International Journal of Pattern Recognition and Artificial Intelligence

A Methodology for Improving the Performance of Non-ranker Feature Selection Filters

Lior Rokach
Department of Information Systems Engineering, Ben-Gurion University of the Negev, Israel
liorrk@bgu.ac.il

Barak Chizi
Department of Industrial Engineering, Tel-Aviv University, Israel
barak.chizi@gmail.com

Feature selection is the process of identifying relevant features in the dataset and discarding everything else as irrelevant and redundant. Since feature selection reduces the dimensionality of the data, it enables the learning algorithms to operate more effectively and rapidly. In some cases, classification performance can be improved; in other instances, the obtained classifier is more compact and can be easily interpreted. There is much work done on feature selection methods for creating ensemble of classifiers. Thus, these works examine how feature selection can help ensemble of classifiers to gain diversity. This paper examines a different direction, i.e. whether ensemble methodology can be used for improving feature selection performance. In this paper we present a general framework for creating several feature subsets and then combine them into a single subset. Theoretical and empirical results presented in this paper validate the hypothesis that this approach can help finding a better feature subset.

1. Introduction

Feature selection is a common issue on statistics, pattern recognition and machine learning [7,21]. The aim of feature selection is to distil the most useful subset of features from a given subset.

There are two main strategies for performing feature selection. The first known as *filter* [17] operates independent of any learning algorithm – undesirable features are filtered out of the data before learning begins. These algorithms use heuristics based on general characteristics of the data to evaluate the merit of feature subsets.

The second strategy argues that the bias of a particular induction algorithm should be taken into account when selecting features. The second strategy, known as *wrapper* [17], uses a learning algorithm along with a statistical re-sampling technique such as cross-validation to select the best feature subset for this specific learning algorithm.

Filter methods can be further divided into ranker and non-rankers. Rankers are methods that employ some criterion to score each feature and provide a ranking. From this ordering, several feature subsets can be chosen, either manually of setting

a threshold. A non-ranker methods provide only a selected subset of the features without providing any ranking.

The main advantages of the wrapper methods are: the fact that it generates reliable evolutions and that it can be used for any induction algorithm. Nevertheless the fact that the wrapper procedure repeatedly executes the inducer, is considered major drawback. For this reason, wrappers may not scale well to large datasets containing many features. Filters methods usually run faster and can scale well to larger datasets. However their predictive performance is usually inferior to that of the wrapper methods.

The aim of this paper is theoretically and experimentally examine whether ensemble feature subsets can be used for improving the predictive performance of non-ranker feature selection filters methods without significantly increase the execution time .

The rest of this paper is organized as follows: Section 2 reviews related works in the field of feature selection and the usage of ensemble of feature selectors. Section 3 formulate the problem. Section 4 presents a new algorithm framework suggested to the problem discussed here. Section 5 reports the experiments carried out on a real case study. Section 6 discuss the usefulness of each method in view of the obtained results. Finally, Section 7 concludes the work and presents further research in the field.

2. Related work

First important aspects of features selection algorithms will be presented in Section 2.1. Then in Section 2.2 we will discuss the ensemble methodology and the recent researches that combine feature selection with the ensemble methodology.

2.1. *Feature Selection Algorithms*

Feature selection algorithms search through the space of feature subsets in order to find the best subset. This subset search has two major properties [18]:

- Search Organization - How the search space of all possible feature subsets is searched. Because it is not practical to perform an exhaustive search of all possible feature subsets, there is a need of a heuristic search. Several heuristic searches have been examined for feature selection, including: greedy hill climbing methods [21] (with forward selection [16,5], backward elimination or stepwise bi-directional search), best first search [26] (similar to hill climbing but with backtracking capabilities) and genetic algorithms [35].
- Evaluation Strategy - How feature subsets are evaluated. As mentioned above there are two main evaluation strategies: filters and wrappers. Several evaluation methods have been developed in the literature. In this paper we mainly use the Correlation-based Feature Subset Selection (CFS) as a subset evaluator [11]. CFS Evaluates the worth of a subset of attributes by

considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred. Beside the CFS, we will consider consistency subset evaluator [19].

2.2. *Ensembles and Feature Selection*

The main idea of ensemble methodology is to combine a set of models, each of which solves the same original task, in order to obtain a better composite global model, with more accurate and reliable estimates or decisions than can be obtained from using a single model. The idea of building a predictive model by integrating multiple models has been under investigation for a long time.

In the past few years, experimental studies conducted by the machine-learning community show that combining the outputs of multiple classifiers reduces the generalization error [29]. Ensemble methods are very effective, mainly due to the phenomenon that various types of classifiers have different "inductive biases". Indeed, ensemble methods can effectively make use of such diversity to reduce the variance-error [32], without increasing the bias-error. In certain situations, an ensemble can also reduce bias-error, as shown by the theory of large margin classifiers [2].

A common strategy for manipulating the training set is to manipulate the input attribute set. The idea is to simply give each classifier a different projection of the training set. Ensemble feature selection methods [25] extend traditional feature selection methods by looking for a set of feature subsets that will promote disagreement among the base classifiers. Ho [12] has shown that simple random selection of feature subsets may be an effective technique for ensemble feature selection because the lack of accuracy in the ensemble members is compensated for by their diversity. Tsymbal and Puuronen [31] presented a technique for building ensembles of simple Bayes classifiers in random feature subsets.

The hill climbing ensemble feature selection strategy [6], randomly construct the initial ensemble. Then, an iterative refinement is performed based of hill-climbing search in order to improve the accuracy and diversity of the base classifiers. For all the feature subsets, an attempt is made to switch (include or delete) each feature. If the resulting feature subset produces better performance on the validation set, that change is kept. This process is continued until no further improvements are obtained.

The Genetic Ensemble Feature Selection strategy uses genetic search for ensemble feature selection [25]. It begins with creating an initial population of classifiers where each classifier is generated by randomly selecting a different subset of features. Then, new candidate classifiers are continually produced by using the genetic operators of crossover and mutation on the feature subsets. The final ensemble is composed of the most fitted classifiers.

An approach for constructing an ensemble of classifiers using rough set theory

was presented in [14]. The method searches for a set of reducts, which include all the indispensable attributes. A reduct represents the minimal set of attributes which has the same classification power as the entire attribute set.

Oliveira et al. [22] suggest creating a set of feature selection solutions using a genetic algorithm. Then they create a Pareto-optimal front in relation to two different objectives: accuracy on a validation set and number of features. Following that they select the best feature selection solution. Masulli and Rovetta [20] have employed ensemble methodology for feature selection. Nevertheless their method was specifically developed for micro-array data, and no general framework was proposed.

