ARE BETTER FEATURE SELECTION METHODS ACTUALLY BETTER?
Discussion, Reasoning and Examples

Petr Somol, Jana Novovičová
Inst. of Information Theory and Automation, Pattern Recognition Dept., Pod vodárenskou věží 4, Prague 8, 18208, Czech Republic
somol@utia.cas.cz, novovic@utia.cas.cz

Pavel Pudil
Prague University of Economics, Faculty of Management, Jarošovská 1117/II, Jindřichův Hradec, 37701, Czech Republic
pudil@fm.vse.cz

Keywords: Feature Selection, Subset Search, Search Methods, Performance Estimation, Classification Accuracy.

Abstract: One of the hot topics discussed recently in relation to pattern recognition techniques is the question of actual performance of modern feature selection methods. Feature selection has been a highly active area of research in recent years due to its potential to improve both the performance and economy of automatic decision systems in various applicational fields, with medical diagnosis being among the most prominent. Feature selection may also improve the performance of classifiers learned from limited data, or contribute to model interpretability. The number of available methods and methodologies has grown rapidly while promising important improvements. Yet recently many authors put this development in question, claiming that simpler older tools are actually better than complex modern ones – which, despite promises, are claimed to actually fail in real-world applications. We investigate this question, show several illustrative examples and draw several conclusions and recommendations regarding feature selection methods' expectable performance.

1 INTRODUCTION

Dimensionality reduction (DR) concerns with the task of finding low dimensional representation for high dimensional data. DR is an important step in data preprocessing in pattern recognition applications. It is sometimes the case that such tasks as classification of the data represented by so called feature vectors, can be carried out in the reduced space more accurately than in the original space. There are two main ways of doing DR depending on the resulting features: DR by *feature selection* (FS) and DR by *feature extraction* (FE). The FS approach does not attempt to generate new features, but tries to select the "best" ones from the original set of features. The FE approach defines a new feature vector space in which each new feature is obtained by transformations of the original features. FS leads to savings in measurement cost and the selected features retain their original physical interpretation, important e.g., in medical applications. On the other hand, transformed features generated by FE may provide a better discriminative ability than the best subset of given features, but these new features may not have a clear physical meaning. A typical feature selection process consists of four basic steps: *feature subset selection*, *feature subset evaluation, stopping criterion*, and *result validation*. Based on the *selection criterion* choice, feature selection methods may roughly be divided into three types: the *filter* (Yu and Liu, 2003; Dash et al., 2002), the *wrapper* (Kohavi and John, 1997) and the *hybrid* (Das, 2001; Sebban and Nock, 2002; Somol et al., 2006). The filter model relies on general characteristics of the data to evaluate and select feature subsets without involving any mining algorithm. The wrapper model requires one predetermined mining algorithm and uses its performance as the evaluation criterion. It attempts to find features better suited to the mining algorithm aiming to improve mining performance. This approach tends to be more computationally expensive than the filter approach. The hybrid model attempts to take advantage of the two approaches by exploiting their different evaluation criteria in different search stages. The hybrid approach is recently proposed to handle large datasets.

In recent years FS seems to have become a topic attracting an increasing number of researchers. Among the possible reasons the main one is certainly the importance of FS (or FE) as an inherent part of classification or modelling system design. Another reason, however, may be the relatively easy accessibility of the topic to the general research community. Apparently, many papers have been published in which any substantial advance is difficult to identify. One is tempted to say that the more papers on FS that are published, the fewer important contributions actually appear.

Certainly many key questions remain unanswered and key problems remain unsolved to satisfaction. For example, not enough is known about error bounds of many popular feature selection criteria, especially about their relation to classifier generalization performance. Despite the huge number of methods in existence, it is still a very hard problem to perform FS satisfactorily, e.g., in the context of gene expression data, with enormous dimensionality and very few samples. Similarly, in text categorization the standard way of FS is to completely omit context information and to resort to much more limited FS based on individual feature evaluation. In medicine these problems tend to become emphasized, as the available datasets are often incomplete (missing feature values in sample vectors), continuous and categorical data is to be treated at once, and the notion of feature itself may be difficult to interpret.

