79.80 ± 0.02	73.64 ± 0.04	63.73 ± 0.06	80.53 ± 0.04	78.04 ± 0.02
Arrhyth	(50)	65.20 ± 0.34	68.76 ± 0.35	64.72 ± 0.17	66.88 ± 0.44	68.37 ± 0.17	65.12 ± 0.19	65.86 ± 0.02	65.66 ± 0.02	68.92 ± 0.31
Madelon	(50)	60.24 ± 0.11	61.50 ± 0.16	60.15 ± 0.10	60.36 ± 0.08	60.76 ± 0.12	58.44 ± 0.29	61.00 ± 0.01	60.39 ± 0.01	60.52 ± 0.11
Multi-fts	(50)	88.04 ± 0.03	92.59 ± 0.08	91.05 ± 0.06	91.43 ± 0.05	91.65 ± 0.06	89.01 ± 0.06	92.33 ± 0.13	91.71 ± 0.12	92.94 ± 0.04
Advertis	(50)	94.43 ± 0.02	95.68 ± 0.02	94.93 ± 0.04	94.98 ± 0.02	95.29 ± 0.02	94.83 ± 0.02	95.43 ± 0.02	95.07 ± 0.01	95.63 ± 0.02
Average		80.81 ± 11.73	81.94 ± 11.43	77.43 ± 13.27	81.43 ± 11.77	82.09 ± 11.44	78.51 ± 12.72	79.62 ± 12.90	81.14 ± 11.93	81.80 ± 11.78

- From the efficiency point-of-view, maxRel method, being the simplest method, is the most efficient. CMIM, CIFE, Avg-CMIM and DDC are the least efficient, due to the computation of class-dependent correlation. MID and MIQ are more efficient, as the redundancy factor only involves two variables.

As indicated in [21], the test error becomes over optimistic when the entire dataset is used for feature selection. Although this may result in bias for estimated prediction error (or classification performance), its impact on feature selection becomes much lower due to the following reasons. (1) It is also mentioned in [21] that such bias is offset to some extent by the fact that the training dataset is a part of the whole dataset. Especially, in this work the training data cover most (90%) of the entire dataset, as 10-fold cross-validation is used in our experiment, where the training data is enough to cover the diversity of the entire dataset. Thus, the training dataset and the entire dataset have almost the same underlying distributions. And this makes the feature selections conducted on the training dataset and the entire dataset output similar sets of features. (2) Although the estimated classification performance may become biased (too optimistic), such bias does not favor any specific feature selection method. From this point of view, the comparison between different feature selection methods is still fair, as long as these methods are performed on the same dataset. Empirical study shows that for the proposed method feature selections performed on the whole dataset and the training dataset output very similar sets of features (over all about 95% of the selected features are in common in our experiments).

4. Proposed method

The observations from previous section suggest that a good feature selection method should:

- include both relevance and redundancy factors;
- model redundancy using both class-dependent and class-independent correlation;
- take the average redundancy over all the previously selected features.

Based on this analysis and the general formulation of mutual information-based feature selection (Eq. (7)), we derive the following formulation:

$$x_m = \operatorname*{argmax}_x \left\{ \gamma_m I(y;x) - \frac{1}{m-1} \sum_i^{m-1} \{\eta R_y(x,x_i) + (1-\eta) I(x;x_i)\} \right\}, \tag{15}$$

where γ_m is a weighting coefficient ($\gamma_m \geq 0$) for relevance factor at iteration m, and η is a coefficient ($0 \leq \eta \leq 1$) for redundancy factor which is formulated as a convex combination of both class-dependent and class-independent correlation.

The convex combination makes sure that both class-dependent and class-independent correlations contribute to the objective function in a regularized way. Both $R_y(x,x_i)$ and $I(x;x_i)$ contribute to the objective function negatively. In addition, the balance between relevance and redundancy can be controlled by γ_m since the coefficients of class-dependent and class-independent correlations sum to one.

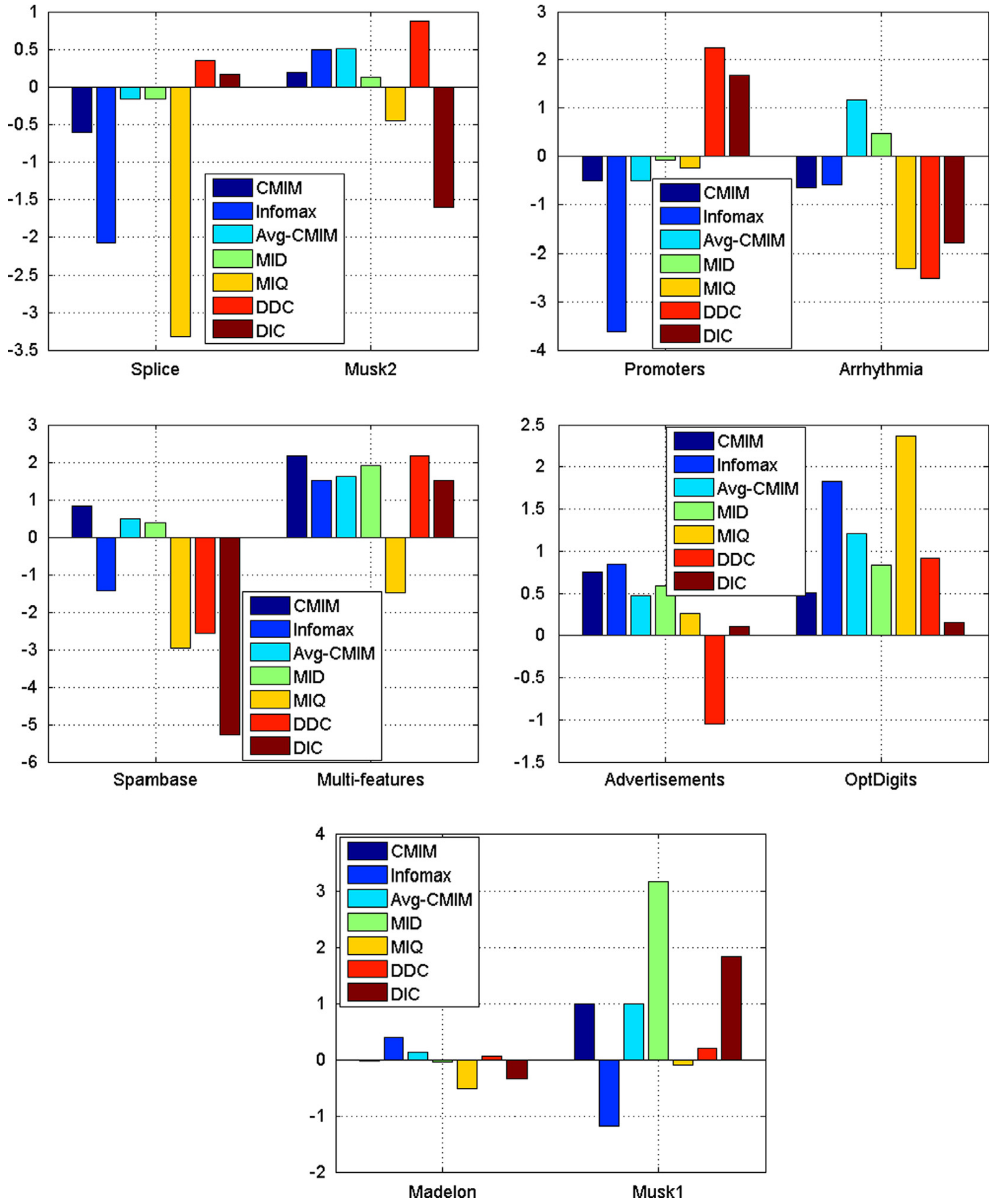

Fig. 2. Margins between the performance (%) of different feature selection methods and that of the baseline (maxRel) method on different UCI datasets when using SVM as a classifier. The performance of each method is according to the values in Table 3. A higher margin indicates a better performance over the baseline method.