Several researchers examined the possibility of using multi-objective genetic algorithms (MOGA) to create ensemble of classifiers based on feature selection. MOGAs are based on the Pareto dominance concept. Oliveira et al. [23] proposed a hierarchical multi-objective genetic algorithm that especially effective when classifiers have to work with very low error rates. Their algorithm works in two levels. In the first level, feature selections methods are used to generate a set of classifiers minimizing two criteria: misclassification rate and number of features. In the second level the algorithm chooses the best set of classifiers and combine them by maximizing the following two criteria: accuracy of the ensemble and a measure of diversity. Radtke et al. [24] examined a similar approach, however in the second level instead of optimizing a diversity metric, they optimize the number of active classifiers in order to reduce computation time during classification.

Recently Torkkola and Tuv [30] and Tuv and Torkkola [33] examined the idea of using ensemble classifiers such as decision trees in order to create a better features ranker. They have showed that this ensemble can be very effective in variable ranking for problems with up to a hundred thousand input attributes. Note that this approach uses inducers for obtaining the ensemble. Thus, it concentrates on wrapper feature selectors.

There is much work done on feature selection methods for creating ensemble of classifiers. These works examine how feature selection can help ensemble of classifiers to gain diversity. Nevertheless there is hardly works that examine the other way around, i.e. how can ensemble of feature selectors improve the feature selection results.

3. Problem Definition and Theoretical Observations

The problem of feature selection ensemble is that of finding the best feature subset by combining a given set of feature selectors, such that if a specific inducer is run on it, the generated classifier will have the highest possible accuracy. Following Kohavi and John [17] we adopt the definition of optimal feature subset with respect to a particular inducer.

Definition 1. *Given an inducer I, a training set S with input feature set* $A = \{a_1, a_2, ..., a_n\}$ *and target feature* y *from a fixed and unknown distribution D over*

the labeled instance space, the subset $B \subseteq A$ is said to be optimal if the expected generalization error of the induced classifier $I(\pi_{B\cup y}S)$ will be minimized over the distribution D.

where $\pi_{B\cup y}S$ represents the corresponding projection of S and $I(\pi_{B\cup y}S)$ represent a classifier which was induced by activating the induction method I onto dataset $\pi_{B\cup y}S$.

Definition 2. *Given an inducer I, a training set S with input feature set $A = \{a_1, a_2, ..., a_n\}$ and target feature y from a fixed and unknown distribution D over the labeled instance space, and an optimal subset B, a Feature Selector FS is said to be consistent if it selects an attribute $a_i \in B$ with probability $\mathrm{p} > 1/2$ and it selects an attribute $a_j \notin B$ with probability $\mathrm{q} < 1/2$.*

Definition 3. *Given a set of feature subsets $B_1, ..., B_\omega$ the majority combination of features subsets is a single feature subset that contains any attribute a_i such that $f_c(a_i, B_1, ..., B_\omega) > \frac{\omega}{2}$ where $f_c(a_i, B_1, ..., B_\omega) = \sum_{j=1}^{\omega} g(a_i, B_j)$ and $g(a_i, B_j) =$*

$$\begin{cases} 1 \; a_i \in B_j \\ 0 \; otherwise \end{cases}$$

Definition 3 refers to a simple majority voting, in which attribute a_i is included in the combined feature subset if it appears in at least half of the base feature subsets $B_1, ..., B_\omega$, where ω is the number of base feature subsets. Note that $f_c(a_i, B_1, \ldots, B_\omega)$ counts the number of base feature subsets in which a_i included.

Lemma 1. *A majority combination of feature subsets obtained from a given a set of independent and consistent feature selectors $FS_1, \ldots, FS_\omega$ (where ω is the number of feature selectors) converges to the optimal feature subset when $\omega \to \infty$.*

Proof. For ensuring that for attributes for which $a_i \in B$ are actually selected we need to show that:

$$\lim_{\omega\to\infty, p>1/2} p\left(f_c(a_i) > \frac{\omega}{2}\right) = 1 \tag{1}$$

We denote by $p_{j,i} > 1$ the probability of FS_j to select a_i. We denote by $p_i = \min(p_{j,i})$. Note that $p_i > \frac{1}{2}$. Because the feature selectors are independent we can use approximation binomial distribution, i.e.:

$$\lim_{\omega\to\infty} p\left(f_c(a_i) > \frac{\omega}{2}\right) \leq \lim_{\omega\to\infty, p_i>1/2} \sum_{k=0}^{\frac{\omega}{2}} \binom{\omega}{k} p_i^k (1-p_i)^{\omega-k} \tag{2}$$

Due to the fact that $\omega \to \infty$ we can use the central limit theorem in which, $\mu = \omega p_i, \sigma = \sqrt{\omega p_i(1-p_i)}$:

$$\lim_{\omega\to\infty,p_i>1/2}\sum_{k=0}^{\frac{\omega}{2}}\binom{\omega}{k}p_i^k(1-p_i)^{\omega-k}=\lim_{\omega\to\infty p_i>1/2}p\left(Z>\frac{\frac{\omega}{2}-p_i\omega}{\sqrt{\omega p_i(1-p_i)}}\right)=$$

$$\lim_{\omega\to\infty,p_i>1/2}p\left(Z>\frac{\sqrt{\omega}(1/2-p_i)}{\sqrt{p_i(1-p_i)}}\right)=p\left(Z>-\infty\right)=1 \quad (3)$$

For ensuring that for attributes for which $a_i \notin B$ are actually selected we need to show that:

$$\lim_{\omega\to\infty}p\left(f_c(a_i)<\frac{\omega}{2}\right)=0 \quad (4)$$

We denote by $q_{j,i}<1/2$ the probability of FS_j to select a_i. We denote by $q_i=\max(q_{j,i})$. Note that $q_i<\frac{1}{2}$. Because the feature selectors are independent we can use approximation binomial distribution, i.e.:

$$\lim_{\omega\to\infty}p\left(f_c(a_i)<\frac{\omega}{2}\right)\geq\lim_{\omega\to\infty,q_i<1/2}\sum_{k=0}^{\frac{\omega}{2}}\binom{\omega}{k}q_i^k(1-q_i)^{\omega-k} \quad (5)$$