Among many criticisms of the current FS development there is one targeted specifically at the effort of finding more effective search methods, capable of yielding results closer to optimum with respect to some chosen criterion. The key argument against such methods is their alleged tendency to "over-select" features, or to find feature subsets fitted too tightly to training data, what degrades generalization. In other words, more search-effective methods are supposed to cause a similar unwanted effect as classifier over-training. Indeed, this is a serious problem that requires attention.

In recent literature the problem of "over-effective" FS has been addressed many times (Reunanen, 2003; Raudys, 2006). Yet, the effort to point out the problem (which seems to have been ignored, or at least insufficiently addressed before) now seems to have led to the other extreme notion of claiming that most of FS method development is actually contra-productive. This is, that older methods are actually superior to newer methods, mainly due to better over-fitting resistance.

The purpose of this paper is to discuss the issue of comparing actual FS methods' performance and to show experimentally what impact of the more effective search in newer methods can be expected.

1.1 FS Methods Overview

Before giving overview of the main methods to be discussed further we should note that it is not generally agreed in literature what the term "FS method" does actually describe. The term "FS method" is equally often used to refer to a) the complete framework that includes everything needed to select features, or b) the combination of search procedure and criterion or c) just the bare search procedure. In the following we will focus mainly on comparing the standard search procedures, which are not criterion- or classifier dependent. The widely known representatives of such "FS methods" are:

- Best Individual Features (IB) (Jain et al., 2000),
- Sequential Forward Selection (SFS), Sequential Backward Selection (SBS), (Devijver and Kittler, 1982),
- "Plus *l*-take away *r*" Selection (+L-R) (Devijver and Kittler, 1982),
- Sequential Forward Floating Selection (SFFS), Sequential Backward Floating Selection (SFBS) (Pudil et al., 1994),
- Oscillating Search (OS) (Somol and Pudil, 2000).

Many other methods exist (in all senses of the term "FS Method"), among others generalized versions of the ones listed above, various randomized methods, methods related to use of specific tools (FS for Support Vector Machines, FS for Neural Networks) etc. For overview see, e.g., (Jain et al., 2000; Liu and Yu, 2005). The selection of methods we are going to investigate is motivated by their interchangeability – any one of them can be used with the same given criterion, data and classifier. This makes experimental comparison easier.

2 PERFORMANCE ESTIMATION PROBLEM

FS methods comparison seems to be understood ambiguously as well. It is very different whether we compare concrete method properties or the final classifier performance determined by use of particular methods under particular settings. Certainly, final classifier performance is the ultimate quality measure. However, misleading conclusions about FS may be easily drawn when evaluating nothing else, as classifier performance depends on many more different aspects then just the actual FS method used. Nevertheless, in the following we will adapt classifier accuracy as the main means of FS method assessment.

There seems to be a general agreement in the literature that wrapper-based FS enables creation of more accurate classifiers than filter-based FS. This claim is nevertheless to be taken with caution, while using actual classifier accuracy as the FS criterion in wrapper-based FS may lead to the very negative effects mentioned above (overtraining). At the same time the weaker relation of filter-based FS criterion functions to particular classifier accuracy may help better generalization. But these effects can be hardly judged before the building of classification system has actually been accomplished.

In the following we will focus only on wrapper-based FS. Wrapper-based FS can be accomplished (and accordingly its effect can be evaluated) using one of the following methods:

