Int. J. Appl. Math. Comput. Sci., 2006, Vol. 16, No. 4, 503–511

CORRELATION–BASED FEATURE SELECTION STRATEGY IN CLASSIFICATION PROBLEMS

KRZYSZTOF MICHALAK, HALINA KWAŚNICKA

Wrocław University of Technology, Institute of Applied Informatics
ul. Wybrzeże Wyspiańskiego 27, 50–370 Wrocław, Poland
e-mail: michalak@zacisze.wroc.pl, halina.kwasnicka@pwr.wroc.pl

In classification problems, the issue of high dimensionality, of data is often considered important. To lower data dimensionality, feature selection methods are often employed. To select a set of features that will span a representation space that is as good as possible for the classification task, one must take into consideration possible interdependencies between the features. As a trade-off between the complexity of the selection process and the quality of the selected feature set, a pairwise selection strategy has been recently suggested. In this paper, a modified pairwise selection strategy is proposed. Our research suggests that computation time can be significantly lowered while maintaining the quality of the selected feature sets by using mixed univariate and bivariate feature evaluation based on the correlation between the features. This paper presents the comparison of the performance of our method with that of the unmodified pairwise selection strategy based on several well-known benchmark sets. Experimental results show that, in most cases, it is possible to lower computation time and that with high statistical significance the quality of the selected feature sets is not lower compared with those selected using the unmodified pairwise selection process.

Keywords: feature selection, pairwise feature evaluation, feature correlation, pattern classification

1. Introduction

In many applications of computational methods, the problem of high dimensionality of data appears. Since high dimensional data are often hard to analyze, various methods are employed to reduce data dimensionality.

In the case of classification problems, data are often given as a set of vectors with each element of each vector being a value of some feature $f_i \in F = \{f_1, \ldots, f_k\}$. If we assume that the features are real-valued, we can introduce a set of vectors $V = \{v_1, v_2, \ldots, v_n\} \subset \mathbb{R}^k$, a set of classes C and a classifier $K : \mathbb{R}^k \to C$. Obviously,

$$\forall\ v_i \in V, j \in \{1, \ldots, k\} \cdot v_{ij} \in f_j. \tag{1}$$

Dimensionality reduction can be performed by selecting a subset $F' \subset F$. However, it is not easy to decide which features should be selected so that the quality of classification made using the representation space consisting of the selected features is as good as possible.

Usually, the feature selection process involves a quantitative criterion $Q(F_i)$ that measures the capability of the feature set F_i to discriminate between the classes. Depending of the number of features in the evaluated set F_i, selection techniques are divided into univariate and multivariate ones. The advantage of univariate methods is low computational complexity. However, they do not take into account any possible interdependencies between features. Thus, multivariate approaches may select more appropriate feature sets when features are correlated. Apart from higher computational complexity, the disadvantage of multivariate methods is that, when the sample size is small, they are more likely to suffer from the effects of high dimensionality.

Two main categories of feature selection methods are filters and wrappers (Kohavi and John, 1997). Filter methods rely solely on the properties of the features to choose the best feature set. On the other hand, wrappers evaluate feature sets based on the performance of a preselected classification algorithm on a training data set.

Both the filter and the wrapper approaches require a search procedure that iterates over the space of possible feature sets. Some basic strategies of feature selection are individual ranking (Kittler, 1978), forward search and backward search.

Individual ranking starts with an empty set $F_0 = \emptyset$. In each step, one best individually ranked feature f' is added,

$$F_n = F_{n-1} \cup \{f'\}, \tag{2}$$

where

$$f' = \arg \max_{F_{n-1} \cap \{f_i\} = \emptyset} Q(f_i). \tag{3}$$

Individual ranking does not take into account the existence of any dependencies between features and may therefore give poor results.

Forward search also starts with an empty set $F_0 = \emptyset$. In each step, one feature f' is added which maximizes the criterion Q together with previously selected features,

$$F_n = F_{n-1} \cup \{f'\}, \tag{4}$$

where

$$f' = \arg \max_{F_{n-1} \cap \{f_i\} = \emptyset} Q(F_{n-1} \cup \{f_i\}). \tag{5}$$

Forward search takes into consideration at least some of the potential interdependencies between features, but the required feature set is constructed in a greedy manner which may also produce suboptimal results (Cover and van Campenhout, 1977; Pudil *et al.*, 1994).

Backward search starts with the set of all features $F_0 = F$, and in each step it removes one feature f' which, when removed from the selected features set, maximizes the criterion Q,

$$F_n = F_{n-1} \setminus \{f'\}, \tag{6}$$

where

$$f' = \arg \max_{F_{n-1} \cap \{f_i\} \neq \emptyset} Q(F_{n-1} \setminus \{f_i\}). \tag{7}$$

Backward search is also a greedy approach, so it may produce suboptimal results. Also, it is much more computationally complex than forward search, as it requires the criterion Q to be evaluated on a representation space of much higher dimensionality than the one used in forward search.

