Large-scale attribute selection using wrappers

Martin Gütlein, Eibe Frank, Mark Hall, Andreas Karwath

Abstract— **Scheme-specific attribute selection with the wrapper and variants of forward selection is a popular attribute selection technique for classification that yields good results. However, it can run the risk of overfitting because of the extent of the search and the extensive use of internal cross-validation. Moreover, although wrapper evaluators tend to achieve superior accuracy compared to filters, they face a high computational cost. The problems of overfitting and high runtime occur in particular on high-dimensional datasets, like microarray data.**

We investigate Linear Forward Selection, a technique to reduce the number of attributes expansions in each forward selection step. Our experiments demonstrate that this approach is faster, finds smaller subsets and can even increase the accuracy compared to standard forward selection. We also investigate a variant that applies explicit subset size determination in forward selection to combat overfitting, where the search is forced to stop at a precomputed "optimal" subset size. We show that this technique reduces subset size while maintaining comparable accuracy.

I. Introduction

Until recently, classification tasks with more than 50 attributes were considered to have a high dimensionality. This is no longer the case. The number of different applications with thousands of attributes is rising, as exemplified by microarray or text classification, and creates a need for techniques that are able to handle a much larger number of attributes. While performing a search for a good attribute subset, it is necessary to evaluate attributes and sets of attributes. *Wrappers* are a popular type of evaluator: they calculate a score for a subset by inducing a classifier using only those attributes. Wrappers tend to lead to superior accuracy, but need high computational effort, compared to so-called *filter* methods. Filters use statistical characteristics of the data for evaluation that are independent of the classifier.

In the attribute selection methods presented in this paper, we modify the standard search technique known as forward selection to yield a computationally efficient wrapper-based attribute selection method for high-dimensional data. To this end we reduce the number of attribute extensions in each step of the forward search. Our experimental results show that this approach leads to competitive results, requires less runtime, and results in less overfitting compared to complete forward selection.

Previous research indicates that extensive search using the wrapper suffers from overfitting [see 15, 7, 18]. Our experiments confirm this, especially in datasets with many irrelevant attributes and a small number of instances, such as microarray data. To further reduce the amount of overfitting we modify our new search techniques based on ideas presented in [20]: we precompute an "optimal" subset size based on cross-validation, and then perform a forward search up to that particular size. Our experiments show that this approach is competitive to a classical wrapper-driven forward selection, and leads to smaller attribute subsets.

This paper is organized as follows. Section II presents related work on improving the wrapper. Our new methods are introduced in Sections III and IV. Section V provides an analysis of the experimental results, followed by Section VI, which summarizes our findings.

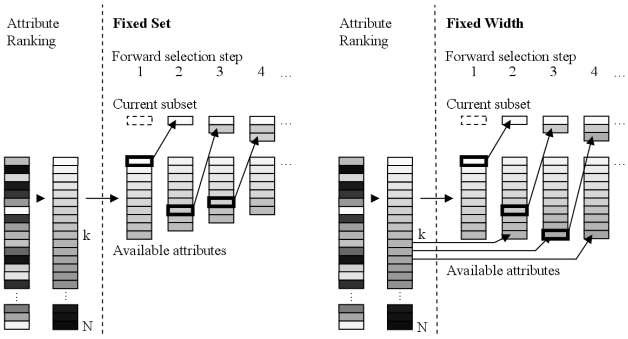
II. Related work on speeding up the wrapper

To reduce the number of subset evaluations, [11] propose a forward search approach that works in two steps. In the first step, all attributes are ranked. This can be done either with a filter method, or with the wrapper. In the second step the algorithm builds N attribute subsets: the first set is the top-ranked attribute, followed by the two top-ranked attributes, the three top-ranked attributes, and so on. These subsets are evaluated using the wrapper, or a filter method that can evaluate sets of attributes. The authors use this technique to compare various filter techniques to the wrapper. With $2 \times N$ evaluations, this algorithm known as Rank-Search is quite fast, but chooses relatively large subsets (see results in Section V-C).

A similar method, called BIRS, is presented in [24]. Again, an initial ranking is produced based on a filter method or the wrapper. The second step constructs attribute subsets and uses the wrapper for evaluation. Similar to the previous method, the algorithm starts with the top-ranked attribute and regards the remaining attributes in order of the ranking, but it only adds an attribute if it improves the current subset significantly. This method also requires $2 \times N$ evaluations, but generates smaller subsets than Rank-Search (see Section V-C). Different criteria for deciding whether a newly expanded subset is significantly better than the current subset were studied in [4].

Further research on the wrapper was done using randomized search. RVE, a randomized wrapper algorithm that is designed for datasets with a large number of irrelevant attributes, is presented in [26]. Huerta et. al [12] first apply fuzzy logic for pre-selecting attributes in microarray data, and then apply a genetic algorithm that uses a wrapper. The same idea of first building a pool of promising attributes and then applying a genetic algorithm to that pool is used in [27].

Martin Gütlein, who was supported by the EU grant HEALTH-F5-2008-200787, and Andreas Karwath are with the Department of Computer Science, Albert-Ludwigs-Universität Freiburg, Germany ({guetlein,karwath}@informatik.uni-freiburg.de). Eibe Frank is with the Department of Computer Science, University of Waikato, Hamilton, New Zealand (eibe@cs.waikato.ac.nz). Mark Hall is with Pentaho Corporation, Orlando, Florida, USA (mhall@pentaho.org).