- Re-substitution – In each step of the FS algorithm all data is used both for classifier training and testing. This has been shown to produce strongly optimistically biased results.
- Data split – In each step of the FS algorithm the same part of the data is used for classifier training and the other part for testing. This is the correct way of classifier performance estimation, yet it is often not feasible due to insufficient size of available data or due to inability to prevent bias caused by unevenly distributed data in the dataset (e.g., it may be difficult to ensure that with two-modal data distribution the training set won't by coincidence represent one mode and the testing set the other mode)
- 1-Tier Cross-Validation (CV) – Data is split to several parts. Then in each FS step a series of tests is performed, with all but one data part used for classifier training and the remaining part used for testing. The average classifier performance is then considered to be the result of FS criterion evaluation. Because in each test a different part of data is used for testing, all data is eventually utilized, without actually testing the classifier on the same data on which it had been trained. This is significantly better than re-substitution, yet it still produces optimistically biased results because all data is actually used to govern the FS process.
- Leave-one-out – can be considered a special case of 1-Tier CV with the finest data split granularity, thus the number of tests in one FS step is equal to the number of samples while in each test all but one sample are used for training with the one sample used for testing. This is computationally more expensive, better utilizes the data, but suffers the same problem of optimistic bias.
- 2-Tier CV – Defined to enable less biased estimation of final classifier performance than it is possible with 1-Tier CV. The data is split to several parts, FS is then performed repeatedly in 1-Tier CV manner on all but one part, which is eventually used for classifier accuracy estimation. This process yields a sequence of possibly different feature subsets, thus it can be used only for assessment of FS method effectivity and not for actual determination of the best subset. The average classifier performance on independent test data parts is then considered to be the measure of FS method quality. This is computationally demanding.

In our experiments we accept 2-Tier CV as satisfactory for the purpose of FS methods performance evaluation and comparison. Due to the fact that 2-Tier CV yields a series of possibly different feature subsets, we define an additional measure to be called *consistency*, that expresses the stability, or robustness of FS method with respect to various data splits.

Definition: Let $Y = \{f_1, f_2, \ldots, f_{|Y|}\}$ be the set of all features and let $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ be a system of $n > 1$ feature subsets $S_j = \{f_i | i = 1, \ldots, d_j, f_i \in Y, d_j \in \langle 1, |Y| \rangle\}, j = 1, \ldots, n$. Denote $\mathcal{F}_f$ the system of subsets in $\mathcal{S}$ containing feature f, i.e.,

$$\mathcal{F}_f = \{S | S \in \mathcal{S}, f \in S\}. \tag{1}$$

Let F_f be the number of subsets in $\mathcal{F}_f$ and X the subset of Y representing all features that appear anywhere in system $\mathcal{S}$, i.e.,

$$X = \{f | f \in Y, F_f > 0\}. \tag{2}$$

Then the *consistency* $C(\mathcal{S})$ of feature subsets in system $\mathcal{S}$ is defined as:

$$C(\mathcal{S}) = \frac{1}{|X|} \sum_{f \in X} \frac{F_f - 1}{n - 1}. \tag{3}$$

Properties of $C(\mathcal{S})$**:**

1. $0 \le C(\mathcal{S}) \le 1$.
2. $C(\mathcal{S}) = 0$ if and only if all subsets in $\mathcal{S}$ are disjunct from each other.
3. $C(\mathcal{S}) = 1$ if and only if all subsets in $\mathcal{S}$ are identical.

The higher the value, the more similar are the subsets in system to each other. For $C(\mathcal{S}) \approx 0.5$ on average each feature present in $\mathcal{S}$ appears in about half of all subsets. When comparing FS methods, higher *consistency* of subsets produced during 2-Tier CV is clearly advantageous. However, it should be considered a complementary measure only as it does not have any straight relation to the key measure of classifier generalization ability.

Remark: In experiments, if the best performing FS method also produces feature subsets with high *consistency*, its superiority can be assumed well founded.

3 EXPERIMENTS

To illustrate the differences between simpler and more complex FS methods we have collected experimental results under various settings: for two different classifiers, three FS search algorithms and eight datasets with dimensionalities ranging from 13 to 65 and number of classes ranging from 2 to 6. We used 3 different mammogram datasets as well as wine and wave datasets from UCI Repository (Asuncion and Newman, 2007), satellite image dataset from ELENA database (ftp.dice.ucl.ac.be), speech data from British Telecom and sonar data (Gorman and Sejnowski, 1988). For details see Tables 1 to 8.

Note that the choice of classifier and/or FS setup may not be optimal for each dataset, thus the reported results may be inferior to results reported in the literature; the purpose of our experiments is mutual comparison of FS methods only. All experiments have been done with 10-fold Cross-Validation used to split the data into training and testing parts (to be denoted "Outer CV" in the following), while the training parts have been further split by means of another 10-fold CV into actual training and validation parts for the purpose of feature selection and classifier training (to be denoted "Inner CV").