More sophisticated methods of feature selection include genetic algorithms (Kwaśnicka and Orski, 2004), in which the population consists of potential feature sets and the fitness is calculated using the criterion Q. Other approaches are hybrid methods (Das, 2001; Xing *et al.*, 2001).

Recently, a pairwise selection strategy was proposed (Pękalska *et al.*, 2005). Pairwise selection takes into consideration at least some possible interdependencies between features and has reasonable computational complexity. In this selection strategy, the selection procedure begins with an empty set $F_0 = \emptyset$. Then, in each step of the iterative process, the best pair of features is added to the set of selected features F_n,

$$F_n = F_{n-1} \cup \{f', f''\}, \tag{8}$$

where

$$\{f', f''\} = \arg \max_{\substack{i \neq j \\ F_{n-1} \cap \{f_i, f_j\} = \emptyset}} Q(F_{n-1} \cup \{f_i, f_j\}). \tag{9}$$

Similarly to forward search, pairwise selection has one major advantage. Namely, it takes into account possible relationships between features. However, it is also far more computationally complex.

In this paper, a modification in the pairwise selection procedure is proposed. It is shown that the new approach substantially shortens computation time while producing equally good results. In Section 2, we present the new method of feature selection: in Section 3, experimental results are summarized: and Section 4 concludes the paper.

2. Selection Strategy

The method proposed in this paper uses a predetermined classifier to evaluate the quality of the feature set, and therefore this approach is a wrapper method. Let c_i denote the actual class to which the vector $v_i \in V$ belongs, and c'_i be the class chosen by the classifier. Assume that the data set V is partitioned into a training set V_{train} and a test set V_{test}. The aim of the feature selection process is to select a set F' containing a predefined number l of features such that the classifier $K : R^l \to C$ trained using vectors from the training set $V_{\text{train}}\lceil_{F'}$ will give the possibly lowest classification error E on the test set V_{test}:

$$E(F') = \frac{1}{|V_{\text{test}}|} \sum_{v_i \in V_{\text{test}}} H(c'_i, c_i), \tag{10}$$

where c_i signifies the actual class to which v_i belongs, and $c'_i = K(v_i\lceil_{F'})$ means the classification result given by the classifier K,

$$H(a, b) = \begin{cases} 1 & \text{if} \quad a \neq b, \\ 0 & \text{if} \quad a \neq b. \end{cases}$$

Consequently, we define the criterion Q used in the selection process as

$$Q(F_i) = 1 - E(F_i). \tag{11}$$

In the unmodified pairwise selection process (Pękalska *et al.*, 2005), the feature set is expanded by iteratively adding pairs of features satisfying the condition (9).

To reduce the number of required operations, we suggest a modification in the pairwise selection strategy. In our method, the features are evaluated individually or in a pairwise manner depending on the value of the correlation between the given feature and all other features. Formally, consider each feature f_i as a random variable. Given the training vector set V_{train}, we can compute a sample estimate of the correlation coefficient σ_{ij} between the features f_i and f_j using

$$\sigma_{ij} = \frac{COV(f_i, f_j)}{\sqrt{VAR(f_i)VAR(f_j)}}. \tag{12}$$

Each selection step is performed as follows:

$S = \emptyset$
$Q_{\max} = 0$
for each $i = 1, \ldots, k$, $f_i \notin F_{n-1}$
  if exists $j \in \{i+1, \ldots, k\}$ such that $\sigma_{ij} > \theta$
    for each $j = i+1, \ldots, k$, $i \neq j$, $f_j \notin F_{n-1}$
      if $Q_{\max} < Q(F_{n-1} \cup \{f_i, f_j\})$
        $S = \{f_i, f_j\}$
        $Q_{\max} = Q(F_{n-1} \cup \{f_i, f_j\})$
      end if
    end for
  else
    if $Q_{\max} < Q(F_{n-1} \cup \{f_i\})$
      $S = \{f_i\}$
      $Q_{\max} = Q(F_{n-1} \cup \{f_i\})$
    end if
  end if
end for
$F_n = F_{n-1} \cup S$.

This procedure ensures that any feature that correlates with any other feature at the level of at least θ (θ is a parameter with a fixed value) will be evaluated in a pairwise manner, while the features that have no significant correlation with any other feature will be evaluated individually. In each case, the criterion Q defined by (11) is used for evaluating the set of features together with the previously selected features. Obviously, if there is only one more feature required, i.e., $|F_{n-1}| = l - 1$ (where l is the required number of features), no pairwise evaluation is performed.