Due to the fact that $\omega\to\infty$ we can use the central limit theorem again this time: $\mu=\omega q_i, \sigma=\sqrt{\omega q_i(1-q_i)}$:

$$\lim_{\omega\to\infty,q_i<1/2}\sum_{k=0}^{\frac{\omega}{2}}\binom{\omega}{k}q_i^k(1-q_i)^{\omega-k}=\lim_{\omega\to\infty,q_i<1/2}p\left(Z>\frac{\frac{\omega}{2}-q_i\omega}{\sqrt{\omega q_i(1-q_i)}}\right)=$$

$$\lim_{\omega\to\infty,q_i<1/2}p\left(Z>\frac{\sqrt{\omega}(1/2-q_i)}{\sqrt{q_i(1-q_i)}}\right)=p\left(Z>\infty\right)=0 \quad (6)$$

□

4. Independent Algorithmic Framework

Roughly speaking, the feature selectors in the ensemble can be created dependently or independently. In the dependent framework the outcome of a certain feature selector affect the creation of the next feature selector. Alternatively each feature selector is built independently and their results are combined in some fashion. In this paper we concentrate on independent framework. Figure 1 presents the proposed algorithmic framework. This simple framework gets as an input the following arguments:

(1) A Training set (S) – A labeled dataset used for feature selectors.
(2) A set of feature selection algorithms $\{FS_1,\ldots,FS_\xi\}$ – A feature selection algorithm is an algorithm that obtains a training set and outputs a subset of relevant features. Recall that in this paper we employ non-wrapper and non-ranker feature selectors.
(3) Ensemble Size (ω)
(4) Ensemble generator (G) – This component is responsible for generating a set of ω pairs of feature selection algorithms and their corresponding training sets. We refer to G as a class that implements a method called "genrateEnsemble".

(5) Combiner (C) – The combiner is responsible to create the subsets and combine them into a single subset. We refer to C as a class that implements the method "combine".

The proposed algorithm simply uses the ensemble generator to create a set of pairs of feature selection algorithms and their corresponding training sets. Then it call the combine method in C to execute the feature selection algorithm on its corresponding dataset and then combine the various feature subsets into a single subset.

Require: S, $\{FS_1, \ldots, FS_\xi\}$, G, C
Ensure: A combined feature subset.
1: $\{(S_1, FS_1), \ldots, (S_\omega, FS_\omega)\} \leftarrow$ G.genrateEnsemble $(S, (FS_1, \ldots, FS_\xi), \omega)$
2: Return C.combine $(\{(S_1, FS_1), \ldots, (S_\omega, FS_\omega)\})$

Fig. 1. Pseudo-code of Independent Algorithmic Framework for Feature Selection

4.1. *Combining Procedure*

We begin by describing two implementations for the combiner component. In the literature there are two types of methods to combine the results of the ensemble members: weighting methods and meta-learning methods. In this paper we concentrate on weighting methods. The weighting methods are best suited for problems where the individual members have comparable success or when we would like to avoid problems associated with added learning (such as over-fitting or long training time).

4.1.1. *Simple Weighted Voting*

Figure 2 presents an algorithm for selecting a feature subset based on the weighted voting of feature subsets. As this is an implementation of the abstract combiner used in Figure 1, the input of the algorithm is a set of pairs; every pair is built from one feature selector and a training set. It executes the feature selector on its associated training set to obtain a feature subset. Then the algorithm employs some weighting method and attaches a weight to every subset. Finally it uses a weighted voting to decide which attribute should be included in the final subset. We considered the following methods for weighting the subsets:

(1) **Majority Voting** – In this weighting method the same weight is attached to every subset such that the total weights is 1, i.e. if there are ωsubsets then the weight is simply$1/\omega$. Note that the inclusion of a certain attribute in the final result requires that this attribute will appear in at least $\omega/2$ subsets. This method should have a low false positive rate, because selecting an irrelevant

attribute will take place only if at least $\omega/2$ feature selections methods will decide to select this attribute.

(2) **"Take-It-All"** – In this weighting method all subsets obtain a weight that is greater than 0.5. This leads to the situation in which any attribute that has been in at least one of the subsets will be included in the final result. This method should have a low false negative rate, because loosing a relevant attribute will take place only if all feature selections methods will decide to filter out this attribute.

(3) **"Smaller is Heavier"** – The weight for each selector is defined by its bias to smallest subset. Selectors that tend to provide small subset will gain more weight than selectors that tend to provide large subset. This approach is inspired by the fact that the precision rate of selectors tend to decrease as the size of the subset increases. This approach can be used to avoid noise caused by feature selectors that tend to select most of the possible attributes. More specifically the weights are defined as (note that in this case the weights are normalized and sum up to 1):

$$w_i = \frac{\frac{|B_i|}{\sum_{j=1}^{\omega}|B_j|}}{\sum_{k=1}^{\omega}\frac{|B_k|}{\sum_{j=1}^{\omega}|B_j|}} \tag{7}$$

Require: $\{(S_1, FS_1), \ldots, (S_\omega, FS_\omega)\}$
Ensure: A Combined feature subset
1: **for all** $(S_i, FS_i) \in F$ **do**
2: $B_i = \mathrm{FS}_i.\mathrm{getSelectedFeatures}(S_i)$
3: **end for**
4: $\{w_1, \ldots, w_\omega\} = getWeight\,(\{B_1, ..., B_\omega\})$
5: $B \leftarrow \emptyset$
6: **for all** $a_j \in A$ **do**
7: totalWeight=0
8: **for** $i = 1$ to ω **do**
9: **if** $a_j \in B_i$ **then**
10: totalWeight $\leftarrow$ totalWeight+W_i
11: **end if**
12: **end for**
13: **if** totalWeight> 0.5 **then**
14: B $\leftarrow B \cup a_j$
15: **end if**
16: **end for**
17: Return B

Fig. 2. Pseudo-code of combining procedure

4.1.2. *Naïve Bayes Weighting using Artificial Contrasts*

Using Bayesian approach a certain attribute should be filtered out if:

$P(a_i \notin B\,|B_1,...,B_\omega) > 0.5$ or $P(a_i \notin B\,|B_1,...,B_\omega) > P(a_i \in B\,|B_1,...,B_\omega)$ where $B \subseteq A$ denote the set of relevant features

By using the Bayes Theorem we obtain:

$$P(a_i \notin B\,|B_1,...,B_\omega) = \frac{P(B_1,...,B_\omega\,|a_i \notin B)P(a_i \notin B)}{P(B_1,...,B_\omega)} \tag{8}$$