The application of SFS and SFFS was straightforward. The OS algorithm as the most flexible procedure has been used in two set-ups: slower randomized version and faster deterministic version. In both cases the *cycle depth* set to 1 [see (Somol and Pudil, 2000) for details]. The randomized version, denoted in the following as OS(1,r3), is called repeatedly with random initialization as long as no improvement has been found in last 3 runs. The deterministic version, denoted as OS(1,IB) in the following, is initialized by means of Individually Best (IB) feature selection.

The problem of determining optimal feature subset size was solved in all experiments by brute force. All algorithms were applied repeatedly for all possible feature sizes whenever needed. The final result has been determined as that with the highest classification accuracy (and lowest subset size in case of ties).

3.1 Notes on Obtained Results

All tables clearly show that more modern methods are capable of finding criterion values closer to optimum – see column Inner-CV in each table.

The effect pointed out by Reunanen (Reunanen, 2003) of the simple SFS outperforming all more complex procedures (regarding the ability to generalize) takes place in Table 4, column Outer-CV, with Gaussian classifier. Note the low *consistency* in this case. Conversely, Table 2 shows no less outstanding performance of OS with 3-Nearest Neighbor classifier (3-NN) with better *consistency* and smallest subsets found, while Table 3 shows top performance of SFFS with both Gaussian and 3-NN classifiers. Although it is impossible to draw decisive conclusions from the limited set of experiments, it should be of interest to extract some statistics (all on independent test data – results in the column Outer-CV):

- Best result among FS methods for each given classifier: SFS 11×, SFFS 17×, OS 11×.
- Best achieved overall classification accuracy for each dataset: SFS 1×, SFFS 5×, OS 2×.

Average classifier accuracies:

- Gaussian: SFS 0.652, SFFS 0.672, OS 0.663.
- 1-NN: SFS 0.361, SFFS 0.361, OS 0.349.
- 3-NN: SFS 0.762, SFFS 0.774, OS 0.765.

4 DISCUSSION AND CONCLUSIONS

With respect to FS we can distinguish the following entities which all affect the resulting classification performance: search algorithms, stopping criteria, feature subset evaluation criteria, data and classifier. The impact of the FS process on the final classifier performance (with our interest targeted naturally at its generalization performance, i.e., its ability to classify previously unknown data) depends on all of these entities.

When comparing pure search algorithms as such, then there is enough ground (both theoretical and experimental) to claim that newer, often more complex methods, have better potential of finding better solutions. This often follows directly from the method definition, as newer methods are often defined to improve some particular weakness of older ones. (Unlike IB, SFS takes into account inter-feature dependencies. Unlike SFS, +L-R does not suffer the nesting problem. Unlike +L-R, Floating Search does not depend on pre-specified user parameters. Unlike Floating Search, OS may avoid local extremes by means of randomized initialization etc.) Better solution, however, means in this context merely being closer to optimum with respect to the adopted criterion. This may not tell much about final classifier quality, while criterion choice has proved to be a considerable problem in itself. Vast majority of practically used criteria have only insufficient relation to correct classification rate,

Table 1: Classification performance as result of wrapper-based Feature Selection on wine data.

