KYBERNETIKA — VOLUME *47* (2011), NUMBER 3, PAGES 401–425

IMPROVING FEATURE SELECTION PROCESS RESISTANCE TO FAILURES CAUSED BY CURSE-OF-DIMENSIONALITY EFFECTS

Petr Somol, Jiří Grim, Jana Novovičová, and Pavel Pudil

The purpose of feature selection in machine learning is at least two-fold – saving measurement acquisition costs and reducing the negative effects of the curse of dimensionality with the aim to improve the accuracy of the models and the classification rate of classifiers with respect to previously unknown data. Yet it has been shown recently that the process of feature selection itself can be negatively affected by the very same curse of dimensionality – feature selection methods may easily over-fit or perform unstably. Such an outcome is unlikely to generalize well and the resulting recognition system may fail to deliver the expectable performance. In many tasks, it is therefore crucial to employ additional mechanisms of making the feature selection process more stable and resistant the curse of dimensionality effects. In this paper we discuss three different approaches to reducing this problem. We present an algorithmic extension applicable to various feature selection methods, capable of reducing excessive feature subset dependency not only on specific training data, but also on specific criterion function properties. Further, we discuss the concept of criteria ensembles, where various criteria vote about feature inclusion/removal and go on to provide a general definition of feature selection hybridization aimed at combining the advantages of dependent and independent criteria. The presented ideas are illustrated through examples and summarizing recommendations are given.

Keywords: feature selection, curse of dimensionality, over-fitting, stability, machine learning, dimensionality reduction

Classification: 62H30, 62G05, 68T10

1. INTRODUCTION

A broad class of decision-making problems can be solved by the *learning approach*. This can be a feasible alternative when neither an analytical solution exists nor the mathematical model can be constructed. In these cases the required knowledge can be gained from the past data which form the so-called “learning” or “training” set. Then the formal apparatus of statistical pattern recognition can be used to learn the decision-making.

A common practice in multidimensional classification methods is to apply a feature selection (FS) procedure as the first preliminary step. The aim is to avoid over-fitting in the training phase since, especially in the case of small and/or high-dimensional data, the classifiers tend to adapt to some specific properties of training data which are not typical for the independent test data. The resulting classifier then poorly generalizes and the classification accuracy on independent test data decreases [5]. By choosing a small subset of "informative" features, we try to reduce the risk of over-fitting and thus improve the generalizing property of the classifier. Moreover, FS may also lead to data acquisition cost savings as well as to gains in processing speed. An overview of various feature selection approaches and issues can be found in [17, 24, 25, 36].

In most cases, a natural way to choose the optimal subset of features would be to minimize the probability of a classification error. As the exact evaluation of error probability is usually not viable, we have to minimize some estimates of classification error or at least some estimates of its upper bound, or even some intuitive probabilistic criteria like entropy, model-based class distances, distribution divergences, etc. [5, 20]. Many existing feature selection algorithms designed with different evaluation criteria can be categorized as *filter* [1, 4, 51] *wrapper* [20], *hybrid* [3, 38, 41] or *embedded* [12, 13, 14, 21, 28, 29]. Filter methods are based on performance evaluation functions calculated directly from the training data such as *distance, information, dependency, and consistency,* [5, 25] and select feature subsets without involving any learning algorithm. Wrapper methods require one predetermined learning algorithm and use its estimated performance as the evaluation criterion. The necessity to estimate the classification performance in each FS step makes wrappers considerably slower than filters. Hybrid methods primarily attempt to obtain wrapper-like results in filter-like time. Embedded methods incorporate FS into modeling and can be viewed as a more effective but less general form of wrappers. In order to avoid biased solutions the chosen criterion has to be evaluated on an independent validation set. The standard approach in wrapper-based FS is thus to evaluate classifier accuracy on training data by means of cross-validation or leave-one-out estimation. Nevertheless, the problem of over-fitting applies to FS criteria and FS algorithms as well [31] and cannot be fully avoided by means of validation, especially when the training data is insufficiently representative (due to limited size or due to bias caused by the faulty choice of training data). It is well known that different optimality criteria may choose different feature subsets [5] and the same criterion may choose different subsets for differently sampled training data [31]. In this respect the "stability" of the resulting feature subsets becomes a relevant viewpoint [22, 42]. To summarize, although the key purpose of FS is to reduce the negative impact of the curse of dimensionality on classification accuracy, the FS process itself may be affected by the very same curse of dimensionality with serious negative consequences in the final pattern recognition system.

In this paper we suggest several ways of reducing FS over-fitting and stability problems. In subset-size-optimizing scenarios we suggest putting more preference on effective reduction of the resulting subset size instead of criterion maximization performance only. In accordance with [31] we suggest placing less emphasis on the notion of optimality with respect to the chosen criterion (see Section 2). In analogy to the idea of multiple classifier systems [19] that has proved capable of considerable classification accuracy improvement, we suggest employing ensembles of FS criteria instead of single criterion to prevent feature over-selection (see Section 3). We suggest FS process hybridization in order to improve generalization ability of wrapper-based FS approaches [20] as well as to save computational time (see Section 4). Section 5 summarizes the presented ideas and concludes the paper.

Let us remark, that there is a similar problem studied in statistics which is based on penalizing the models fit to data by the number of their parameters. In this way the complexity of statistical models can be optimized by means of special well-known criteria like Akaikes information criterion (AIC) or Bayes information criterion (BIC). However, in the case of feature selection methods, the resulting subset of features is only a preliminary step to model fitting. Thus, instead of optimizing a single penalized criterion, we only use some very specific properties of feature selection procedures to avoid the negative consequences of possible over-fitting tendencies.

1.1. Feature Subset Selection Problem Formulation

We shall use the term "pattern" to denote the D-dimensional data vector $\mathbf{z} = (z_1, \ldots, z_D)^T$ of measurements, the components of which are the measurements of the features of the entity or object. Following the statistical approach to pattern recognition, we assume that a pattern $\mathbf{z}$ is to be classified into one of a finite set of M different classes $\Omega = \{\omega_1, \omega_2, \ldots, \omega_M\}$. We will focus on the supervised case, where the classification rule is to be learned from training data consisting of a sequence of pattern vectors $\mathbf{z}$ with known class labels.

Given a set Y of $D = |\mathrm{Y}|$ features, let us denote $\mathcal{X}_d$ the set of all possible subsets of size d, where d represents the desired number of features (if possible $d \ll D$). Let us denote $\mathcal{X}$ the set of all possible subsets of Y, of any size. Let $J(\mathrm{X})$ be a criterion function that evaluates feature subset $\mathrm{X} \in \mathcal{X}$. Without any loss of generality, let us consider a higher value of J to indicate a better feature subset. Then the traditional feature selection problem formulation is: Find the subset $\tilde{\mathrm{X}}_d$ for which

$$J(\tilde{\mathrm{X}}_d) = \max_{\mathrm{X} \in \mathcal{X}_d} J(\mathrm{X}). \tag{1}$$

Let the FS methods that solve (1) be denoted as *d-parametrized* methods. The feature selection problem can be formulated more generally as follows: Find the subset $\tilde{\mathrm{X}}$ for which

$$J(\tilde{\mathrm{X}}) = \max_{\mathrm{X} \in \mathcal{X}} J(\mathrm{X}). \tag{2}$$

Let the FS methods that solve (2) be denoted as *d-optimizing* methods. Most of the traditional FS methods are d-parametrized, i. e., they require the user to decide what cardinality the resulting feature subset should have. The d-optimizing FS procedures aim at optimizing both the feature subset size and its contents at once, provided the suitable criterion is available (classifier accuracy estimates in FS wrappers [20] can be used while monotonic probabilistic measures [5] can not). For more details on FS criteria see [5, 20].

Remark 1.1. It should be noted that if no external knowledge is available, determining the correct subspace dimensionality is, in general, a difficult problem depending on the size of training data as well as on model complexity and as such is beyond the scope of this paper.

1.2. Sub-optimal Search Methods

Provided a suitable FS criterion function [5, 20] has been chosen, feature selection is reduced to a search problem that detects an optimal feature subset based on the selected measure. Then the only tool needed is the search algorithm that generates a sequence of feature subsets to be evaluated by the respective criterion (see Figure 1). A very large number of various methods exists. Despite the advances in optimal search [26, 46], for larger than moderate-sized problems we have to resort to sub-optimal methods. (Note that the number of candidate feature subsets to be evaluated increases exponentially with increasing problem dimensionality.) Deterministic heuristic *sub-optimal methods* implement various forms of hill climbing to produce satisfactory results in polynomial time. Unlike *sequential selection* [5], *floating search* does not suffer from the nesting problem [30] and finds good solutions for each subset size [27, 30]. *Oscillating search* and *dynamic oscillating search* can improve existing solutions [43, 45]. Stochastic (randomized) methods like *random subspace* [23], *evolutionary algorithms* [15], *memetic algorithms* [52] or swarm algorithms like *ant colony* [16] may be better suited to over-come local extrema, yet may take longer to converge. The *Relief* algorithm [47] iteratively estimates feature weights according to their ability to discriminate between neighboring patterns. Deterministic search can be notably improved by randomization as in *simulated annealing* [10], *tabu search* [48], randomized *oscillating search* [45] or in combined methods [9]. The fastest and simplest approach to FS is the Best Individual Feature (BIF), or *individual feature ranking*. It is often the only applicable approach in problems of very high dimensionality. BIF is standard in text categorization [37, 50], genetics [35], etc. BIF may be preferable not only in scenarios of extreme computational complexity, but also in cases when FS stability and over-fitting issues hinder considerably the outcome of more complex methods [22, 33]. In order to simplify the presentation of the key paper ideas we will first focus on a family of related FS methods based on the sequential search (hill-climbing) principle.