By substituting $R_y(x,x_i)$ with Eq. (3), the above equation can be written as

$$x_m = \underset{x}{\operatorname{argmax}}\left\{\lambda_m I(y;x) + \frac{1}{m-1}\sum_{i}^{m-1}\{\eta I(y;x|x_i) - (1-\eta)I(x;x_i)\}\right\}, \qquad (16)$$

where $\lambda_m = \gamma_m - \eta$. Eq. (16) is the formulation of the proposed feature selection method. How to determine the values for η and λ_m is explained below.

In Eq. (15), if we expand $I(x;x_i)$ by Eq. (12), the redundancy factor becomes

$$\eta R_y(x,x_i) + (1-\eta)I(x;x_i) = R_y(x,x_i) + (1-\eta)I(x;x_i|y). \qquad (17)$$

The above equation reveals that η is actually a discount coefficient for $I(x;x_i|y)$, which is part of class-independent correlation that is often considered irrelevant for feature selection. This means that the proposed method considers both $R_y(x,x_i)$ and $I(x;x_i|y)$, but it puts a larger weight on the former. This is intuitively correct, since the latter is not directly related to class y (see Fig. 1), but it may still affect feature selection.

Selecting the value for λ_m is more difficult for the following reasons:

- λ_m is not intuitive and it can have any value.
- A value of λ_m which is well-suited for a particular dataset may not be suitable for other datasets.

We propose an adaptive setting for λ_m based on the following reasoning. In early iterations where only few features have been selected, there is still a large potential for gaining new information about the class, unlike in later iterations where most information about the class has been gained from previously selected features.

This means that we should focus on the relevance factor (i.e., using a larger value of λ_m) in early iterations, and focus on the redundancy factor (i.e., using a smaller value of λ_m) in later iterations.

The amount of information about class y that has not been gained from the features in S_{m-1} is measured by $H(y|S_{m-1})$ which is defined as

$$H(y|S_{m-1}) = H(y) - \sum_{i}^{m-1} I(y;x_i) + R_y(x_{m-1}, S_{m-2}),$$

where $H(y)$ is the information entropy [12] of y. We approximate $H(y|S_{m-1})$, which is difficult to calculate due to $R_y(x_{m-1}, S_{m-2})$, as follows:

$$H(y|S_{m-1}) \approx \tilde{H}(y|S_{m-1}) = H(y) - \sum_{i}^{m-1} I(y;x_i).$$

The shift in focus from the relevance factor to the redundancy factor as discussed earlier is possible when λ_m is proportional to the percentage of information about y that has not been gained from S_{m-1},

$$\lambda_m \propto \frac{\tilde{H}(y|S_{m-1})}{H(y)} = 1 - \frac{\sum_i^{m-1} I(y;x_i)}{H(y)}.$$

However, $\tilde{H}(y|S_{m-1})$ may significantly underestimate $H(y|S_{m-1})$ especially during later iterations, and it may even have a negative value. As such, we take the exponential value and set the lower-bound with a constant c where $c \geq 0$,

$$\lambda_m = \max\left\{c, \exp\left(1 - \frac{\sum_i^{m-1} I(y;x_i)}{H(y)}\right)\right\}. \tag{18}$$

In other words, c is a minimum weighting coefficient for the relevance factor. Selecting a value for c is easier than directly selecting a value for λ_m.

In summary, the proposed method (Eq. (16)) has two parameters. The first parameter, η, is a discount coefficient for the part of class-independent correlation that is not related to the class variable. The second parameter, c, is a minimum weighting coefficient for the relevance factor. In this paper we set $c = 0.4$, and it works well despite the different datasets used in the experiments. The value for η is chosen from {0.2, 0.4, 0.6, 0.8} that performs best according to two-fold cross-validation. In general, the parameters c and η can be automatically selected by using grid search technique. Similar to the hyper-parameter optimization for SVM with RBF kernel [22], a subset of the space is firstly designed by discretizing parameter values. For instance, the subset of the parameter space can be defined by the cross product of the parameter values. Then, exhaustive search can be performed on a held-out validation dataset with the guidance of classification accuracy. Finally, the parameter values with the highest accuracy are selected for the algorithm.

Table 5

Margins (mean ± std. dev.) between the performance of the proposed method and other methods when using SVM. The performance of each method is according to the values in Table 3. The statistical significance of the difference between two approaches is tested by one-sided paired *t*-test.

Datasets	maxRel	CMIM [6]	CIFE [7]	Avg-CMIM	MID [2]	MIQ [8]	DDC [11]	DIC [11]
Promoters	0.86 ± 0.70[a]	1.42 ± 0.41[a]	5.41 ± 1.41[a]	1.36 ± 0.92[a]	0.84 ± 1.12	1.28 ± 0.94[a]	−0.65 ± 0.3[b]	−0.06 ± 0.5
Spambase	0.97 ± 0.02[a]	0.33 ± 0.03[a]	2.58 ± 0.06[a]	0.65 ± 0.03[a]	0.76 ± 0.03[a]	4.11 ± 0.03[a]	3.71 ± 0.21[a]	6.43 ± 0.23[a]
Splice	−0.13 ± 0.05[b]	0.46 ± 0.04[a]	1.84 ± 0.04[a]	0.02 ± 0.04	0.02 ± 0.02	3.19 ± 0.07[a]	−0.49 ± 0.03[b]	−0.3 ± 0.05
OptDigits	1.69 ± 0.06[a]	1.19 ± 0.03[a]	−0.09 ± 0.05[b]	0.52 ± 0.02[a]	0.85 ± 0.04[a]	−0.64 ± 0.04[b]	0.8 ± 0.63[a]	1.55 ± 0.55[a]
Musk1	3.30 ± 0.17[a]	2.15 ± 0.16[a]	4.31 ± 0.34[a]	1.93 ± 0.29[a]	0.09 ± 0.34	3.49 ± 0.34[a]	3.1 ± 0.22[a]	1.46 ± 0.3[a]
Musk2	0.37 ± 0.03[a]	0.16 ± 0.02[a]	−0.14 ± 0.01[b]	−0.13 ± 0.02[b]	0.22 ± 0.01[a]	0.84 ± 0.01[a]	−0.52 ± 0.23[b]	1.96 ± 0.21[b]
Arrhythmia	1.17 ± 0.26[a]	0.27 ± 1.47	1.93 ± 0.31[a]	−0.47 ± 0.25[b]	0.63 ± 0.21[a]	3.34 ± 0.20[a]	3.68 ± 0.11[a]	2.96 ± 0.31[a]
Madelon	0.07 ± 0.06[a]	0.09 ± 0.11	−0.15 ± 0.22	−0.01 ± 0.10	0.06 ± 0.06	0.89 ± 0.14[a]	0.19 ± 0.13[a]	0.6 ± 0.2[a]
Multi-features	2.84 ± 0.07[a]	0.56 ± 0.06[a]	1.27 ± 0.03[a]	0.93 ± 0.06[a]	0.85 ± 0.05[a]	4.17 ± 0.03[a]	0.56 ± 0.38[a]	1.22 ± 0.26[a]
Advertisements	0.77 ± 0.03[a]	0.03 ± 0.04	−0.03 ± 0.04	0.30 ± 0.05[a]	0.16 ± 0.03[a]	0.58 ± 0.05[a]	1.86 ± 0.52[a]	0.7 ± 0.12[a]
#win/tie/loss	9/0/1	7/3/0	6/2/2	6/2/2	6/4/0	9/0/1	7/0/3	7/2/1

The three numbers at the bottom of each column are the number of datasets for which the proposed method performs better/equally/worse than the compared method.