(a) The fixed-set technique (b) The fixed-width technique

Fig. 1. Linear Forward Selection

III. Limiting the Number of Subset Expansions in Forward Selection

We consider a frequently used forward selection algorithm called Sequential Forward Selection (*SFS*). SFS performs a simple hill-climbing search. Starting with the empty subset, it evaluates all possible single-attribute expansions of to the current subset. The attribute that leads to the best score is added permanently. The search terminates when no single-attribute expansion improves on the current best score. Like [14], we define improvement as an accuracy enhancement of at least ϵ, compared to the current score (we use $\epsilon = 0.0001$).

We further applied our approach to best-first search [14], as well as Sequential Floating Forward Selection [22], and a combination of both. For clarity, we omit these approaches in this paper, as they lead to similar results. Results for these methods can be found in [9].

In the classical SFS approach, the number of evaluations grows quadratically with the number of attributes N: the number of evaluations in each step is equal to the number of remaining attributes that are not in the currently selected subset. The currently selected subset grows with each step, until the algorithm terminates. In the first step, we perform N subset evaluations, in the second step $N-1$ and so on. Thus the upper bound on the number of evaluations is $\sum_{i=0}^{N}(N-i) = \frac{1}{2} \times N(N+1)$.

This quadratic growth can be problematic for datasets with a large number of attributes. In our first approach, called *Linear Forward Selection*, we limit the number of attributes that are considered in each step so that it does not exceed a certain user-specified constant. This drastically reduces the number of evaluations, and therefore improves the runtime performance of the algorithm. We investigate the following two methods for limiting the number of attributes.

1) Fixed Set: Here, we initially rank all attributes and simply select the top-k ranked attributes as input to forward selection. The initial ranking is performed by evaluating each attribute individually and ranking the attributes according to their scores. Scores can obtained using a wrapper evaluator on a per-attribute basis (as we do for the experiments in this paper) or using a filter criterion. Only the k best attributes are employed in the subsequent forward selection and the rest is discarded. This very simple method reduces the upper bound on the number of evaluations considered during the search process to $\frac{1}{2} \times k(k+1)$, regardless of the original number of attributes. The motivation for this approach is to take away most of the irrelevant attributes. Then the algorithm is able to focus on the remaining, relevant attributes—those with sufficiently high scores. A disadvantage is that we may lose weakly relevant attributes that perform poorly on their own, but may enhance overall classification performance when coupled with other attributes. Also, k may not be large enough to comprise all relevant attributes. However, the experimental results in Section V-A show that the performance of this method is competitive to a full forward selection on the datasets we tested. Note that the overall number of evaluations is linear in N because all attributes need to be considered for the ranking. We call this first variant of Linear Forward Selection the "fixed-set technique" because the available attributes are reduced to a fixed set of size k. The method is illustrated in Figure 1(a): the number of potential subset extensions decreases with each step, as in SFS, while the currently selected subset grows.

2) Fixed Width: This method keeps the number of extensions in each forward selection step constant to a fixed width k (see Figure 1(b)). Again, an initial ranking is calculated based on the single attribute evaluation scores, and the search starts with the top-k attributes. However, in each of the subsequent forward selection steps, we increase the number of attributes that are considered by one, by adding the next best attribute in the ranking to the set of candidate expansions. This ensures that the set of candidate expansions consists of the individually best k attributes that have not been selected so far during the search. This increases the theoretical upper bound for the number of evaluations in the forward search process to $N \times k - \frac{1}{2} \times k(k-1)$. The second term is necessary because the number of available attributes is less then k in the last k steps of the search. This approach handles strongly relevant attributes first, but as the search proceeds, more attributes with individually weaker scores are taken into account.

IV. Explicit Subset Size Determination in Forward Selection

Ng [20] estimates theoretical error bounds for the standard wrapper approach when used in conjunction with exhaustive search. He develops a search algorithm called ORDERED-FS that has a lower error bound in the case of many irrelevant attributes. ORDERED-FS randomly splits the dataset D into a set D_{Train} for training and a hold-out set D_{Test}. However, in contrast to the standard wrapper, attribute sets are evaluated on the *training* data D_{Train} to decide on expansions during the search. Only the *best* subset of each subset size is evaluated on the hold-out data, and the best of these is what is output by the algorithm. Hence this method only uses the test data to choose between subset sizes.

[20] shows that ORDERED-FS has a smaller sample complexity than the standard wrapper, i.e. it finds a hypothesis faster and with fewer training examples than the

standard wrapper approach. [20] suggests that this holds for standard wrapper evaluation via cross-validation as well, as it asymptotically yields a constant improvement over a test set. However, performing an exhaustive search makes this algorithm intractable on most real-world data.

A. *Forward Selection with Size Determination*

We adapt ORDERED-FS by using m-fold cross-validation rather than a single train/test split and forward selection rather than exhaustive search. The algorithm performs m forward selections, one for each of the training sets in the cross-validation. The training data is used to decide which attribute is added in each iteration of forward selection, and the test data is only used to evaluate the "best" m best subsets of a particular size. To determine the "optimal" subset size, we average the m scores on the test data for each subset size, and choose the size with the highest average. Then, a final forward selection is performed on the complete dataset to find a subset of that optimal size. The resulting attribute set is output by the algorithm.

The m runs of forward selection may stop at different subset sizes. We restart all those runs that have a smaller subset size than the largest one found, and force them to continue to that size. Similary, the termination criterion for the final forward search is the optimal subset size. Thus the evaluation score may decrease during the search.

