Feature Subset Selection Using a New Definition of Classifiability [⋆]

Ming Dong [a], Ravi Kothari [b,∗]

[a] *Computer Science Department, Wayne State University, Detroit, MI 48202, USA*

[b] *IBM - India Research Lab., Block I, Indian Institute of Technology, Hauz Khas, New Delhi 110016, INDIA*

Abstract

The performance of most practical classifiers improves when correlated or irrelevant features are removed. Machine based classification is thus often preceded by subset selection – a procedure which identifies relevant features of a high dimensional data set. At present, the most widely used subset selection technique is the so-called "wrapper" approach in which a search algorithm is used to identify candidate subsets and the actual classifier is used as a "black box" to evaluate the fitness of the subset. Fitness evaluation of the subset however requires cross-validation or other resampling based procedure for error estimation necessitating the construction of a large number of classifiers for each subset. This significant computational burden makes the wrapper approach impractical when a large number of features are present.

In this paper, we present an approach to subset selection based on a novel definition of the *classifiability* of a given data. The classifiability measure we propose characterizes the relative ease with which some labeled data can be classified. We use this definition of classifiability to systematically add the feature which leads to the most increase in classifiability. The proposed approach does not require the construction of classifiers at each step and therefore does not suffer from as high a computational burden as a wrapper approach. Our results over several different data sets indicate that the results obtained are at least as good as that obtained with the wrapper approach.

Key words: Feature selection, dimensionality reduction, classification.

[⋆] This work was done while the authors were at the University of Cincinnati

[∗] Corresponding author.

1 Introduction

The goal of statistical pattern classification is to assign a class label to an input x on the basis of N labeled (possibly noisy) training patterns $\{(x^{(i)}, t^{(i)})\}_{i=1}^{N}$. Here, $x^{(i)} \in \Re^n$ denotes the input, $t^{(i)} \in \{\omega_1, \omega_2, \ldots, \omega_c\}$ denotes the class label (or the target) corresponding to $x^{(i)}$, and c is the total number of classes. In high dimensional spaces (n is large), features often tend to be *correlated* or *irrelevant* leading to a deterioration of classification performance. For example, the performance of the Naive-Bayes classifier is relatively insensitive to irrelevant features but deteriorates rapidly with correlated features (Langley et al., 1992; Duda and Hart, 1973). On the other hand, the performance of classifiers that rely on some form of distance (for example, a nearest-neighbor classifier which assigns a class label based on the class labels of a certain number of training patterns closest to the input) deteriorates rapidly with irrelevant features. Even when the effect of irrelevant or correlated features is limited (or unexplored for a particular classifier), having fewer inputs can at least lead to simplified or quicker classifier construction (Hartman et al., 1990).

Because of the above considerations, *feature subset selection* is typically used before pattern classification to reduce the number of features. Subset selection requires the definition of a *fitness criteria* to decide on the relevant merits of a subset and a *search criteria* to examine the different subsets. The large number of possible subsets ($2^n - 1 \approx 2^n$) makes an exhaustive search impractical. The *branch and bound* approach works with a monotonic fitness criterion to provide the best subset of a given size (Narendra and Fukunaga, 1977; Fukunaga, 1990) without searching through all subsets. When the computational expense of branch and bound is too large to be acceptable, sequential selection of features as done in *forward selection* or *backward elimination* can be carried out. Unlike branch and bound procedure, forward selection and backward elimination may not find the best subset of a given size. At present, the most widely used method is the so-called *wrapper* approach which uses *hill climbing* (or some other *greedy* search strategy) and the error rate of the classifier itself as the fitness criteria (Kohavi and John, 1997). Since the classifier (for which the lower dimensional subspace is being prepared) is itself used to provide the fitness of a specific subset, features most relevant to the classifier can be chosen. However, because a classifier is constructed for the evaluation of each subset (often several classifiers have to be constructed for each subset; for example when the error rate has to be estimated through cross validation or resampling based methods), the wrapper approach is extremely slow and impractical for high dimensional or very large data sets.

In this paper, we present an approach to subset selection based on a novel definition of the *classifiability* of a given data. The classifiability, as we define it, characterizes the relative ease with which some labeled data can be classified. We use the proposed definition of classifiability to systematically examine each of the remaining features and add the feature which leads to the most increase in classifiability. We stop adding features when the classifiability stops increasing. The proposed approach does not require the construction of multiple classifiers at each step and is thus faster than wrapper approach. On the other hand, our results over several different data sets indicate that the result obtained are at least as good as that obtained with the wrapper approach.

We have laid out the rest of the paper as follows. In Section 2, we discuss the wrapper approach to subset selection in greater detail. We also briefly discuss a less widely used (and less effective) approach - the filter approach. In Section 3, we present a short overview of some existing methods of characterizing the difficulty of a classification problem and then present our definition of classifiability (Dong and Kothari, 2001). In Section 4, we present the algorithm for subset selection based on the proposed classifiability measure. In Section 5 we present some experimental results and compare those results with that obtained with the wrapper approach. In Section 6, we present our conclusions.