We hypothesize that, for a sufficiently low threshold θ, all significant relationships between the features will be exploited by the pairwise part of the search. As the complexity of selecting l features in a pairwise manner is $O(l^2)$ and the complexity of selecting l features individually is $O(l)$, evaluating some of the features individually should improve the performance of the selection strategy for sufficiently large l.

3. Experiments

To validate the proposed approach, we performed a number of experiments in which the classification error and computation time for the pairwise selection strategy proposed in (Pękalska *et al.*, 2005) and for the correlation-based strategy were compared. The correlation-based selection strategy is parametrized by a parameter $\theta \in [0, 1]$, which is used to decide whether a given feature should be evaluated individually or in a pairwise manner. In the experiments, this parameter was set to $\theta = 0.5$. The experiments were performed using the following data sets: Mushroom, Waveform and Waveform with noise – all from the UCI Repository of Machine Learning Databases (Blake and Merz, 2006), and Gaussian – an artificial data set with some of the features forming correlated pairs, as described in (Pękalska *et al.*, 2005). A summary of the data sets is presented in Table 1. In this table, the number of samples used for training and the maximum number of features that were selected using tested selection strategies are also given.

Table 1. Data summary.

Data set	Total samples	Training samples	Total feat.	Max. selected features
V	$\lvert V\rvert$	$\lvert V_{\text{train}}\rvert$	k	$l_{\max}$
Gaussian	10 000	100	20	20
	10 000	100	40	20
	10 000	100	60	20
	10 000	100	80	20
	10 000	100	100	20
Mushroom	8124	100	20	20
	8124	200	20	20
	8124	400	20	20
	8124	1000	20	20
	8124	2000	20	20
	8124	4000	20	20
Waveform	5000	35	21	21
	5000	350	21	21
	5000	3500	21	21
Waveform	5000	35	21	21
w/noise	5000	350	21	21
	5000	3500	21	21

The Gaussian data set is constructed so that it contains k features of which only $q \leq k$ are informative. The informative features are drawn in pairs from the Gaussian distribution with the class means $\mu_1 = [0, \ 0]^T$ and $\mu_2 = \frac{\sqrt{2}}{2}[r, \ -r]^T$ for some $r > 0$. The covariance matrix for both classes is

$$\Sigma_1 = \Sigma_2 = \begin{bmatrix} v+1 & v-1 \\ v-1 & v+1 \end{bmatrix},$$

where v is a parameter with a fixed value. The remaining $k - q$ uninformative features are drawn from the spherical Gaussian distribution $N(0, \frac{v}{\sqrt{2}}I)$. In our experiments, the number of informative features was always $q = 20$. The remaining $k - q$ features were used to simulate the classification of noisy data. The distribution parameters were set to $r = 3$ and $v = \sqrt{40}$.

In the experiments, the feature selection strategies were tested using two Bayes classifiers: the NLC and the NQC (Duda *et al.*, 2001). The first of these classifiers was used by other authors to evaluate the pairwise selection

method (Pękalska *et al.*, 2005). Both classifiers are defined on the Euclidean space $\mathbb{R}^l$. The NLC classifier is defined as

$$f(x) = \left[x - \frac{1}{2}(m_1 + m_2)\right]^T S^{-1}(m_1 - m_2) + \log\frac{p_1}{p_2}, \tag{13}$$

where m_1 and m_2 are the estimated means of the classes, S is the estimated covariance matrix, and p_1 and p_2 are the prior probabilities of the classes.

The NQC classifier is defined as

$$\begin{aligned} f(x) = {} & \frac{1}{2}(x - m_1)^T S_1^{-1}(x - m_1) \\ & - \frac{1}{2}(x - m_2)^T S_2^{-1}(x - m_2) \\ & + \frac{1}{2}\log\frac{|S_1|}{|S_2|} + \log\frac{p_1}{p_2}, \end{aligned} \tag{14}$$

where m_1 and m_2 are the estimated means of the classes, S_1 and S_2 are the estimated class covariance matrices, and p_1 and p_2 are the prior probabilities of the classes.

Both classifiers are binary classifiers with the classification boundary $f(x) = 0$. Thus, in the experiments, the data were always partitioned into two classes and the class membership was determined by the sign of the value returned by the classifier.

In the experiments, the selection of the $l = 1, \ldots, l_{\max}$ features was performed for each data set presented in Table 1. For each number of features, the mean classification error obtained using pairwise selection (E_p) and the correlation-based method (E_c) was recorded. Also, the mean computation times (T_p and T_c) were recorded. For each number of features, the average values from 30 runs were recorded. Tables 2–9 present the average ratios E_c/E_p and T_c/T_p calculated for all numbers of features and for each data set. As the pairwise selection strategy allows only selecting an even number of features, only even numbers were taken into consideration when calculating the values presented in the tables.

As the above results suggest, the most significant improvement is the reduction in computation time. As was expected, the new method performs better in the presence of noise in data (compare the results for the Waveform data sets, with and without noise, and for the Gaussian data sets with different numbers of uninformative features).