Wine data: 13 features, 3 classes containing 59, 71 and 48 samples, UCI Repository									
		Inner 10-f. CV		Outer 10-f. CV		Subset Size		Consis-	Run Time
Classifier	FS Method	Mean	*St.Dv.*	Mean	*St.Dv.*	Mean	*St.Dv.*	tency	h:m:s.ss
Gaussian	SFS	0.599	*0.017*	0.513	*0.086*	3.1	*1.221*	0.272	00:00:00.54
	SFFS	0.634	*0.029*	0.607	*0.099*	3.9	*1.136*	0.370	00:00:02.99
	OS(1,r3)	0.651	*0.024*	**0.643**	*0.093*	3.1	*0.539*	0.463	00:00:34.30
1-NN scaled	SFS	0.355	*0.071*	**0.350**	*0.064*	1	*0*	1	00:00:00.98
	SFFS	0.358	*0.073*	**0.350**	*0.064*	1	*0*	1	00:00:02.27
	OS(1,r3)	0.285	*0.048*	0.269	*0.014*	1.1	*0.3*	0.5	00:00:15.61
3-NN scaled	SFS	0.983	*0.005*	0.960	*0.037*	6.5	*1.118*	0.545	00:00:01.10
	SFFS	0.986	*0.005*	**0.965**	*0.039*	6.6	*0.917*	0.5	00:00:03.75
	OS(1,r3)	0.986	*0.004*	0.955	*0.035*	6.1	*0.7*	0.505	00:00:45.68

Table 2: Classification performance as result of wrapper-based Feature Selection on mammogram data.

Mammogram data, 65 features, 2 classes containing 57 (benign) and 29 (malignant) samples, UCI Rep.									
		Inner 10-f. CV		Outer 10-f. CV		Subset Size		Consis-	Run Time
Classifier	FS Method	Mean	*St.Dv.*	Mean	*St.Dv.*	Mean	*St.Dv.*	tency	h:m:s.ss
Gaussian	SFS	0.792	*0.028*	0.609	*0.101*	9.6	*3.382*	0.156	00:12:07.74
	SFFS	0.842	*0.030*	**0.658**	*0.143*	12.8	*2.227*	0.179	00:46:59.06
	OS(1,IB)	0.795	*0.017*	0.584	*0.106*	7.2	*2.638*	0.139	01:29:10.24
1-NN scaled	SFS	0.335	*0.002*	**0.337**	*0.024*	1	*0*	1	00:00:30.05
	SFFS	0.335	*0.002*	**0.337**	*0.024*	1	*0*	1	00:00:59.72
	OS(1,IB)	0.335	*0.002*	**0.337**	*0.024*	1	*0*	1	00:01:45.63
3-NN scaled	SFS	0.907	*0.032*	0.856	*0.165*	15.3	*6.001*	0.361	00:00:31.10
	SFFS	0.937	*0.017*	0.896	*0.143*	7.7	*3.770*	0.206	00:03:03.16
	OS(1,IB)	0.935	*0.014*	**0.907**	*0.119*	5.3	*0.781*	0.543	00:04:18.10

Table 3: Classification performance as result of wrapper-based Feature Selection on sonar data.

Sonar data, 60 features, 2 classes containing 103 (mine) and 105 (rock) samples, Gorman & Sejnowski									
		Inner 10-f. CV		Outer 10-f. CV		Subset Size		Consis-	Run Time
Classifier	FS Method	Mean	*St.Dv.*	Mean	*St.Dv.*	Mean	*St.Dv.*	tency	h:m:s.ss
Gaussian	SFS	0.806	*0.019*	0.628	*0.151*	20.2	*12.156*	0.283	00:08:41.83
	SFFS	0.853	*0.016*	**0.656**	*0.131*	22.8	*8.738*	0.326	01:51:46.31
	OS(1,IB)	0.838	*0.018*	0.649	*0.066*	21.5	*10.366*	0.315	03:36:04.92
1-NN scaled	SFS	0.511	*0.004*	**0.505**	*0.010*	1	*0*	1	00:01:51.78
	SFFS	0.511	*0.004*	**0.505**	*0.010*	1	*0*	1	00:03:10.47
	OS(1,IB)	0.505	*0.001*	**0.505**	*0.010*	1	*0*	1	00:08:06.63
3-NN scaled	SFS	0.844	*0.025*	0.618	*0.165*	15.2	*7.139*	0.273	00:02:15.84
	SFFS	0.870	*0.016*	**0.660**	*0.160*	18.9	*7.120*	0.293	00:12:26.01
	OS(1,IB)	0.864	*0.016*	0.622	*0.151*	15.8	*5.474*	0.247	00:25:55.39

while their relation to classifier generalization performance can be put into even greater doubt.