[a] The proposed method achieves a statistically significant better performance than the compared method according to the one-sided paired *t*-test at 5% level of significance.

[b] A worse performance by the proposed method according to the *t*-test.

Table 6

Margins (mean ± std. dev.) between the performance of the proposed method and other methods when using NB. The performance of each method is according to the values in Table 4. The statistical significance of the difference between two approaches is tested by one-sided paired *t*-test.

Datasets	maxRel	CMIM [6]	CIFE [7]	Avg-CMIM	MID [2]	MIQ [8]	DDC [11]	DIC [11]
Promoters	2.12 ± 0.31[a]	1.82 ± 0.39[a]	3.62 ± 0.55[a]	1.80 ± 0.25[a]	1.44 ± 0.25[a]	1.50 ± 0.32[a]	3.2 ± 0.41[a]	2.01 ± 0.17[a]
Spambase	−2.52 ± 0.05[b]	−3.77 ± 0.05[b]	9.88 ± 0.12[a]	−4.14 ± 0.05[b]	−4.76 ± 0.06[b]	7.27 ± 0.03[a]	−4.34 ± 0.08[b]	−3.45 ± 0.1[b]
Splice	0.13 ± 0.01[a]	0.06 ± 0.01[a]	0.56 ± 0.02[a]	0.03 ± 0.01[a]	0.02 ± 0.01[a]	2.07 ± 0.01[a]	1.51 ± 0.22[a]	1.18 ± 0.03[a]
OptDigits	2.62 ± 0.07[a]	1.77 ± 0.07[a]	2.14 ± 0.07[a]	1.68 ± 0.06[a]	0.51 ± 0.07[a]	−1.44 ± 0.07[b]	0.77 ± 0.3[a]	0.32 ± 0.11[a]
Musk1	2.79 ± 0.13[a]	0.74 ± 0.35[a]	7.49 ± 0.41[a]	1.76 ± 0.20[a]	−0.34 ± 0.23[b]	8.43 ± 0.36[a]	4.17 ± 0.11[a]	3.86 ± 0.07[a]
Musk2	−5.43 ± 0.03[b]	−1.59 ± 0.01[b]	12.77 ± 0.05[a]	−1.87 ± 0.01[b]	−1.75 ± 0.02[b]	4.40 ± 0.04[a]	14.31 ± 0.2[a]	−2.49 ± 0.09[a]
Arrhythmia	3.72 ± 0.40[a]	0.15 ± 0.43	4.20 ± 0.25[a]	2.04 ± 0.50[a]	0.55 ± 0.32[a]	3.79 ± 0.25[a]	3.06 ± 0.31[a]	3.26 ± 0.2[a]
Madelon	0.28 ± 0.05[a]	−0.98 ± 0.12[b]	0.37 ± 0.15[a]	0.16 ± 0.05[a]	−0.24 ± 0.15[b]	2.08 ± 0.30[a]	−0.48 ± 0.22[b]	0.13 ± 0.03
Multi-features	4.90 ± 0.03[a]	0.35 ± 0.06[a]	1.89 ± 0.04[a]	1.51 ± 0.02[a]	1.29 ± 0.03[a]	3.93 ± 0.08[a]	0.61 ± 0.32[a]	1.23 ± 0.13[a]
Advertisements	1.20 ± 0.02[a]	−0.04 ± 0.01[b]	0.70 ± 0.05[a]	0.65 ± 0.01[a]	0.34 ± 0.01[a]	0.80 ± 0.03[a]	0.2 ± 0.36[a]	0.56 ± 0.17[a]
#win/tie/loss	8/0/2	5/1/4	10/0/0	8/0/2	6/0/4	9/0/1	8/0/2	8/1/1

The three numbers at the bottom of each column are the number of datasets for which the proposed method performs better/equally/worse than the compared method.

[a] The proposed method achieves a statistically significant better performance than the compared method according to the one-sided paired *t*-test at 5% level of significance.

[b] Worse performance by the proposed method according to the *t*-test.

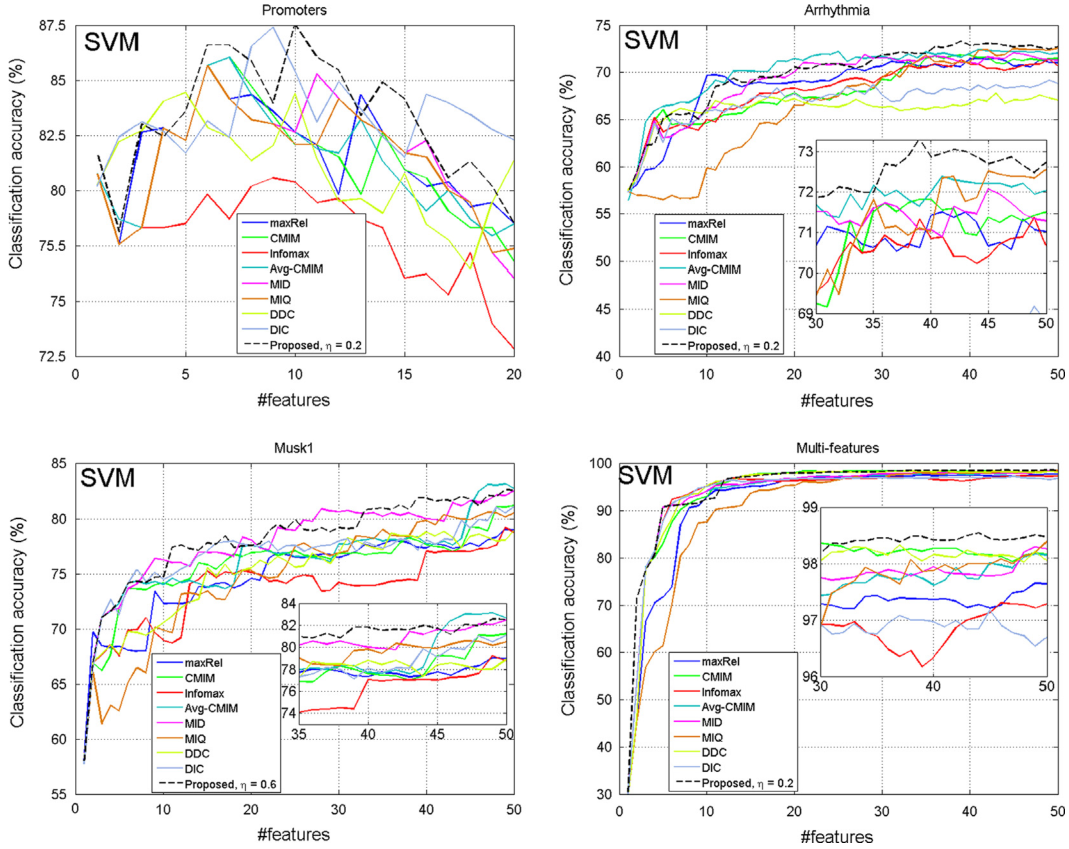

Fig. 3. Average accuracy (from five repetitions of 10-fold cross-validation) of SVM when using different feature selection methods and varying number of features on four UCI datasets [1]: *Promoters*, *Arrhythmia*, *Musk*1, and *Multi-features*.