2 The Wrapper Approach to Subset Selection

The wrapper approach to feature subset selection is based on using the classifier as a "black box". A search algorithm (such as hill climbing) is used to search for a "good" subset and the classifier is used to find the error rate with a particular subset. However, the true error rate of the classifier with a given subset is hard to compute and an estimate obtained using *cross validation* or bootstrap based methods (Efron and Tibshirani, 1993, 1995) has to be used in lieu of the true error rate. When sufficient bootstrap samples are used the error estimate is usually reliable (Efron and Tibshirani, 1993, 1995).

Typically, a "state vector" of length n (i.e. of the same length as the number of features) is defined. A 1 in the state vector implies inclusion of the corresponding feature and a 0 implies exclusion. To minimize time, wrapper algorithms typically use forward selection, i.e. they start from an empty list of features and add relevant features as they are discovered. The following sequence of steps, adopted from Kohavi and John (1997), illustrate a typical wrapper approach to subset selection based on hill climbing.

(1) Let $v \leftarrow$ empty set of features.

(2) Expand v. Typically, this generates new states by adding or deleting a single feature from v. For example, if $n = 3$ and $v = (0\ 0\ 0)$, then expansion of v might lead to the following states: (1 0 0), (0 1 0), and (0 0 1).

(3) Use the classifier and an error estimation procedure (such as bootstrapping) to find the fitness of each subset that resulted from the expansion of v.

(4) Let v' be the subset with the highest fitness.

(5) If fitness of v' is greater than that of v, $v \leftarrow v'$ and goto step 2. Else terminate with v as the final subset.

There are of course many variations to the above algorithm. For example, it is known that *hill climbing* may get trapped in a local minima. Consequently, better search methods, such as *best-first search* may be used (Russell and Norvig, 1995; Goldberg, 1989). Additionally, one can formulate alternate *operators* to expand v.

Despite the variations, the central aspect of the wrapper approach is that since the classifier is used in the selection process, one can get an accurate estimate of the performance with a given subset. On the other hand if a mechanism other than the classifier is chosen for evaluating the subsets, then a subset which provides poor performance with the actual classifier may be chosen. Of course, this implies that at each pass of the wrapper algorithm requires the construction of $(E * |v|)$ number of classifiers. Here $|v|$ denotes the number of child states of v and E denotes the number of independent classifiers that must be constructed with a given subset to obtain an estimate of the error. For example, if sufficient data is available and a simple estimation procedure such as k-fold cross validation is used, then $E = k$. When sufficient data is not available, and resampling based procedures such as bootstrapping is used then one might require 100—200 classifiers resulting in $E = 100$ or $E = 200$. Clearly, this results in an enormous computational expense and is not feasible for large data sets.

A less widely used approach is the so-called *filter* approach. Algorithms based on the filter approach typically do not consider the classifier (or error estimates obtained from the target classifier) for subset selection. The Relief algorithm (Kira and Rendell, 1992) for example, assigns a “relevance” to each feature. Relevance values are updated by selecting a pattern at random from the data set and finding the difference between it and two “nearest” patterns to the chosen pattern – one of the same class and the other of the opposite class. Due to the random sampling involved in Relief, it is likely that results exhibit a large variance unless the algorithm is run for a very long time. Other approaches in this category include the FOCUS algorithm (**?**), the decision tree based feature selection Cardie (1993), and the PRESS algorithm (Neter et al., 1990).

3 A New Definition of Classifiability

Prior to presenting the proposed measure of classifiability we present a short overview of some of the existing approaches to characterizing the difficulty of a classification problem.

A classical approach to measuring the classifiability uses Fisher’s Discriminant

Ratio (FDR) which in a two-class situation can be defined as,

$$\text{FDR} = \max_{i \in \{1,2,\ldots,n\}} \left\{ \frac{(\mu 1_i - \mu 2_i)^2}{\sigma 1_i^2 + \sigma 2_i^2} \right\} \tag{1}$$

where, $\mu 1_i$ and $\mu 2_i$ denote the mean and $\sigma 1_i^2$ and $\sigma 2_i^2$ denote the variance of the two classes along the i^{th} feature. The maximum along any feature is then used to characterize the problem. Of course, an underlying assumption in FDR is that the class distribution is normal along the individual features which in general is not valid.