When comparing feature selection methods as a whole (under specific criterion-classifier-data settings) the advantages of more modern search algorithms may diminish considerably. Reunanen (Reunanen, 2003) points out, and our experiments confirm, that a simple method like SFS may lead to better classifier generalization. The problem we see with the ongoing discussion is that this is often claimed to be the general case. But this is not true, as confirmed by our experiments.

According to our experiments the "better" methods (being more effective in optimizing criteria) also tend to be "better" with respect to final classifier generalization ability, although this tendency is by no means universal and often the difference is negligible. No clear qualitative hierarchy can be recognized among standard methods, perhaps with the exception of mostly inferior performance of IB (not shown

Table 4: Classification performance as result of wrapper-based Feature Selection on mammogram data.

WPBC data, 31 features, 2 classes containing 151 (nonrecur) and 47 (recur) samples, UCI Repository									
		Inner 10-f. CV		Outer 10-f. CV		Subset Size		Consis-	Run Time
Classifier	FS Method	Mean	*St.Dv.*	Mean	*St.Dv.*	Mean	*St.Dv.*	tency	h:m:s.ss
Gaussian	SFS	0.807	*0.011*	**0.756**	*0.088*	9.2	*4.534*	0.241	00:00:21.24
	SFFS	0.818	*0.012*	0.698	*0.097*	15.4	*5.731*	0.441	00:04:07.81
	OS(1,r3)	0.826	*0.010*	0.682	*0.062*	12.6	*5.219*	0.356	00:34:07.20
1-NN	SFS	0.251	*0.020*	**0.237**	*0.018*	1	*0*	1	00:00:14.93
scaled	SFFS	0.251	*0.020*	**0.237**	*0.018*	1	*0*	1	00:00:39.71
	OS(1,r3)	0.332	*0.021*	**0.237**	*0.018*	7.3	*4.776*	0.169	00:03:19.70
3-NN	SFS	0.793	*0.013*	0.712	*0.064*	9.4	*5.869*	0.226	00:00:15.56
scaled	SFFS	0.819	*0.008*	**0.722**	*0.086*	11.7	*4.797*	0.322	00:01:48.94
	OS(1,r3)	0.826	*0.007*	0.687	*0.083*	11	*3.550*	0.325	00:14:44.24

Table 5: Classification performance as result of wrapper-based Feature Selection on mammogram data.

WDBC data, 30 features, 2 classes containing 357 (benign) and 212 (malignant) samples, UCI Rep.									
		Inner 10-f. CV		Outer 10-f. CV		Subset Size		Consis-	Run Time
Classifier	FS Method	Mean	*St.Dv.*	Mean	*St.Dv.*	Mean	*St.Dv.*	tency	h:m:s.ss
Gaussian	SFS	0.962	*0.007*	0.933	*0.039*	10.8	*6.539*	0.303	00:00:22.21
	SFFS	0.972	*0.005*	0.942	*0.042*	10.6	*2.653*	0.36	00:03:24.90
	OS(1,r3)	0.973	*0.004*	**0.943**	*0.039*	10.3	*2.147*	0.366	00:36:36.49
1-NN	SFS	0.373	*0.000*	**0.373**	*0.004*	1	*0*	1	00:01:33.07
scaled	SFFS	0.421	*0.022*	**0.373**	*0.004*	1	*0*	1	00:03:26.00
	OS(1,r3)	0.435	*0.001*	**0.373**	*0.004*	7.6	*2.871*	0.202	00:25:31.84
3-NN	SFS	0.981	*0.002*	0.967	*0.020*	15.3	*4.451*	0.456	00:01:32.19
scaled	SFFS	0.983	*0.001*	**0.970**	*0.019*	13.7	*4.220*	0.414	00:08:16.72
	OS(1,r3)	0.985	*0.002*	0.959	*0.025*	13.4	*3.072*	0.421	01:41:02.62

Table 6: Classification performance as result of wrapper-based Feature Selection on speech data.