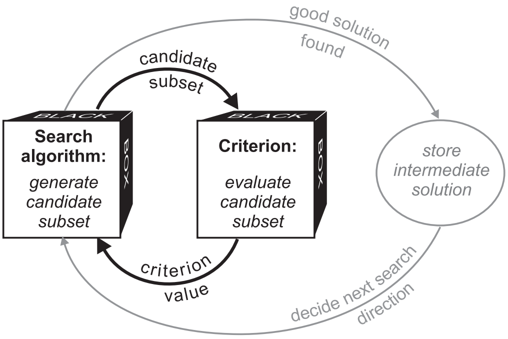
good solution found
candidate subset
BLACK
Search algorithm:
generate candidate subset
BOX
BLACK
Criterion:
evaluate candidate subset
BOX
store intermediate solution
criterion value
decide next search direction

Fig. 1. Feature selection algorithms can be viewed as black box procedures generating a sequence of candidate subsets with respective criterion values, among which intermediate solutions are chosen.

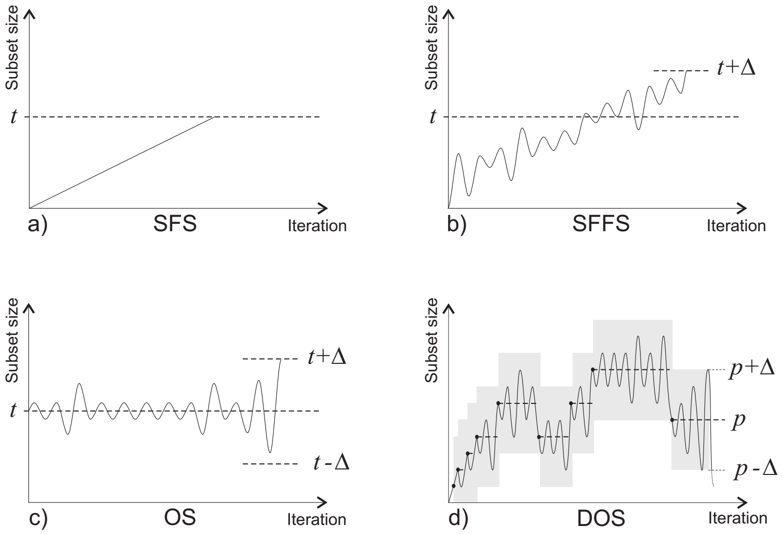

Fig. 2. Comparing subset search methods' course of search. a) Sequential Forward Selection, b) Sequential Forward Floating Selection, c) Oscillating Search, d) Dynamic Oscillating Search.

1.2.1. Decomposing Sequential Search Methods

To simplify the discussion of the schemes to be proposed let us focus only on the family of sequential search methods. Most of the known sequential FS algorithms share the same "core mechanism" of adding and removing features (c-tuples of c features) to/from a working subset. The respective algorithm steps can be described as follows:

Definition 1.1. Let $ADD_c()$ be the operation of adding the *most significant* feature c-tuple $\mathcal{T}_c^+$ to the working set X_d to obtain X_{d+c}:

$$X_{d+c} = X_d \cup \mathcal{T}_c^+ = ADD_c(X_d), \qquad X_d, X_{d+c} \subseteq Y \tag{3}$$

where

$$\mathcal{T}_c^+ = \arg \max_{\mathcal{T}_c \in Y \setminus X_d} \mathcal{J}^+(X_d, \mathcal{T}_c) \tag{4}$$

with $\mathcal{J}^+(X_d, \mathcal{T}_c)$ denoting the evaluation function form used to evaluate the subset obtained by adding $\mathcal{T}_c$, where $\mathcal{T}_c \subseteq Y \setminus X_d$, to X_d.

Definition 1.2. Let $RMV_c()$ be the operation of removing the *least significant* feature c-tuple $\mathcal{T}_c^-$ from the working set X_d to obtain set X_{d-c}:

$$X_{d-c} = X_d \setminus \mathcal{T}_c^- = RMV_c(X_d), \qquad X_d, X_{d-c} \subseteq Y \tag{5}$$

where

$$\mathcal{T}_c^- = \arg \max_{\mathcal{T}_c \in X_d} \mathcal{J}^-(X_d, \mathcal{T}_c) \tag{6}$$

with $\mathcal{J}^-(X_d, \mathcal{T}_c)$ denoting the evaluation function form used to evaluate the subset obtained by removing $\mathcal{T}_c$, where $\mathcal{T}_c \subseteq X_d$, from X_d.

In standard sequential FS methods the impact of feature c-tuple adding (or removal) in one algorithm step is evaluated directly using a single chosen FS criterion function $J(\cdot)$, usually filter- or wrapper-based (see Section 1):

$$\mathcal{J}^+(X_d, \mathcal{T}_c) = J(X_d \cup \mathcal{T}_c), \qquad \mathcal{J}^-(X_d, \mathcal{T}_c) = J(X_d \setminus \mathcal{T}_c)\ . \tag{7}$$

1.2.2. Simplified View of Sequential Search Methods

In order to simplify the notation for a repeated application of FS operations we introduce the following useful notation

$$\begin{aligned} X_{d+2c} = ADD_c(X_{d+c}) = ADD_c(ADD_c(X_d)) &= ADD_c^2(X_d)\ , \\ X_{d-2c} = RMV_c(RMV_c(X_d)) &= RMV_c^2(X_d)\ , \end{aligned} \tag{8}$$

and more generally

$$X_{d+\delta c} = ADD_c^\delta(X_d), \qquad X_{d-\delta c} = RMV_c^\delta(X_d) \tag{9}$$

Using this notation we can now outline the basic idea behind standard sequential FS algorithms very simply. For instance:

(Generalized) *Sequential Forward Selection*, (G)SFS [5, 49] yielding a subset of t features, $t = \delta c$, evaluating feature c-tuples in each step (by default $c = 1$):

1. $X_t = ADD_c^\delta(\emptyset)$.

(Generalized) *Sequential Forward Floating Selection*, (G)SFFS [30] yielding a subset of t features, $t = \delta c$, $t < D$, evaluating feature c-tuples in each step (by default $c = 1$), with optional search-restricting parameter $\Delta \in [0, D - t]$. Throughout the search all so-far best subsets of δc features, $\delta = 1, \ldots, \lfloor \frac{t+\Delta}{c} \rfloor$ are kept:

1. Start with $X_c = ADD_c(\emptyset)$, $d = c$.
2. $X_{d+c} = ADD_c(X_d)$, $d = d + c$.
3. Repeat $X_{d-c} = RMV_c(X_d)$, $d = d - c$ as long as it improves solutions already known for the lower d.
4. If $d < t + \Delta$ go to 2, otherwise return the best known subset of t features as result.

(Generalized) *Oscillating Search*, (G)OS [45] yielding a subset of t features, $t < D$, evaluating feature c-tuples in each step (by default $c = 1$), with optional search-restricting parameter $\Delta \geq 1$:

1. Start with arbitrary initial set X_t of t features. Set cycle depth to $\delta = 1$.
2. Let $X_t^\downarrow = ADD_c^\delta(RMV_c^\delta(X_t))$.
3. If $X_t^\downarrow$ better than X_t, let $X_t = X_t^\downarrow$, let $\delta = 1$ and go to 2.

4. Let $X_t^{\uparrow} = RMV_c^{\delta}(ADD_c^{\delta}(X_t))$.
5. If $X_t^{\uparrow}$ better than X_t, let $X_t = X_t^{\uparrow}$, let $\delta = 1$ and go to 2.
6. If $\delta c < \Delta$ let $\delta = \delta + 1$ and go to 2.

(Generalized) *Dynamic Oscillating Search*, (G)DOS [43] yielding a subset of optimized size p, evaluating feature c-tuples in each step (by default $c = 1$), with optional search-restricting parameter $\Delta \geq 1$:

1. Start with $X_p = ADD_c^3(\emptyset)$, $p = 3c$, or with arbitrary X_p, $p \in \{1, \ldots, D\}$. Set cycle depth to $\delta = 1$.
2. While computing $ADD_c^{\delta}(RMV_c^{\delta}(X_t))$ if any intermediate subset of i features, X_i, $i \in \{p - \delta c, p - (\delta - 1)c, \ldots, p\}$ is found better than X_p, let it become the new X_p with $p = i$, let $\delta = 1$ and restart step 2.
3. While computing $RMV_c^{\delta}(ADD_c^{\delta}(X_t))$; if any intermediate subset of j features, X_j, $j \in \{p, p+c, \ldots, p+\delta c\}$ is found better than X_p, let it become the new X_p with $p = j$, let $\delta = 1$ and go to 2.
4. If $\delta c < \Delta$ let $\delta = \delta + 1$ and go to 2.

Obviously, other FS methods can be described using the notation above as well.

See Figure 2 for visual comparison of the respective methods' course of search. Note that (G)SFS, (G)SFFS and (G)OS have been originally defined as d-parametrized, while (G)DOS is d-optimizing. Nevertheless, many d-parametrized methods evaluate subset candidates of various cardinalities throughout the course of search and thus in principle permit d optimization as well.

2. THE PROBLEM OF FRAGILE FEATURE SUBSET PREFERENCE AND ITS RESOLUTION

In FS algorithm design it is generally assumed that *any* improvement in the criterion value leads to better feature subset. Nevertheless, this principle has been challenged [31, 33, 34] showing that the strict application of this rule may easily lead to over-fitting and consequently to poor generalization performance even with the best available FS evaluation schemes. Unfortunately, there seems to be no way of defining FS criteria capable of avoiding this problem in general, as no criterion can substitute for possibly non-representative training data.

Many common FS algorithms (see Section 1.2) can be viewed as generators of a sequence of candidate feature subsets and respective criterion values (see Figure 1). Intermediate solutions are usually selected among the candidate subsets as the ones with the highest criterion value discovered so far. Intermediate solutions are used to further guide the search. The solution with the highest overall criterion value is eventually considered to be the result. In the course of the search the candidate feature subsets may yield fluctuating criterion values while the criterion values of intermediate solutions usually form a nondecreasing sequence. The search generally continues as long as intermediate solutions improve, no matter how significant the improvement is and often without respect to other effects like excessive subset size

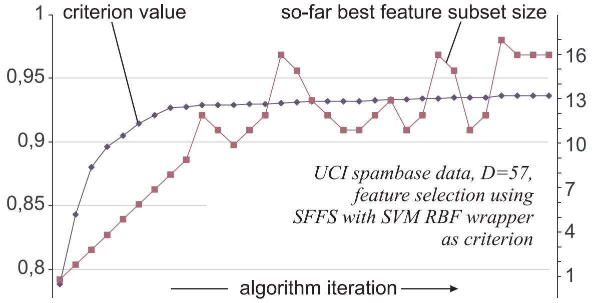

Fig. 3. In many FS tasks very low criterion increase is accompanied by fluctuations in selected subsets; both in size and contents

increase, although it is known that increasing model dimensionality in itself increases the risk of over-fitting.

