PERGAMON

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Pattern Recognition 38 (2005) 41–49

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

An SVM classifier incorporating simultaneous noise reduction and feature selection: illustrative case examples

R. Kumar, V.K. Jayaraman, B.D. Kulkarni*

Chemical Engineering Division, National Chemical Laboratory, Dr. Homi Bhabha Road, Pune 411 008, India

Received 7 November 2003; received in revised form 1 June 2004; accepted 1 June 2004

Abstract

A hybrid technique involving symbolization of data to remove noise and use of conditional entropy minima to extract relevant and non-redundant features is proposed in conjunction with support vector machines to obtain more robust classification algorithm. The technique tested on three data sets shows improvements in classification efficiencies.

Keywords: Symbolization; Conditional entropy; SVM; Classification

1. Introduction

Classification tasks continue to interest researchers and every year several new algorithms are proposed claiming improved accuracy. While the majority of these learning algorithms perform well on domains with relevant information, they degrade in adverse situations like: data with high noise content, small sample sizes relative to number of features, irrelevant or redundant information and non-linearity. Markovitch [1] identifies irrelevant, noisy and redundant information as detrimental elements leading to the inaccuracies in prediction. The efficiency of the learning algorithms decreases on domains with irrelevant and redundant features [2–9]. Moreover, as the number of features used for classification task grows, the number of training samples required for statistical model fitting and/or supervised learning systems grows exponentially [10,11], a situation highly undesirable in low sample size situations. Improved performance may be achieved by discarding such noisy, irrelevant and redundant information [12–14]. Literature suggests the use of feature preprocessing schemes like Feature Extraction, Feature Construction and Feature Selection to deal with this problem [8]. Feature extraction schemes (like Principal component analysis, Linear discriminant analysis, Locally linear embedding, Isomap, Multidimensional scaling, etc.) carry out linear/non-linear transformation of data and project it to a lower dimensional space in such a way that most of the information is retained while discarding the noisy component of data. Feature construction attempts to simplify hypothesis search by adding newer features with additional information [15]. These two approaches try to solve the problem of irrelevant information in the feature space by changing the representation. Feature selection is a special case of feature extraction involving selection of a subset of features that describe the hypothesis at least as well as the original set [16,17]. Feature extraction makes N measurements to obtain M-dimensional data $(N \gg M)$. Feature selection, on the other hand, discards $(N\text{-}M)$ irrelevant features requiring to collect only relevant attributes reducing the cost of data collection. The benefits of feature selection thus include a reduction in the amount of data needed to achieve learning, improved predictive accuracy, more compact and easily understood knowledgebase and reduced execution time. The last two factors are of particular importance in the area of

* Corresponding author. Tel.: +91-20-25893095; fax: +91-20-25893041.
E-mail address: bdk@ems.ncl.res.in (B.D. Kulkarni).

0031-3203/$30.00 © 2004 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2004.06.002

commercial and industrial data mining making it the more preferred. It is desirable that the overall scheme should also be capable of handling noisy component of the features selected.

2. Previous work

The literature regarding the feature selection methods and applications is widespread across many fields, including document classification, data mining, object recognition, biometrics, remote sensing and computer vision. It is relevant to any task where the number of features is larger than the number of training samples, or too large to be computationally feasible. Existing feature selection methods for machine learning typically fall into two broad categories: wrappers and filters [18].

The wrapper approaches are heuristic search procedures that evaluate the quality of the feature subset by using the prediction accuracy of the target-learning scheme. They include techniques such as the sequential forward and backward feature selection [19], the greedy variants of hill climbers [20], best-first search [3], beam search [21] and the randomized algorithms like Simulated Annealing [22] and Genetic algorithms [23,24]. Wrappers often give better results (in terms of the final predictive accuracy of a learning algorithm) than filters because feature selection is optimized for the particular learning algorithm used. However, since a learning algorithm is employed to evaluate each and every set of features considered, wrappers are prohibitively expensive to run, and can be intractable for large databases containing many features. Furthermore, since the feature selection process is tightly coupled with a learning algorithm, wrappers are less general than filters and must be rerun when switching from one learning algorithm to another.

The Filter approach evaluates the features independent of the classifiers and attempts to remove the irrelevant features from the feature set before it is used by the learning algorithm [8]. The examples of feature evaluating measures are intrinsic properties of the data, probabilistic distance measures, probabilistic dependence measures, interclass distance measures, information theoretic measures like entropy etc. [23]. *FOCUS* [25], cross-entropy filter [26] and RELIEF and its variants [27,28], decision tree filter [29] are some of the well-known filter schemes. These measures capture the relationship of the feature with the target concept. Filter approaches are computationally less expensive and more general in nature but return a large feature subset. Also, some of the filter algorithms previously described do not handle noise in data (Focus), and others require that the level of noise be roughly specified by the user a priori [30].