However calculating the above probability as-is might be difficult. Thus we are using the Naïve Bayes combination. This is a well-known combining method due to its simplicity and its relatively outstanding results. According to the naïve Bayes assumption, the results of the feature selectors are independent given the fact that the attribute a_i is not relevant. Thus, using this assumption we obtain:

$$\frac{P(B_1,...,B_\omega\,|a_i \notin B)P(a_i \notin B)}{P(B_1,...,B_\omega)} = \frac{P(a_i \notin B)\prod_{j=1}^{\omega} P(B_j\,|a_i \notin B\,)}{P(B_1,...,B_\omega)} \tag{9}$$

Using Bayes Theorem again:

$$\frac{P(a_i \notin B)\prod_{j=1}^{\omega} P(B_j|a_i \notin B\,)}{P(B_1,...,B_\omega)} = \frac{P(a_i \notin B)\prod_{j=1}^{\omega} \frac{P(a_i \notin B|B_j\,)}{P(a_i \notin B)} P(B_j)}{P(B_1,...,B_\omega)} = \frac{\prod_{j=1}^{\omega} P(B_j)\prod_{j=1}^{\omega} P(a_i \notin B|B_j\,)}{P(B_1,...,B_\omega)\cdot P^{\omega-1}(a_i \notin B)} \tag{10}$$

Thus a certain attribute should be filtered out if:

$$\frac{\prod_{j=1}^{\omega} P(B_j)\prod_{j=1}^{\omega} P(a_i \notin B|B_j\,)}{P(B_1,...,B_\omega)\cdot P^{\omega-1}(a_i \notin B)} > \frac{\prod_{j=1}^{\omega} P(B_j)\prod_{j=1}^{\omega} P(a_i \in B|B_j\,)}{P(B_1,...,B_\omega)\cdot P^{\omega-1}(a_i \in B)} \tag{11}$$

or after omitting the common term from both sides:

$$\frac{\prod_{j=1}^{\omega} P(a_i \notin B|B_j\,)}{P^{\omega-1}(a_i \notin B)} > \frac{\prod_{j=1}^{\omega} P(a_i \in B|B_j\,)}{P^{\omega-1}(a_i \in B)} \tag{12}$$

Assuming that the a-priori probability for a_i to be relevant is equal to that of not being relevant:

$$\prod_{j=1}^{\omega} P(a_i \notin B\,|B_j\,) > \prod_{j=1}^{\omega} P(a_i \in B\,|B_j\,) \tag{13}$$

Using the complete probability theorem:

$$\prod_{j=1}^{\omega} P(a_i \notin B\,|B_j\,) > \prod_{j=1}^{\omega} (1 - P(a_i \notin B\,|B_j\,)) \tag{14}$$

Because we are using non-ranker feature selectors the above probability is estimated using:

$$P(a_i \notin B \,|B_j\,) \approx \begin{cases} P(a \notin B \,|a \in B_j\,)\; if\, a_i \in B_j \\ P(a \notin B \,|a \notin B_j\,)\; if\, a_i \notin B_j \end{cases} \tag{15}$$

Note that $P(a \notin B \,|a \in B_j\,)$ does not refer to a specific attribute, but to the general bias of the feature selector j. In order to estimate the remaining probabilities, we are adding to the dataset a set of ϕ contrast attributes that are known to be truly irrelevant and analyzing the number of artificial features ϕ_j included in the subset $B_j$obtained by the feature selector j:

$$P(a \in B_j \,|a \notin B\,) = \frac{\phi_j}{\phi} \quad ; \quad P(a \notin B_j \,|a \notin B\,) = 1 - \frac{\phi_j}{\phi} \tag{16}$$

The artificial contrast variables are obtained by randomly permuting the values of the original n attributes across m instances. Generating just random attributes from some simple distribution, such as Normal Distribution, is not sufficient, because the values of original attributes may exhibit some special structure. Using Bayes Theorem:

$$P(a \notin B \,|a \in B_j\,) = \frac{P(a \notin B)P(a \in B_j \,|a \notin B\,)}{P(a \in B_j)} = \frac{P(a \notin B)}{P(a \in B_j)}\frac{\phi_j}{\phi} \tag{17}$$

$$P(a \notin B \,|a \notin B_j\,) = \frac{P(a \notin B)P(a \notin B_j \,|a \notin B\,)}{P(a \notin B_j)} = \frac{P(a \notin B)}{1 - P(a \in B_j)}(1 - \frac{\phi_j}{\phi}) \tag{18}$$

where $P(a \in B_j) = \frac{|B_j|}{n+\phi}$

4.2. *Feature Ensemble Generator*

In order to make the ensemble more effective, there should be some sort of diversity between the feature subsets. Diversity may be obtained through different presentations of the input data or variations in feature selector design.

4.2.1. *Multiple Feature Selectors*

In this approach we simply use a set of different feature selection algorithms. The basic assumption is that using different algorithms have different inductive biases' and thus they will create different feature subsets.

In this paper we examined the proposed method mainly using the Correlation-based Feature Subset Selection (CFS) as a subset evaluator [11]. As for the search organization the following methods have been examined: Best First Search [26], Forward Selection Search [16] by using Gain Ratio [28], Chi-Square [15], OneR classifier [13], and Information Gain [27].

Beside the CFS, we also have considered other evaluation methods such as consistency subset evaluator [19] and the the wrapper subset evaluator with simple classifiers (K-nearest neighbors [1], logistic regression [4] and naïve bayes [9])

4.2.2. *Bagging*

The most well-known independent method is bagging (bootstrap aggregating). In this case each feature selector is executed on a sample of instances taken with replacement from the training set. Usually each sample size is equal to the size of the original training set. Note that since sampling with replacement is used, some of the instances may appear more than once in the same sample and some may not be included at all. So the training samples are different from each other, but are certainly not independent from statistics point of view

5. Experimental Study

In order to illustrate the theoretical results shown above, a comparative experiment has been conducted on benchmark data sets. The following subsections describe the experimental set-up and the obtained results.

5.1. *Dataset Used*

The selected algorithms were examined on 10 data sets of which have been selected manually from the UCI Machine Learning Repository. The datasets chosen vary across a number of dimensions such as: the number of target classes, the number of instances, the number of input features and their type (nominal, numeric).

5.2. *Algorithms Used*

Table 1 presents 10 feature ensemble alternatives examined in this experiment. The first column indicates the abbreviation used to denote each alternative. All multiple generators used 5 different feature selection algorithms. Recall that all the algorithms use Correlation-based Feature Subset Selection (CFS) as a subset evaluator. The algorithms differ by their search method: Best First Search (BFS), Forward Selection Search by using Gain Ratio, Chi-Square, OneR classifier, and Information Gain. The bagging approach was used by employing the CFS with BFS as a search method.