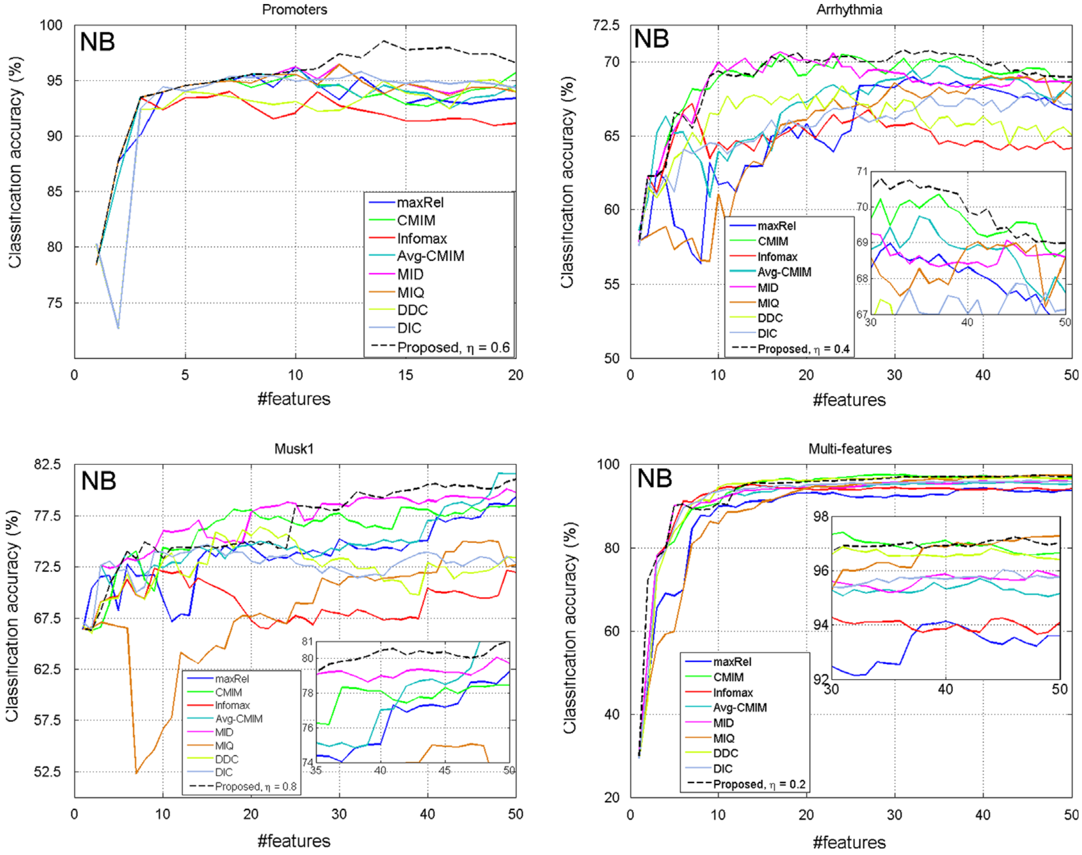

Fig. 4. Average accuracy (from five repetitions of 10-fold cross-validation) of naive Bayes (NB) when using different feature selection methods and varying number of features on four UCI datasets [1]: *Promoters*, *Arrhythmia*, *Musk*1, and *Multi-features*.

5. Experiments

5.1. UCI datasets

We repeat the experimental procedure in Section 3 with the proposed feature selection method. The last column in Tables 3 and 4 shows the performance of the proposed method when using SVM and NB as a classifier, respectively. We also measure the margins between the performance of the proposed method and that of other methods for each dataset when using SVM (see Table 5) and NB (see Table 6). The last row of Tables 5 and 6 shows the number of datasets for which the proposed method performs better, equally, or worse than each of the compared methods according to a statistical hypothesis testing, namely the one-sided paired *t*-test at 5% level of significance.

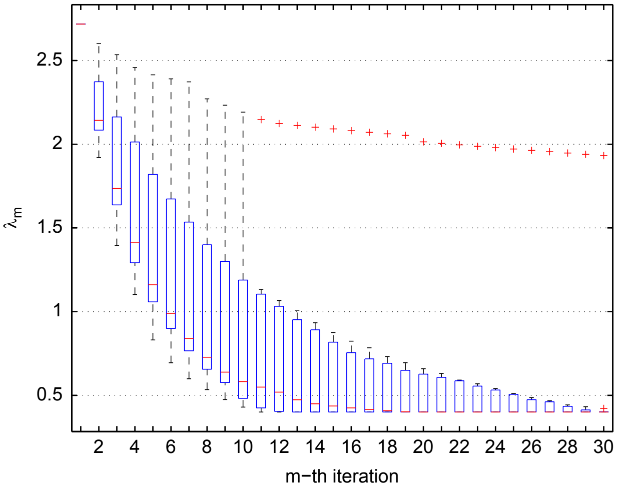

Fig. 5. Boxplot for λ_m when selecting m-th features from ten UCI datasets.

When SVM is used as a classifier, the proposed method has the best average performance compared to other methods (see Table 3). Table 4 shows that the overall proposed method is competitive to CMIM and MID when NB is used, and for many datasets it outperforms both. Tables 5 and 6 show that the proposed method outperforms the other methods for most datasets. So far, the performance of a feature selection method has been based on the overall performance when using from 1 up to n features (see Algorithm 1). Figs. 3 and 4 show the performance of SVM and naive Bayes when using different feature selection methods with varying number of features for four datasets: *Promoters*, *Arrhythmia*, *Musk*1, and *Multi-features*. The four datasets are chosen because they cover a diverse range of characteristics, in terms of the number of features and the number of examples.

Figs. 3 and 4 show that the proposed method (dashed lines) tends to outperform the other methods, especially during later iterations when many features have already been selected (see the insets in Figs. 3 and 4). This indicates the importance of considering both class-dependent and class-independent correlation during later iterations, when the redundancy of the selected features is potentially large.