Note that the algorithm just presented is an attribute subset search technique that can theoretically be used with any subset evaluator. Its computational cost is higher than the cost of a simple forward selection because of the two stages involved (m-fold cross-validation + final search). We are using the standard cross-validation based wrapper for the final search. To reduce the runtime a fast wrapper evaluator is employed for computing the optimal size, as described in Section V-B.

Note also that we are not the first to propose an ORDERED-FS-inspired method to yield a practical algorithm. [28] evaluate a hybrid filter-wrapper method on the leukemia dataset, where they rank all attributes based on a filter criterion. They observe that using too many of the top-ranked attributes leads to overfitting. Facing the problem of how many attributes to choose, they adapt the ORDERED-FS search approach to calculate the best cardinality n using leave-one-out cross-validation. In each fold, [28] calculates a filter-based ranking and evaluates all possible subset sizes with the wrapper on the test data. Then n is set to the subset size that leads, on average, to the best accuracy. [28] shows that selecting the top n attributes of the initial ranking on the whole training data leads to good results. However, in contrast to our method this approach can essentially be classified as a filter method for attribute selection: the wrapper is only used to decide how many of the filter-ranked attributes to use to build the final classifier, and attribute dependencies are thus only taken into account in a very limited fashion.

```
Perform $m$-fold cross-validation split on data $D$:
$D \rightarrow (D^{(1)}_{Train}, D^{(1)}_{Test}), (D^{(2)}_{Train}, D^{(2)}_{Test}), \dots (D^{(m)}_{Train}, D^{(m)}_{Test})$

                                        ▷ STEP 1: COMPUTE OPT.-SIZE
for all folds $i = 1\ to\ m$ do
    Generate ranking $R_{D^{(i)}_{Train}}$ on training data $D^{(i)}_{Train}$
    $S_i = LinearForwardSelection(D^{(i)}_{Train}, R_{D^{(i)}_{Train}}, k)$
proceed all $i$ forward selections until $|S_i| = max_{1 \leq i \leq m}|S_i|$

for all folds $i = 1\ to\ m$ do
    for all subsets $S'_i = S_i$ and preceding subsets of $S_i$ do
        $score_{S'_i} = evaluate(S'_i, D^{(i)}_{Train}, D^{(i)}_{Test})$
$avgScore_n$ = mean score for subset size $n$
$optSize$ = subset size $n$ with max $avgScore_n$

                          ▷ STEP 2: FORW.-SELECTION UP TO OPT.-SIZE
Generate ranking $R_D$ on data $D$
$S = LinearForwardSelectionToSize(D, optSize, R_D, k)$
return $S$
```

Fig. 2. Linear forward selection with explicit subset size determination. $LinearForwardSelection(D, R, k)$ denotes a forward selection with a limited number of k attributes, based on the ranking R, using either the *fixed-set* or the *fixed-width* technique from Section III. $LinearForwardSelectionToSize(...)$ uses a given subset size as termination criterion and outputs a subset of that size, $evaluate(S, D_1, D_2)$ delivers the accuracy of the classifier on the data D_2, trained on the data D_1, using only the attributes in S.

B. *Linear Forward Selection with Subset Size Determination*

ORDERED-FS addresses the overfitting problem in the context of many irrelevant attributes. Even though the results of our adapted version are promising (see Section V-B), the search still tends to overfit in the case where the data has a small number of instances. Like standard wrapper-based forward selection, it can also be relatively slow. We therefore combine it with Linear Forward Selection, the method for reducing the number of subset expansions from Section III.

Figure 2 shows the pseudo code for Linear Forward Selection with Subset Size Determination. As in Section III, limiting the number of subset extensions is based on an initial ranking of the attributes. We perform one ranking for each of the $m + 1$ forward selections. Since each forward selection runs on different training data, each ranking will be different, and the runs may thus operate with different attributes. Alternatively, one could use a single ranking generated on the complete data. In doing so, all forward searches on the inner folds would search different training data, but with the same attribute ranking. However, this approach yields similar accuracy (see [9] for details).

V. Experimental Results

We have implemented our new algorithms within the WEKA framework [6], which provides various attribute selection techniques such as forward selection and the wrapper subset evaluator.

1) Evaluating the Search Result: All experiments in this paper are based on a stratified 5-fold cross-validation. Thus, each attribute selection method is applied five times on sub samples of the training set. The resulting attribute subsets are evaluated on the corresponding test sets, which have not

Datasets	Field	#Attr	#Inst	#Class
CNS	microarray-data [21]	7129	60	2
DLBCL	microarray-data [23]	7399	240	2
Leukemia	microarray-data [8]	7129	72	2
Lung-Cancer	microarray-data [3]	7129	96	2
MLL-Leukemia	microarray-data [1]	12582	72	3
Prostate-Cancer	microarray-data [25]	12600	102	2
20News[small]	text-classification [2]	2572	180	20
Reuters[small]	text-classification [17]	2748	438	7
Arrhythmia	ECG & patient-data [10]	279	452	13
Coil-20[small]	image-recognition [19]	1024	240	20
Internet[small]	img-size, url-phrases [16]	1558	1093	2
Splice-Site	dna-data [5]	408	2000	2

TABLE I
THE DATASETS.

been seen by the method previously. Using 5-fold cross-validation instead of 10-fold cross-validation leads to reasonable estimates [see 13], with only slightly increased variance. We repeat the cross-validation at least 2 times and up to 5 times, depending on the size of the dataset concerned and the number of attributes it contains. Due to time limitations, it was not possible to apply 5 repetitions of cross-validation in all cases. However, the same number of cross-validation repetitions as well as identical sub samples were used for comparing different methods on the same dataset. We use a paired *t*-test to identify significant differences (significance level is 0.05).