Another noticeable observation from these works is that there is no algorithm that performs optimally on all domains, as shown by variability in experimental results. This is understandable as feature selection is a highly domain specific task. Finding the optimal set of features is usually intractable [4] and many problems related to feature selection have been shown to be NP-hard [31,32]. For most practical problems, an optimal solution can only be guaranteed if a monotonic criterion for evaluating features can be found, but this assumption however rarely holds in the real world. As a result, we are forced to find heuristic solutions that represent a trade-off between solution quality (w.r.t. generalization, predictive accuracy) and time.

3. Proposed system

This work describes an efficient and robust scheme that discards the noisy, irrelevant and redundant information present in data, while still retaining the discriminating power of the data. A combination of filter and wrapper approaches is suggested to get improved accuracy, efficiency and better generalization. Here filter provides an intelligent starting feature subset for a wrapper—a process that is likely to result in a shorter, and hence faster search for the wrapper. The proposed scheme applies the method of data symbolization for solving the dual problem of filtering and noise reduction. Data symbolization involves discretization of the raw data features into a stream of limited set of values called symbols, which retain dominant deterministic features while suppressing measurement noise. Further the conditional entropy of class label with respect to the feature attribute (converted into symbolic form) is computed to determine whether the feature is decidedly correlated to the class or not. Here lower the conditional entropy, higher is the coupling. Similarly we can find the degree of coupling of a feature to other features. Quantities such as the correlation coefficient or the correlation function often do not provide unequivocal indication of the coupling feature variables and class information (since these can be sparse and noisy). Symbolization scheme, on the other hand, works even in presence of external noise [33,34]. The data symbolization method can be applied to deterministic or stochastic, linear or non-linear systems, without any a priori assumptions about the nature of the underlying dynamical process and has a practical advantage of simplifying and speeding up subsequent computations as data space is changed from continuum to discrete form.

Outline of the proposed scheme is as follows:

1. Convert the data into symbolic form.
2. Compute the conditional entropy of the class information with respect to all features one-by-one. Here conditional entropy is used as relevance filter. We therefore threshold the relevance values to divide the feature set into relevant and irrelevant features. This can be done either by thresholding the conditional entropy value directly or by selecting the lowest n values and discarding the remaining features. This comprises subset 1.
3. Compute the conditional entropy of the feature (with highest coupling with class information) with respect to

all remaining features (in subset 1) one-by-one. Features showing high correlation (low values of conditional entropy) are discarded. Here too we either use a direct threshold or select the highest *n* values and discard the remaining features. We get subset 2.

4. Arrange subset 2 features in ascending order of their conditional entropy with class information. Take feature with second lowest conditional entropy and compute its conditional entropy with respect to features with conditional entropy higher than it. Features are chosen in similar way as in step 3.
5. Repeat step 4 until the last feature.
6. Features obtained from step 5 are used as an input to the wrapper scheme using support vector machines (SVMs) as the learning algorithm. Support vector machines based on rigorous statistical learning theory has many desirable properties such as nonlinear learning, improved generalization performance etc. [35,36]. Here SVM is assisted by genetic algorithm and quasi Newton algorithm for optimal tuning of parameters [37]. In wrapper step we have carried an exhaustive enumeration of all possible feature subsets in filtered space.

In the following sections, we discuss in brief the process of data symbolization, calculation of the conditional entropy and SVM methodology for classification problem. Thereafter the performance estimation and parameter tuning of SVM is discussed. Finally the data sets used, results obtained and conclusions are discussed, respectively.

4. Symbolization

Symbolization implies coarse graining. Typically the range of each original feature (or the range of some transform of the original data such as the first differences between successive values) is partitioned into a finite number of discrete cells and assigning different symbols to each cell [38,39]. Each original value of an attribute is thus uniquely mapped to a particular symbol depending on the domain in which the measurement falls. Thus

$$S_i = \begin{bmatrix} 1 & X_{\min} < X_i < X_{C_1} \\ 2 & Xc_1 < X_i < X_{C_2} \\ 3 & X_{C_2} < X_i < X_{C_3} < X_{\max} \\ \vdots & \\ \ldots & n\ symbols \end{bmatrix}$$

Here X_{C_1}, X_{C2}, X_{C3} are critical points, defining the boundaries of cells. $1, 2, 3, \ldots, n$ are the symbols. The number of symbols used, *n*, is referred as the *symbol-set*. In the simplest (binary) case $n=2$. The number of symbols determines how much of the original information is retained. A higher value of *n* takes into account more details of original data, along with the effects of any measurement noise that might be present. For example, when *n* equals the number of distinct values in the data, the symbolized data and the original data are equivalent in the sense that they contain the same information (that is, there is no loss of information due to symbolization). Also for any symbol set size, the placement of critical points affects the characteristics of the symbolic description of the data. Partitioning of data should be carried out carefully, as poor choice of partition locations may lead to loss of meaningful information [40,41]. Thus, even though symbolization minimizes the effects of noise in data, it also causes the loss of meaningful information during the process. It is necessary that the loss of information during the process be at its minimum. We implement this trade-off (between the reduction of noise and loss of meaningful information) by partitioning the data in conjunction with the information entropic analysis [42–45] as described in below.