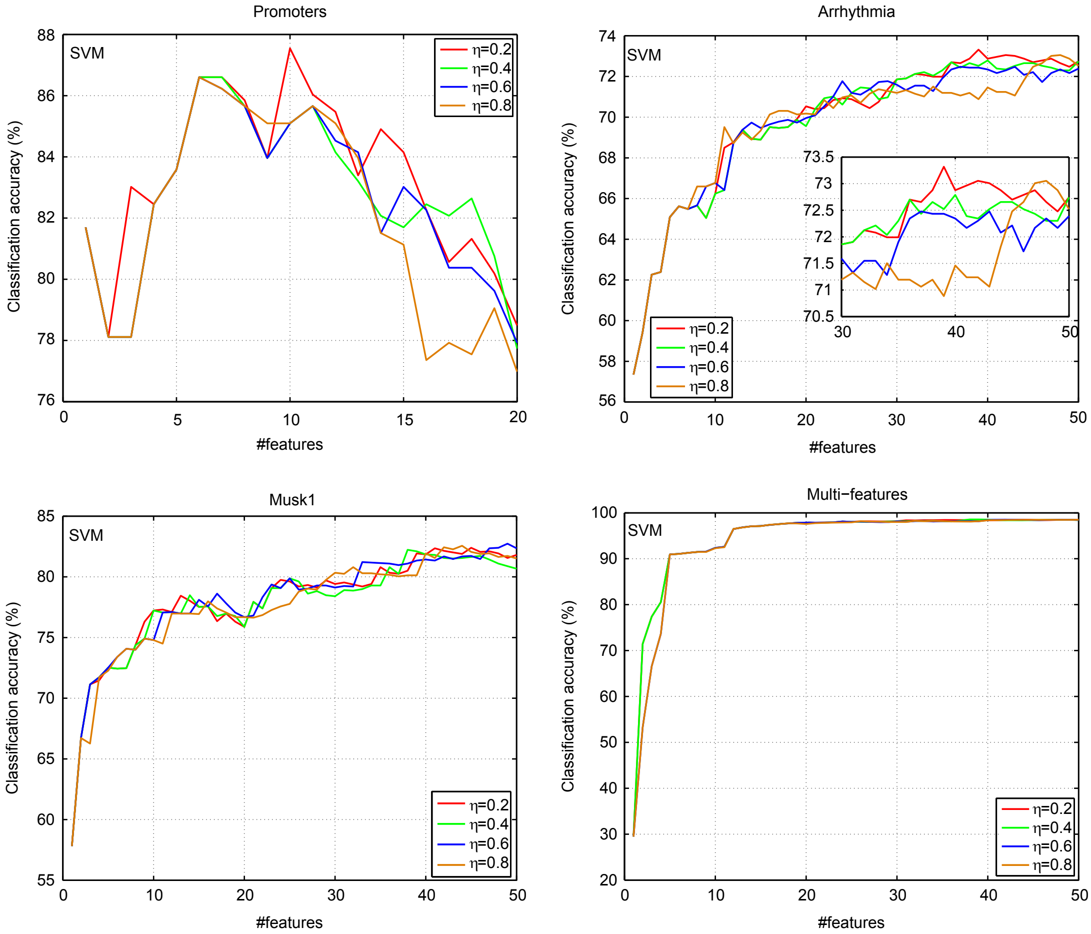

Fig. 6. Average accuracy of SVM when using the proposed feature selection method with different values for $\eta \in \{0.2, 0.4, 0.6, 0.8\}$, on four UCI datasets [1]: *Promoters*, *Arrhythmia*, *Musk*1, and *Multi-features*.

Fig. 5 shows the boxplot for the values of λ_m (Eq. (18)) in the proposed method. As shown in the figure, the value of λ_m gradually decreases with more iterations until it reaches the minimum limit (which is 0.4 in this paper). This adaptive setting for λ_m has proved to be effective, judging by the good performance of the proposed method in the experiments.

5.2. Different values for parameter η

Figs. 6 and 7 show the effects of varying η on the performance of SVM and naive Bayes with the proposed feature selection method. From the figures, we can see that changing η has a noticeable impact for some datasets, such as *Promoters* and *Musk*1 with NB, and *Arrhythmia* with SVM. The impact is more noticeable during later iterations. This is consistent with previous observation that the advantage of the proposed method is more pronounced during later iterations.

Recall that dependency is a combination of two factors: relevance and redundancy. One possible explanation for the good performance of the proposed method is that because class-independent correlation overestimates class-dependent correlation, incorporating class-independent correlation in the proposed method penalizes the dependency by imposing a larger weight on the redundancy factor (see Eq. (15)), and this avoids selecting a set of features that 'overfit' the training data. Tables 7 and 8 show the overall performance of the proposed method when using different η. Interestingly, $\eta = 0.2$ (which means a larger weight on class-independent correlation than on class-dependent correlation) is effective for most datasets when either SVM or NB is used as a classifier. This reiterates the importance of class-independent correlation for feature selection.

5.3. Object recognition

Object recognition is an active research topic in computer vision. Its task is to determine whether an image contains occurrences of certain object categories (e.g., cars). A popular approach in recent years is based on local features [23]

Table 7
Performance (mean accuracy ± std. dev.) of the proposed method when using SVM as a classifier and different values for η. Shown underlined is the best performance for each dataset.

Datasets	$\eta = 0.2$	$\eta = 0.4$	$\eta = 0.6$	$\eta = 0.8$
Promoters	83.31 ± 1.12	82.72 ± 1.29	82.55 ± 1.00	81.94 ± 0.82
Spambase	86.46 ± 0.02	86.45 ± 0.02	86.44 ± 0.04	86.61 ± 0.01
Splice	81.17 ± 0.04	81.24 ± 0.04	81.21 ± 0.06	81.20 ± 0.06
OptDigits	81.44 ± 0.05	81.29 ± 0.06	81.25 ± 0.06	81.42 ± 0.03
Musk1	77.96 ± 0.29	77.73 ± 0.31	78.03 ± 0.24	77.62 ± 0.28
Musk2	90.32 ± 0.02	90.33 ± 0.03	90.42 ± 0.02	90.63 ± 0.01
Arrhythmia	69.78 ± 0.14	69.68 ± 0.19	69.66 ± 0.23	69.55 ± 0.26
Madelon	61.43 ± 0.12	61.52 ± 0.18	61.56 ± 0.13	61.48 ± 0.12
Multi-features	94.51 ± 0.05	94.47 ± 0.05	93.73 ± 0.05	93.72 ± 0.06
Advertisements	95.80 ± 0.02	95.77 ± 0.03	95.71 ± 0.02	95.71 ± 0.02