2) Wrapper Subset Evaluation: Two different variants of the wrapper were used to compute the accuracy of the induced classifier, i.e. the evaluation score of a subset; note that both variants are applied to the *training* sub samples of the (outer) 5-fold cross-validation.

- *The wrapper using cross-validation* corresponds to the WEKA-implementation using default settings. It evaluates subsets as proposed by [14] and performs an inner stratified 5-fold cross-validation on the data that is used by the search. The classifier is applied on each fold, i.e. it is built on the training set and accuracy is estimated by classifying the test set. As long as the standard deviation divided by the mean exceeds 1%, the cross-validation is repeated up to 5 times. This evaluator is used by default in most of our experiments.
- *The simple wrapper* assigns the training accuracy of the wrapped classifier as the subset evaluation score. The classifier is trained on the complete training data and the same data is used to estimate accuracy. This is much faster than using a cross-validation, but leads to a more optimistic score. This evaluator is only used in Section V-B.

3) Classifiers: The experiments in this paper are performed with naive Bayes (NB) and C4.5, two different classifiers that are often used to perform attribute selection with the wrapper. Both algorithms are used as implemented in WEKA, and all parameters set to their default values.

4) Datasets: We use 12 high-dimensional datasets from different fields, shown in Table I. The number of attributes ranges from 279 up to 12600. Datasets that are available as separate training and test sets have been joined together. As well as the new techniques presented in this paper, we also applied complete forward selection using the wrapper for comparison purposes. It was therefore necessary to reduce the number of instances for some of the datasets. This was done by removing stratified samples of the data. Preprocessed data is marked with the tag [small]. Additionally, the textual information of the text classification datasets was converted into numerical attributes, using the string converter in WEKA. The minimum frequency of each term was restricted to 10 in the 20 Newsgroups dataset, and to 7 in the Reuters-21578 dataset. Splice-Site is a dataset created from DNA data, which was used in [5] to apply backward feature elimination. We processed the raw DNA sequences by transforming each position into 4 binary attributes, one for each nucleotide.

A. Linear Forward Selection

We first compare the fixed-set and fixed-width methods for Linear Forward Selection, as well as a full forward selection. Figure 3 shows the results we obtained for different values of k—10, 50, 100, 200. "All" refers to the full forward selection search. Each group of 8 bars are sorted by k, in each case showing the fixed-set technique first, followed by the fixed-width technique for the same value k. We show classification accuracy, number of subset evaluations performed, and size of the final attribute sub set. The bars in the graph for accuracy show both accuracy on the training data (lighter shades) and test data (darker shades). This is possible because training accuracy was never greater than test accuracy in any of the runs of 5-fold cross-validation. We consider the difference in the training and test accuracy as a measure of overfitting.

Comparing to the results for "all" shows that limiting the number of attributes generally has a very beneficial effect on forward selection: the number of evaluations decreases dramatically and accuracy on the test data remains competitive. Increasing k often only increases accuracy on the training data, not the test data, indicating overfitting. In some cases, like the DLBCL data, accuracy on the test data actually decreases. However, the text datasets and Coil-20[small] show that $k = 200$ may not be large enough in some cases, because the highest accuracy is obtained using a full search. Despite the high sensitivity of the standard *t*-test in this setting, we only observed significant differences in test accuracy on the two text classification datasets: using naive Bayes on Reuters[small], it is significantly better to use all attributes than to impose a limit (this holds for most values of k); on 20News[small], $k = 200$ with the fixed-width technique leads to significant better accuracy than $k = 10$ for the fixed-set technique; and, applying C4.5, there is one significant loss when using a small value for k instead of a full search on each of the two text datasets.

The main benefit of Linear Forward Selection is the reduction in computational effort. Compared to $k = 10$, using all attributes requires 9.5 times more subset evaluations