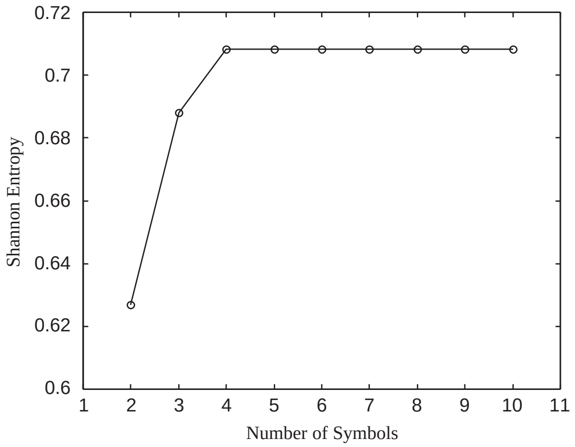

Fig. 1. Shannon entropy vs. number of symbols for equal interval binning.

For analysis, the symbol series is transformed into symbolic sequences by defining a finite length (L) template that can be moved along symbol series one-step at a time, each step revealing new sequence (see Fig. 1).

For convenience of reference and identification every short sequence is uniquely denoted by just one integer

$$\ell = \sum_{i=1}^{L} M^{L-i} S_i, \tag{1}$$

where M is number of different symbols and L is length of symbolic sequence. This symbol sequence series (or coded series) can be characterized using information theoretical measures such as Shannon entropy defined as [34]

$$E = -\frac{1}{L} \sum_{\ell} P_\ell \ln P_\ell, \tag{2}$$

where P_ℓ is the probability of finding a particular sequence ℓ. It is defined as number of times this sequence can be found in the symbolic series divided by the number of all short sequences. The Shannon entropy is a gauge to quantify the information content in the symbolic series. The optimal number of symbols that should be used for maximizing the information content and minimizing the effect of noise can be obtained by maximizing entropy E, with respect to (a) the number of critical points and (b) the placements of these critical points. The entire process of symbolization is illustrated below for a data with one single feature:

0.4966	0.8998	0.8216	0.6449
0.8180	0.6602	0.3420	0.2897
0.3412	0.5341	0.7271	0.3093
0.8385	0.5681	0.3704	0.7027
0.5466	0.4449	0.6946	0.6213

The procedure is illustrated for the number of cells equal to $n = 3$ (equal size), and for symbol sequence length of $L = 4$. The above data can be converted into the symbolized data for three discrete cells as

2 3 3 2 3 2 1 1 1 2 3 1 3 2 1
3 2 1 2 2

Now for $L = 4$, the sequences are:

2 3 3 2; 3 3 2 3; . . . ; 2 1 2 2

The equivalent code for first sequence as obtained from Eq. (1):

(3^3)*2 + (3^2)*3 + (3^1)*3 + (3^0)*2 = 92

By repeating this coding step for all remaining sequences, we get the following coded series:

92 117 110 88 103 67 41 45 55
87 101 61 105 74 61 104 71

This whole process is repeated for $n = (2, 3, 4, \ldots, 10)$. The resultant entropies calculated using Eq. (2) show that a maximum entropy (0.7083) was obtained for number of symbols $n = 4$ (Fig. 1). Thus symbolized data for $n = 4$ would be optimal. In case of multi-feature data, we need to follow the same procedure for each feature independent of other features. Test data (or online data) is symbolized with same cell boundaries as used for training. If test data exceed the range covered by training data (on either side), then it is assigned the lowest and highest symbol in accordance of the boundary crossed.

4.1. Conditional entropy

For two signals $\{X\}$ and $\{Z\}$ the conditional information entropy is defined as [34]

$$E(Z/X) = -\frac{1}{N_x} \sum_{S_x} \sum_{S_z} P(S_z/S_x) \ln P(S_z/S_x), \tag{3}$$

where N_x is the total number of different S_x values that are observed and $P(S_z/S_x)$ is the probability for the variable Z to take symbolic value of S_z when the variable X occupies the symbolic value of S_x.

One can interpret the conditional entropy as the amount of uncertainty remaining about Z after X has been observed. Lower the conditional entropy higher the correlation between two variables.

5. SVM classification

SVMs based on the tenets of statistical learning theory is now being routinely used for several binary and multi class classification tasks in different fields. Computational biologists have also employed SVM for carrying important tasks such as structural classifications of proteins [46–50]. The methodology, algorithms and software are now readily available [51–53]. We therefore provide a very brief treatment of the binary SVM classification algorithm in this section. Methodologies for extending the analysis to multi-class problems are available in literature [54,55]. Starting with a set of input–output training pairs

$$(\mathbf{x}_1, y_1),\ (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N) \quad \mathbf{x} \in \Re^d,\ y \in \Re.$$