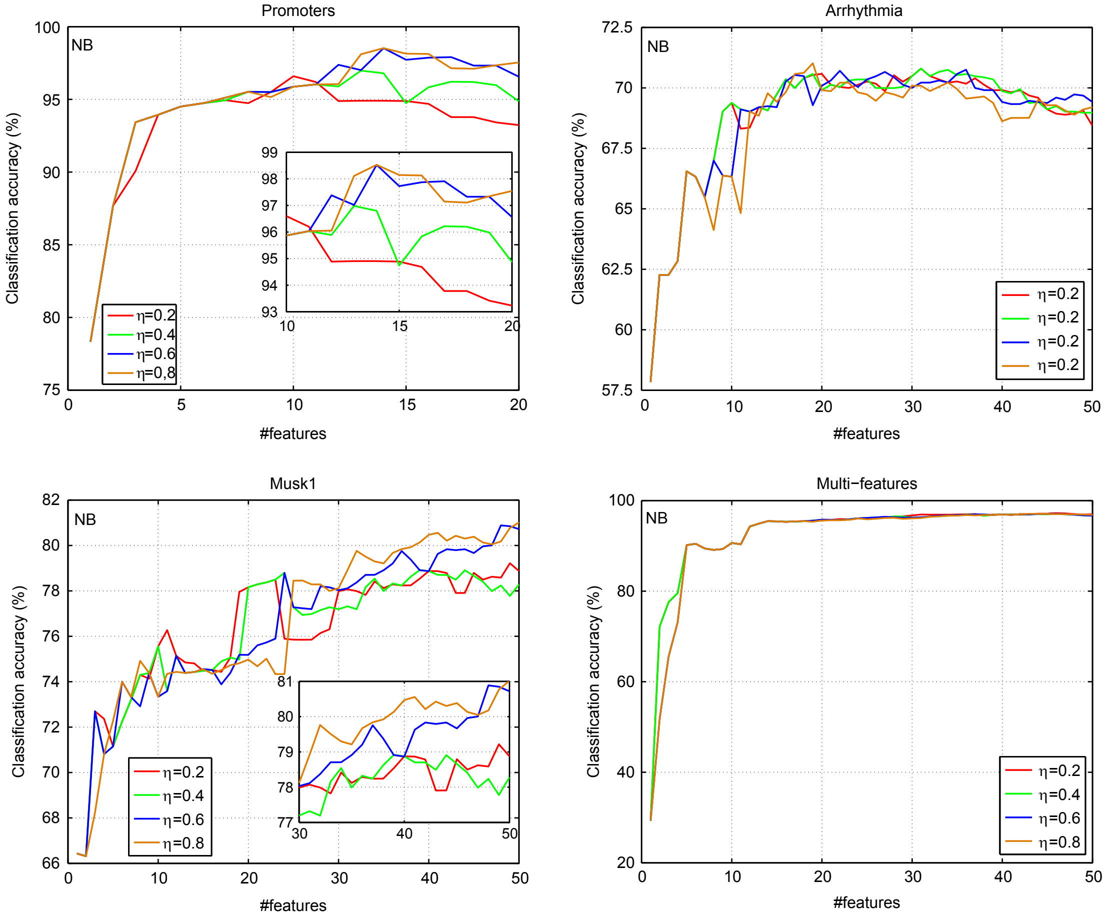

Fig. 7. Average accuracy of naive Bayes (NB) when using the proposed feature selection method with different values for $\eta \in \{0.2, 0.4, 0.6, 0.8\}$, on four UCI datasets [1]: *Promoters*, *Arrhythmia*, *Musk*1, and *Multi-features*.

that capture the salient regions (corners or blobs) in an image, thus an image is viewed as a set of salient regions. The descriptions of the salient regions are used as the features. This approach often involves a large number of features, therefore it is suitable for testing feature selection methods.

In this experiment we use MILES (Multiple-Instance Learning via Embedded Instance Selection) algorithm [24] for object recognition which consists of the following three steps:

1. Collect the descriptions of all the salient regions from all the training images, $C=\{\mathbf{x}^{(k)}:k=1,\ldots,t\}$ where t is the total number of salient regions detected from training images.
2. Represent each (training and test) image $\mathbf{B}$ as a t-dimensional feature vector, where its value at the k-th dimension is $\max_{\mathbf{x}_j\in\mathbf{B}}\exp(-\|\mathbf{x}_j-\mathbf{x}^{(k)}\|^2/\sigma^2)$, where $\|\cdot\|$ is an Euclidean distance, $\mathbf{x}^{(k)}\in C$, and σ is a predetermined parameter.
3. Apply L1-norm SVM with the new representation of images (i.e., examples) to learn the object concept.

Because each image might have many salient regions and different images have different sets of salient regions, the total number of salient regions (i.e., features) involved is usually very large, and this often creates a problem for storage and processing as acknowledged in [24]. A simple solution to this problem is to use only a subset of the original feature set. Here we investigate the effectiveness of various feature selection methods (maxRel, CMIM, MID, and the proposed method) when they are applied to object recognition which involves a large number of features. The other methods (MIQ, CIFE, and Avg-CMIM) are not considered as they did not perform well in Sections 3 and 5.1. maxRel is also chosen as it is the baseline method in this experiment.

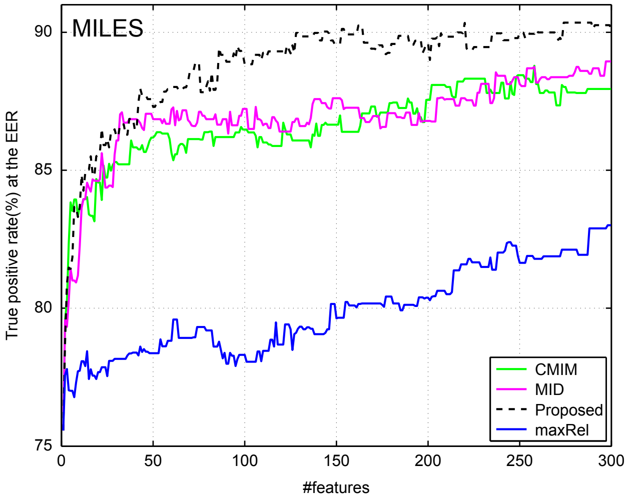

Fig. 9. Performance (true positive rate (%) at the equal-error-rate) of MILES with different feature selection methods and varying number of features used.

Table 8
Performance (mean accuracy ± std. dev.) of the proposed method when using NB as a classifier and different values for η. Shown underlined is the best performance for each dataset.

Datasets	$\eta=0.2$	$\eta=0.4$	$\eta=0.6$	$\eta=0.8$
Promoters	93.29 ± 0.50	94.20 ± 0.36	94.92 ± 0.34	94.92 ± 0.47
Spambase	80.47 ± 0.02	80.40 ± 0.02	80.27 ± 0.03	79.93 ± 0.04
Splice	91.22 ± 0.01	91.22 ± 0.01	91.23 ± 0.01	91.21 ± 0.01
OptDigits	78.72 ± 0.07	78.03 ± 0.08	77.67 ± 0.08	77.41 ± 0.05
Musk1	76.24 ± 0.36	76.20 ± 0.43	76.46 ± 0.29	76.57 ± 0.17
Musk2	76.43 ± 0.03	77.16 ± 0.03	77.48 ± 0.03	78.04 ± 0.02
Arrhythmia	68.85 ± 0.24	68.92 ± 0.31	68.79 ± 0.31	68.41 ± 0.29
Madelon	60.52 ± 0.11	60.49 ± 0.12	60.42 ± 0.10	60.32 ± 0.06
Multi-features	92.94 ± 0.04	92.86 ± 0.05	92.09 ± 0.06	92.04 ± 0.06
Advertisements	95.63 ± 0.02	95.56 ± 0.02	95.51 ± 0.02	95.47 ± 0.01

Table 9
Performance (true positive rate (%) at the equal-error-rate) of MILES algorithm [24] with different feature selection methods and number of features used. The last two columns show the average and the best performance of each method when using 300 features or less. Displayed in bold are the best performance in each category (column). For comparison, MILES with full feature set (with more than 10,000 features) achieves 94.5% on this dataset as reported in [24].