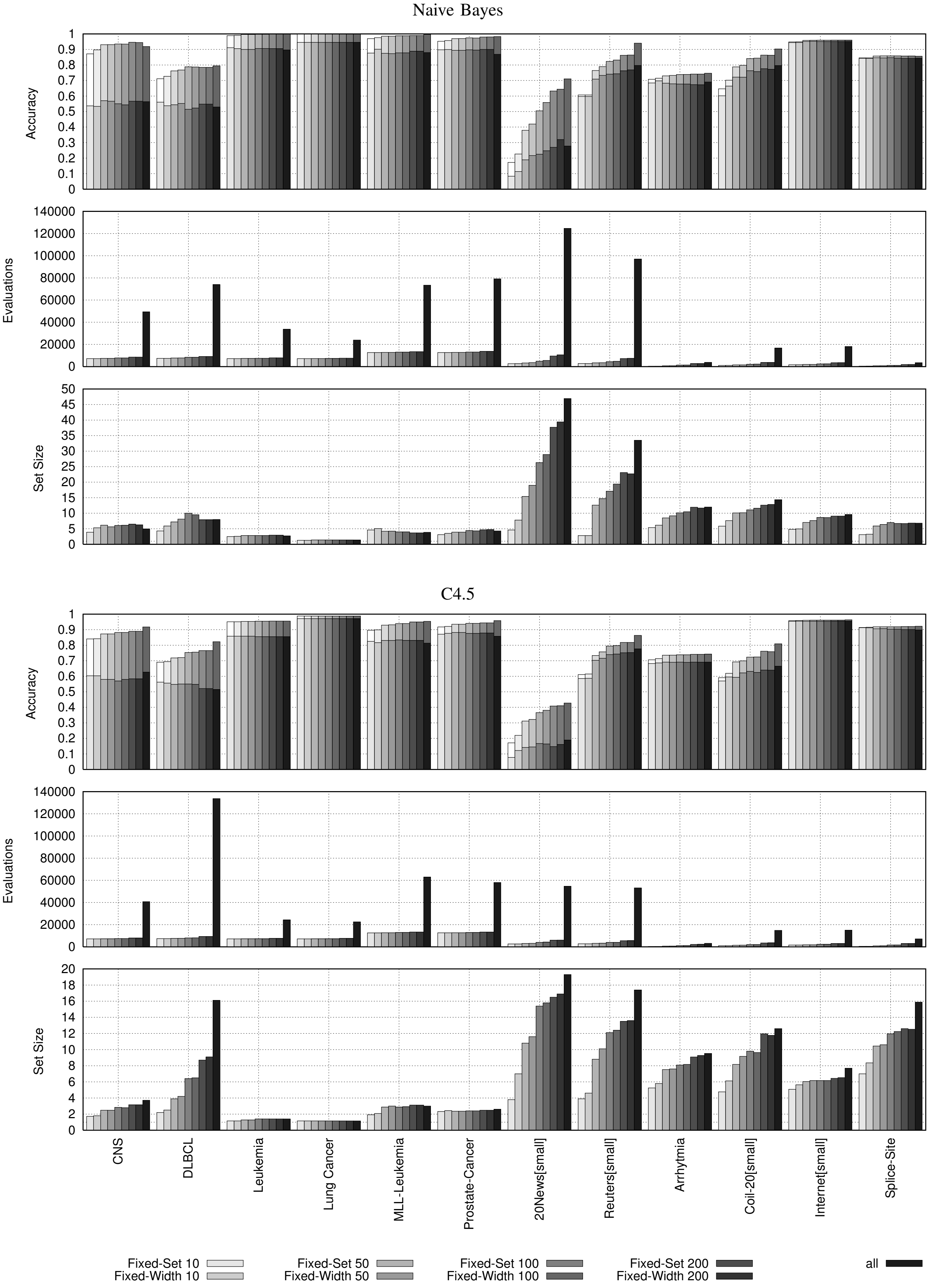


Fig. 3. SFS results for different values of k (explained in Section V-A)

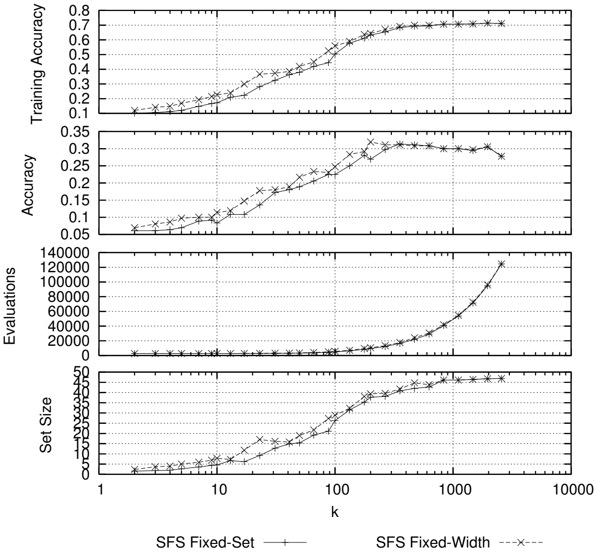


Fig. 4. SFS and SFS-BF results using naive Bayes for the *20News[small]* dataset.

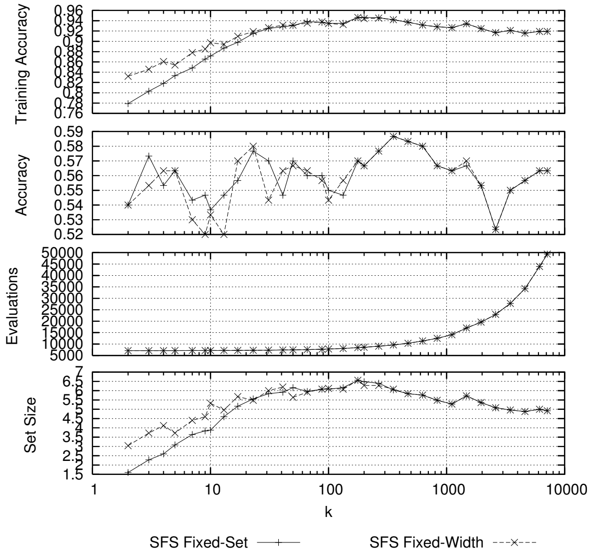


Fig. 5. SFS and SFS-BF results using naive Bayes for the *CNS* dataset.

on average. The real speedup is even higher, as evaluating larger attribute subsets requires more computational effort than smaller ones. However, another substantial benefit of Linear Forward Selection is that it can reduce subset size: small values of k generally result in fewer selected attributes, especially for C4.5.