Friedman and Rafsky (1979) proposes a Minimum Spanning Tree (MST) is constructed from all the patterns in a sample. The fraction of patterns of opposite classes connected by an edge in the MST is then used as a measure of the classifiability of the sample. Of course as noted in Ho and Baird (2002), even in a simple linearly separable classification problem there might exist a large number of patterns of opposite classes that are close to each other. In effect, as might become evident later, the distribution (structure) of the pattern distribution is ignored in this formulation. In some other approaches, the deviation from linear separability is used a basis for characterizing the classifiability (Smith, 1968; Hoekstra and Duin, 1996).

In some other approaches, the deviation from linearity as measured from the value of the objective function used to obtain a linear classifier is used as a measure of classfiability (Smith, 1968; Hoekstra and Duin, 1996). However, these measures do not accurately capture the classifiability since they rely primarily on the number of misclassifications (or more generally on the value of the objective function) and disregard the distribution of the error. For additional details and a comparative review we refer the reader to Ho and Baird (2002).

Our definition of classifiability is motivated by the fact that a n-dimensional classification problem may be visualized in $(n+1)$ dimensions using the class label as the $(n+1)^{\text{th}}$ dimension. For example, Figure 1 shows a classification problem in 2 dimensions with the corresponding visualization in 3 dimensions.

The class label may thus be viewed as defining a surface which is "rough" when patterns of different classes are near each other and "smooth" when patterns of the same class are adjacent to each other. Naturally, classification is considerably more complicated when the "class label surface" is rough. Consequently, if the smoothness (or roughness) of the class label surface can be quantified, then a natural measure of classifiability is obtained.

This intuitive notion is nicely captured by the second order joint conditional density function $f(\omega_i, \omega_j | d)$, i.e., the probability of going from class ω_i to class ω_j within a distance d[1]. We develop the proposed measure of classifiability

[1] This definition is similar to that used in image processing to characterize the texture of images (Haralick, 1980; Rao, 1990). In the context of image processing,

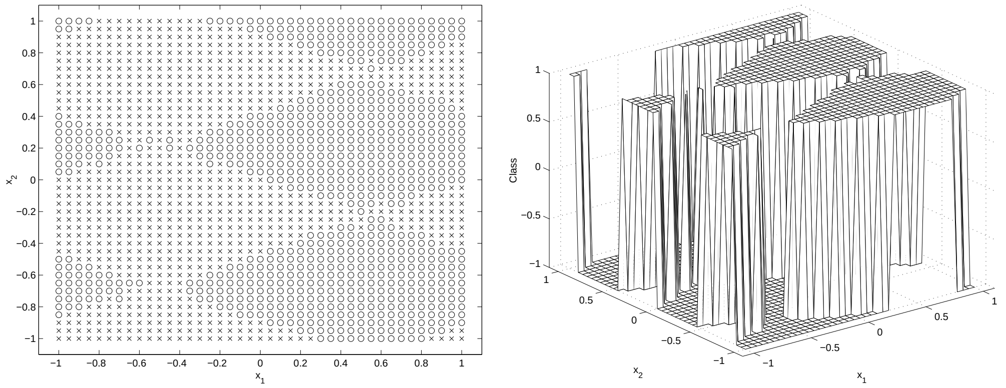

Fig. 1. A 2-class classification problem (left panel) and the visualization in 3-dimensions (right panel).

as follows. For simplicity, and without loss of generality, we consider a 2-class classification problem.

Consider a given training pattern $x^{(i)}$. Let y be a training pattern in the neighborhood (within a distance d) of $x^{(i)}$. One can then define a joint probability matrix for pattern $x^{(i)}$ as,

$$J^{(i)} = \begin{bmatrix} P(\omega_1|y,\omega_1|x^{(i)}) & P(\omega_2|y,\omega_1|x^{(i)}) \\ P(\omega_1|y,\omega_2|x^{(i)}) & P(\omega_2|y,\omega_2|x^{(i)}) \end{bmatrix} \tag{2}$$

Since y and $x^{(i)}$ are independent, this simplifies to,

$$J^{(i)} = \begin{bmatrix} P(\omega_1|y)P(\omega_1|x^{(i)}) & P(\omega_2|y)P(\omega_1|x^{(i)}) \\ P(\omega_1|y)P(\omega_2|x^{(i)}) & P(\omega_2|y)P(\omega_2|x^{(i)}) \end{bmatrix} \tag{3}$$

Note that the matrix $J^{(i)}$ defined in equation (2) will be strongly diagonal when patterns in the neighborhood of $x^{(i)}$ belong to the same class as $x^{(i)}$. Neighboring patterns (i.e. within a distance d) belonging to the same class correspond to a smooth class label surface or easier classification. As the class label surface becomes more rough, the off-diagonal entries become larger.