The SVM decision function in terms of an appropriately defined kernel function can be obtained as

$$f(\mathbf{x}) = \sum_{i=1}^{N} y_i \alpha_i K_\theta(\mathbf{x}_i, \mathbf{x}) + b, \tag{4}$$

where N is the sample size and $K_\theta(\mathbf{x}_i, \mathbf{x})$ is the kernel function mapping the input vectors into a feature space and θ a set of parameters and b is bias. The coefficients α_i are obtained by solving the following quadratic optimization problem:

$$\mathbf{w}(\alpha) = \sum_{i=1}^{N} \alpha_i - (1/2) \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K_\theta(\mathbf{x}_i, \mathbf{x}_j) \tag{5}$$

subject to the constraints

$$0 \leqslant \alpha_i \quad i = 1, \ldots, N \tag{6}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{7}$$

If in the above hard margin SVM optimization problem (no explicit provision for penalizing training errors) the equality is satisfied for the points $\mathbf{x}_i$ with the corresponding $\alpha_i > 0$, these nonzero points are called as support vectors. If the separating hyperplane is allowed to pass through the origin by taking $b = 0$, then the equality constraint in Eq. (6) disappears and the problem formulation is called as hard margin SVM without threshold.

In case of non-separable training patterns, the training errors are allowed and the problem formulation in that case is called as soft margin SVM. The inequality constraint in

Eq. (5) is slightly modified as $0 \leqslant \alpha_i \leqslant C$ where C is viewed as a constant penalizing the training errors (i.e. regularization parameter). Soft margin SVM can also be considered as a special case of hard margin SVM [56] with the modified kernel function as

$$K \leftarrow K + \frac{I}{C}.$$

The kernel function appearing in the problem can be selected by using the Mercer's theorem. In our work, we have used Gaussian radial basis function (RBF) kernel of the following form:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\sum_i \frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{2\sigma_i^2}\right), \qquad (8)$$

where σ is the kernel width parameter.

If we take σ to be constant and allow some training error, then kernel parameter C is to be optimized/tuned along with σ to minimize the generalization error. A number of attractive error bounds have been proposed in the literature [56,57] including the more popular radius/margin bound [58].

Vapnik et al. [56] use gradient descent algorithm, while Keerthi et al. [58] use the quasi-Newton updates for automatic tuning. Both these methods can converge to sub-optimal local solutions [59], requiring the use of better methods such as the use of a hybrid framework based on the combination of quasi-Newton (with BFGS update) [60,61] and genetic algorithms. A detailed stepwise procedure for tuning the SVM parameters (to minimize radius/margin bound) is discussed in [37].

6. Experiments

In order to evaluate the proposed method we conducted three experiments over three different data sets. Two data sets "Ionosphere Data", "Wine Recognition Data" are selected from the UCI repository of machine learning databases http://www.ics.uci.edu/~mlearn/MLrepository. The Third data set is the Colon-cancer data from (http://microarray.princeton.edu/oncology).

The ionosphere data set has 34 attributes for a total 351 instances for a binary classification task corresponding to "Good" radar returns and "Bad" returns. The data set is split randomly into two sets: 150 patterns for training and 201 for testing.

The wine data set represents 13 chemical constituents of 178 Italian wines derived from three different cultivars. We have solved here all two-class problems, thus discarding the 48 instances corresponding to third class in wine data. The remaining 130 data points are divided into 80 train and 50 testing instances.

The Colon-cancer data set consists of 62 samples of colon epithelial cells from colon-cancer patients. The samples consist of tumor biopsies collected from tumors, and normal biopsies collected from healthy part of the colons of the same patient. The number of genes in the data set is 2000. The data set is split into two sets: 30 patterns for training and 32 for testing.

7. Results and discussions

The proposed methodology as applied to classification of data sets is illustrated in Fig. 2. The original data can be directly subjected to classification using the SVM algorithm. This classifier is designated as F1 in the flow diagram. The original data may contain some noise and outliers, which can be removed through the process of symbolization. The data so obtained can then be classified and designated as F2. The symbolized data can be further processed to identify the irrelevant and redundant features by computing conditional entropies. Figs. 3–5 show the plots of conditional entropy of class information with respect to all features one-by-one, for the three data sets. As mentioned previously, here conditional entropy is used as relevance filter. We therefore threshold the relevance values to divide

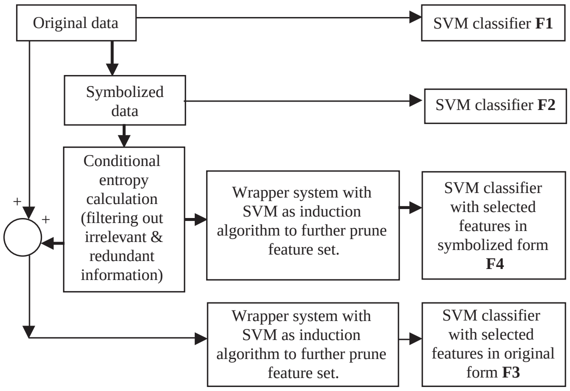
Original data
SVM classifier **F1**
Symbolized data
SVM classifier **F2**
Conditional entropy calculation (filtering out irrelevant & redundant information)
+
+
Wrapper system with SVM as induction algorithm to further prune feature set.
SVM classifier with selected features in symbolized form **F4**
Wrapper system with SVM as induction algorithm to further prune feature set.
SVM classifier with selected features in original form **F3**

Fig. 2. Flow diagram illustrating methodology.