Speech data, 15 features, 2 classes containing 682 (yes) and 736 (no) samples, British Telecom									
		Inner 10-f. CV		Outer 10-f. CV		Subset Size		Consis-	Run Time
Classifier	FS Method	Mean	*St.Dv.*	Mean	*St.Dv.*	Mean	*St.Dv.*	tency	h:m:s.ss
Gaussian	SFS	0.773	*0.008*	0.770	*0.052*	9.6	*0.917*	0.709	00:00:03.28
	SFFS	0.799	*0.008*	**0.795**	*0.042*	9.3	*0.458*	0.684	00:00:20.51
	OS(1,r3)	0.801	*0.008*	0.793	*0.041*	9.5	*0.5*	0.642	00:02:46.16
1-NN	SFS	0.522	*0.001*	**0.519**	*0.002*	1	*0*	1	00:01:27.25
scaled	SFFS	0.521	*0.001*	**0.519**	*0.002*	1	*0*	1	00:03:07.95
	OS(1,r3)	0.556	*0.011*	**0.519**	*0.002*	8.6	*2.577*	0.526	00:22:55.49
3-NN	SFS	0.946	*0.003*	0.935	*0.030*	7	*1.483*	0.487	00:01:33.55
scaled	SFFS	0.948	*0.003*	**0.939**	*0.030*	6.7	*1.1*	0.509	00:05:54.57
	OS(1,r3)	0.949	*0.003*	0.937	*0.029*	7	*1.095*	0.537	01:08:39.20

here). It has been shown that different methods become the best performing tools in different contexts, with no reasonable way of predicting the winner in advance (note, e.g., OS in Table 1 – gives best result with Gaussian classifier but worst result with k-NN).

Our concluding recommendation can be stated as follows: only in the case of strongly limited time should one resort to the simplest methods. Whenever possible try variety of methods ranging from SFS to more complex ones. If one method only has to be chosen, than we would stay with Floating Search as the best general compromise between performance, generalization ability and search speed.

4.1 Quality of Criteria

The performance question of more complex FS methods is directly linked to another question: How well do the available criteria describe the quality of evaluated subsets ? The contradicting experimental results

Table 7: Classification performance as result of wrapper-based Feature Selection on satellite land image data.

Satimage data, 36 features, 6 classes with 1072, 479, 961, 415, 470 and 1038 samples, ELENA database									
		Inner 10-f. CV		Outer 10-f. CV		Subset Size		Consis-	Run Time
Classifier	FS Method	Mean	*St.Dv.*	Mean	*St.Dv.*	Mean	*St.Dv.*	tency	h:m:s.ss
Gaussian	SFS	0.509	*0.016*	0.516	*0.044*	19	*7*	0.643	00:05:21.77
	SFFS	0.525	*0.011*	**0.528**	*0.034*	13.7	*3.743*	0.474	00:41:25.60
	OS(1,IB)	0.527	*0.010*	0.517	*0.055*	12.2	*3.311*	0.410	01:57:06.71
1-NN	SFS	0.234	*0.000*	**0.234**	*0.001*	1.6	*1.2*	0.244	03:05:20.63
scaled	SFFS	0.234	*0.000*	**0.234**	*0.001*	1	*0*	0.444	08:00:19.17
	OS(1,IB)	0.234	*0.000*	0.217	*0.001*	1.2	*0.6*	0.222	19:32:09.52
3-NN	SFS	0.234	*0.000*	**0.234**	*0.001*	1	*0*	1	03:16:08.09
scaled	SFFS	0.234	*0.000*	**0.234**	*0.001*	1	*0*	1	07:51:08.98
	OS(1,IB)	0.234	*0.000*	**0.234**	*0.001*	1.1	*0.3*	0.296	19:09:44.29

Table 8: Classification performance as result of wrapper-based Feature Selection on wave data.