The classifiability measure for patterns distributed in the neighborhood of a pattern $x^{(i)}$ is thus defined by,

$$\begin{aligned} C(x^{(i)}) = {} & P(\omega_1|y)P(\omega_1|x^{(i)}) + P(\omega_2|y)P(\omega_2|x^{(i)}) \\ & -P(\omega_2|y)P(\omega_1|x^{(i)}) - P(\omega_1|y)P(\omega_2|x^{(i)}) \end{aligned} \tag{4}$$

and the overall classifiability L for the entire data can be defined by,

$$L = \sum_i P(y)\, C(x^{(i)}) \tag{5}$$

the gray level intensities serve the role that the class label serves here.

where y, as before, is a pattern in the neighborhood of a pattern $x^{(i)}$.

Computationally, it is easy to compute the classifiability. One can simply consider a training pattern – say $x^{(i)}$ – and populate $J^{(i)}$ based on fraction of neighboring patterns in the different classes. This provides $J^{(i)}$ and thus $C(x^{(i)})$. $P(y)$ is simply given by the ratio of patterns in the neighborhood of $x^{(i)}$ over N.

It is easy to see that $0 \leq L \leq 1$ and a higher value of L implies greater classifiability. In the next section, we use this definition of classifiability for subset selection.

4 Classifiability Based Subset Selection

The proposed measure of classifiability provides an efficient measure for the subset selection. Our specific method is based on forward selection, where at each stage we add the feature which gives the largest increase in classifiability. The complete algorithm is shown below. In the algorithm we have used the shorthand notation $L(v)$ to denote the classifiability as computed with all the features in v and we have used ϵ as a user specified parameter representing the minimum acceptable increase in classifiability with each added feature.

```
Let $v = \{\emptyset\}$ and let $s = \{x_1, x_2, \ldots, x_n\}$
for i = 1 to length(s) do
  Find $\arg\max_i L(v')$ where $v' = v \cup x_i$
    if $(L(v') - L(v)) > \epsilon$
      $v = v \cup x_i$
      $s = s - x_i$
      $i = 1$
      continue
    else
      break
    end if
end for
Return $v$ as the final subset
```

In the proposed algorithm for feature subset selection, the need for constructing multiple classifiers does not arise since the classifiability (for a fixed d) does not depend on random sampling, initial conditions or other factors that can alter results from one run to another. Therefore, unlike as in the wrapper based approach multiple classifiers do not need to be constructed.

It is easy to compute the worst case running time for the proposed algorithm. In the worst case, all the input features may have to be included in the subset. In that situation, the maximum number of times that the classifiability has to be computed is given by $\sum_{i=1}^{n} i = n(n+1)/2$. Each time the classifiability is computed one has to compute $(N-1)$ distances to count the points which are within the neighborhood d of $x^{(i)}$. This has to be done for each of the N points. So the total complexity of the proposed subset technique in the worst case is given by $O(N^2n^2)$. In practice, the size of the subset is much lesser than n and the complexity in practice is significantly less. In addition, when the distance computation are done in parallel (or using some efficient data structures), then the actual complexity can be quite modest.

5 Experimental Results

We present our experimental results in 4 separate groups. The first group consists of a single simulation and is intended to highlight that our definition of classifiability is in fact robust to the addition of irrelevant features. The second group of simulations is based on 8 separate data sets. These data sets are the ones used with the wrapper approach (Kohavi and John, 1997) and thus allow for a direct comparison of the proposed method with the wrapper approach. The third group of simulations is based on two large data sets (one with 60 features and one with 649 features) that are widely available but were not used in the wrapper approach reported earlier (Kohavi and John, 1997). This group provides further evidence of the effectiveness and efficiency of the proposed method. The last group of simulations is used to show the effect of varying neighborhood when selecting feature subset based on classifiability measure. We found out that the classifiability is quite robust to the change of the neighborhood size. So is the feature selection results.

In all our simulations, we normalized each feature to lie in the range $[a, a+1]$, where a is some constant. We achieved this by dividing values of a feature by the difference between the maximum and the minimum values of the feature. Also, in all our simulations (except those in the last group) neighborhood size d is set to be $3\times$ the RMS distance of each pattern from its nearest neighbor. In our distance computation, we use the Euclidean distance when a feature has a numeric value. For symbolic features we fixed the distances between two dissimilar symbolic features to be 1.

5.1 Simulation Group I

We present results with a synthetic data set to illustrate that ideally (i.e. when we have a large number of samples) the classifiability of a data set does not change when we add some irrelevant features. In this simulation, there are three attributes and two classes labeled class 1 and class 2. The first attribute for each class is obtained by sampling from a Gaussian distribution ($\mu_1 = 0$ and $\sigma = 1$ for class 1, and $\mu_2 = 1$ and $\sigma = 1$ for class 2). The second and third attributes are random numbers uniformly generated in $[0, 1]$. We constructed three different data sets by using attribute 1 only, attribute 1 and 2 and attribute 1, 2 and 3. The last two data sets thus contain some irrelevant features. For each data set, we calculate the classifiability 5 times (with a different sample each time) and the results are summarized in Table 1. We can clearly see that the difference between the classifiability of first data set and the classifiability of the second and third data set decreases as the sample size increases. In the limit where a large number of samples are present, the classifiability will be constant for all three data sets.