Algorithms	50	100	150	200	250	300	Mean	Max
MILES + maxRel	78.37	78.31	79.63	80.38	81.64	83.01	79.89	83.01
MILES + CMIM [6]	86.18	86.47	86.65	87.55	88.44	87.94	86.63	88.79
MILES + MID [2]	86.85	87.22	87.61	86.79	88.12	88.94	86.93	88.94
MILES + Proposed	**87.30**	**88.84**	**89.58**	**89.59**	**89.99**	**90.11**	**88.79**	**90.35**

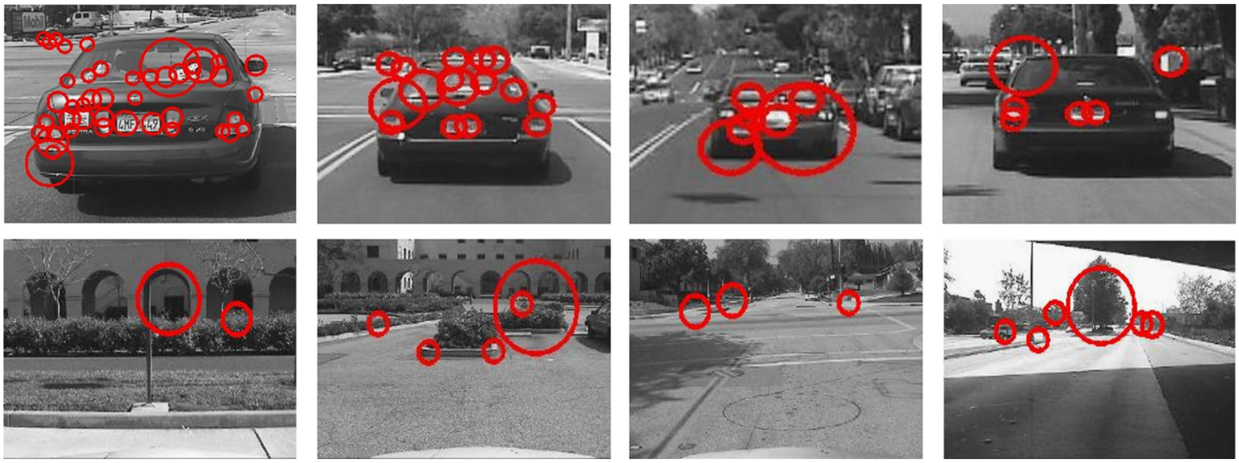

Fig. 8. Some sample images for object recognition with car labels (top row) and road background labels (bottom row). Each detected salient region is represented by a red circle in the images. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

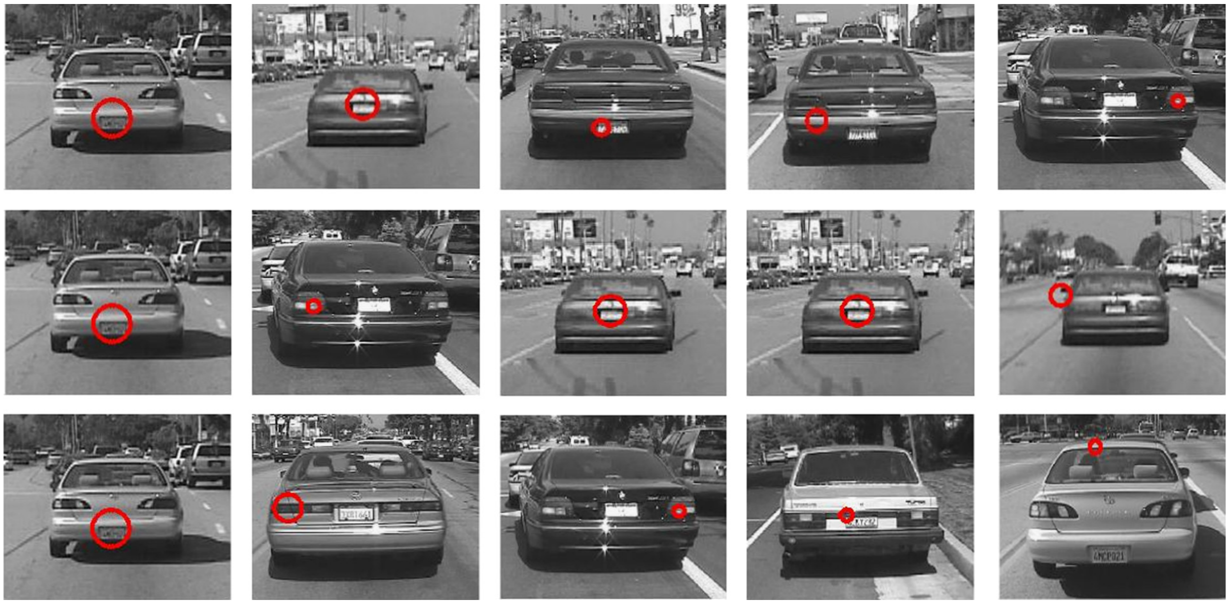

Fig. 10. The best five features (the best one being on the leftmost) selected by different feature selection methods: CMIM (first row), MID (second row), and the proposed method (last row).

We follow the experimental procedure and the parameter settings as those used in [24]. The training and test images are from Caltech cars dataset[3] which is widely used for object recognition. This dataset is also the most challenging for MILES as indicated in [24]. For training, there are 400 images which contain cars (cars images) and 685 images which depict roads without the presence of any cars (road background images). Some examples of the training images are shown in Fig. 8. For testing, there are also 400 car images and 685 road background images, but they are separate from the training images. The task is to determine for any given test image whether it contains any cars.

There are a total of 13,693 salient regions[4] detected from the training images (see Fig. 8). A square patch is extracted from each salient region and is resized to an 11×11 patch. The 121-dimensional vectors of pixel values from the 11×11 patches are then reduced to 15-dimensional vectors by applying Principal Component Analysis (PCA). The scale and the location (x, y coordinate) of salient regions add three additional dimensions, thus the description of each salient region is an 18-dimensional feature vector.

We compare the performance of different feature selection methods as follows. Firstly, each feature selection method is used to select the best 300 features (i.e., salient regions) from training images. Next, we apply MILES using increasing number of features, from 1 up to 300. We use at most 300 features as we consider them to be sufficient for this dataset. Following [24], we use the true positive rate (%) at the equal-error-rate (EER) point as the performance measure. EER is a point at which the false-rejection rate (i.e., 1 – true positive rate) is equal to the false-acceptance rate.

Fig. 9 plots the performance of different feature selection methods against the number of features used. We also plot the performance of the baseline (maxRel) method which is the simplest mutual information-based feature selection method. Table 9 summarizes the performance of different feature selection methods. From Fig. 9, we can see that the proposed method consistently outperforms the other methods, and it achieves the best average and maximum performance (see Table 9). Meanwhile, there is no clear winner between CMIM and MID, although MID achieves a slightly better average and maximum performance than CMIM. From the table, we can also see that the margin between the performance of the proposed method and that of the baseline method is significant (~10%). The best performance in Fig. 9 is 90.35%, achieved by the proposed method when using 274 features. Compared to the performance of MILES using the complete set of features (94.5% as reported in [24] using more than 10,000 features), the proposed method has proved to be effective in selecting a small set of the most informative features from a large feature set.