Waveform data, 40 features, 3 classes containing 1692, 1653 and 1655 samples, UCI Repository									
		Inner 10-f. CV		Outer 10-f. CV		Subset Size		Consis-	Run Time
Classifier	FS Method	Mean	*St.Dv.*	Mean	*St.Dv.*	Mean	*St.Dv.*	tency	h:m:s.ss
Gaussian	SFS	0.505	*0.002*	**0.493**	*0.015*	2.1	*0.3*	0.222	00:08:38.86
	SFFS	0.506	*0.003*	0.492	*0.016*	2.4	*0.663*	0.185	00:42:36.39
	OS(1,IB)	0.506	*0.002*	0.489	*0.015*	2.7	*1.005*	0.222	01:57:58.04
1-NN	SFS	0.356	*0.009*	**0.331**	*0.000*	1	*0*	1	07:29:40.76
scaled	SFFS	0.356	*0.009*	**0.331**	*0.000*	1	*0*	1	16:09:52.71
	OS(1,IB)	0.331	*0.000*	**0.331**	*0.000*	1	*0*	1	35:55:50.53
3-NN	SFS	0.826	*0.002*	0.810	*0.024*	17.4	*2.332*	0.411	08:08:17.25
scaled	SFFS	0.829	*0.003*	0.808	*0.020*	17.4	*1.020*	0.475	38:46:26.60
	OS(1,IB)	0.830	*0.002*	**0.816**	*0.016*	17.1	*2.022*	0.593	95:12:19.24

seem to suggest, that the criterion used (classifier accuracy on testing data in this case) does not relate well enough to classifier generalization performance. Although we do not present any filter-based FS results here, the situation with filters seems similar. Thus, uneven performance of more complex FS methods may be viewed as a direct consequence of insufficient criteria. In this view it is difficult to claim that more complex FS methods are problematic per se.

4.2 Does It Make Sense to Develop New FS Methods?

Our answer is undoubtedly yes. Our current experience shows that no clear and unambiguous qualitative hierarchy can be established within the existing framework of methods, i.e., although some methods perform better than others more often, this is not the case always and any method can prove to be the best tool for some particular problem. Adding to this pool of methods may thus bring improvement, although it is more and more difficult to come up with new ideas that have not been utilized before. Regarding the performance of search algorithms as such, developing methods that yield results closer to optimum with respect to any given criterion may bring considerably more advantage in future, when better criteria may have been found to better express the relation between feature subsets and classifier generalization ability.

ACKNOWLEDGEMENTS

The work has been supported by EC project No. FP6-507752, the Grant Agency of the Academy of Sciences of the Czech Republic project A2075302, grant AV0Z10750506, and Czech Republic MŠMT grants 2C06019 and 1M0572 DAR.

REFERENCES

Asuncion, A. and Newman, D. (2007). UCI machine learning repository, http://www.ics.uci.edu/ ∼mlearn/ mlrepository.html.

Das, S. (2001). Filters, wrappers and a boosting-based hybrid for feature selection. In *Proceedings of the 18th International Conference on Machine Learning*, pages 74–81.

Dash, M., Choi, K., P., S., and Liu, H. (2002). Feature selection for clustering - a filter solution. In *Proceedings of the Second International Conference on Data Mining*, pages 115–122.

Devijver, P. A. and Kittler, J. (1982). *Pattern Recognition: A Statistical Approach*. Prentice-Hall International, London.

Gorman, R. P. and Sejnowski, T. J. (1988). Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1:75–89.

Jain, A. K., Duin, R. P. W., and Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:4–37.

Kohavi, R. and John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324.

Liu, H. and Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):491–502.

Pudil, P., Novovičová, J., and Kittler, J. (1994). Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125.

Raudys, S. (2006). Feature over-selection. *Lecture Notes in Computer Science*, (4109):622–631.

Reunanen, J. (2003). Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 3(3):1371–1382.

Sebban, M. and Nock, R. (2002). A hybrid filter/wrapper approach of feature selection using information theory. *Pattern Recognition*, 35:835–846.

Somol, P., Novovičová, J., and Pudil, P. (2006). Flexible-hybrid sequential floating search in statistical feature selection. *Lecture Notes in Computer Science*, (4109):632–639.

Somol, P. and Pudil, P. (2000). Oscillating search algorithms for feature selection. In *Proceedings of the 15th IAPR Int. Conference on Pattern Recognition, Conference B: Pattern Recognition and Neural Networks*, pages 406–409.

Yu, L. and Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th International Conference on Machine Learning*, pages 56–63.