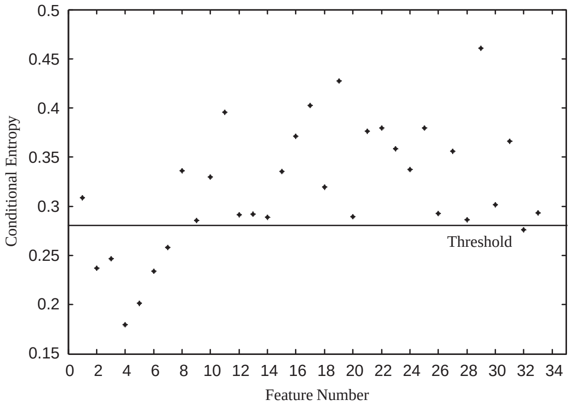

Fig. 3. Conditional entropy of class information with respect to all features one-by-one for ionosphere data.

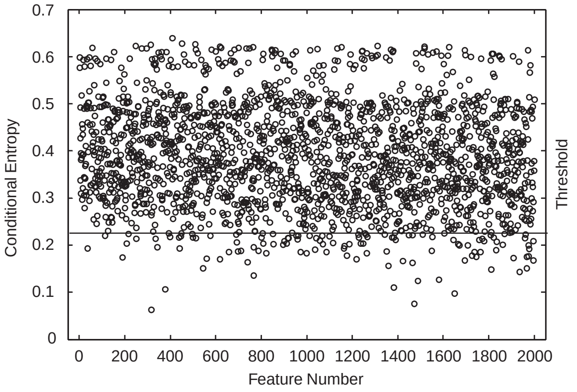

Fig. 4. Conditional entropy of class information with respect to all features one-by-one for colon cancer data.

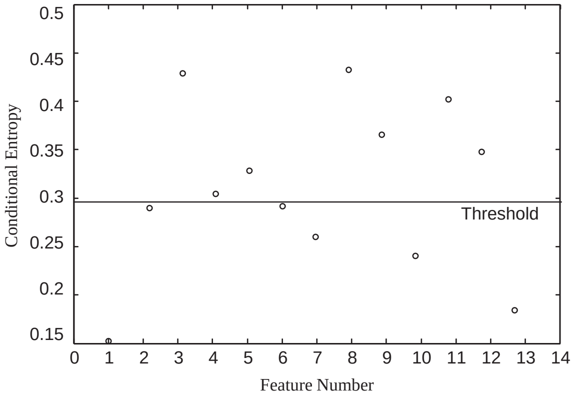

Fig. 5. Conditional entropy of class information with respect to all features one-by-one for wine data.

Table 1
SVM Classifier results for non-noisy case

Classifier type	Ionosphere data	Wine data	Colon cancer data
	Test error (%)	Test error (%)	Test error (%)
F1	12.44	5	18.75
F2	12.44	0	18.75
F3	5.97	2.5	15.63
F4	4.98	0	9.38

Table 2
SVM Classifier results for noisy case $(\mathrm{SNR} = 3)$

Classifier type	Ionosphere data	Wine data	Colon cancer data
	Test error (%)	Test error (%)	Test error (%)
F1	16.42	12.5	15.63
F2	16.42	2.5	12.50
F3	16.42	12.5	15.63
F4	14.93	2.5	15.63

the feature set into relevant and irrelevant features. This is done by thresholding the conditional entropy value directly. Lower values of conditional entropies signify importance of that feature and those above a certain threshold can be considered as irrelevant. In Figs. 3–5 horizontal solid lines represent the user defined threshold value. In Fig. 3, conditional entropy equal to 0.27 represents user-defined threshold and features $\{2, 3, 4, 5, 6, 7\}$ with conditional entropy lower than this threshold constitutes the relevant set. Thus relevance filter alone has reduced the number of features from 34 to 6. In Fig. 5, conditional entropy equal to 0.295 represents user-defined threshold and features $\{1, 2, 6, 7, 10, 13\}$ represents the relevant set. In this case too decrease in relevant features is significant from 13 to 6. Similar trend is seen for colon cancer data (Fig. 4), where conditional entropy equal to 0.23 represents user-defined threshold. In the same way redundancy is removed by computing feature–feature conditional entropies. Sometimes even after removal of irrelevant and redundant information we are left with a large number of features. These can be further pruned using wrapper with SVM as induction algorithm to obtain those select few that are more important. Once the relevant attributes are identified we can use the numerical values associated with them for purpose of classification. This classifier is designated as F3 whereas use of symbols for these attributes gives classifier F4.

In case of wine data we identify attributes 1, 10 and 13 while for ionosphere data the attributes 2,4 and 5 are found to be important. Similarly the colon cancer data give attributes (560, 1745, 765) as optimal sets. To carry out symbolization we used the equal-size intervals over a feature range and thus do not search for the optimum locations of critical points. Shannon entropy is thus maximized with respect to the number of symbols only. The results indicate that even this simplified approach gave excellent results.