Figures 1–16 present the results for different numbers of selected features. For the pairwise selection strategy, only the results for an even number of selected fea-

Table 2. Results for the NLC classifier obtained when selecting $l = 2, \ldots, 20$ features for the Gaussian data set ($|V| = 10000$, $|V_{\text{train}}| = 100$).

Total features	Error ratio	Time ratio
k	E_c/E_p	T_c/T_p
20	1.0000	1.0994
40	0.9378	0.5072
60	0.9053	0.3255
80	0.9119	0.2433
100	0.8430	0.1883

Table 3. Results for the NLC classifier obtained when selecting $l = 2, \ldots, 20$ features for the Mushroom data set ($|V| = 8124, k = 20$).

Training samples	Error ratio	Time ratio
$\lvert V_{\text{train}}\rvert$	E_c/E_p	T_c/T_p
100	0.9929	0.6091
200	0.9961	0.5778
400	0.9989	0.5306
1000	0.9955	0.4657
2000	0.9978	0.4451
4000	0.9896	0.4305

Table 4. Results for the NLC classifier obtained when selecting $l = 2, \ldots, 20$ features for the Waveform data set ($|V| = 5000, k = 21$).

Training samples	Error ratio	Time ratio
$\lvert V_{\text{train}}\rvert$	E_c/E_p	T_c/T_p
35	0.9948	0.8182
350	0.9964	0.7561
3500	0.9996	0.6588

Table 5. Results for the NLC classifier obtained when selecting $l = 2, \ldots, 20$ features for the Waveform w/noise data set ($|V| = 5000, k = 21$).

Training samples	Error ratio	Time ratio
$\lvert V_{\text{train}}\rvert$	E_c/E_p	T_c/T_p
35	0.9930	0.4842
350	0.9942	0.3486
3500	0.9979	0.2974

Table 6. Results for the NQC classifier obtained when selecting $l = 2, \ldots, 20$ features for the Gaussian data set ($|V| = 10000$, $|V_{\text{train}}| = 100$).

Total feat.	Error ratio	Time ratio
k	E_c/E_p	T_c/T_p
20	1.0000	0.5394
40	0.9604	0.4714
60	0.9110	0.3339
80	0.9088	0.2359
100	0.9194	0.2037

Table 7. Results for the NQC classifier obtained when selecting $l = 2, \ldots, 20$ features for the Mushroom data set ($|V| = 8124, k = 20$).

Training samples $\lvert V_{\text{train}}\rvert$	Error ratio E_c/E_p	Time ratio T_c/T_p
100	1.0123	0.6048
200	0.9961	0.5778
400	0.9849	0.5284
1000	1.0019	0.4770
2000	1.0013	0.4432
4000	1.0575	0.4269

Table 8. Results for the NQC classifier obtained when selecting $l = 2, \ldots, 20$ features for the Waveform data set ($|V| = 5000, k = 21$).

Training samples $\lvert V_{\text{train}}\rvert$	Error ratio E_c/E_p	Time ratio T_c/T_p
35	0.9744	0.8212
350	0.9987	0.7211
3500	0.9995	0.6812

Table 9. Results for the NQC classifier obtained when selecting $l = 2, \ldots, 20$ features for the Waveform w/noise data set ($|V| = 5000, k = 21$).

Training samples $\lvert V_{\text{train}}\rvert$	Error ratio E_c/E_p	Time ratio T_c/T_p
35	0.9773	0.4431
350	0.9730	0.3552
3500	0.9941	0.3254

tures are available. Apart from data points, one-variance ranges were marked in the charts. Clearly, the classification errors given by both methods are very similar, except for the Gaussian data set with 20 informative and 80 uninformative features, where the new method produces a substantially lower classification error. Apparently, the new method is more effective in reducing the influence of uninformative features on the classification process.

From the presented results it is clear that the new selection strategy is much faster than the classic pairwise approach. To prove that the classification errors produced by the new method are, on average, no higher than the errors obtained using the unmodified pairwise selection strategy, we computed the p-value of the hypothesis that the new method gives on average worse results than the traditional one. Let m_p and m_c denote the mean classification error yielded by the pairwise selection strategy and the correlation-based strategy, respectively. Let n_p and n_c denote the number of the best results given by each method. Assume that

$$m_p < m_c, \tag{15}$$

i.e., that the new method produces statistically worse results than the traditional one.

As the averages of 30 measurements have approximately normal distributions, the probability $P(k)$ of getting k best results using the new method would then have the Bernoulli distribution with the success probability $q < 1/2$.