5.3. *Evaluation Method*

Based on the problem formulation described above, the main goal of the feature selection is to minimize the generalization error of a particular inducer.

J48 algorithm is used as the induction algorithm. J48 is a java version of the well-known C4.5 algorithm [28].

In order to estimate the generalization error 10-fold cross-validation procedure was used. Each dataset was randomly divided into 10 equal parts in order to provide 10 different iterations of feature selection and classification. For each iteration 1/10 of the dataset was used as the test set and 9/10 of the dataset was used as train. All experiments were performed in the WEKA framework [34].

Table 1. Accuracy Results of Various Ensemble Alternatives

Alternative	Generator	Combiner
BMV10	Bagging of size 10	Majority Voting
BMV5	Bagging of size 5	Majority Voting
BSH10	Bagging of size 10	"Smaller is Heavier"
BSH5	Bagging of size 5	"Smaller is Heavier"
BTA10	Bagging of size 10	"Take-It-All"
BTA5	Bagging of size 5	"Take-It-All"
MMV	Multiple	Majority Voting
MNB	Multiple	Naïve Bayes Weighting using Artificial Contrasts
MSH	Multiple	"Smaller is Heavier"
MTA	Multiple	"Take-It-All"

5.4. *Predictive Power Results*

Table 2 summarizes the experimental results of the various feature ensemble implementations. It can be seen that the MTA (Multiple Take-It-All) implementation achieved on average the best results. On the other hand MNB (Multiple Using Naïve Bayes) obtained on average the worst results. Still there is one dataset (Bridges) in which MNB had significantly outperforms MTA. All other methods achieved on average comparable results.

For benchmarking the ensemble approach, all the feature selection algorithms mentioned above were separately experimented on the same datasets. Moreover we examined the result obtained with no feature selection. Table 3 summarizes the comparison of MTA with these algorithms. The second column indicates the accuracy obtained by employing J4.8 algorithm on the original feature set (i.e. without running any feature selection). The third column refers to the accuracy of the MTA approach (as it also appears in Table 2). The subsequent columns (columns 4 to 8) indicate the accuracy obtained by employing a single filter feature selection approach followed by J4.8 algorithm. The filter feature selection methods that were examined are the same methods that were used in generation of the ensemble (see Section 4.2.1). The superscript "+" indicates that the accuracy rate of MTA was significantly higher than the corresponding algorithm at confidence level of 5%. The "–" superscript indicates the accuracy was significantly lower.

When observing the results, two important observations appear: First, as seen on Table 3, MTA did much better than other methods for feature selection. Applying t-test (paired two sample for means) validates the above observation, by providing $p < 0.05$ for all methods vs. MTA. Statistically the empirical results validate the theoretical results shown on Section 3. Ensemble method may yield better feature subset when the goal is to improve classification accuracy.

Another important observation from Table 3 indicates that employing MTA before using induction algorithm provides better results than not using MTA. Using

Table 2. Accuracy Results of Various Ensemble Alternatives

Dataset	BMV10	BMV5	BSH10	BSH5	BTA10	BTA5	MMV	MNB	MSH	MTA
Arrhythmia	68.13	67.93	69.51	67.87	64.44	65.25	66.89	66.18	67.34	68.71
Audiology	72.02	71.49	72.02	71.49	73.35	73.48	71.81	64.98	71.89	77.54
Balance	78.18	78.18	78.18	78.18	78.18	78.18	77.61	78.18	77.61	78.18
Bridges	57.88	58.43	57.88	57.60	58.18	58.73	58.45	62.62	58.17	58.43
Car	77.50	77.50	77.50	77.50	78.05	77.50	77.50	77.50	77.50	86.39
Kr-vs-kp	90.34	90.34	90.34	90.34	90.34	90.34	90.34	71.15	90.34	90.69
Letter	85.90	85.90	85.90	85.90	85.90	85.90	85.90	85.53	85.90	85.97
Pendigits	95.18	95.14	95.03	95.14	95.21	95.21	95.20	95.29	95.20	95.63
Soybean	88.95	89.08	88.78	89.08	88.61	88.65	88.52	87.79	88.52	88.44
Spambase	92.03	91.94	91.94	91.94	92.44	92.03	92.34	91.46	92.27	92.38
Splice	93.30	93.30	93.30	93.30	93.59	93.26	93.30	89.72	93.30	93.30

Table 3. Classification Results

Datasets	J48 Without FS	MTA	CFS-BFS	CFS-Forward Selection Search-Gain Ratio	CFS-Forward Selection Search-Chi Square	CFS-Forward Selection Search-OneR	CFS-Forward Selection Search-Information Gain
Arrhythmia	61.84^{+}	68.71	68.19	68.58	66.89	66.04	64.93^{+}
Audiology	75.00	77.53	72.01^{+}	73.27^{+}	71.40^{+}	71.73^{+}	71.89
Balance	74.40^{+}	78.17	77.61^{+}	78.17	77.61^{+}	77.61^{+}	77.61^{+}
Bridges	55.55^{+}	58.43	57.59^{+}	58.19	58.16	58.43	58.16
Car	89.45	86.38	77.50^{+}	77.50^{+}	77.50^{+}	86.38	77.50^{+}
Kr-vs-kp	99.63^{-}	90.68	90.33	71.15^{+}	90.33	90.33	90.33
Letter	85.92	85.97	85.89	85.37^{+}	85.89	85.97	85.89
Pendigits	95.93	95.62	95.04	95.25	95.19	95.62	95.19
Soybean	88.12	88.44	88.57	88.44	88.44	88.44	88.45
Spambase	92.20	92.37	92.00	90.51^{+}	92.37	90.88	92.06
Splice	91.88^{+}	93.29	93.29	89.72^{+}	93.29	93.29	93.29

t-test (paired two sample for means) validates the above by providing $p < 0.05$ for J48 without any feature selection procedure vs. MTA. When handling noisy data sets which involved irrelevant and redundant information, feature selection can provides better classification accuracy. Nevertheless, this improvement is not guaranteed. Some feature selection techniques might reduce the accuracy in certain datasets (see for instance the Audiology dataset). However in this experimental study, it becomes evident that MTA has almost never reduced the accuracy of the inducer (the only dataset in which MTA has significantly reduced accuracy was Kr-vs-kp). Thus, the MTA can be referred as a more reliable preprocessing step for induction algorithm.