Therefore, in this section we present a workaround targeted specifically at improving the robustness of decisions about feature inclusion/removal in the course of feature subset search.

2.1. The Problem of Fragile Feature Preference

In many FS tasks it can be observed that the difference between criterion values of successive intermediate solutions decreases in time and often becomes negligible. Yet minimal change in criterion value may be accompanied by substantial changes in subset contents. This can easily happen, e. g., when many of the considered features are important but dependent on each other to various degrees with respect to the chosen criterion, or when there is large number of features carrying limited but nonzero information (this is common, e. g., in text categorization [37]). We illustrate this phenomenon in Figure 3, showing the process of selecting features on *spambase* data [8] using SFFS algorithm [30] and estimated classification accuracy of Support Vector Machine (SVM, [2]) as criterion [20]. Considering only those tested subset candidates with criterion values within 1% difference from the final maximum achieved value, i. e., values from $[0.926, 0.936]$, their sizes fluctuate from 8 to 17. This sequence of candidate subsets yields *average Tanimoto distance* ATI [18, 42] as low as 0.551 on the scale $[0, 1]$ where 0 marks disjunct sets and 1 marks identical sets. This suggests that roughly any two of these subsets differ almost in half of their contents. Expectedly, notable fluctuations in feature subset contents following from minor criterion value improvement are unlikely to lead to reliable final classification system. We will refer to this effect of undue criterion sensitivity as feature *over-evaluation*. Correspondingly, Raudys [31] argues that to prevent over-fitting it may be better to consider a subset with slightly lower than the best achieved criterion value as a FS result.

2.2. Tackling The Problem of Fragile Feature Subset Preferences

Following the observations above, we propose to *treat as equal* (effectively indistinguishable) all subsets known to yield *primary criterion* value within a pre-defined (very small) distance from the maximum known at the current algorithm stage [40]. Intermediate solutions then need to be selected from the treated-as-equal subset groups using a suitable complementary *secondary criterion.* A good complementary criterion should be able to compensate for the primary criterion's deficiency in distinguishing among treated-as-equal subsets. Nevertheless, introducing the secondary criterion opens up alternative usage options as well, see Section 2.2.2.

The idea of the secondary criterion is similar to the principle of penalty functions as used, e. g., in two-part objective function consisting of goodness-of-fit and number-of-variables parts [12]. However, in our approach we propose to keep the evaluation of primary and secondary criteria separated. Avoiding the combination of two criteria into one objective function is advantageous as it a) avoids the problem of finding reasonable combination parameters (weights) of potentially incompatible objective function parts and b) enables to use the secondary criterion as supplement only in cases when the primary criterion response is not decisive enough.

Remark 2.1. The advantage of separate criteria evaluation comes at the cost of necessity to specify which subset candidates are to be treated as equal, i. e., to set a threshold depending on the primary criterion. This, however, is transparent to define (see below) and, when compared to two-part objective functions, allows for finer control of the FS process.

2.2.1. Complementary Criterion Evaluation Mechanism

Let $J_1(\cdot)$ denote the primary FS criterion to be maximized by the chosen FS algorithm. Let $J_2(\cdot)$ denote the secondary (complementary) FS criterion for resolving the "treated-as-equal" cases. Without any loss of generality we assume that a higher J_2 value denotes a more preferable subset (see Section 2.2.2 for details). Let $\tau \in [0, 1]$ denote the *equality threshold* parameter. Throughout the course of the search, two pivot subsets, $\mathrm{X}^{\max}$ and $\mathrm{X}^{\mathrm{sel}}$, are to be updated after each criterion evaluation. Let $\mathrm{X}^{\max}$ denote the subset yielding the maximum J_1 value known so far. Let $\mathrm{X}^{\mathrm{sel}}$ denote the currently selected subset (intermediate solution). When the search process ends, $\mathrm{X}^{\mathrm{sel}}$ is to become the final solution.

The chosen backbone FS algorithm is used in its standard way to maximize J_1. It is the mechanism proposed below that simultaneously keeps selecting an intermediate result $\mathrm{X}^{\mathrm{sel}}$ among the currently known "treated-as-equal" alternatives to the current $\mathrm{X}^{\max}$, allowing $\mathrm{X}^{\mathrm{sel}} \neq \mathrm{X}^{\max}$ if $\mathrm{X}^{\mathrm{sel}}$ is better than $\mathrm{X}^{\max}$ with respect to J_2 while being only negligibly worse with respect to J_1, i. e., provided

$$J_1(\mathrm{X}^{\mathrm{sel}}) \geq (1 - \tau) \cdot J_1(\mathrm{X}^{\max}) \wedge J_2(\mathrm{X}^{\mathrm{sel}}) > J_2(\mathrm{X}^{\max}) \; . \tag{10}$$

Algorithmic Extension: Whenever the backbone FS algorithm evaluates a feature subset X (depicting any subset evaluated at any algorithm stage), the following update sequence is to be called:

1. If $J_1(\mathrm{X}) \leq J_1(\mathrm{X}^{\max})$, go to 4.
2. Make X the new $\mathrm{X}^{\max}$.
3. If $J_1(\mathrm{X}^{\mathrm{sel}}) < (1-\tau) \cdot J_1(\mathrm{X}^{\max}) \vee J_2(\mathrm{X}^{\mathrm{sel}}) \leq J_2(\mathrm{X}^{\max})$, make X also the new $\mathrm{X}^{\mathrm{sel}}$ and stop this update sequence.
4. If $\big(J_1(\mathrm{X}) \geq (1-\tau) \cdot J_1(\mathrm{X}^{\max}) \wedge J_2(\mathrm{X}) > J_2(\mathrm{X}^{\mathrm{sel}})\big) \vee \big(J_2(\mathrm{X}) = J_2(\mathrm{X}^{\mathrm{sel}}) \wedge J_1(\mathrm{X}) > J_1(\mathrm{X}^{\mathrm{sel}})\big)$, make X the new $\mathrm{X}^{\mathrm{sel}}$ and stop this update sequence.

The proposed mechanism does not affect the course of the search of the primary FS algorithm; it only adds a form of lazy solution update. Note that the presented mechanism is applicable with a large class of FS algorithms (cf. Sect 2).

Remark 2.2. Note that in a single backbone FS algorithm run it is easily possible to collect solutions for an arbitrary number of τ values. The technique does not add any additional computational complexity burden to the backbone FS algorithm.

Remark 2.3. Note that to further refine the selection of alternative solutions it is possible to introduce another parameter σ as an equality threshold with respect to the criterion J_2. This would prevent selecting set X_1^{sel} at the cost of X_2^{sel} if

$$J_2(\mathrm{X}_1^{\mathrm{sel}}) > J_2(\mathrm{X}_2^{\mathrm{sel}}) \,, \tag{11}$$

but

$$J_2(\mathrm{X}_1^{\mathrm{sel}}) \leq (1+\sigma) \cdot J_2(\mathrm{X}_2^{\mathrm{sel}}) \wedge J_1(\mathrm{X}_2^{\mathrm{sel}}) > J_1(\mathrm{X}_1^{\mathrm{sel}}) \geq (1-\tau) \cdot J_1(\mathrm{X}^{\max}) \,. \tag{12}$$

We do not adopt this additional mechanism in the following so as to avoid the choice of another parameter σ.

2.2.2. Complementary Criterion Usage Options

The J_2 criterion can be utilized for various purposes. Depending on the particular problem, it may be possible to define J_2 to *distinguish better among subsets* that J_1 fails to distinguish reliably enough.

The simplest yet useful alternative is to utilize J_2 for emphasising the *preference of smaller subsets.* To achieve this, J_2 is to be defined as

$$J_2(\mathrm{X}) = -|\mathrm{X}| \,. \tag{13}$$

Smaller subsets not only mean a lower measurement cost, but more importantly in many problems the forced reduction of subset size may help to reduce the risk of over-fitting and improve generalization (see Section 2.3).

More generally, J_2 can be used to incorporate *feature acquisition cost* minimization into the FS process. Provided a weight (cost) $w_i, i = 1, \ldots, D$ is known for each feature, then the appropriate secondary criterion can be easily defined as

$$J_2(\mathrm{X}) = -\sum_{x_i \in \mathrm{X}} w_i \,. \tag{14}$$

Table 1. FS with reduced feature preference fragility for various τ – lower-dimensional data examples. Bullets mark cases where $\tau > 0$ led to improvement.

SFFS		dermatology, $D = 34$, 6 classes, 358 samples			spectf, $D = 44$, 2 classes, 267 samples			spambase, $D = 57$, 2 classes, 4601 samples		
crit.	τ	feat. subs. size	train acc.	test acc.	feat. subs. size	train acc.	test acc.	feat. subs. size	train acc.	test acc.
SVM RBF	0	8	.977	.917	2	.827	.761	16	.937	.883
	0.001	dtto	dtto	dtto	dtto	dtto	dtto	dtto	dtto	dtto
	0.005	dtto	dtto	dtto	dtto	dtto	dtto	10	.934	.879
	0.01	dtto	dtto	dtto	dtto	dtto	dtto	9	.930	.884 ●
	0.02	7	.966	.922 ●	dtto	dtto	dtto	8	.921	.870
	0.03	6	.955	.922 ●	dtto	dtto	dtto	6	.912	.872
	0.04	5	.944	.933 ●	1	.797	.791 ●	5	.904	.871
	0.05	dtto	dtto	dtto	dtto	dtto	dtto	4	.896	.866
3NN	0	16	.994	.933	11	.948	.769	30	.930	.871
	0.001	dtto	dtto	dtto	dtto	dtto	dtto	24	.930	.872 ●
	0.005	dtto	dtto	dtto	dtto	dtto	dtto	20	.926	.871 ●
	0.01	dtto	dtto	dtto	dtto	dtto	dtto	18	.923	.876 ●
	0.02	6	.983	.950 ●	dtto	dtto	dtto	14	.913	.867
	0.03	5	.966	.939 ●	7	.925	.776 ●	9	.905	.856
	0.04	dtto	dtto	dtto	dtto	dtto	dtto	8	.896	.828
	0.05	dtto	dtto	dtto	6	.910	.716	7	.887	.807

Table 2. FS with reduced feature preference fragility for various τ – higher-dimensional data examples. Bullets mark cases where $\tau > 0$ led to improvement.