The upper bound on the p-value of the null hypothesis that the new method statistically produces worse results than the traditional one can be calculated with respect to the results obtained as the probability p of getting at least n_c better results using the new method. Considering the above, this probability can be calculated as

$$p = 1 - P(k < n_c), \tag{16}$$

$$P(k < n_c) = \sum_{i=0}^{n_c-1} \binom{n_c + n_p}{i} q^i (1-q)^{n_c+n_p-i}, \tag{17}$$

$$P(k < n_c) \geq \frac{1}{2^{n_c+n_p}} \sum_{i=0}^{n_c-1} \binom{n_c + n_p}{i}, \tag{18}$$

$$p \leq 1 - \frac{1}{2^{n_c+n_p}} \sum_{i=0}^{n_c-1} \binom{n_c + n_p}{i}. \tag{19}$$

Tables 10 and 11 present the values of n_c and n_p calculated as the number of times each method gave a lower classification error than the other for the data sets described in Table 1, using the NLC and NQC classifiers, respectively. When both methods gave identical classification errors, neither n_c, nor n_p was incremented. In Tables 10 and 11, the upper p-value bounds for the null hypothesis are also given.

The overall upper p-value bound for the NLC classifier is $6.91 \cdot 10^{-47}$, and for the NQC classifier it is $7.25 \cdot 10^{-33}$.

4. Conclusion

In this paper we have proposed a modification of pairwise selection of features. We tested the new method on several data sets commonly used as benchmark data in machine learning. The results suggest that the new method requires less computation time than the traditional approach when selecting a given number of features. The experiments also show with very high statistical significance that the average classification error obtained when using the correlation-based selection strategy should not be higher than the classification error obtained when using the traditional approach. The new feature selection method was tested using two classifiers — NLC and NQC. A further study is necessary to evaluate this method with other classifiers.

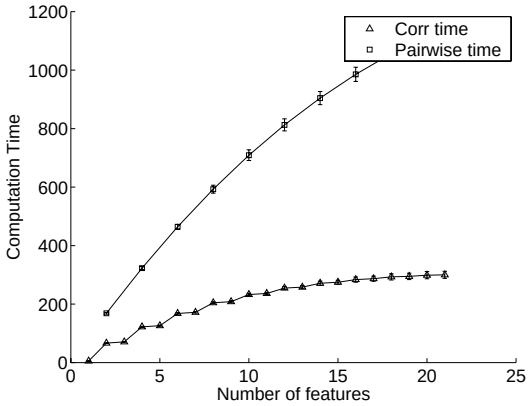

Fig. 1. Computation time observed for the NQC classifier on the Waveform w/noise data set for $|V_{\text{train}}| = 3500$.

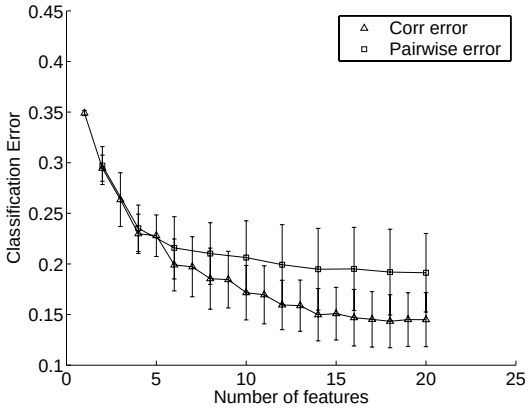

Fig. 2. Classification error observed for the NLC classifier on the Gaussian data set for $k = 100$.

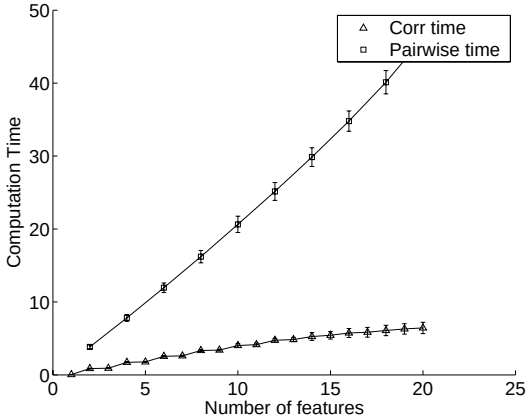

Fig. 3. Computation time observed for the NLC classifier on the Gaussian data set for $k = 100$.

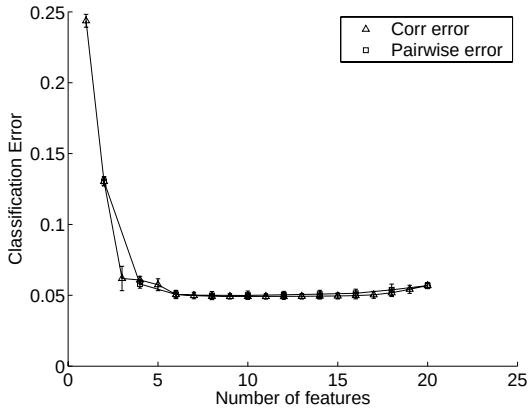

Fig. 4. Classification error observed for the NLC classifier on the Mushroom data set for $|V_{\text{train}}| = 4000$.