5.5. *Dimensionality Reduction Results*

Table 4 presents the number of features selected by each one of the methods. Not surprisingly MTA has selected the largest feature subset. This is can be explained by the fact that MTA takes any feature that appears at least once. As for BTA5 and BTA10 (which also use the "Take-It-All" strategy), their feature subset size

is smaller than the MTA, because the random bagging can be considered as pre-filtering which results with smaller base feature subsets.

In most of the cases there is correlation between the classification accuracy and the feature subsets. This observation indicates that filters methods usually over-filter the feature set, namely these methods remove relevant features that can contribute to the classifier performance.

5.6. *Comparing to Wrapper approach*

In this section we compare the performance of the MTA (Multiple-Take-It-All) and MMV (Majority Voting) with the performance of a GA-based feature selector using wrapper approach as a subset evaluator. The wrapper evaluator was set to perform 5 folds while using the J4.8 as the base classifier. Following [35] we set the GA according to the following parameter settings:

Probability of crossover: 0.6 Number of generations: 20 Probability of mutation: 0.001 Population Size: 50

Moreover we also examine the performance of revised versions of MTA and MMV that employ wrapper as a subset evaluator (instead of CFS) and denote them as MTA-W and MMV-W respectively.

Table 5 presents the obtained results. For each method we provide the obtained accuracy and the number of selected features. The superscript "-" indicates that the accuracy rate of MTA was significantly lower than the corresponding algorithm at confidence level of 5%. The results indicate that in most of the cases the wrapper methods outperform the filters ensemble. However it is interesting to note that MTA-W and MMV-W have obtained comparable results to GA-based feature selection.

Using only classification accuracy as the only performance measure, is in practice not necessarily optimal, as the feature selectors that are given more execution time may have a higher accuracy, but also an overall higher cost. Thus, there is a trade-off between the execution time and the classification accuracy. Table 6 provides information regarding the execution time of the feature selection and the subsequent induction of the classifier. GA method is around 500 times slower than MTA. Thus, the practitioner should decide if she is ready to compromise accuracy for getting the results faster. Note that the execution time of MMV is faster than MTA, because MMV creates smaller feature subsets which results with a shorter training time.

5.7. *Filtering irrelevant attributes*

Following Guyon [10] we tested the ability of the various proposed feature selectors to filtering irrelevant attributes. For this purpose we added to every training date 100 random features distributed similarly to the real features in the same way we created the artificial contrast variables (see Section 4.1.2). In what follows we refer to such features as probes to distinguish them from the real features. This will allow us to rank algorithms according to their ability to filter out irrelevant

features. The method with smallest number of random probes in the feature set wins. Table 7 presents the mean number of features selected by MTA and MMV, the mean number of probes selected and the accuracy over 10-fold cross validation. The results indicate that MMV is usually more efficient in filtering random probes. However this capability is gained by compromising the classifier accuracy.

5.8. *Using various subset evaluators*

In this section we repeat the above experimental study using the same search organization methods but but instead of using a single subset evaluator (CFS) for all base feature selectors, we are using the following five different subset evaluators:

- Consistency Subset evaluator
- CFS
- Wrapper Subset evaluator with the K-nearest neighbors classifier
- Wrapper Subset evaluator with simple logistic regression
- Wrapper Subset evaluator with simple naïve bayes classifier

Note that in evaluators 3 to 5 (wrapper evaluation methods) we have not used the targeted classifier (J4.8) but simple classifiers that were selected due to their training speedup when compared to more discriminant approaches.

Table 8 presents the results obtained by using various subset evaluators. As oppose to the first experiment (see Table 2), the new results indicate that the difference between MTA and MMV are now negligible. Both MNB and MSH obtained almost the same accuracy performance as MTA. However MTA used 3 or 4 times more features than the MNB and MSH respectively.

6. Discussion

It appears from the experimental study that the MTA method (Multiple-Take-It-All) obtained the best results because it compensates in part the weakness of filter feature selection methods. This is especially true if the base attribute selectors select only a small portion of the original features and accidently filter out unnoticeable but still relevant features (over-filtering). The over-filtering can be explained by the fact the filters methods does not take into account the targeted classifier. The experimental study indicates that majority voting (MMV) might not be sufficient to compensate over filtering. This argument is supported by the results presented in Table 5. When wrapper approach is used instead of filtering and subsequently over-filtering is less likely, then MMV-W performs as well as MTA-W. Thus MTA is preferred approach if either the cost or the chance of losing relevant information is high . On the other hand, MMV is preferred if the dataset has many irrelevant features because the chance that an irrelevant feature will be selected by at least half of the feature selectors is low. The results presented in Table 7 supports this argument by showing that MMV was capable to filter out the irrelevant features in 7 of 11 cases (as oppose of 4 of 11 cases with MTA).

The relatively poor results of MNB (Multiple Using Naïve Bayes) in Table 2 might be explained by the fact that the feature selectors are not really independent as MNB requires. Recall that all feature selectors used the CFS as a subset evaluator. Thus all base feature selectors are biased to the same subset. However when different subset evaluators have been used, MNB provided comparable accuracy but with a smaller feature subset.

The results of MSH and MMV in Table 2 are almost identical. This can be explained by the fact that if the base feature subsets are similar in size then MSH behaves similarly to MMV. In fact if the size of all subsets is identical ($|B_j| = |B_i|$) then the weights of all subsets is $\frac{1}{\omega}$. The MSH strategy might provide a different result when all feature subsets have similar accuracy performance but have different sizes. However when various subset evaluators are used then the results of MSH depart from the results of MMV (see in Table 8). Thus, MSH provides a potential contribution when the base feature selectors are diverse.

7. Conclusions

This paper examines theoretically and experimentally whether ensemble of feature subsets can be used for improving feature selection performance.

Theoretically, it has been shown that using an ensemble method provides better subset than using other feature selection technique. As mentioned on Lemma 1, the probability of choosing "good" attribute for the subset is much higher than on other feature selection technique. More, this probability becomes higher as the size of the ensemble become bigger. Empirically, the experiments shown on this paper provides validation for the theoretical results. This paper examines several methods for ensemble feature selection.

From the empirical study, we can conclude that ensemble approach can efficiently achieve high degree of dimensionality reduction and enhance or maintain predictive accuracy with selected features. Selecting the most suitable ensemble method for a given problem depends on the dataset characteristics (such as the portion of irrelevant features), the base feature selectors and the decision maker preferences regarding the possibility of losing relevant information.

Additional issues to be further studied include: examining the effectiveness of the proposed methodology using other inducers like neural networks and developing a similar methodology for ranker feature selector filters.