To study the effect of noise on classification efficiency, random noise was generated and added to each attribute in the data set before the application of learning algorithms. The classification results for various data sets for noisy and non-noisy cases are presented in Tables 1 and 2. From these results it is clear that the feature selection procedure gives significant improvement to the classifier's performance, both in noisy and non-noisy cases. Also the classification obtained is generally better for symbolized data than for original data. The reason for these improvements is that irrelevant, redundant and noisy information of the data is removed by the combined effect of feature selection and symbolization which minimizes the impact of small amplitude details in the measurements that are not related to the dynamics that dominate the large-scale events.

Table 3
SVM Classifier CPU-TIME for non-noisy case: (seconds)

Classifier type	Ionosphere data	Wine data	Colon cancer data
F1	2.8290	0.6560	3.0000
F2	2.7810	0.5630	2.9530
F3	2.1250	0.4370	0.2340
F4	2.0320	0.4360	0.2190

Table 4
SVM Classifier CPU-TIME for noisy case: (seconds)

Classifier type	Ionosphere data	Wine data	Colon cancer data
F1	2.8600	0.5160	3.1250
F2	2.7970	0.4380	2.9220
F3	2.0470	0.5000	0.2350
F4	1.7350	0.4220	0.2340

Table 5
KNN Classifier results for non-noisy case

Classifier type	Ionosphere data	Wine data	Colon cancer data
	Test error (%)	Test error (%)	Test error (%)
F1	11.99	0	9
F3	8.06	1	8

Additionally a significant advantage of having to deal with only few distinct values is gained. This has a practical advantage of simplifying and speeding up subsequent computations (e.g. wrapper step). The time taken for one time training and testing of SVM classifiers for all three data sets in both non-noisy and noisy cases is presented in Tables 3 and 4. It is clear from the results that the feature selection and symbolization has lessened the computation time considerably. The computational time reported is the CPU time required to carry out one iteration of training and testing of SVM classifier. All the simulations were carried out on Pentium IV, 512 MB RAM machine.

To test the generalization of the method we classify data sets using *k* nearest neighbors (KNN) algorithm with selected features. Note that the data used with KNN is without any symbolization, as symbolization is irrelevant in case of the distance based KNN method. The results for KNN are presented in Tables 5 and 6. As can be seen, performance of the KNN classifier is comparable for both cases: data with total number of features and data with reduced number of features.

The main contribution of this work is the simultaneous execution of feature selection and noise reduction with better generalization of feature selection concept. In the present work we have considered attributes individually while calculating the feature-class and feature–feature couplings and higher order correlations are ignored. These can be included but need large data samples, computation time and cost.

Table 6
KNN Classifier results for noisy case

Classifier type	Ionosphere data	Wine data	Colon cancer data
	Test error (%)	Test error (%)	Test error (%)
F1	15.42	3	9
F3	16.92	3	10

8. Conclusions

In this work, we have presented a novel algorithm to simultaneously discard noisy, irrelevant and redundant information. The benefits of such a preprocessing include a reduction in amount of data needed to achieve learning, improved accuracy and efficiency. Also the hybrid scheme is more general than filter and wrapper alone as demonstrated by KNN results.

Acknowledgements

Financial assistance from Department of Science and Technology (DST), New Delhi is acknowledged gratefully.

References

[1] S. Markovitch, Information filtering: selection mechanisms in learning systems, Ph.D. Thesis, EECS Department, University of Michigan, 1989.

[2] G.H. John, Enhancements to the data mining process, Ph.D. Thesis, Computer Science Department, School of Engineering, Stanford University, 1997.

[3] R. Kohavi, G. John, Wrappers for feature subset selection, Artificial Intel. (special issue on relevance) 97 (1–2) (1996) 273–324.

[4] R. Kohavi, D. Sommerfield, Feature subset selection using the wrapper method: overfitting and dynamic search space topology, in: Proceedings of the First International Conference on Knowledge Discovery and Data Mining,AAAI Press1995.

[5] D. Koller, M. Sahami, Toward optimal feature selection, in: Proceedings of the 13th International Conference on Machine Learning (ML), Bari, Italy, 1996, pp. 284–292.

[6] P. Langley, Elements of Machine Learning, in: B.M. Morgan (Ed.), Morgan Kaufmann, Los Altos, CA, 1996.

[7] P. Langley, S. Sage, Induction of selective Bayesian Classifiers, in: Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence, Seattle, WA, Morgan Kaufmann, Los Altos, CA, 1994, pp. 399–406.

[8] H. Liu, H. Motoda, Feature Extraction Construction and Selection, A Data Mining Perspective, Kluwer Academic Publisher, Norwell, MA, 1998.

[9] A.L. Rendell, R. Sheshu, Learning hard concepts through constructive induction: framework and rationale, Comput. Intel. 6 (1990) 247–270.

[10] R.O. Duda, P.E. Hart, Classification and Scene Analysis, Wiley, New York, 1973.

[11] A. Jain, B. Chandrasekaran, Dimensionality and sample size considerations, in: Krishnaiah, I.N. Kanal (Eds.), Pattern Recognition Practice, Vol. 2, North-Holland, Amsterdam, 1982, pp. 835–855.