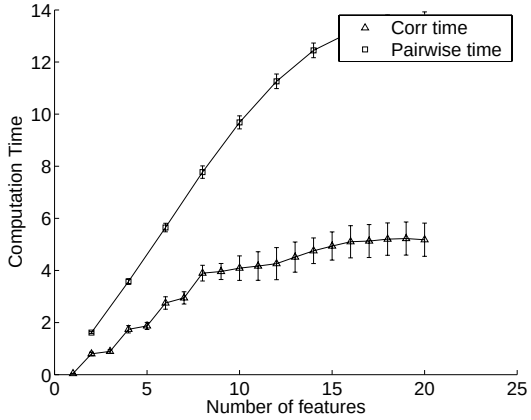

Fig. 5. Computation time observed for the NLC classifier on the Mushroom data set for $|V_{\text{train}}| = 4000$.

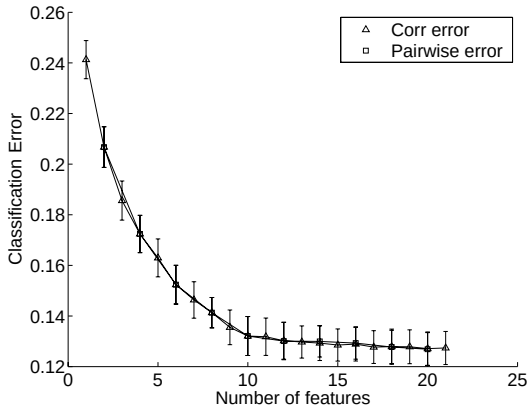

Fig. 6. Classification error observed for the NLC classifier on the Waveform data set for $|V_{\text{train}}| = 3500$.

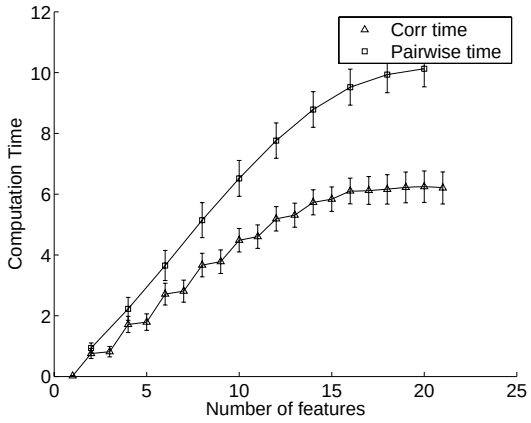

Fig. 7. Computation time observed for the NLC classifier on the Waveform data set for $|V_{\mathrm{train}}| = 3500$.

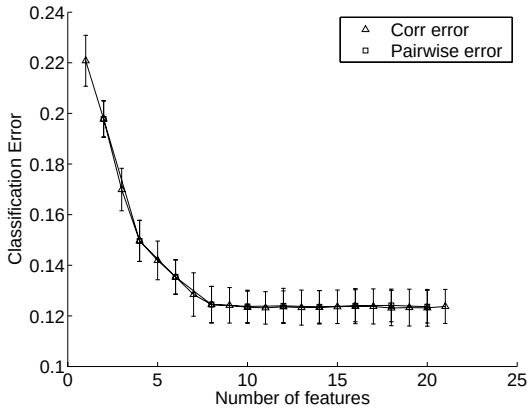

Fig. 8. Classification error observed for the NLC classifier on the Waveform w/noise data set for $|V_{\mathrm{train}}| = 3500$.

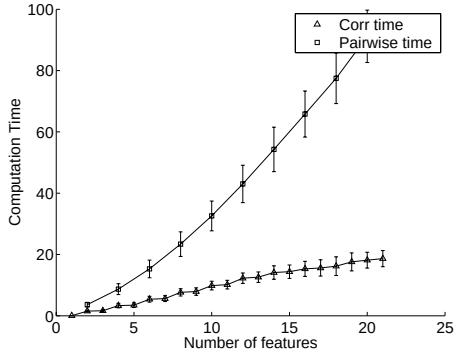

Fig. 9. Computation time observed for the NLC classifier on the Waveform w/noise data set for $|V_{\mathrm{train}}| = 3500$.

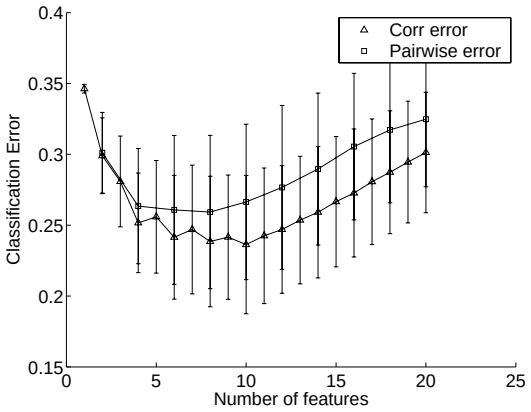

Fig. 10. Classification error observed for the NQC classifier on the Gaussian data set for $k = 100$.