From the simulation, we also observe that the classifiability varies with the sample size although the effect is not very significant. This can be explained by the third criterion we mentioned in Section 3. The proposed measure of classifiability, like other empirical techniques, provides more accurate estimates with increasing sample size.

Table 1
Classifiability L of three data sets with different sample size. All results are reported as mean ± standard deviation computed from 5 independent trials.

# of Patterns	L with 1 Attr	L with 2 Attr	L with 3 Attr
200	0.6081 ± 0.0253	0.5109 ± 0.0260	0.3722 ± 0.0187
400	0.5556 ± 0.0710	0.4960 ± 0.0340	0.3858 ± 0.0358
800	0.5628 ± 0.0510	0.5214 ± 0.0354	0.4411 ± 0.0403
1600	0.5378 ± 0.0163	0.5043 ± 0.0094	0.4302 ± 0.0144

5.2 Simulation Group II

We present the results with 8 different data sets to illustrate that the proposed feature subset selection criterion can achieve similar or better performance compared with wrapper approach. We used two typical classification algorithms, ID3 (Quinlan, 1986) and Naive-Bayes (Langley et al., 1992; Duda and Hart, 1973; Fukunaga, 1990) to evaluate the classification accuracy on the original data set (with all features) and on the subset as chosen by us. Results are reported based on 10 fold cross-validation irrespective of whether there is separate testing set or not. When separate training and testing data sets are present, we simply merge them into one data set. All instances with missing value are discarded. These test conditions are identical to the ones used with the wrapper approach (Kohavi and John, 1997) and thus allow for a direct comparison.

All data sets can be obtained from the MLC++ Machine Learning Library (www.sgi.com/tech/mlc) along with some additional documentation. We provide the salient characteristics of each of the data sets.

Cleve Data Set: The first data set in this group is the Cleve data set, which has a total of 14 (8 symbolic, 6 numeric) attribute and two classes: sick or healthy. Our proposed algorithm chooses 5 attributes (see Table 2). The cross-validation accuracy of feature subset is improved for both classification algorithms (see Table 3). For comparison, the wrapper approach chooses 2.6 (ID3) or 3.1 (Naive Bayes)[2] chosen are shown as fractional attributes and results in better classification performance.

Corral Data Set: The Corral data set is an artificial data set. It has 6 attributes: $A0$, $A1$, $B0$, $B1$, Irr and $Correlated$. The target concept is $(A0 \cap A1) \cup (B0 \cap B1)$. Irr is an irrelevant attribute, and $Correlated$ is an attribute highly correlated with the class label, but with a 25% error rate. Our proposed algorithm chooses $A0, A1, B0, B1$ and $Correlated$ as the feature subset (see Table 2). For ID3, the cross-validation accuracy of feature subset is the same, while the accuracy is improved for Naive-Bayes classifier (see Table 3)). For comparison, the wrapper approach will choose only 1 attribute and results in worse classification performance.

Crx Data Set: The Crx data set is based on credit card applications. There are a total of 15 attributes and 2 classes. As Table 2 and 3 show, we choose 4 attributes and the accuracy increases for both the classification algorithms. For comparison, the wrapper approach chooses 2.9 (ID3) or 1.6 (Naive Bayes) attributes and results in Better (ID3) or worse (Naive-Bayes) classification performance.

M of n $3-7-10$ **Data Set:** This data set is again an artificial data set. It has 10 attributes and 7 of which (numbers $2, 3, 4, 5, 6, 7, 8$) are relevant to the class label. Table 2 shows that our algorithm chooses 7 features, $3, 4, 5, 6, 7, 8, 9$, as the subset and Table 3 shows accuracy is improved. The wrapper approach results 0 attributes chosen and a corresponding decrease in accuracy.

MONK's Problem: For the next three simulations, we consider the well known MONK's data sets. The MONK's data sets are actually three subproblems. The domains for all MONK's problems are the same. There are 432 instances that belong to two classes and each instance is described by 7 attributes $(a1, \ldots, a7)$. Among the 7 attributes, there is one ID attribute (a unique symbol for each instance), which is not related to classification and is ignored in our simulations.

MONK-1: The target concept associated with the MONK-1 problem is $(a1 == a2)$ `OR` $(a5 == 1)$. Table 2 summarizes the results obtained. We choose totally three attributes in the order of $a5$, $a1$, $a2$, which is a good match with the target concept. The cross-validation accuracy is shown in Table 3. For comparison purpose, wrapper approach will choose only 1

[2] The number of features reported are the number of features chosen averaged over 10 independent trials (Kohavi and John, 1997). Hence the fractional number of features.

attribute and results in worse classification performance.