[12] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, New York, 1972.

[13] A. Mucciardi, E.E. Gose, A comparison of seven techniques for choosing subsets of pattern recognition properties, IEEE Trans. Comput. 20 (9) (1971) 1023–1031.

[14] J.M. Steppe, K.W. Bauer, Improved feature screening in feedforward neural networks, Neurocomputing 13 (1996) 47–58.

[15] C.J. Matheus, L.A. Rendell, Constructive induction on decision trees, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI),1989, pp. 645–650.

[16] G.H. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: International Conference on Machine Learning,1994, pp. 121–129.

[17] P. Langely, Selection of relevant features in machine learning, in: AAAI Fall Symposium on Relevance,1994, pp. 140–144.

[18] R. Kohavi, Wrappers for performance enhancement and oblivious decision graphs, Ph.D. Thesis, Stanford University, 1995.

[19] P.A. Devijver, J. Kittler, Pattern Recognition: A Statistical Approach, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[20] R. Caruana, D. Freitag, Greedy attribute selection, in: Proceedings of the 11th International Conference on Machine Learning, New Brunswick, NJ, USA, 1994, pp. 28–36.

[21] D.W. Aha, R.L. Bankert, A comparative evaluation of sequential feature selection algorithms, in: D. Fisher, H. Lenz (Eds.), Proceedings of the 5th International Workshop on Artificial Intelligence and Statistics, Ft. Lauderdale, FL, 1995, pp. 1–7.

[22] S. Kirkpatrick, C. Gelatt, M. Vecci, Optimization by simulated annealing, Science 220 (1983) 671–680.

[23] J. Doak, An evaluation of feature selection methods and their application to computer security, CSE Technical Report 92-18, University of California at Davis, 1992.

[24] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Pub. Co., 1989.

[25] H. Almauallium, T.G. Dietterich, Learning with many irrelevant features, Proceedings of Ninth National Conference on Artificial Intelligence, Vol. 2, AAAI Press, Anaheim, CA, 1991, pp. 547–552.

[26] D. Koller, M. Sahami, Toward optimal feature selection, in: Proceedings of the 13th International Conference on Machine Learning (ML), Bari, Italy, 1996, pp. 284–292.

[27] K. Kira, L.A. Rendell, A practical approach to feature selection, in: D. Sleeman, J. Edwards (Eds.), Proceedings of Ninth International Conference on Machine Learning, Alberdeen, Italy, Morgan Kaufmann, Los Altos, CA, 1992, pp. 249–256.

[28] I. Kononenko, Estimating attributes: analysis and extensions of RELIEF, in: European Conference on Machine Learning,1994, pp. 171–182.

[29] C. Cardie, Using decision trees to improve case-based learning, in: Proceedings of the Tenth International Conference on Machine Learning, Amherst, MA, Morgan Kaufmann, Los Altos, CA, 1993, pp. 25–32.

[30] H. Liu, H. Setiono, A probabilistic approach to feature selection: a filter solution, machine learning, in: Proceedings of the 13th International Conference on Machine Learning,Morgan Kaufmann, Los Altos, CA, 1996.

[31] A.L. Blum, R.L. Rivest, Training a 3-node neural network is NP-complete, Neural Networks 5 (1992) 117–127.

[32] L. Hyafil, R.L. Rivest, Constructing optimal binary decision trees is NP-complete, Informat. Process. Lett. 5 (1) (1976) 15–17.

[33] R.K. Azad, J.S. Rao, R. Ramaswamy, Information-entropic analysis of chaotic time series: determination of time-delays and dynamical coupling, Chaos Solitons Fractals 14 (2002) 633–641.

[34] M. Lehrman, A.B. Rechester, R.B. White, Symbolic analysis of chaotic signals and turbulent fluctuations, Phys. Rev. Lett. 78 (1) (1997) 54–57.

[35] C. Cortes, V. Vapnik, Support vector networks, Machine Learn. 20 (1995) 273–297.

[36] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[37] Abhijit Kulkarni, V.K. Jayaraman, B.D. Kulkarni, Support vector classification with parameter tuning assisted by agent based technique, Comput. Chem. Eng. 28 (2004) 311–318.

[38] G. Brida, L.F. Punzo, Symbolic time series analysis and economic regimes, IDEE Working papers series, 2001, pp. 2001–03.

[39] C.E.A. Finney, J.B. Green, C.S. Daw, Symbolic time series analysis of engine combustion measurements, SAE paper no. 980624, 1998.

[40] E.M. Bollt, T. Stanford, Y.C. Lai, K. Zyczkowski, Validity of threshold-crossing analysis of symbolic dynamics from chaotic time series, Phys. Rev. Lett. 85 (16) (2000) 3524–3527.

[41] E.M. Bollt, T. Stanford, Y.C. Lai, K. Zyczkowski, What symbolic dynamics do we get with a misplaced partition?: on the validity of threshold crossings analysis of chaotic time-series, Physica D 154 (2001) 259–286.