Closer inspection of the results shows that there is a difference between datasets with few class values and those with more, most likely reflecting the fact that the latter group of datasets has a more complicated structure that is more difficult to classify. We now consider these two groups of datasets separately, using specific results that illustrate the general trends.

1) Datasets with many Class Values: Figure 4 displays results for increasing values of k, obtained on the 20News[small] dataset using naive Bayes. The right-most set of results corresponds to a full forward selection. "Training accuracy" refers to accuracy on the training data, and "Accuracy" is the accuracy obtained on the test sets of the outer 5-fold cross-validation.

The accuracy curve shown in this figure is typical for datasets with many class values. 20News[small], as well as Reuters[small], Coil-20[small], and Arrhythmia, differ from the other 8 datasets because they have between 7 and 20 class values, and the behaviour is similar in these cases: accuracy for small values of k is poor; it increases with k until the graph flattens out. For this particular example, using the fixed-width technique with $k = 200$ yields the highest accuracy. From that point on accuracy tends to decrease slightly, but it remains similar. Note that the number of evaluations for $k = 200$ is about 7.5 times less compared to a complete forward search.

On all datasets with many class values we considered, accuracy plateaus when k is in the low hundreds. Proceeding from there, it decreases a bit in the above example, but it may slightly increase on other datasets: on Coil-20[small], using naive-Bayes, maximum accuracy is obtained for the second-to-last value of k.

The fixed-set and fixed-width techniques mainly differ for small values of k (≤ 10). Fixed-width is superior in that region, as it enables selection of more attributes; fixed-set is too restrictive. This does not hold for high values of k: the more attributes are available, the lower are the chances that a few additional attribute make a difference. Note that the difference in computational effort is negligible.

2) Datasets with few Class Values: The 6 microarray datasets, as well as Internet[small] and Splice-Site, have very few class values. Apart from the MLL data, which has 3 classes, they all have 2 classes. The main result for these datasets is that the best accuracy can generally be achieved for very small values of k. In 11 of 16 experiments (8 each for naive Bayes and C4.5), the highest accuracy was achieved for $k \leq 10$. For larger values of k the accuracy stays almost constant or decreases. We observed only two datasets where the accuracy clearly increases up to slightly larger values for k, namely $k = 30$ for Prostate-Cancer (using C4.5), and $k = 50$ for Internet[small].

Figure 5 shows an example where the accuracy exhibits no clear tendency, with peaks at $k = 23$ and $k = 354$. The curve fluctuates significantly, although these results were averaged from a 5-fold cross-validation repeated 5 times. Many of the experiments on microarray data show similar behaviour. As far as accuracy is concerned, it is difficult to decide on an "optimal" value for k in these cases. However, it is clear that Linear Forward Selection is preferable to a complete forward search. Note that about half of the microarray datasets exhibit a decrease in accuracy for large values of k.

B. Explicit Subset Size Determination in Forward Selection

We now investigate the technique introduced in Section IV, which attempts to avoid overfitting by limiting the use of

Standard SFS vs SFS with Subset Size Detn.

		few classes			many classes		
k	t	wins	Δacc	Δsize	wins	Δacc	Δsize
10	s	11/0/5	0.003	-0.1	3/0/5	0.001	1.5
10	w	9/0/7	0.003	-0.4	4/0/4	0.006	4.1
50	s	9/1/6	0.002	-0.9	2/1/5	-0.002	0.3
50	w	9/0/7	0.003	-0.9	3/0/5	0.002	0.8
100	s	11/0/5	0.009	-1.3	2/0/6	-0.012	-1.1
100	w	11/0/5	0.008	-1.1	3/0/5	-0.007	-2.4
200	s	10/0/6	0.004	-1.5	5/0/3	-0.009	-4.2
200	w	10/0/6	0.004	-1.7	2/0/6	-0.019	-4.3
all		8/1/7	0.001	-2.3	2/0/6	-0.025	-5.8

TABLE II

COMPARISON OF ACCURACY FOR SFS (USING FIXED-SET AND FIXED-WIDTH) AND EXPLICIT SIZE DETERMINATION, BOTH NAIVE BAYES AND C4.5 WERE USED, GROUPED BY DATASET TYPES

cross-validation for evaluating individual subsets; rather, 5-fold cross-validation is used to estimate the "optimal" subset size, by performing 5 forward selection runs using the *simple* wrapper evaluator, which uses the resubstitution error for evaluation. The wrapper evaluator using (5-fold) cross-validation is only used in the final run of forward selection, once the "optimal" size has been determined.[1]

1) Explicit Subset Size Determination using all Attributes: We first apply this method in standard forward selection, before investigating the effect on Linear Forward Selection. Measuring accuracy, this yields 10 wins and 13 losses, with one draw, but no statistically significant differences. This is shown in the last row of Table II, which distinguishes between the two groups of datasets discussed above, and also reports the average difference in accuracy and subset size. As we perform 6 forward selections rather than one, the number of evaluations is much higher than in a standard forward selection, and thus not shown in the table. However, because the simple wrapper is used for the first 5 forward selections, the computational effort is only about 1.5 times higher than in a normal forward selection. More importantly, the modified method yields subsets that are about one third smaller on average. This is the main advantage of forward selection with explicit subset size determination.

2) Limiting the Number of Attributes: Table II also shows the effect of explicit subset size determination on the two types of Linear Forward Selection, for different values of k: as k increases, the difference in subset size generally increases; for larger values of k, explicit subset size determination leads to smaller subsets, without significant impact on accuracy. The overall effect is thus positive.