DOS(15)		gisette, $D = 5000$, 2 classes, 1000 samples			madelon, $D = 500$ 2 classes, 2000 samples			xpxinsar, $D = 57$ 7 classes, 1721 samples		
crit.	τ	feat. subs. size	train acc.	test acc.	feat. subs. size	train acc.	test acc.	feat. subs. size	train acc.	test acc.
SVM RBF	0	10	.922	.856	21	.841	.804	12	.873	.863
	0.001	9	.921	.860 ●	dtto	dtto	dtto	dtto	dtto	dtto
	0.005	7	.918	.862 ●	17	.837	.817 ●	9	.871	.867 ●
	0.01	5	.914	.854	15	.833	.812 ●	7	.866	.897 ●
	0.02	3	.906	.852	13	.825	.816 ●	6	.864	.896 ●
	0.03	dtto	dtto	dtto	dtto	dtto	dtto	5	.856	.871 ●
	0.04	2	.890	.856 ●	12	.811	.793	4	.840	.845
	0.05	dtto	dtto	dtto	dtto	dtto	dtto	dtto	dtto	dtto
3NN	0	15	.958	.904	18	.891	.844	16	.847	.854
	0.001	dtto	dtto	dtto	dtto	dtto	dtto	14	.847	.854 ●
	0.005	13	.954	.898	13	.888	.842	12	.844	.848
	0.01	11	.950	.892	9	.883	.850 ●	10	.840	.847
	0.02	8	.940	.892	7	.877	.848 ●	9	.837	.825
	0.03	6	.930	.874	6	.869	.847 ●	5	.823	.842
	0.04	5	.922	.89	5	.858	.854 ●	dtto	dtto	dtto
	0.05	4	.914	.87	dtto	dtto	dtto	4	.812	.837

2.3. Experimental Results

We illustrate the potential of the proposed methodology on a series of experiments where J_2 was used for emphasising the *preference of smaller subsets* (see Section 2.2.2). For this purpose we used several data-sets from UCI repository [8] and one data-set – xpxinsar satellite – from Salzburg University. Table 1 demonstrates the results obtained using the extended version (see Section 2.2) of the Sequential Forward Floating Search (SFFS, [30]). Table 2 demonstrates results obtained using the extended version (see Section 2.2) of the Dynamic Oscillating Search (DOS, [43]). For simplification we consider only single feature adding/removal steps (c-tuples with $c = 1$). Both methods have been used in wrapper setting [20], i. e., with estimated classifier accuracy as FS criterion. For this purpose we have used a Support Vector Machine (SVM) with Radial Basis Function kernel [2] and 3-Nearest Neighbor classifier accuracy estimates. To estimate final classifier accuracy on independent data we split each dataset to equally sized parts; the training part was used in 3-fold Cross-Validation manner to evaluate wrapper criteria in the course of FS process, the testing part was used only once for independent classification accuracy estimation.

We repeated each experiment for different *equality thresholds* τ, ranging from 0.001 to 0.05 (note that due to the wrapper setting both considered criteria yield values from $[0, 1]$). Tables 1 and 2 show the impact of changing equality threshold to classifier accuracy on independent data. The first row ($\tau = 0$) equals standard FS algorithm operation without the extension proposed in this paper. The black bullet points emphasize cases where the proposed mechanism has led to an improvement, i. e., the selected subset size has been reduced with better or equal accuracy on independent test data. Underlining emphasizes those cases where the difference from the ($\tau = 0$) case has been confirmed by statistical significance t-test at significance level 0.05. Note that the positive effect of nonzero τ can be observed in a notable number of cases. Note in particular that in many cases the number of features could be reduced to less than one half of what would be the standard FS method's result (cf. in Table 1 the dermatology–3NN case and in Table 2 the gisette–SVM, xpxinsar–SVM and madelon–3NN cases). However, it can be also seen that the effect is strongly case dependent. It is hardly possible to give a general recommendation about the suitable τ value, except that improvements in some of our experiments have been observed for various τ values up to roughly 0.1.

Remark 2.4. Let us note that the reported statistical significance test results in this paper are of complementary value only as our primary aim is to illustrate general methodological aspects of feature selection and not to study concrete tasks in detail.

3. CRITERIA ENSEMBLES IN FEATURE SELECTION

It has been shown repeatedly in literature that classification system performance may be considerably improved in some cases by means of a classifier combination [19]. In multiple-classifier systems FS is often applied separately to yield different subsets for each classifier in the system [7, 11]. Another approach is to select one feature subset to be used in all co-operating classifiers [6, 32].

In contrary to such approaches, we propose to utilize the idea of combination to eventually produce one feature subset to be used with one classifier [39]. We propose to combine FS criteria with the aim of obtaining a feature subset that has better generalization properties than subsets obtained using a single criterion. In the course of the FS process we evaluate several criteria simultaneously and, at any selection step, the best features are identified by combining the criteria output. In the following we show that subsets obtained by combining selection criteria output using voting and weighted voting are more stable and improve the classifier performance on independent data in many cases. Note that this technique follows a similar goal as the one presented in Section 2.

3.1. Combining Multiple Criteria

Different criterion functions may reflect different properties of the evaluated feature subsets. Incorrectly chosen criterion may easily lead to the wrong subset (cf. feature over-evaluation, see Section 2.1). Combining multiple criteria is justifiable from the same reasons as traditional multiple classifier systems. It should reduce the tendency to over-fit by preferring features that perform well with respect to several various criteria instead of just one and consequently enable to improve the generalization properties of the selected subset of features. The idea is to reduce the possibility of a single criterion to exploit too strongly the specific properties of training data, that may not be present in independent test data.

In the following we discuss several straight-forward approaches to criteria combination by means of re-defining $\mathcal{J}^+$ and $\mathcal{J}^-$ in expression (7) for use in Definitions 1.1 and 1.2. We will consider ensembles of arbitrary feature selection criteria $J^{(k)}$, $k = 1, \ldots, K$. In Section 3.2 concrete example will be given for ensemble consisting of criteria $J^{(k)}$, $k = 1, \ldots, 4$, standing for the estimated accuracy of $(2k-1)$-Nearest Neighbor classifier.

3.1.1. Multiple Criterion Voting

The most universal way to realize the idea of criterion ensemble is to implement a form of voting. The intention is to reveal stability in feature (or feature c-tuple $\mathcal{T}_c$) preferences, with no restriction on the principle or behavior of the combined criteria $J^{(k)}$, $k = 1, \ldots, K$. Accordingly, we will redefine $\mathcal{J}^+$ and $\mathcal{J}^-$ to express averaged feature c-tuple ordering preferences instead of directly combining criterion values.

In the following we define $\mathcal{J}^+_{\text{order}}$ as replacement of $\mathcal{J}^+$ in Definition 1.1. The following steps are to be taken separately for each criterion $J^{(k)}$, $k = 1, \ldots, K$ in the considered ensemble of criteria. First, evaluate all values $J^{(k)}(X_d \cup \mathcal{T}_{c,i})$ for fixed k and $i = 1, \ldots, T$, where $T = \binom{D-d}{c}$, and $\mathcal{T}_{c,i} \subseteq Y \setminus X_d$. Next, order these values descending with possible ties resolved arbitrarily at this stage and encode the ordering using indexes $i_j, j = 1, \ldots, T, i_j \in [1, T]$ where $i_m \neq i_n$ for $m \neq n$:

$$J^{(k)}(X_d \cup \mathcal{T}_{c,i_1}) \geq J^{(k)}(X_d \cup \mathcal{T}_{c,i_2}) \geq \cdots \geq J^{(k)}(X_d \cup \mathcal{T}_{c,i_T}) \; . \tag{15}$$

Next, express feature c-tuple preferences using coefficient $\alpha_j^{(k)}$, $j = 1, \ldots, T$, defined to take into account possible feature c-tuple preference ties as follows:

$$\alpha_{i_1}^{(k)} = 1 \tag{16}$$

$$\alpha_{i_j}^{(k)} = \begin{cases} \alpha_{i_{j-1}}^{(k)} & \text{if } J^{(k)}(X_d \cup \mathcal{T}_{c,i_{(j-1)}}) = J^{(k)}(X_d \cup \mathcal{T}_{c,i_j}) \\ \alpha_{i_{j-1}}^{(k)} + 1 & \text{if } J^{(k)}(X_d \cup \mathcal{T}_{c,i_{(j-1)}}) > J^{(k)}(X_d \cup \mathcal{T}_{c,i_j}) \end{cases} \quad \text{for } j \geq 2 \, .$$

Coefficient $\alpha_j^{(k)}$ can be viewed as a feature c-tuple index in a list ordered according to criterion $J^{(k)}$ values, where c-tuples with equal criterion value all share the same position in the list.

Now, having collected the values $\alpha_j^{(k)}$ for all $k = 1, \ldots, K$ and $j = 1, \ldots, D - d$ we can transform the criteria votes to a form usable in Definition 1.1 by defining:

$$\mathcal{J}^{+}_{\text{order}}(X_d, \mathcal{T}_{c,i}) = -\frac{1}{K} \sum_{k=1}^{K} \alpha_i^{(k)} \, . \tag{17}$$

The definition of $\mathcal{J}^{-}_{\text{order}}$ is analogous.