[42] J. Kurths, A. Voss, P. Saparin, A. Witt, H.J. Kleiner, N. Wessel, Quantitative analysis of heart-rate variability, Chaos 5 (1) (1995) 88–94.

[43] U. Schwarz, O. Benz, J. Kurths, A. Witt, Analysis of the solar spike events by means of symbolic dynamics methods, Astronomy Astrophys. 277 (1993) 215–224.

[44] X.Z. Tang, E.R. Tracy, Data compression and information retrieval via symbolization, Chaos 8 (3) (1998) 688–696.

[45] X.Z. Tang, E.R. Tracy, A.D. Boozer, A. DeBrauw, R. Brown, Symbol sequence statistics in noisy chaotic signal reconstruction, Phys. Rev. E 51 (5) (1995) 3871–3889.

[46] M.P.S. Brown, W.N. Grundy, D. Lin, N. Cristianini, C.W. Sugnet, T.S. Furey, M. Ares, D. Haussler, Knowledge-based analysis of microarray gene expression data by using support vector machines, Proc. Natl. Acad. Sci. USA 97 (2000) 2–267.

[47] R. Burbidge, M. Trotter, B. Buxton, S. Holden, Drug design by machine learning: support vector machines for pharmaceutical data analysis, Comput. Chem. 26 (2001) 5–14.

[48] Y.D. Caia, S.L. Linb, Support vector machines for predicting rRNA-, RNA-, and DNA-binding proteins from amino acid sequence, Biochim. Biophys. Acta (BBA)—Proteins Proteomics 1648 (1–2) (2003) 127–133.

[49] S. Hua, Z. Sun, A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach, J. Mol. Biol. 308 (2001) 397–407.

[50] Y.F. Sun, X.D. Fan, Y.D. Li, Identifying splicing sites in eukaryotic RNA: support vector machine approach, Comput. Biol. Med. 33 (1) (2003) 17–29.

[51] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowledge Discovery 2 (1998) 121–167.

[52] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge University Press, Cambridge, UK, 2000.

[53] S. Gunn, Support vector machines for classification and regression, ISIS Group Technical Report, University of Southampton, 1998.

[54] J. Weston, C. Watkins, Support vector machines for multi-class pattern recognition, in: M. Verleysen (Ed.), Proceedings of the Seventh ESANN, D. Facto Press, Brussels, 1999, pp. 219–224.

[55] J. Zhu, T. Hastie, Kernel logistic regression and the import vector machine, in: Advances in Neural Information Processing Systems,Vancouver, British Columbia, Canada, 2001, pp. 1081–1088.

[56] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, Choosing multiple parameters for support vector machines, Machine Learn. 46 (2001) 131–160 Online version is at: http://www-connex.lip6.fr/˜chapelle/.

[57] T. Joachims, Estimating the generalization performance of a SVM efficiently, in: Proceedings of the International Conference on Machine Learning,Morgan Kaufman, Los Altos, CA, 2000.

[58] K. Duan, S.S. Keerthi, A.N. Poo, Evaluation of simple performance measures for tuning SVM hyperparameters, Neurocomputing 51 (2003) 41–59.

[59] S.S. Keerthi, Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms, IEEE Trans. Neural Networks 13 (2002) 1225–1229.

[60] T.F. Edgar, D.M. Himmelblau, Optimization of Chemical Processes, McGraw-Hill, New York, 1988.

[61] G.V. Reklaitis, A. Ravindran, K.M. Ragsdell, Engineering Optimization Methods and Applications, Wiley, New York, 1983.

About the Author—RAKESH KUMAR is a project assistant in the chemical engineering division of National Chemical Laboratory, Pune, India. He obtained his bachelor’s degree from Indian Institute of technology, Kharagpur.

About the Author—V.K. JAYARAMAN is a senior scientist in the chemical engineering division of the National Chemical Laboratory, Pune, India (jayaram@che.ncl.res.in). His interests include chemical and bio-reaction engineering, applications of artificial-intelligence tools in engineering, process modeling, optimization and control. Jayaraman has been visiting faculty to Indian universities and has taught many core chemical engineering courses to graduate students. He obtained his bachelor’s and master’s degrees in chemical engineering from the Univ. of Madras and his Ph.D. while working at National Chemical Laboratory. He has over 50 international publications. He has recently received the Herdillia award from the Indian Institute of Chemical Engineers (IIChE) for excellence in basic research.

About the Author—B.D. KULKARNI is a senior scientist and heads the chemical engineering division of the National Chemical Laboratory (NCL), Pune, India. He has been with NCL for over 25 years. His interests are stochastic processes, non-linear systems, chemical reaction engineering, applications of artificial-intelligence tools in engineering, process modeling, optimization and control. A fellow of the Indian National Science Academy, National Academy of Sciences, National Academy of Engineering, and Third World Academy of Sciences, he has received numerous awards for his work. He has published three books and over 200 technical papers in prestigious international journals. Kulkarni obtained his bachelor’s and master’s degrees in chemical engineering from Laxminarayan Institute of Technology in Nagpur, India, and received his Ph.D. degree while working at NCL.