Figures 6 and 7 show the behaviour in more detail, for CNS (based on fixed-set) and 20News (based on fixed-width). In CNS, using explicit size determination is clearly beneficial across the board. In the case of 20News it actually leads to larger—but also more accurate—subsets for small k, but smaller subsets for large k. In the latter case, we conjecture that the difference is primarily due to the fact that the simple wrapper is used during the search for determining the best subset size.

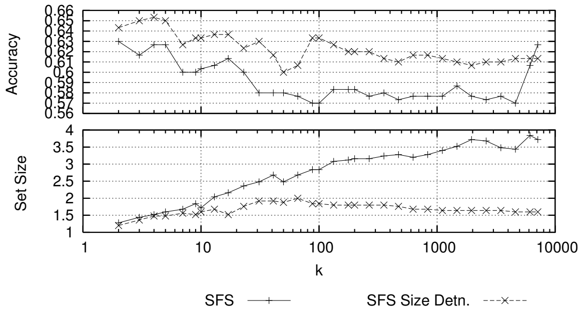


Fig. 6. Comparison of SFS (fixed-set) to explicit subset size determination using the C4.5 classifier and the *CNS* dataset.

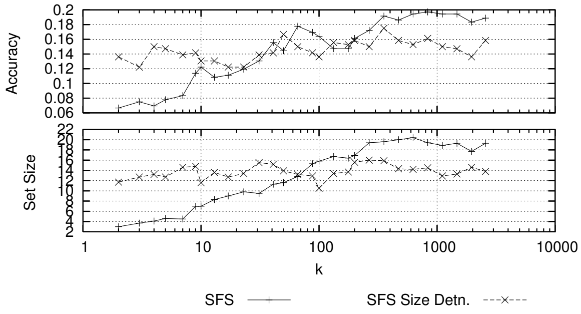


Fig. 7. Comparison of SFS (fixed-width) to explicit subset size determination using the C4.5 classifier and the *20News[small]* dataset.

C. Comparison to related work

We now compare Linear Forward Selection, BIRS and Rank-Search, discussed in Section II. Table III shows a summary, comparing them to the fixed-set method—without subset size determination.

We configured BIRS as recommended by [4]: ranking is performed with a filter method (Symmetrical Uncertainty), and the wrapper used during the search performs a 5-fold cross-validation. Compared to BIRS, Linear Forward Selection (with $k = 100$) yields competitive accuracy and smaller subset sizes. The subsets chosen by BIRS are about twice as large on average, and larger in 19 of 24 experiments. The win/loss statistics are in favour of BIRS, but none of the differences in accuracy are statistically significant. Moreover, BIRS performs more subset evaluations (on the other hand, the evaluator used for BIRS is faster).

We configured Rank-Search [11] to produce its initial ranking by evaluating attributes individually using the wrapper. Nevertheless, only 10 of 24 experiments finished the 5 times 5-fold cross-validation. This is because very large attribute subsets are evaluated during the search. However, on some datasets this extra search effort pays off, especially for Splice-Site using naive Bayes, where it yields an improvement in accuracy of about 10% (93.68 percent). The main drawbacks of Rank-Search are the large attribute subsets it finds and the high computational effort required.

[1] Using the simple wrapper produces comparable, but slightly worse results [9].

	few classes			many classes		
Algorithms	wins	Δacc	Δsize	wins	Δacc	Δsize
BIRS - SFS 10	9/7	0.009	5.6	7/1	0.109	24.1
BIRS - SFS 50	11/5	0.009	4.3	7/1	0.055	18.8
BIRS - SFS 100	11/5	0.008	4.0	6/2	0.016	14.9
Rank - SFS 10	4/2	0.025	265.4	3/1	0.078	167.1
Rank - SFS 50	4/2	0.027	263.8	3/1	0.046	165.4
Rank - SFS 100	4/2	0.028	263.3	2/2	0.020	164.2
Rank - BIRS	3/3	0.016	260.7	3/1	0.001	153.6

TABLE III

Comparison of SFS (fixed-set) to BIRS and Rank-Search (Only 10/24 experiments have finished in time for Rank-Search: Arrhythmia, Coil-20[small], Splice-Site for both algorithms; CNS, Lung-Cancer, Leukemia, Internet[small] for naive Bayes only)

VI. Summary and Future Work

We have presented two variants of Linear Forward Selection, a simple technique for tackling high-dimensional datasets with wrapper-based forward selection. Both variants, fixed-width and fixed-set search, are preferable to standard forward selection, primarily because of the dramatic reduction in runtime, but also because they can produce smaller subsets without marked changes in accuracy. We have also shown that Linear Forward Selection is competitive with other approaches used for speeding up wrapper-based search. However, the parameter k needs to be set to a sufficiently large value; larger values are required for structurally more complex learning problems.

We have also investigated explicit subset size determination in forward selection, inspired by ORDERED-FS [20]. Compared to the standard method, it generally produces smaller subset without degrading accuracy. Moreover, it can be successfully combined with Linear Forward Selection to yield a fast method for obtaining small and accurate subsets of attributes for high-dimensional data.

References

[1] S.A. Armstrong, J.E. Staunton, L.B. Silverman, R. Pieters, M.L. den Boer, M.D. Minden, S.E. Sallan, E.S. Lander, T.R. Golub, and S.J. Korsmeyer. MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature Genetics*, 30(1):41–47, 2002.