MONK-2: The target concept associated with the MONK-2 problem is: exactly two of $(a1 == 1, a2 == 1, a3 == 1, a4 == 1, a5 == 1, a6 == 1)$. Table 2 shows the results obtained. Proposed algorithm choose all 6 attributes. The cross-validation accuracy is shown in Table 3. For comparison purpose, wrapper approach will choose only 0 attribute and results in worse classification performance.

MONK-3: The target concept associated with the MONK-3 problem is $(a5 == 3$ `AND` $a4 == 1)$ `OR` $(a5 \neq 1$ `AND` $a2 \neq 3)$. 5% noise is added to the training set. Results obtained are shown in Table 2. Totally 4 attributes are chosen. The cross-validation accuracy is shown in Table 3. For comparison purpose, wrapper approach chooses 2 attributes and results in better (ID3) or same (Naive Bayes) classification performance.

Pima Data Set: The last data set is the Pima data set. It has 2 classes, 8 attributes and a total of 768 instance. Tables 2 and 3 show that 3 attributes are chosen and accuracy increases. For comparison, the wrapper approach chooses 1 (ID3) or 3.8 (Naive Bayes) attributes and results in better (ID3) or worse (Naive Bayes) classification performance.

Table 2
The number of features in the original data set and the number of features retained in the subset. The features retained in the subset are also shown in the order of selection.

	Data set	Original	Subset	Features #
1	Cleve	13	5	10,13,12,3,9
2	Corral	6	5	6,1,2,3,4
3	Crx	15	4	8,9,13,10
4	mofn-3-7-10	10	7	4,9,5,8,3,6,7
5	monk1	6	3	5,1,2
6	monk2	6	6	3,6,1,2,4,5
7	monk3	6	4	2,5,4,1
8	pima	8	3	2,8,1

From these 8 simulations, we can see clearly that proposed approach performs at least as well as the wrapper approach when using real data sets, while does much better than the wrapper approach when using artificial data sets. Figure 2 provides a pictorial comparison of the proposed method and the wrapper based approach to subset selection when the Naive-Bayes classifier is used. In either case, the proposed approach is faster than the wrapper approach [3].

[3] The differences in CPU time are trivial for these data sets because it usually takes only several seconds (0-5 sec) for both approaches on today's PC. The slight differences were also affected by many factors other than the algorithm itself, for

Table 3
Cross-validation accuracy for ID3 and Naive-Bayes classifier with the entire data set (all features) and the subset (selected features). All results are reported as mean ± standard deviation computed from 10 independent trials.

Data set		Full Set		Subset	
		ID3	Naive-Bayes	ID3	Naive-Bayes
1	Cleve	73.31 ± 4.26	83.51 ± 1.38	76.23 ± 2.25	84.17 ± 1.82
2	Corral	96.92 ± 2.05	80.83 ± 8.79	96.92 ± 2.05	86.03 ± 3.75
3	Crx	81.16 ± 1.04	77.68 ± 1.56	85.65 ± 1.3	84.06 ± 1.33
4	mofn-3-7-10	83.67 ± 2.19	87.33 ± 1.63	84.33 ± 1.22	89.33 ± 1.56
5	monk1	95.12 ± 1.36	74.97 ± 1.95	100.00 ±0.00	74.97 ± 1.95
6	monk2	46.05 ± 3.02	66.22 ± 2.80	46.05 ± 3.02	66.22 ± 2.80
7	monk3	100.00 ± 0.00	97.22 ± 0.47	100.00 ± 0.00	97.22 ± 0.47
8	pima	70.56 ± 1.66	75.90 ± 1.88	71.73 ± 1.38	73.43 ± 1.57