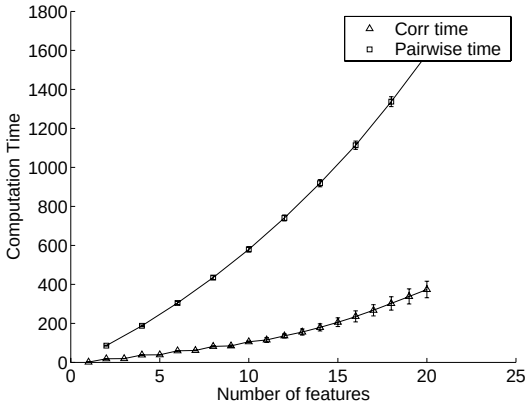

Fig. 11. Computation time observed for the NQC classifier on the Gaussian data set for $k = 100$.

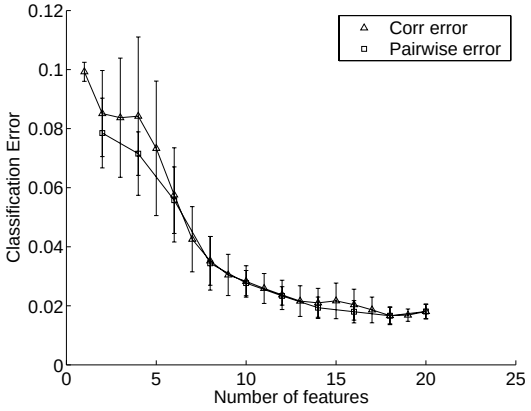

Fig. 12. Classification error observed for the NQC classifier on the Mushroom data set for $|V_{\mathrm{train}}| = 4000$.

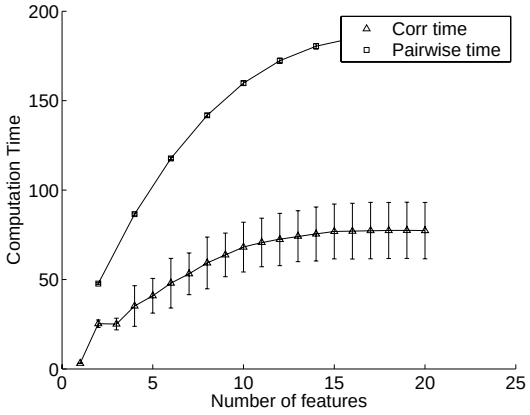

Fig. 13. Computation time observed for the NQC classifier on the Mushroom data set for $|V_{\mathrm{train}}| = 4000$.

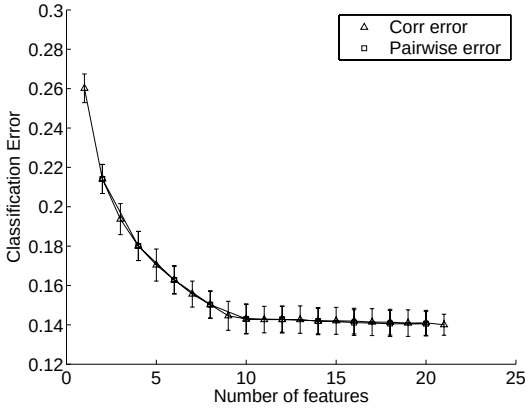

Fig. 14. Classification error observed for the NQC classifier on the Waveform data set for $|V_{\mathrm{train}}| = 3500$.

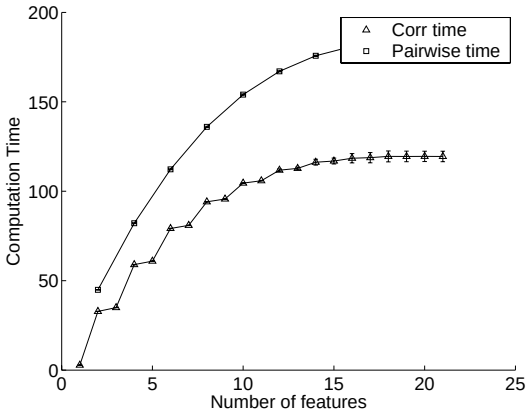

Fig. 15. Computation time observed for the NQC classifier on the Waveform data set for $|V_{\mathrm{train}}| = 3500$.