Acknowledgments

The author would like to thank Prof. Oded Maimon for his helpful and inspiring comments.

References

1. Aha, D., and D. Kibler (1991) "Instance-based learning algorithms", Machine Learning, vol.6, pp. 37-66

2. Bartlett P. and Shawe-Taylor J., Generalization Performance of Support Vector Machines and Other Pattern Classifiers, In "Advances in Kernel Methods, Support Vector Learning", Bernhard Scholkopf, Christopher J. C. Burges, and Alexander J. Smola (eds.), MIT Press, Cambridge, USA, 1998.
3. Buhlmann, P. and Yu, B., "Boosting with L_2 loss: Regression and classification," Journal of the American Statistical Association, 98, 324–338. 2003.
4. le Cessie, S. and van Houwelingen, J.C. (1992). Ridge Estimators in Logistic Regression. Applied Statistics, Vol. 41, No. 1, pp. 191–201.
5. Chizi B., Maimon O. and Smilovici A., On Dimensionality Reduction of High Dimensional Data Sets, in Frontiers in Artificial Intelligence and Applications. IOS press, pp. 230-236, 2002.
6. Cunningham P., and Carney J., Diversity Versus Quality in Classification Ensembles Based on Feature Selection, In: R. L. de Mántaras and E. Plaza (eds.), Proc. ECML 2000, 11th European Conf. On Machine Learning,Barcelona, Spain, LNCS 1810, Springer, 2000
7. Devijver, P. and Kittler, J. *Pattern Recognition: A Statistical Approach*. Prentice Hall International, 1982.
8. Dimitriadou E., Weingessel A., Hornik K., A cluster ensembles framework, Design and application of hybrid intelligent systems, IOS Press, Amsterdam, The Netherlands, 2003.
9. Duda R. and Hart P., Pattern Classification and Scene Analysis. Wiley, New York, 1973.
10. Guyon I., Gunn S., Ben-Hur A., Dror G., Design and Analysis of the NIPS 2003 variable selection Challenge, In: Feature Extraction: Foundation and applications, Physcia-Verlag, Springer, 2006.
11. Hall, M. Correlation- based Feature Selection for Machine Learning. University of Waikato, 1999.
12. Ho T. K. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8):832–844, 1998.
13. Holte R.C., Very simple classification rules perform well on most commonly used datasets. Machine Learning, Vol. 11, pp. 63-91, 1993.
14. Hu, X., Using Rough Sets Theory and Database Operations to Construct a Good Ensemble of Classifiers for Data Mining Applications. ICDM01. pp. 233-240, 2001.
15. Kass G. V., An exploratory technique for investigating large quantities of categorical data. Applied Statistics, 29(2):119-127, 1980.
16. Kittler J., Feature set search algorithms, In: C. H. Chen, Ed. , Pattern Recognition and Signal Processing pp. 41 - 60. Sijthoff and Noordhoff, Alphen ann den Rijn, The Netherlands, 1978
17. Kohavi R. and John, G. Wrappers for feature subset selection. *Artificial Intelligence*, special issue on relevance, 97(1– 2): 273– 324, 1996
18. Langley, P. and Sage, S., Scaling to domains with irrelevant features. In R. Greiner, (Ed.), Computational Learning Theory and Natural Learning Systems, volume 4. MIT Press, 1994.
19. Liu H. and Setiono R., A probabilistic approach to feature selection, pages 319–327. Morgan Kaufmann, 1996.
20. Masulli, F. and Rovetta, S. Random Voronoi ensembles for gene selection in DNA microarray data, in Udo Seiffert and Lakhmi C. Jain, editors, Bioinformatics using Computational Intelligence Paradigms, World Scientific Publishing, Singapore, 2003
21. Miller, A. Subset Selection in Regression. Chapman and Hall, New York, 1990.
22. Oliveira L.S., Sabourin R., Bortolozzi F., and Suen C.Y. A Methodology for Feature

Selection using Multi-Objective Genetic Algorithms for Handwritten Digit String Recognition, International Journal of Pattern Recognition and Artificial Intelligence, 17(6):903-930, 2003.

23. Oliveira L.S., Morita M. E., Sabourin R., Bortolozzi F., Multi-objective Genetic Algorithms to Create Ensemble of Classifiers: in Proc. of the Third International Conference on Evolutionary Multi-Criterion Optimization,(EMO 2005), Guanajuato, Mexico, March 9-11, 2005, Lecture Notes in Computer Science 3410 Springer 2005, ISBN 3-540-24983-4, pp. 592-606.
24. Radtke P. V.W., Sabourin R., Wong T., Intelligent Feature Extraction for Ensemble of Classifiers, 8th International Conference on Document Analysis and Recognition (ICDAR 2005), pp. 866-870, Seoul, South Korea, August 29-September 1st, ISBN 0-7695-2420-6, 2005
25. Opitz, D., Feature Selection for Ensembles, In: Proc. 16th National Conf. on Artificial Intelligence, AAAI, pages 379-384, 1999.
26. Pearl, J. Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley, 1984. p. 48.
27. Quinlan, J., Simplifying decision trees, International Journal of Man-Machine Studies, 27, 221-234, 1987.
28. Quinlan, J. *C4.5: Programs for machine learning*. Morgan Kaufmann, Los Altos, California, 1993.
29. Quinlan, J. Induction of decision trees. *Machine Learning*, 1: 81– 106, 1996
30. Torkkola, K. and Tuv, E. Variable selection using ensemble methods. *IEEE Intelligent Systems*, 2005, (Vol. 20, No. 6): 68-70.
31. Tsymbal A., and Puuronen S., Ensemble Feature Selection with the Simple Bayesian Classification in Medical Diagnostics, In: Proc. 15thIEEE Symp. on Computer-Based Medical Systems CBMS'2002, Maribor, Slovenia,IEEE CS Press, 2002.
32. Tumer, K. and Ghosh J., Error Correlation and Error Reduction in Ensemble Classifiers, Connection Science, Special issue on combining artificial neural networks: ensemble approaches, 8 (3-4): 385-404, 1999.
33. Tuv, E. and Torkkola, K. Feature filtering with ensembles using artificial contrasts. In *Proceedings of the SIAM 2005 Int. Workshop on Feature Selection for Data Mining*, Newport Beach, CA, April 23 2005, pp. 69-71.
34. Witten I. H. and Frank E., Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.
35. Yang J. and Honavar V., Feature subset selection using a genetic algorithm. IEEE Intelligent Systems (Sepcial Issue on Feature Transformation and Subset Selection), 13(2):44–49, 1998.