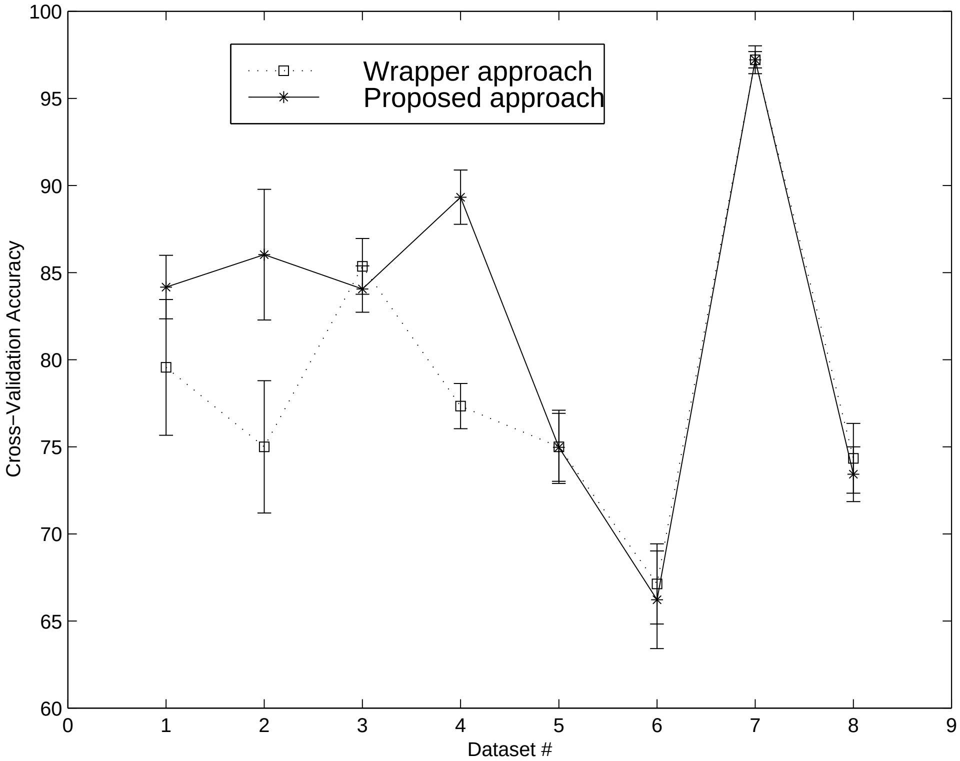

Fig. 2. Comparison with wrapper approach (Naive Bayes classifier).

example, the number of features selected, the status of CPU. Hence the CPU time is not reported here. The CPU time on two large data sets is reported in the following section.

5.3 Simulation Group III

In this group of simulations, we present the results with two widely used data sets each with a large number of features. As such this group provides further evidence of the effectiveness and efficiency of the proposed subset selection strategy. As before, we used ID3 (Quinlan, 1986) and Naive-Bayes (Langley et al., 1992; Duda and Hart, 1973; Fukunaga, 1990) to evaluate the classification accuracy on the original data set (with all features) and on the subset as chosen by us. Results are reported based on 10 fold cross-validation irrespective of whether there is separate testing set or not. When separate training and testing data sets are present, we simply merge them into one data set. All instances with missing values are discarded. The simulation was done on the Pentium IV 1.8 GHz PC with 256M memory running Microsoft Windows XP Professional Edition.

We present a brief description of each of the data sets and summarize the subset selected for each data set in Table 4 and the cross-validation accuracy obtained with the original and reduced data sets in Table 5.

Sonar Data Set: The first simulation is based on Sonar data set, which has 208 instances and 2 classes. Out of 60 possible attributes, our algorithm chooses 2, shown in Table 4. From Table 5, we can see clearly that the cross-validation accuracy of feature subset is improved for both classification algorithms.

Multi-Feature Digit Data Set: The second simulation is based on Multi-Feature Digit data set, which has 1000 instances and 5 classes. Out of 649 possible attributes, our algorithm chooses 10 as shown in Table 4. The cross-validation accuracy is shown in Table 5.

Once again, it is clear that the proposed algorithm succeeds in selecting a subset with far fewer features than the original data set while improving the testing accuracy. We can also clearly see the efficiency of proposed method based on CPU time reported in Table 4.

5.4 Simulation Group IV

This group of simulations is intended to estimate the effect of neighborhood size when doing subset selection based on the proposed classifiability measure. For that purpose, we chose data sets Crx and Pima and varied the neighborhood size from $2 * r$ to $8 * r$ as shown in the second column of Table 6 (r is the RMS distance of each pattern from its nearest neighbor). As the results in Table 6 show, there exists a large range of values of d for which the same feature subset is selected. For example, the subset for the Crx data set always contains features 8, 9, 13 and 10 when the neighborhood size d equals $2r$, $3r$ or $5r$. When $d = 8r$, those four features were picked again with priority although two more features (6 and 12) were included.

Table 4
The number of features in the original data set, the number of features retained in the subset and corresponding CPU time. The features retained in the subset are shown in the order of selection.

		Prop. Method			Wrapper Approach		
Data set	Original	Subset	Features #	CPU (sec)	Subset	Features #	CPU (sec)
Sonar	59	2	12,16	3	11	11,19 18,37 27,43 55,29 9,41 42	107
Digit	649	10	362,48 475,133 358,582 86,638 645,359	2150	9	361, 295 132, 294 152, 289 643, 359 47	4807

Table 5
Cross-validation accuracy for ID3 and Naive-Bayes classifier with the entire data set (all features) and the subset (selected features).