Fig. 10 shows the best five features selected by different feature selection methods. These features are the first five features selected by each method, and they are regarded by each method as most useful for describing the class variable. From this figure we can see the advantage of the proposed method over the other methods. The third and the fourth features selected by MID represent the same part of cars (i.e., the number plate). Similarly, the first and the second features selected by CMIM also represent the same part. In contrast, the proposed method selects a diverse range of features which give more information about the training examples and consequently yield a better performance over the other methods.

6. Conclusions and future work

There are three main contributions in this paper:

1. We present a general formulation of mutual information-based feature selection.
2. We compare and analyze the performance of eight different mutual information-based methods.
3. We present an effective feature selection method that takes into account both class-dependent and class-independent correlation among features.

[3] The dataset is available at http://www.robots.ox.ac.uk/~vgg/data.html.

[4] We use the salient region detector Version 1.5 implemented by Kadir [25] with the following parameters: scale is from 8 to 50, 16 bins, threshold for saliency 0.7, and threshold for inter-scale saliency 1.6. The software is available from http://www.robots.ox.ac.uk/~timork/salscale.html.

In this paper we have not addressed how to determine the number of features to be selected. Clearly there is a trade-off between the information gain and the complexity of the selected feature set. To this end, some researchers have used heuristics [2,11] to determine when to stop the feature selection process, or measure the 'merit' of a feature set [26] to select the best feature set. A more principled way to determine the number of features needs further investigation.

Conflict of interest

The authors declare that they have no conflict of interest.

Acknowledgment

National ICT Australia (NICTA) is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Center of Excellence program.

References

[1] A. Asuncion, D.J. Newman, UCI Machine Learning Repository ⟨http://archive.ics.uci.edu/ml⟩, 2010.

[2] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 1226–1238.

[3] K.S. Balagani, V.V. Phoha, On the feature selection criterion based on an approximation of multidimensional mutual information, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (7) (2010) 1342–1343.

[4] H. Liu, X. Wu, S. Zhang, Feature selection using hierarchical feature clustering, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, 2011, pp. 979–984.

[5] M.A. Kerroum, A. Hammouch, D. Aboutajdine, Textural feature selection by joint mutual information based on Gaussian mixture model for multispectral image classification, Pattern Recognition Letters 31 (10) (2010) 1168–1174.

[6] F. Fleuret, Fast binary feature selection with conditional mutual information, Journal of Machine Learning Research 5 (2004) 1531–1555.

[7] D. Lin, X. Tang, Conditional infomax learning: an integrated framework for feature extraction and fusion, in: Proceedings of European Conference on Computer Vision, 2006, pp. 68–82.

[8] C. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data, in: Proceedings of International Conference on Computational Systems Bioinformatics, 2003, pp. 523–528.

[9] L. Yu, H. Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, in: Proceedings of International Conference on Machine Learning, 2003, pp. 856–863.

[10] M. Vidal-Naquet, S. Ullman, Object recognition with informative features and linear classifier, in: Proceedings of IEEE International Conference on Computer Vision, vol. 1, 2003, pp. 281–288.

[11] G. Qu, S. Hariri, M. Yousif, A new dependency and correlation analysis for features, IEEE Transactions on Knowledge and Data Engineering 17 (2005) 1199–1207.

[12] T. Cover, J. Thomas, Elements of Information Theory, Wiley, 1991.

[13] J.Y. Ching, A.K.C. Wong, K.C.C. Chan, Class-dependent discretization for inductive learning from continuous and mixed-mode data, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (7) (1995) 641–651.

[14] S. Kotsiantis, D. Kanellopoulos, Discretization techniques: a recent survey, GESTS International Transactions on Computer Science and Engineering 32 (1) (2006) 47–58.

[15] U. Fayyad, K. Irani, Multi-interval discretization of continuous-valued attributes for classification learning, in: Proceedings of the International Joint Conference on Uncertainty in AI, 1993, pp. 1022–1027.

[16] B. Zhang, G. Ye, Y. Wang, J. Xu, G. Herman, Finding shareable informative patterns and optimal coding matrix for multiclass boosting, in: 2009 IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 56–63.

[17] N. Kwak, C.-H. Choi, Input feature selection by mutual information based on Parzen window, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (12) (2002) 1667–1671.

[18] K.K. Ang, Z.Y. Chin, H. Zhang, C. Guan, Mutual information-based selection of optimal spatial–temporal patterns for single-trial EEG-based BCIs, Pattern Recognition 45 (6) (2012) 2137–2144.

[19] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines ⟨http://www.csie.ntu.edu.tw/∼cjlin/libsvm⟩, 2001.

[20] I. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, second edn., Morgan Kaufmann, 2005.

[21] C. Ambroise, G.J. McLachlan, Selection bias in gene extraction on the basis of microarray gene-expression data, Proceedings of the National Academy of Sciences 99 (10) (2002) 6562–6566.

[22] C.-W. Hsu, C.-C. Chang, C.-J. Lin, et al., A practical guide to support vector classification, 2003, http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf, 2009.

[23] T. Tuytelaars, K. Mikolajczyk, Local invariant feature detectors: a survey, Foundations and Trends in Computer Graphics and Vision 3 (3) (2008).

[24] Y. Chen, J. Bi, J.Z. Wang, MILES: multiple-instance learning via embedded instance selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 1931–1947.

[25] T. Kadir, M. Brady, Scale, saliency and image description, International Journal on Computer Vision 45 (2001) 83–105.

[26] M. Hall, L.A. Smith, Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper, in: Proceedings of Florida Artificial Intelligence Symposium, 1999, pp. 235–239.

Gunawan Herman holds a Ph.D. in computer science from the University of New South Wales. He received B.E. degree in software engineering from the University of New South Wales in 2005. His research interests include computer vision, pattern recognition, and machine learning.

Bang Zhang holds a Ph.D. in computer science from the University of New South Wales. He is currently a research engineer in National ICT Australia (NICTA). He received the M.S. degree in computer science from the University of New South Wales in 2006, and the B.S. degree in computer science from the University of Sun Yat-Sen in 2004. His research interests include pattern recognition, image processing and machine learning.

Yang Wang received his Ph.D. degree in computer science from National University of Singapore in 2004. He is currently a senior researcher in National ICT Australia (NICTA). Before joining NICTA in 2006, he worked at the Institute for Infocomm Research, Rensselaer Polytechnic Institute, and Nanyang Technological University. His research interests include video analysis, sensor networks, pattern classification, biomedical engineering, medical imaging, and computer vision.

Getian Ye received his Ph.D. degree from School of Information Technology and Electrical Engineering at the University of New South Wales, Australia. He is currently working as a senior research engineer in Canon Information System Research Australia (CiSRA). Before joining CiSRA in 2010, he worked as a researcher at National ICT Australia (NICTA). His research interests include image processing, pattern recognition, and computer vision.

Fang Chen holds a Ph.D. in Communications and Electronic Systems and an MBA. She is currently the research group manager for the Making Sense of Data research theme in National ICT Australia (NICTA), Sydney. She is also a Conjoint Professor at the University of New South Wales. Her main research interests are human machine interaction, especially in multimodal systems, cognitive load modeling, speech processing, natural language processing, user interface design and evaluation.