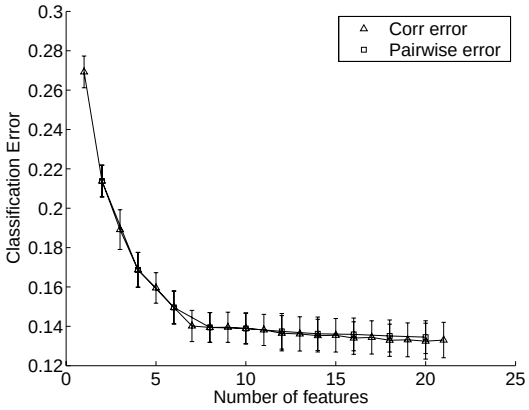

Fig. 16. Classification error observed for the NQC classifier on the Waveform w/noise data set for $|V_{\mathrm{train}}| = 3500$.

Table 10. Upper p-value bounds obtained in the tests for the NLC classifier.

Data set	Training samples $\lvert V_{\mathrm{train}}\rvert$	Total feat. k	n_c	n_p	p-value
Gaussian	100	20	0	0	—
	100	40	111	26	$5.24 \cdot 10^{-14}$
	100	60	147	34	$3.01 \cdot 10^{-18}$
	100	80	148	52	$3.63 \cdot 10^{-12}$
	100	100	185	36	$1.21 \cdot 10^{-25}$
Mushroom	100	20	65	39	$6.92 \cdot 10^{-3}$
	200	20	53	44	0.21
	400	20	70	47	0.02
	1000	20	64	52	0.15
	2000	20	47	40	0.26
	4000	20	81	46	$1.21 \cdot 10^{-3}$
Waveform	35	21	79	59	0.05
	350	21	71	55	0.09
	3500	21	31	22	0.14
Waveform w/noise	35	21	105	70	$4.98 \cdot 10^{-3}$
	350	21	88	77	0.22
	3500	21	63	48	0.09

Table 11. Upper p-value bounds obtained in the tests for the NQC classifier.

Data set	Training samples $\lvert V_{\mathrm{train}}\rvert$	Total feat. k	n_c	n_p	p-value
Gaussian	100	20	0	0	—
	100	40	111	47	$1.89 \cdot 10^{-7}$
	100	60	172	38	$7.36 \cdot 10^{-22}$
	100	80	172	32	$1.12 \cdot 10^{-24}$
	100	100	177	45	$5.29 \cdot 10^{-20}$
Mushroom	100	20	79	69	0.23
	200	20	76	65	0.20
	400	20	91	84	0.33
	1000	20	84	100	0.90
	2000	20	65	113	1.00
	4000	20	89	121	0.99
Waveform	35	21	112	90	0.07
	350	21	57	46	0.16
	3500	21	41	32	0.17
Waveform w/noise	35	21	121	65	$2.44 \cdot 10^{-5}$
	350	21	80	24	$1.61 \cdot 10^{-8}$
	3500	21	79	30	$1.48 \cdot 10^{-6}$

References

Blake C. and Merz C.(2006): *UCI Repository of Machine Learning Databases*. — Available at: `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Cover T.M. and van Campenhout J.M. (1977): *On the possible ordering in the measurement selection problem*. — IEEE Trans. Syst. Man Cybern., SMC-07(9), pp. 657–661.

Das S. (2001): *Filters, wrappers and a boosting-based hybrid for feature selection.* — Int. Conf. *Machine Learning*, San Francisco, Ca, USA, pp. 74–81.

Duda R., Hart P. and Stork D. (2001): *Pattern Classification.* — New York: Wiley.

Kittler J. (1978): *Pattern Recognition and Signal Processing.* — The Netherlands: Sijhoff and Noordhoff, pp. 4160.

Kohavi R. and John G.H. (1997): *Wrappers for feature subset selection.* — Artif. Intell., Vol. 97, Nos. 1–2, pp. 273–324.

Kwaśnicka H. and Orski P. (2004): *Genetic algorithm as an attribute selection tool for learning algorithms*, Intelligent Information Systems 2004, New Trends in Intelligent Information Processing and Web Mining, Proc. Int. IIS: IIP WM04 Conf. — Berlin: Springer, pp. 449–453.

Pękalska E., Harol A., Lai C. and Duin R.P.W. (2005): *Pairwise selection of features and prototypes*, In: Computer Recognition Systems (Kurzyński M., Puchała E., Woźniak M., Zolnierek, Eds.). — Proc. 4-th Int. Conf. *Computer Recognition Systems, CORES'05*, Advances in Soft Computing, Berlin: Springer, pp. 271–278.

Pudil P., Novovicova J. and Kittler J. (1994): *Floating search methods in feature selection*. — Pattern Recogn. Lett., Vol. 15, No. 11, pp. 1119–1125.

Xing E., Jordan M. and Karp R. (2001): *Feature selection for high-dimensional genomic microarray data*. — Proc. Int. Conf. *Machine Learning*, San Francisco, CA, USA, pp. 601–608.

Received: 2 May 2006
Revised: 6 November 2006