Remark 3.1. Note that alternative schemes of combining the information on ordering coming from various criteria can be considered. Note, e. g., that in the expression (16) all subsets that yield equal criterion value get the the same lowest available index. If such ties occur frequently, it might be better to assign an index median within each group of equal subsets so as to prevent disproportionate influence of criteria that tend to yield less distinct values.

3.1.2. Multiple Criterion Weighted Voting

Suppose we introduce an additional restriction ono the values yielded by criteria $J^{(k)}$, $k = 1, \ldots, K$ in the considered ensemble. Suppose each $J^{(k)}$ yields values from the same interval. This is easily fulfilled, e. g., in wrapper FS methods [20] where the estimated correct classification rate is usually normalized to $[0, 1]$. Now the differences between $J^{(k)}$ values (for fixed k) can be treated as weights expressing relative feature c-tuple preferences of criterion k. In the following we define $\mathcal{J}^{+}_{\text{weight}}$ as replacement of $\mathcal{J}^{+}$ in Definition 1.1. The following steps are to be taken separately for each criterion $J^{(k)}$, $k = 1, \ldots, K$ in the considered ensemble of criteria. First, evaluate all values $J^{(k)}(X_d \cup \mathcal{T}_{c,i})$ for fixed k and $i = 1, \ldots, T$, where $T = \binom{D-d}{c}$, and $\mathcal{T}_{c,i} \subseteq Y \backslash X_d$. Next, order the values descending with possible ties resolved arbitrarily at this stage and encode the ordering using indexes $i_j, j = 1, \ldots, T, i_j \in [1, T]$ in the same way as shown in (15). Now, express feature c-tuple preferences using coefficient $\beta_j^{(k)}$, $j = 1, \ldots, T$ defined to take into account the differences between the impact the various feature c-tuples from $Y \setminus X_d$ have on the criterion value:

$$\beta_{i_j}^{(k)} = J^{(k)}(X_d \cup \mathcal{T}_{c,i_1}) - J^{(k)}(X_d \cup \mathcal{T}_{c,i_j}) \text{ for } j = 1, \ldots, T \, . \tag{18}$$

Now, having collected the values $\beta_j^{(k)}$ for all $k = 1, \ldots, K$ and $j = 1, \ldots, T$ we can transform the criteria votes to a form usable in Definition 1.1 by defining:

$$\mathcal{J}^{+}_{\text{weight}}(X_d, \mathcal{T}_{c,i}) = -\frac{1}{K}\sum_{k=1}^{K}\beta_i^{(k)} \ . \tag{19}$$

The definition of $\mathcal{J}^{-}_{\text{weight}}$ is analogous.

3.1.3. Resolving Voting Ties

Especially in small sample data where the discussed techniques are of particular importance it may easily happen that

$$\mathcal{J}^{+}_{\text{order}}(X_d, \mathcal{T}_{c,i}) = \mathcal{J}^{+}_{\text{order}}(X_d, \mathcal{T}_{c,j}) \quad \text{for} \quad i \neq j \ . \tag{20}$$

(The same can happen for $\mathcal{J}^{-}_{\text{order}}$, $\mathcal{J}^{+}_{\text{weight}}$, $\mathcal{J}^{-}_{\text{weight}}$.) To resolve such ties we employ an additional mechanism. To resolve $\mathcal{J}^{+}$ ties we collect in the course of FS process for each feature c-tuple $\mathcal{T}_{c,i}$, $i = 1, \dots, \binom{D}{c}$ the information about all values (17) or (19), respectively, evaluated so far. In case of $\mathcal{J}^{+}$ ties this collected information is used in that the c-tuple with the highest average value of (17) or (19), respectively, is preferred. (Tie resolution for J^{-}_{order}, J^{+}_{weight}, J^{-}_{weight} is analogous.)

3.2. Experimental Results

We performed a series of FS experiments on various data-sets from UCI repository [8] and one data-set (xpxinsar satellite) from Salzburg University. Many of the data-sets have small sample size with respect to dimensionality. In this type of problem any improvement of generalization properties plays a crucial role. To put the robustness of the proposed criterion voting schemes to the test we used the *Dynamic Oscillating Search* algorithm [43] in all experiments as one of the strongest available subset optimizers, with high risk of over-fitting. For simplification we consider only single feature adding/removal steps (c-tuples with $c = 1$).

To illustrate the concept we have resorted to combining classification accuracy of four simple wrappers in all experiments – *k-Nearest Neighbor* (k-NN) classifiers for $k = 1, 3, 5, 7$, as the effects of increasing k are well understandable. With increasing k the k-NN class-separating hyperplane gets smoother – less affected by outliers but also less sensitive to possibly important detail.

Each experiment was run using 2-tier cross-validation. In the "outer" 10-fold cross-validation the data was repeatedly split to 90 % training part and 10 % testing part. FS was done on the training part. Because we used the *wrapper* FS setup, each criterion evaluation involved classifier accuracy estimation on the training data part. To utilize the information in training data better, the estimation was realized by means of "inner" 10-fold cross-validation, i. e., the training data was repeatedly split to 90% sub-part used for classifier training and 10% sub-part used for classifier validation. The averaged classifier accuracy then served as single FS criterion output. Each selected feature subset was eventually evaluated on the 3-NN classifier, trained on the training part and tested on the testing part of the "outer" data split. The

Table 3. ORDER VOTING. Comparing single-criterion and multiple-criterion FS (first and second row for each data-set). All reported classification rates obtained using 3-NN classifier on independent test data. Improvement emphasized in bold (the higher the classification rate and/or stability measures' value the better).

Data	Dim.	Classes	Rel. samp. size	k-NN (k)	Classsif. rate		Subset size d		FS Stability		Time
					Mean	*S.Dv.*	Mean	*S.Dv.*	CW *rel*	ATI	
derm	36	6	1.66	3	.970	*.023*	9.6	*0.917*	.597	.510	3m
				1,3,5,7	**.978**	*.027*	10.7	*1.676*	.534	.486	16m
house	14	5	7.23	3	.707	*.088*	4.9	*1.513*	.456	.478	1m
				1,3,5,7	.689	*.101*	5.4	*1.744*	**.497**	**.509**	5m
iono	34	2	5.16	3	.871	*.078*	5.6	*1.500*	.303	.216	2m
				1,3,5,7	**.882**	*.066*	4.7	*1.269*	**.441**	**.325**	6m
mam	65	2	0.66	3	.821	*.124*	4.2	*1.833*	.497	.343	30s
mo				1,3,5,7	**.846**	*.153*	3	*1.483*	**.519**	**.420**	80s
opt38	64	2	8.77	3	.987	*.012*	9	*1.414*	.412	.297	2m
				1,3,5,7	.987	*.012*	9.5	*1.360*	**.490**	**.362**	6m
sati	36	6	20.53	3	.854	*.031*	14.2	*3.156*	.347	.392	33h
				1,3,5,7	**.856**	*.037*	14.5	*3.801*	**.357**	**.399**	116h
segm	19	7	17.37	3	.953	*.026*	4.7	*1.735*	.610	.550	35m
				1,3,5,7	**.959**	*.019*	4.6	*2.245*	**.625**	**.601**	2h
sonar	60	2	1.73	3	.651	*.173*	12.8	*4.895*	.327	.260	7m
				1,3,5,7	**.676**	*.130*	8.8	*4.020*	**.350**	.260	16m
specf	44	2	3.03	3	.719	*.081*	9.5	*4.522*	.174	.157	4m
				1,3,5,7	**.780**	*.111*	9.8	*3.092*	**.255**	**.237**	15m
wave	40	3	41.67	3	.814	*.014*	17.2	*2.561*	.680	.657	62h
				1,3,5,7	**.817**	*.011*	16.4	*1.356*	**.753**	**.709**	170h
wdbc	30	2	9.48	3	.965	*.023*	10.3	*1.676*	.327	.345	12m
				1,3,5,7	**.967**	*.020*	10.1	*3.176*	**.360**	**.375**	41m
wine	13	3	4.56	3	.966	*.039*	5.9	*0.831*	.568	.594	15s
				1,3,5,7	.960	*.037*	6	*1.000*	**.575**	**.606**	54s
wpbc	31	2	3.19	3	.727	*.068*	9.1	*3.048*	.168	.211	2m
				1,3,5,7	.727	*.056*	7.2	*2.600*	**.189**	.188	5m
xpxi	57	7	4.31	3	.895	*.067*	10.8	*1.939*	.618	.489	5h
				1,3,5,7	.894	*.069*	11.5	*3.233*	**.630**	**.495**	21h

resulting classification accuracy, averaged over "outer" data splits, is reported in Tables 3 and 4.

In both Tables 3 and 4 for each data-set the multiple-criterion results (second row) are compared to the single-criterion result (first row) obtained using 3-NN as wrapper. For each data-set its basic parameters are reported, including its class-averaged dimensionality-to-class-size ratio. Note that in each of the "outer" runs possibly different feature subset can be selected. The stability of feature preferences across the "outer" cross-validation runs has been evaluated using the stability measures: *relative weighted consistency* CW_{rel} and *averaged Tanimoto distance* ATI [18, 42], both yielding values from $[0, 1]$. In CW_{rel} 0 marks the maximum relative randomness and 1 marks the least relative randomness among the feature subsets (see [42] for details), in ATI 0 marks disjunct subsets and 1 marks identical subsets. We also report the total time needed to complete each 2-tier cross-validation single-threaded experiment on an up-to-date AMD Opteron CPU.

Table 3 illustrates the impact of multiple criterion voting (17) as described in Section 3.1.1. Table 4 illustrates the impact of multiple criterion weighted voting (19)

Table 4. WEIGHTED VOTING. Comparing single-criterion and multiple-criterion FS (first and second row for each data-set). All reported classification rates obtained using 3-NN classifier on independent test data. Improvement emphasized in bold (the higher the classification rate and/or stability measures' value the better).