[2] S.D. Bay, D. Kibler, M.J. Pazzani, and P. Smyth. The UCI KDD archive of large data sets for data mining research and experimentation. *ACM SIGKDD Explorations Newsletter*, 2(2):81–85, 2000.

[3] D.G. Beer, S.L.R. Kardia, C.C. Huang, T.J. Giordano, A.M. Levin, D.E. Misek, L. Lin, G. Chen, T.G. Gharib, D.G. Thomas, et al. Gene-expression profiles predict survival of patients with lung adenocarcinoma. *NATURE MEDICINE*, 8(8):817, 2002.

[4] P. Bermejo, J. Gámez, and J.M. Puerta Jose. On incremental wrapper-based attribute selection: experimental analysis of the relevance criteria. In *Proceedings of IPMU'08*, pages 638–645, 2008.

[5] Sven Degroeve, Bernard De Baets, Yves Van de Peer, and Pierre Rouzé. Feature subset selection for splice site prediction. In *ECCB*, pages 75–83, 2002.

[6] Eibe Frank, Mark A. Hall, Geoffrey Holmes, Richard Kirkby, and Bernhard Pfahringer. Weka - a machine learning workbench for data mining. In *The Data Mining and Knowledge Discovery Handbook*, pages 1305–1314. Springer, 2005.

[7] Holger Fröhlich, Olivier Chapelle, and Bernhard Schölkopf. Feature selection for support vector machines by means of genetic algorithms. In *ICTAI*, pages 142–148, 2003.

[8] TR Golub, DK Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, JP Mesirov, H. Coller, ML Loh, JR Downing, MA Caligiuri, et al. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, 286(5439):531–537, 1999.

[9] Martin Gütlein. Large scale attribute selection using wrappers. Thesis for the degree Diploma of Computer Science, Albert-Ludwigs-Universität Freiburg, http://informatik.uni-freiburg.de/∼guetlein/publications/thesis.pdf, 2006.

[10] HA Guvenir, B. Acar, G. Demiroz, and A. Cekin. A supervised machine learning algorithm for arrhythmia analysis. *Computers in Cardiology 1997*, pages 433–436, 1997.

[11] Mark Hall and Geoffrey Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Trans. Knowl. Data Eng.*, 15(6):1437–1447, 2003.

[12] Edmundo Bonilla Huerta, Béatrice Duval, and Jin-Kao Hao. A hybrid ga/svm approach for gene selection and classification of microarray data. In *EvoWorkshops*, pages 34–44, 2006.

[13] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 2:1137–1145, 1995.

[14] R. Kohavi and G.H. John. Wrappers for Feature Subset Selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[15] R. Kohavi and D. Sommerfield. Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology. *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 192–197, 1995.

[16] N. Kushmerick. Learning to remove Internet advertisements. *Proceedings of the third annual conference on Autonomous Agents*, pages 175–181, 1999.

[17] David D. Lewis. Feature selection and feature extraction for text categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 212–217. Defense Advanced Research Projects Agency, Morgan Kaufmann, February 1992.

[18] J. Loughrey and P. Cunningham. Using Early-Stopping to Avoid Overfitting in Wrapper-Based Feature Selection Employing Stochastic Search. Technical report, Technical Report TCD-CS-2005-37). Department of Computer Science, Trinity College Dublin, Dublin, Ireland, 2005.

[19] S.A. Nene, S.K. Nayar, and H. Murase. Columbia Object Image Library (COIL-20). *Techn. Rep. No. CUCS-006-96, dept. Comp. Science, Columbia University*, 1996.

[20] Andrew Y. Ng. On feature selection: Learning with exponentially many irrelevant features as training examples. In *ICML*, pages 404–412, 1998.

[21] S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, L.M. Sturla, M. Angelo, M.E. McLaughlin, J.Y.H. Kim, L.C. Goumnerova, P.M. Black, C. Lau, et al. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415(6870):436–442, 2002.

[22] Pavel Pudil, Jana Novovicová, and Josef Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(10):1119–1125, 1994.

[23] A. Rosenwald, G. Wright, W.C. Chan, J.M. Connors, E. Campo, R.I. Fisher, R.D. Gascoyne, H.K. Muller-Hermelink, E.B. Smeland, J.M. Giltnane, et al. The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *The New England journal of medicine*, 346(25):1937–1947, 2002.

[24] Roberto Ruiz, Jose C. Riquelme, and Jesus S. Aguilar-Ruiz. Incremental wrapper-based gene selection from microarray expression data for cancer classification. *Pattern Recognition*, 2005.

[25] D. Singh, P.G. Febbo, K. Ross, D.G. Jackson, J. Manola, C. Ladd, P. Tamayo, A.A. Renshaw, A.V. DAmico, J.P. Richie, et al. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203–209, 2002.

[26] David J. Stracuzzi and Paul E. Utgoff. Randomized variable elimination. *J. Mach. Learn. Res.*, 5:1331–1362, 2004.

[27] Feng Tan, Xuezheng Fu, Hao Wang, Yanqing Zhang, and Anu G. Bourgeois. A hybrid feature selection approach for microarray gene expression data. In *International Conference on Computational Science (2)*, pages 678–685, 2006.

[28] Eric P. Xing, Michael I. Jordan, and Richard M. Karp. Feature selection for high-dimensional genomic microarray data. In *ICML*, pages 601–608, 2001.