Data set	Full Set		Subset (Prop. Method)		Subset (Wrapper Appr.)	
	ID3	Naive-Bayes	ID3	Naive-Bayes	ID3	Naive-Bayes
Sonar	72.60 ± 1.05	68.75 ± 0.36	76.44 ± 1.32	71.15 ± 0.25	76.92 ± 1.98	68.75 ± 0.31
Mfeat	98.50 ± 0.87	98.50 ± 1.54	96.25 ± 2.35	98.50 ± 0.54	93.40± 1.32	94.50 ± 0.58

In general, there is no definite way of knowing an appropriate value to use for neighborhood d. Typically, d should increase linearly with n – the number of attributes. Indeed, d should be large enough such that at least a few instances are present within that neighborhood of each instance. d should also be small enough such that classifiability is evaluated locally. However, as Table 6 shows, the proposed method is not overly sensitive to the choice of d. In practice, we found out that 3× the RMS distance of each pattern from its nearest neighbor gives good results.

Table 6
Effect of varying d for the Crx and Pima data set. r is the RMS distance of each pattern from its nearest neighbor

Data set	Neigh.(d)	Subset	Features #
Crx	2r	4	8,9 13,10
	3r	4	8,9 13,10
	5r	4	8,9 13,10
	8r	6	8,9 13,10,6,12
Pima	2r	4	2,8,1,4
	3r	3	2,8,1
	5r	3	2,8,1
	8r	2	2,8

6 Conclusions

In this paper, we described a novel subset selection technique based on a definition of classifiability. The proposed definition of classifiability is based on the general notion of proximity (or overlap) of patterns of opposite classes and is thus unbiased to any particular classifier. We used the proposed definition of classifiability to implement a forward selection based subset selection scheme. More specifically, a feature which maximized the classifiability was added to the subset. Results on the 8 data sets reported earlier with the wrapper approach (Kohavi and John, 1997) confirm that the proposed scheme provides a subset with equal or better performance than the wrapper approach without the need for constructing a large (often in the hundreds) number of classifiers. Based on these results, we believe that the proposed method can be of significant utility in machine based classification of high dimension data.

References

Almuallin, H., Dietterich, T.G., 1991. Learning with many irrelevant features. Proc. 9th Nat. Conf. on AI, 547–552.

Cardie, C., 1993. Using decision trees to improve case based learning. Proc. 10th Int. Conf. on Machine Learning, 25–32.

Duda, R.O., Hart, P.E., 1973. Pattern classification and scene analysis. John Wiley.

Dong, M., Kothari, R., 2001. Look-ahead based fuzzy decision tree induction. IEEE Transactions on Fuzzy Systems 9, 461–468.

Efron, B., Tibshirani, T.J., 1993. An introduction to the Bootstrap. Cahpman and Hall, New York, NY.

Efron, B., Tibshirani, T.J., 1995. Cross-validation and the bootstrap: estimating the error rate of a prediction rule. Technical Report, Stanford University.

Friedman, J.H., Rafsky, L.C., 1979. Multivariate generalizations of the Wald-Wolfowitz and Smirnof two-sample tests. Annals of Statistics, 697–717.

Fukunaga, K., 1990. Introduction to statistical pattern recognition. Academic Press, Boston, MA.

Goldberg, D., 1989. Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading, MA.

Haralick, R.M., 1980. Statistical and structural approaches to texture. Proc. of the IEEE 67, 786–804.

Hartman, E.J., Keeler, J.D., Kowalski, J.M., 1990. Layered neural networks with Gaussian hidden units as universal approximators. Neural Computation 2, 210–215.

Ho, T.K., Basu, M., 2002. Complexity measures of supervised classification problems. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 289–300.

Hoekstra, A., Duin, R.P.W., 1996. On the nonlinearity of pattern classifiers. Proc. 13th Int. Conf. on Pattern Recognition, 271–275.

Kira, K., Rendell, L.A., 1992. The feature selection problem: Traditional methods and a new algorithm. Proc. 10th Nat. Conf. on AI, 129–134.

Kohavi, R., John, G.H., 1997. Wrappers for feature subset selection. Artifical Intelligence 97, 273–324.

Narendra, P.M., Fukunaga, K., 1977. A branch and bound algorithm for feature subset selection. IEEE Transactions on Computers 26, 917-922.

Neter, J., Wasserman, W., Kutner, M.H., 1990. Applied linear statistical models. Third Edition. Richard D. Irwin Inc., Englewood Cliffs, New Jersey.

Langley, P., Iba, W., Thompson, K., 1992. An analysis of Bayesian classifiers. Proc. 10th Nat. Conf. on AI, 223–228.

Quinlan, J.R., 1986. Induction of decision trees. Machine Learning 1, 81–106.

Rao, A.R., 1990. A taxonomy for texture description and identification. Springer-Verlag, New York, NY.

Russell, S.J., Norvig, P., 1995. Artificial intelligence: A modern approach. Prentice Hall, Englewood Cliff, NJ.

Smith, F.W., 1968. Pattern classifier design by linear programming. IEEE Transactions on Computers 17, 367–372.