Data	Dim.	Classes	Rel. samp. size	k-NN (k)	Classsif. rate		Subset size d		FS Stability		FS time
					Mean	*S.Dv.*	Mean	*S.Dv.*	CW *rel*	ATI	h:m:s
derm	36	6	1.66	3	.970	*.023*	9.6	*0.917*	.597	.510	3m
				1,3,5,7	**.978**	*.017*	10.3	*1.552*	**.658**	**.573**	17m
house	14	5	7.23	3	.707	*.088*	4.9	*1.513*	.456	.478	1m
				1,3,5,7	**.716**	*.099*	5.6	*2.29*	**.459**	**.495**	3m
iono	34	2	5.16	3	.871	*.078*	5.6	*1.500*	.303	.216	2m
				1,3,5,7	**.897**	*.059*	4.9	*1.758*	**.393**	**.345**	7m
mam mo	65	2	0.66	3	.821	*.124*	4.2	*1.833*	.497	.343	30s
				1,3,5,7	.813	*.153*	2.6.	*1.428*	**.542**	**.390**	43s
opt38	64	2	8.77	3	.987	*.012*	9	*1.414*	.412	.297	90m
				1,3,5,7	**.988**	*.011*	8.6	*1.020*	**.569**	**.423**	08h
sati	36	6	20.53	3	.854	*.031*	14.2	*3.156*	.347	.392	33h
				1,3,5,7	**.856**	*.038*	13.8	*2.182*	**.448**	**.456**	99h
segm	19	7	17.37	3	.953	*.026*	4.7	*1.735*	.610	.550	35m
				1,3,5,7	**.959**	*.019*	4.6	*2.245*	**.644**	**.610**	02h
sonar	60	2	1.73	3	.651	*.173*	12.8	*4.895*	.327	.260	7m
				1,3,5,7	.614	*.131*	10.1	*3.015*	.301	.224	20m
specf	44	2	3.03	3	.719	*.081*	9.5	*4.522*	.174	.157	4m
				1,3,5,7	**.787**	*.121*	9.1	*3.590*	**.285**	**.229**	18m
wave	40	3	41.67	3	.814	*.014*	17.2	*2.561*	.680	.657	62h
				1,3,5,7	.814	*.016*	16.9	*1.700*	**.727**	**.700**	287h
wdbc	30	2	9.48	3	.965	*.023*	10.3	*1.676*	.327	.345	12m
				1,3,5,7	**.967**	*.020*	10.3	*4.267*	**.352**	**.346**	55m
wine	13	3	4.56	3	.966	*.039*	5.9	*0.831*	.568	.594	15s
				1,3,5,7	.960	*.037*	6.6	*1.200*	.567	**.606**	28s
wpbc	31	2	3.19	3	.727	*.068*	9.1	*3.048*	.168	.211	2m
				1,3,5,7	.686	*.126*	6.9	*2.508*	**.211**	.192	4m
xpxi	57	7	4.31	3	.895	*.067*	10.8	*1.939*	.618	.489	5h
				1,3,5,7	.895	*.071*	11	*2.683*	.595	.475	38h

as described in Section 3.1.2. Improvements are emphasized in bold. Underlining emphasizes those cases where the difference from the single-criterion case has been confirmed by statistical significance t-test at significance level 0.05. The results presented in both Tables 3 and 4 clearly show that the concept of criteria ensemble has the potential to improve both the generalization ability (as illustrated by improved classification accuracy on independent test data) and FS stability (sensitivity to perturbations in training data). Note that the positive effect of either (17) or (19) is not present in all cases (in some cases the performance degraded as with *house* dataset in Table 3 and *sonar* and *wpbc* datasets in Table 4) but it is clearly prevalent among the tested datasets. It can be also seen that none of the presented schemes can be identified as the universally better choice.

4. FEATURE SELECTION HYBRIDIZATION

In the following we will finally investigate the hybrid approach to FS that aims to combine the advantages of filter and wrapper algorithms [20]. The main advantage of filter methods is their speed, ability to scale to large data sets, and better resitance to over-fitting. A good argument for wrapper methods is that they tend to give a superior performance for specific classifiers. FS hybridization has been originally defined to achieve best possible (wrapper-like) performance with the time complexity comparable to that of the filter algorithms [25, 41, 44]. In the following we show that apart from reduced search complexity this approach can also improve the generalization ability of the final classification system.

Hybrid FS algorithms can be easily defined in the context of sequential search (see Section 1.2.1). Throughout the course of sequential feature selection process, in each step the filter criterion is used to reduce the number of candidates to be eventually evaluated by the wrapper criterion. The scheme can be applied in any sequential FS algorithms (see Section 1.2) by replacing Definitions 1.1 and 1.2 by Definitions 4.1 and 4.2 given below. For the sake of simplicity let $J_F(.)$ denote the faster but for the given problem less specific *filter* criterion, $J_W(.)$ denote the slower but more problem-specific *wrapper* criterion. The *hybridization coefficient*, defining the proportion of feature subset candidate evaluations to be accomplished by wrapper means, is denoted by $\lambda \in [0,1]$. Note that choosing $\lambda < 1$ reduces the number of J_W computations but the number of J_F computations remains unchanged. In the following $\lfloor \cdot \rceil$ denotes value rounding.

Definition 4.1. For a given current feature set X_d and given $\lambda \in [0,1]$, denote $T^+ = \binom{D-d}{c}$, and let Z^+ be the set of candidate feature c-tuples

$$\mathrm{Z}^+ = \{\mathcal{T}_{c,i} : \mathcal{T}_{c,i} \subseteq \mathrm{Y} \setminus \mathrm{X}_d; i = 1, \ldots, \max\{1, \lfloor \lambda \cdot T^+ \rceil\}\} \tag{21}$$

such that

$$\begin{gathered} \forall \mathcal{T}_c^{'}, \mathcal{T}_c^{''} \subseteq \mathrm{Y} \setminus \mathrm{X}_d, \mathcal{T}_c^{'} \in \mathrm{Z}^+, \mathcal{T}_c^{''} \notin \mathrm{Z}^+ \\ J_F^+(\mathrm{X}_d, \mathcal{T}_c^{'}) \geq J_F^+(\mathrm{X}_d, \mathcal{T}_c^{''}) \ , \end{gathered} \tag{22}$$

where $J_F^+(\mathrm{X}_d, \mathcal{T}_c)$ denotes the pre-filtering criterion function used to evaluate the subset obtained by adding c-tuple $\mathcal{T}_c$ ($\mathcal{T}_c \subseteq \mathrm{Y} \setminus \mathrm{X}_d$) to X_d. Let $\mathcal{T}_c^+$ be the feature c-tuple such that

$$\mathcal{T}_c^+ = \arg \max_{\mathcal{T}_c \in \mathrm{Z}^+} J_W^+(\mathrm{X}_d, \mathcal{T}_c) \ , \tag{23}$$

where $J_W^+(\mathrm{X}_d, \mathcal{T}_c)$ denotes the main criterion function used to evaluate the subset obtained by adding c-tuple $\mathcal{T}_c$ ($\mathcal{T}_c \in \mathrm{Z}^+$) to X_d. Then we shall say that $hADD_c(\mathrm{X}_d)$ is an operation of adding feature c-tuple $\mathcal{T}_c^+$ to the current set X_d to obtain set X_{d+c} if

$$hADD_c(\mathrm{X}_d) \equiv \mathrm{X}_d \cup \mathcal{T}_c^+ = \mathrm{X}_{d+c}, \qquad \mathrm{X}_d, \mathrm{X}_{d+c} \subseteq \mathrm{Y}. \tag{24}$$

Definition 4.2. For a given current feature set X_d and given $\lambda \in [0,1]$, denote $T^- = \binom{d}{c} - 1$, and let Z^- be the set of candidate feature c-tuples

$$\mathrm{Z}^- = \{\mathcal{T}_{c,i} : \mathcal{T}_{c,i} \subset \mathrm{X}_d; i = 1, \ldots, \max\{1, \lfloor \lambda \cdot T^- \rceil\}\} \tag{25}$$

such that

$$\forall \mathcal{T}_c^{'}, \mathcal{T}_c^{''} \subset \mathrm{X}_d, \mathcal{T}_c^{'} \in \mathrm{Z}^-, \mathcal{T}_c^{''} \notin \mathrm{Z}^- \qquad J_F^-(\mathrm{X}_d, \mathcal{T}_c^{'}) \geq J_F^-(\mathrm{X}_d, \mathcal{T}_c^{''}) \ , \tag{26}$$

where $J_F^-(\mathrm{X}_d, \mathcal{T}_c)$ denotes the pre-filtering criterion function used to evaluate the subset obtained by removing c-tuple $\mathcal{T}_c$ ($\mathcal{T}_c \subset \mathrm{X}_d$) from X_d. Let $\mathcal{T}_c^-$ be the feature c-tuple such that

$$\mathcal{T}_c^- = \arg \max_{\mathcal{T}_c \in \mathrm{Z}^-} J_W^-(\mathrm{X}_d, \mathcal{T}_c), \tag{27}$$

where $J_W^-(\mathrm{X}_d, \mathcal{T}_c)$ denotes the main criterion function used to evaluate the subset obtained by removing c-tuple $\mathcal{T}_c$ ($\mathcal{T}_c \in \mathrm{Z}^-$) from X_d. Then we shall say that $hRMV_c(\mathrm{X}_d)$ is an operation of removing feature c-tuple $\mathcal{T}_c^-$ from the current set X_d to obtain set X_{d-c} if

$$hRMV_c(\mathrm{X}_d) \equiv \mathrm{X}_d \setminus \mathcal{T}_c^- = \mathrm{X}_{d-c}, \qquad \mathrm{X}_d, \mathrm{X}_{d-c} \subseteq \mathrm{Y}. \tag{28}$$

Note that in standard sequential FS methods $J_F^+(\cdot)$, $J_F^-(\cdot)$, $J_W^+(\cdot)$ and $J_W^-(\cdot)$ stand for

$$\begin{aligned} J_F^+(\mathrm{X}_d, \mathcal{T}_c) &= J_F(\mathrm{X}_d \cup \mathcal{T}_c) \ , \\ J_F^-(\mathrm{X}_d, \mathcal{T}_c) &= J_F(\mathrm{X}_d \setminus \mathcal{T}_c) \ , \\ J_W^+(\mathrm{X}_d, \mathcal{T}_c) &= J_W(\mathrm{X}_d \cup \mathcal{T}_c) \ , \\ J_W^-(\mathrm{X}_d, \mathcal{T}_c) &= J_W(\mathrm{X}_d \setminus \mathcal{T}_c) \ . \end{aligned} \tag{29}$$

The idea behind the proposed hybridization scheme is applicable in any sequential feature selection method (see Section 1.2.1).