Table 4. Number of features selected by Various Ensemble Alternatives

Dataset	Original Number of Features	BMV10	BMV5	BSH10	BSH5	BTA10	BTA5	MMV	MNB	MSH	MTA
Arrhythmia	280	31.6	28.7	26.9	25.4	92	73.7	36.2	17	45	117.8
Audiology	71	6.6	5.7	5.6	4.3	8.9	7.8	6.3	5	7	51.2
Balance	4	4.3	4.2	4.8	4.3	4.4	4.2	1	1	1	3
Bridges	12	5.5	4.8	3.8	2.8	6.8	5.2	3.2	3	4	12.5
Car	6	2	1.6	2.3	2.4	2.2	2.4	1	1	1	5
Kr-vs-kp	36	3.1	2.7	3.7	3.7	3.6	3.6	3.3	2	3	4.2
Letter	16	9.9	9.5	9.7	9.7	9.3	9.3	9.5	7	9	10.2
Pendigits	16	11	10.2	11.9	11.7	12.7	1.6	13.6	12	13	14.3
Soybean	35	19.8	18.3	19.6	18.7	23.4	23.7	23.4	19	23	32.7
Spambase	58	10.8	10.2	10	10.5	13.8	13.9	11.2	16	11	22.2
Splice	61	6.2	6.1	6.5	6.3	6.2	6.2	6.3	4	6	6.3

Table 5. Comparing Filter Feature Selection Ensemble with Wrapper Methods

Dataset	Original # Featu.	GA Accu.	GA # Featu.	MMV-W Accu.	MMV-W # Featu.	MTA-W Accu.	MTA-W # Featu.	MMV Accu.	MMV # Featu.	MTA Accu.	MTA # Featu.
Arrhythmia	280	70.5752	134.5	69.982	132.1	70.25	136.2	66.89	36.2	68.71	117.8
Audiology	71	76.97	36.7	77.4336	38.1	77.8761	69.2	71.81	6.3	77.54	51.2
Balance	4	76.64	3.7	78.18	3	78.18	3	77.61	1	78.18	3
Bridges	12	60.9524^-	6.2	60.5512^-	5.8	60.5512^-	5.8	58.45	3.2	58.43	12.5
Car	6	92.7662^-	5.2	92.7662^-	5.2	92.3611^-	6.4	77.50	1	86.39	5
Kr-vs-kp	36	98.6546^-	23.4	99.4368^-	35.3	99.4368^-	36.8	90.34	3.3	90.69	4.2
Letter	16	96.11^-	11.5	95.08^-	12.2	96.43^-	13.9	85.90	9.5	85.97	10.2
Pendigits	16	98.9629^-	16	98.9629^-	16	98.9629^-	16	95.20	13.6	95.63	14.3
Soybean	35	91.8009^-	18.5	90.6296	33.2	90.71^-	33.7	88.52	23.4	88.44	32.7
Spambase	58	93.05	34.7	93.93	22.6	93.93	22.6	92.34	11.2	92.38	22.2
Splice	61	94.1693	24.2	93.791	31.5	93.366	32.9	93.30	6.3	93.30	6.3

Table 6. Execution time of Filter Feature Selection Ensemble and Wrapper Methods

Dataset	GA Exec. Time	MMV-W Exec. Time	MTA-W Exec. Time	MMV Exec. Time	MTA Exec. Time
Arrhythmia	12650.5	14523.3	14523.9	16.25	16.52
Audiology	167.41	265.13	269.06	0.67	1.09
Balance	6.06	8.484	8.562	0.17	0.19
Bridges	10.2	11.22	11.32	0.09	0.14
Car	7.67	15.41	12.78	0.25	0.53
Kr-vs-kp	620.74	816.44	739.19	5.55	5.52
Letter	167782.1	136292.92	136307.61	48.75	60.13
Pendigits	10666.47	12799.764	12801.99	24.17	24.7
Soybean	176.09	207.05	209.14	0.72	0.7
Spambase	7339.16	5871.328	5871.73	13.61	13.48
Splice	2638.86	401.37	405.63	21.5	22.31

Table 7. The ability to filter irrelevant attributes

Dataset	Original Number of Features	MMV Num. Features	MMV Num. Probes	MMV Accuracy	MTA Num. Features	MTA Num. Probes	MTA Accuracy
Arrhythmia	280	36.2	0	66.89	122.9	6.2	67.85
Audiology	71	6.3	0.3	68.5841	17.4	2.4	74.3363
Balance	4	1.2	0	76.64	3.2	0.2	78.18
Bridges	12	11.4	3.1	66.6667	18.4	10.1	67.619
Car	6	1.2	0.2	76.5625	39.5	33	91.5509
Kr-vs-kp	36	3.6	0	90.4255	3.8	0.2	90.4255
Letter	16	7	0	83.245	7	0	83.245
Pendigits	16	13.6	0.3	95.8333	14.8	0	96.3337
Soybean	35	23.4	0	88.52	34.2	1.5	88.52
Spambase	58	11.2	0	92.34	22.2	0	92.6972
Splice	61	6.3	0	93.6677	7.4	0	93.4796

Table 8. Using different subset evaluators

Dataset	Original Number of Features	MNB Num. Features	MNB Accuracy	MSH Num. Features	MSH Accuracy	MMV Num. Features	MMV Accuracy	MTA Num. Features	MTA Accuracy
Arrhythmia	280	25.7	68.3628	25.7	68.3628	279	64.3805	279	64.3805
Audiology	71	6.3	68.5841	6.3	68.5841	68.1	77.8761	68.1	77.8761
Balance	4	4	76.64	4	76.64	4	76.64	4	76.64
Bridges	12	4.4	58.0952	4.4	56.1905	12	56.1905	12	56.1905
Car	6	39.5	91.5509	39.5	91.5509	39.5	91.5509	39.5	91.5509
Kr-vs-kp	36	3.6	90.4255	3.8	90.4255	3.6	90.4255	3.8	90.4255
Letter	16	12	95.49	12	95.49	12	95.49	12	95.49
Pendigits	16	13.6	95.8333	14.8	96.3337	13.6	95.8333	14.8	96.3337
Soybean	35	18	89.3119	23.2	90.122	35	91.508	35	91.508
Spambase	58	54	92.9146	10	91.7844	57	92.9798	57	92.9798
Splice	61	7.3	93.4796	7.3	93.4796	61	93.6677	61	94.0752