When applied in sequential FS methods the described hybridization mechanism has several implications: 1. it makes possible to use wrapper based FS in considerably higher dimensional problems as well as with larger sample sizes due to reduced number of wrapper computations and consequent computational time savings, 2. it improves resistance to over-fitting when the used wrapper criterion tends to over-fit and the filter does not, and 3 for $\lambda = 0$ it reduces the number of wrapper criterion evaluations to the absolute minimum of one evaluation in each algorithm step. In this way it is possible to enable monotonic filter criteria to be used in d-optimizing setting, which would otherwise be impossible.

Table 5. Performance of hybridized FS methods with Bhattacharyya distance used as pre-filtering criterion and 5-NN performance as main criterion. *Madelon* data, 500-dim., 2 classes of 1000 and 1000 samples.

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
d-opt.	Dynamic Oscillating Search ($\Delta = 15$)										
train	$\underline{.795}$	.889	$\underline{.903}$	$\underline{.873}$	$\underline{.897}$	$\underline{.891}$	$\underline{.892}$	$\underline{.894}$	.884	.884	.886
test	$\underline{.811}$	$\underline{.865}$	$\underline{.868}$	$\underline{.825}$	$\underline{.854}$	.877	.871	$\underline{.849}$	.873	.873	.875
features	8	27	19	19	19	18	23	13	13	13	16
time	1s	6m	14m	8m	18m	18m	14m	5m	3m	3m	9m
d-par.	Oscillating Search (BIF initialized, $\Delta = 10$), subset size set in all cases to $d = 20$										
train	$\underline{.812}$	$\underline{.874}$	.887	$\underline{.891}$	$\underline{.879}$	$\underline{.902}$	$\underline{.891}$	$\underline{.899}$	$\underline{.889}$	$\underline{.891}$	.884
test	$\underline{.806}$	.859	$\underline{.869}$	.853	.855	$\underline{.864}$	.856	.853	.857	.86	.858
time	9s	6m	1m	2m	4m	7m	9m	14m	10m	10m	13m

Table 6. Performance of hybridized FS methods with Bhattacharyya distance used as pre-filtering criterion and 5-NN wrapper as main criterion. *Musk* data, 166-dim., 2 classes of 1017 and 5581 samples.

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
d-opt.	Dynamic Oscillating Search ($\Delta = 15$)										
train	$\underline{.968}$	.984	.985	.985	.985	.985	$\underline{.986}$	.985	$\underline{.986}$	.985	.985
test	$\underline{.858}$	$\underline{.869}$	$\underline{.862}$	$\underline{.872}$	$\underline{.863}$	$\underline{.866}$	$\underline{.809}$	$\underline{.870}$	$\underline{.861}$	$\underline{.853}$	.816
features	7	7	9	14	16	17	18	7	16	12	12
time	5s	2m	6m	16m	22m	25m	38m	12m	48m	29m	41m
d-par.	Oscillating Search (BIF initialized, $\Delta = 10$), subset size set in all cases to $d = 20$										
train	$\underline{.958}$	$\underline{.978}$	$\underline{.984}$	$\underline{.983}$	$\underline{.985}$	$\underline{.985}$	$\underline{.984}$	$\underline{.985}$	.986	.986	.986
test	$\underline{.872}$	$\underline{.873}$	$\underline{.864}$	$\underline{.855}$	$\underline{.858}$	$\underline{.875}$	$\underline{.868}$	$\underline{.864}$	$\underline{.853}$	$\underline{.846}$	.841
time	1m	4m	33m	11m	62m	32m	47m	70m	63m	65m	31m

Table 7. Performance of hybridized FS methods with Bhattacharyya distance used as pre-filtering criterion and 5-NN performance as main criterion. *Wdbc* data, 30-dim., 2 classes of 357 and 212 samples.

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
d-opt.	Dynamic Oscillating Search ($\Delta = 15$)										
train	$\underline{.919}$	$\underline{.919}$	$\underline{.926}$	$\underline{.926}$	.961	.961	.961	.961	.961	.961	.961
test	$\underline{.930}$	$\underline{.930}$	$\underline{.933}$	$\underline{.933}$	.944	.944	.944	.944	.944	.944	.944
features	3	2	3	3	5	5	3	3	3	3	3
time	1s	1s	1s	2s	7s	10s	11s	19s	26s	28s	26s
d-par.	Oscillating Search (BIF initialized, $\Delta = 10$), subset size set in all cases to $d = 8$										
train	.919	.919	.919	.919	.919	.919	.919	.919	.919	.919	.919
test	.933	.933	.933	.933	.933	.933	.933	.933	.933	.933	.933
time	1s	2s	2s	3s	4s	5s	6s	6s	7s	8s	8s

4.1. Experimental Results

We have conducted a series of experiments on data of various characteristics. These include: low-dimensional low sample size *speech* data from British Telecom, 15-dim., 2 classes of 212 and 55 samples, and *wdbc* data from UCI Repository [8], 30-dim., 2 classes of 357 and 212 samples, moderate-dimensional high sample size *waveform* data [8], 40-dim., first 2 classes of 1692 and 1653 samples, as well as high-dimensional, high sample size data: *madelon* 500-dim., 2 classes of 1000 samples and *musk* data, 166-dim., 2 classes of 1017 and 5581 samples, each form UCI Repository [8].

For each data set we compare FS results of the d-optimizing Dynamic Oscillating Search (DOS) and its d-parametrized counterpart, the Oscillating Search (OS). The two methods represent some of the most effective subset search tools available. For simplification we consider only single feature adding/removal steps (c-tuples with $c = 1$). For OS the target subset size d is set manually to a constant value to be comparable to the d as yielded by DOS. In both cases the experiment has been performed for various values of the hybridization coefficient λ ranging from 0 to 1. In each hybrid algorithm the following feature selection criteria have been combined: (normal) Bhattacharyya distance for pre-filtering (filter criterion) and 5-Nearest Neighbor (5-NN) 10-fold cross-validated classification rate on validation data for final feature selection (wrapper criterion). Each resulting feature subset has been eventually tested using 5-NN on independent test data (50% of each dataset).

The results are demonstrated in Tables 5 to 7. Note the following phenomena observable across all tables: 1. hybridization coefficient λ closer to 0 leads generally to lower computational time while λ closer to 1 leads to higher computational time, although there is no guarantee that lowering λ reduces search time (for counter-example see, e. g., Table 5 for $\lambda = 0.7$ or Table 6 for $\lambda = 0.4$), 2. low λ values often lead to results performing equally or better than pure wrapper results ($\lambda = 1$) on independent test data (see esp. Table 6), 3. d-optimizing DOS tends to yield higher criterion values than d-parametrized OS; in terms of the resulting performance on independent data the difference between DOS and OS shows much less notable and consistent, although DOS still shows to be better performing (compare the best achieved accuracy on independent data over all λ values in each Table), 4. it is impossible to predict the λ value for which the resulting classifier performance on independent data will be maximum (note in Table 5 $\lambda = 0.5$ for DOS and 0.2 for OS, etc.). The same holds for the maximum found criterion value (note in Table 5 $\lambda = 0.2$ for DOS and 0.5 for OS). Note that underlining emphasizes those cases where the difference from the pure wrapper case ($\lambda = 1$) has been confirmed by statistical significance t-test at significance level 0.05.

5. CONCLUSION

We have pointed out that curse of dimensionality effects can seriously hinder the outcome of feature selection process, resulting in poor performance of devised decision rules on unknown data. We have presented three different approaches to tackling this problem.

First, we have pointed out the problem of feature subset preference fragility (over-emphasized importance of negligible criterion value increase) as one of the factors that make many FS methods more prone to over-fitting. We propose an algorithmic workaround applicable with many standard FS methods. Moreover, the proposed algorithmic extension enables improved ways of standard FS algorithms' operation, e. g., taking into account feature acquisition cost. We show just one of the possible applications of the proposed mechanism on a series of examples where two sequential FS methods are modified to put more preference on smaller subsets in the course of a search. Although the main course of search is aimed at criterion maximization, smaller subsets are permitted to be eventually selected if their respective criterion value is negligibly lower than the known maximum. The examples show that this mechanism is well capable of improving classification accuracy on independent data.

Second, it has been shown that combining multiple critera by voting in FS process has the potential to improve both the generalization properties of the selected feature subsets as well as the stability of feature preferences. The actual gain is problem dependent and can not be guaranteed, although the improvement on some datasets is substantial.

The idea of combining FS criteria by voting can be applied not only in sequential selection methods but generally in any FS method where a choice is made among several candidate subsets (generated, e. g., randomly as in genetic algorithms). Additional means of improving robustness can be considered, e. g. ignoring the best and worst result among all criteria etc.

Third, we introduced the general scheme of defining hybridized versions of sequential feature selection algorithms. We show experimentally that in the particular case of combining faster but weaker *filter* FS criteria with slow but possibly more appropriate *wrapper* FS criteria it is not only possible to achieve results comparable to that of wrapper-based FS but in filter-like time, but in some cases hybridization leads to better classifier accuracy on independent test data.

All of the presented approaches have been experimentally shown to be capable of reducing the risk of over-fitting in feature selection. Their application is to be recommended especially in cases of high dimensionality and/or small sample size, where the risk of over-fitting should be of particular concern.

Remark 5.1. Related source codes can be found at `http://fst.utia.cz` as well as at `http://ro.utia.cas.cz/dem.html`.

ACKNOWLEDGEMENT

The work has been primarily supported by a grant from the Czech Ministry of Education 1M0572 DAR. It was partially supported by grant 2C06019 ZIMOLEZ and the GA ČR No. 102/08/0593. We thank the reviewers for the thorough check and valuable recommendations.

(Received July 30, 2010)

REFERENCES

[1] G. Brown: A new perspective for information theoretic feature selection. In: Proc. AISTATS '09, JMLR: W&CP *5* (2009), pp. 49–56.

[2] Ch.-Ch. Chang and Ch.-J. Lin: LIBSVM: A Library for SVM, 2001. `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[3] S. Das: Filters, wrappers and a boosting-based hybrid for feature selection. In: Proc. 18th Int. Conf. on Machine Learning (ICML '01), Morgan Kaufmann Publishers Inc. 2001, pp. 74–81.

[4] M. Dash, K. Choi, P. Scheuermann, and H. Liu: Feature selection for clustering – a filter solution. In: Proc. 2002 IEEE Int. Conf. on Data Mining (ICDM '02), Vol. 00, IEEE Comp. Soc. 2002, p. 115.

[5] P. A. Devijver and J. Kittler: Pattern Recognition: A Statistical Approach. Prentice Hall 1982.

[6] D. Dutta, R. Guha, D. Wild, and T. Chen: Ensemble feature selection: Consistent descriptor subsets for multiple qsar models. J. Chem. Inf. Model. *43* (2007), 3, pp. 989–997.

[7] C. Emmanouilidis et al.: Multiple-criteria genetic algorithms for feature selection in neuro-fuzzy modeling. In: Internat. Conf. on Neural Networks, Vol. 6, 1999, pp. 4387–4392.

[8] A. Frank and A. Asuncion: UCI Machine Learning Repository, 2010.

[9] I. A. Gheyas and L. S. Smith: Feature subset seleciton in large dimensionality domains. Pattern Recognition *43* (2010), 1, 5–13.

[10] F. W. Glover and G. A. Kochenberger, eds.: Handbook of Metaheuristics. Internat. Ser. Operat. Research & Management Science *5*, Springer 2003.

[11] S. Günter and H. Bunke: An evaluation of ensemble methods in handwritten word recog. based on feature selection. In: Proc. ICPR '04, IEEE Comp. Soc. 2004, pp. 388–392.

[12] I. Guyon and A. Elisseeff: An introduction to variable and feature selection. J. Mach. Learn. Res. *3* (2003), 1157–1182.

[13] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, eds.: Feature Extraction – Foundations and Applications. Studies in Fuzziness and Soft Comp. *207* Physica, Springer 2006.

[14] Tin Kam Ho: The random subspace method for constructing decision forests. IEEE Trans. PAMI *20* (1998), 8, 832–844.

[15] F. Hussein, R. Ward, and N. Kharma: Genetic algorithms for feature selection and weighting, a review and study. In: Proc. 6th ICDAR, Vol. 00, IEEE Comp. Soc. 2001, pp. 1240–1244.

[16] R. Jensen: Performing feature selection with ACO. Studies Comput. Intelligence *34*, Springer 2006, pp. 45–73.

[17] Special issue on variable and feature selection. J. Machine Learning Research. http://www. jmlr.org/papers/special/feature.html, 2003.

[18] A. Kalousis, J. Prados, and M. Hilario: Stability of feature selection algorithms: A study on high-dimensional spaces. Knowledge Inform. Systems *12* (2007), 1, 95–116.

[19] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas: On combining classifiers. IEEE Trans. PAMI *20* (1998), 3, 226–239.

[20] R. Kohavi and G. H. John: Wrappers for feature subset selection. Artificial Intelligence *97* (1997), 1–2, 273–324.

[21] I. Kononenko: Estimating attributes: Analysis and extensions of RELIEF. In: Proc. ECML-94, Springer 1994, pp. 171–182.

[22] L. I. Kuncheva: A stability index for feature selection. In: Proc. 25th IASTED Internat. Mul.-Conf. AIAP'07, ACTA Pr. 2007, pp. 390–395.

[23] C. Lai, M. J. T. Reinders, and L. Wessels: Random subspace method for multivariate feature selection. Pattern Recognition Lett. *27* (2006), 10, 1067–1076.

[24] H. Liu and H. Motoda: Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers 1998.

[25] H. Liu and L. Yu: Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. KDE *17* (2005), 4, 491–502.

[26] S. Nakariyakul and D. P. Casasent: Adaptive branch and bound algorithm for selecting optimal features. Pattern Recognition Lett. *28* (2007), 12, 1415–1427.

[27] S. Nakariyakul and D. P. Casasent: An improvement on floating search algorithms for feature subset selection. Pattern Recognition *42* (2009), 9, 1932–1940.

[28] J. Novovičová, P. Pudil, and J. Kittler: Divergence based feature selection for multimodal class densities. IEEE Trans. PAMI *18* (1996), 2, 218–223.

[29] P. Pudil, J. Novovičová, N. Choakjarernwanit, and J. Kittler: Feature selection based on the approximation of class densities by finite mixtures of special type. Pattern Recognition *28* (1995), 9, 1389–1398.

[30] P. Pudil, J. Novovičová, and J. Kittler: Floating search methods in feature selection. Pattern Recognition Lett. *15* (1994), 11, 1119–1125.

[31] Š. J. Raudys: Feature over-selection. In: Proc. S+SSPR, Lecture Notes in Comput. Sci. *4109*, Springer 2006, pp. 622–631.

[32] V. C. Raykar et al.: Bayesian multiple instance learning: automatic feature selection and inductive transfer. In: Proc. ICML '08, ACM 2008, pp. 808–815.

[33] J. Reunanen: A pitfall in determining the optimal feature subset size. In: Proc. 4th Internat. Workshop on Pat. Rec. in Inf. Systs (PRIS 2004), pp. 176–185.

[34] J. Reunanen: Less biased measurement of feature selection benefits. In: Stat. and Optimiz. Perspectives Workshop, SLSFS, Lecture Notes in Comput. Sci. *3940*, Springer 2006, pp. 198–208.

[35] Y. Saeys, I. Inza, and P. Larrañaga: A review of feature selection techniques in bioinformatics. Bioinformatics *23* (2007), 19, 2507–2517.

[36] A. Salappa, M. Doumpos, and C. Zopounidis: Feature selection algorithms in classification problems: An experimental evaluation. Optimiz. Methods Software *22* (2007), 1, 199–212.

[37] F. Sebastiani: Machine learning in automated text categorization. ACM Comput. Surveys *34* (2002), 1, 1–47.

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

[39] P. Somol, J. Grim, and P. Pudil: Criteria ensembles in feature selection. In: Proc. MCS, Lecture Notes in Comput. Sci. *5519*, Springer 2009, pp. 304–313.

[40] P. Somol, J. Grim, and P. Pudil: The problem of fragile feature subset preference in feature selection methods and a proposal of algorithmic workaround. In: ICPR 2010. IEEE Comp. Soc. 2010.

[41] P. Somol, J. Novovičová, and P. Pudil: Flexible-hybrid sequential floating search in statistical feature selection. In: Proc. S+SSPR, Lecture Notes in Comput. Sci. *4109*, Springer 2006, pp. 632–639.

[42] P. Somol and J. Novovičová: Evaluating the stability of feature selectors that optimize feature subset cardinality. In: Proc. S+SSPR, Lecture Notes in Comput. Sci. *5342* Springer 2008, pp. 956–966.

[43] P. Somol, J. Novovičová, J. Grim, and P. Pudil: Dynamic oscillating search algorithms for feature selection. In: ICPR 2008. IEEE Comp. Soc. 2008.

[44] P. Somol, J. Novovičová, and P. Pudil: Improving sequential feature selection methods performance by means of hybridization. In: Proc. 6th IASTED Int. Conf. on Advances in Computer Science and Engrg. ACTA Press 2010.

[45] P. Somol and P. Pudil: Oscillating search algorithms for feature selection. In: ICPR 2000, IEEE Comp. Soc. *02* (2000), 406–409.

[46] P. Somol, P. Pudil, and J. Kittler: Fast branch & bound algorithms for optimal feature selection. IEEE Trans. on PAMI *26* (2004), 7, 900–912.

[47] Y. Sun: Iterative RELIEF for feature weighting: Algorithms, theories, and applications. IEEE Trans. PAMI *29* (2007), 6, 1035–1051.

[48] M.-A. Tahir et al: Simultaneous feature selection and feature weighting using hybrid tabu search/k-nearest neighbor classifier. Patt. Recognition Lett. *28* (2007), 4, 438–446.

[49] A. W. Whitney: A direct method of nonparametric measurement selection. IEEE Trans. Comput. *20* (1971), 9, 1100–1103.

[50] Y. Yang and J. O. Pedersen: A comparative study on feature selection in text categorization. In: Proc. 14th Internat. Conf. on Machine Learning (ICML '97), Morgan Kaufmann 1997, pp. 412–420.

[51] L. Yu and H. Liu: Feature selection for high-dimensional data: A fast correlation-based filter solution. In: Proc. 20th Internat. Conf. on Machine Learning (ICML-03), Vol. *20*, Morgan Kaufmann 2003, pp. 856–863.

[52] Z. Zhu, Y. S. Ong, and M. Dash: Wrapper-filter feature selection algorithm using a memetic framework. IEEE Trans. Systems Man Cybernet., Part B *37* (2007), 1, 70.

Petr Somol, Institute of Information Theory and Automation – Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 4, 182 08 Praha 8. Czech Republic.
e-mail: somol@utia.cas.cz

Jiří Grim, Institute of Information Theory and Automation – Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 4, 182 08 Praha 8. Czech Republic.
e-mail: grim@utia.cas.cz

Jana Novovičová, Institute of Information Theory and Automation – Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 4, 182 08 Praha 8. Czech Republic.
e-mail: novovic@utia.cas.cz

Pavel Pudil, Faculty of Management, Prague University of Economics, Jarošovská 1117/II, 377 01 Jindřichův Hradec. Czech Republic.
e-mail: pudil@fm.vse.cz