Criteria Ensembles in Feature Selection

Petr Somol[1,2], Jiří Grim[1,2], and Pavel Pudil[2,1]

[1] Dept. of Pattern Recognition, Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, 182 08 Prague, Czech Republic
{somol,grim}@utia.cas.cz
http://ro.utia.cas.cz/

[2] Faculty of Management, Prague University of Economics, Czech Republic
pudil@fm.vse.cz
http://www.fm.vse.cz

Abstract. In feature selection the effect of over-fitting may lead to serious degradation of generalization ability. We introduce the concept of combining multiple feature selection criteria in feature selection methods with the aim to obtain feature subsets that generalize better. The concept is applicable with many existing feature selection methods. Here we discuss in more detail the family of sequential search methods. The concept does not specify which criteria to combine – to illustrate its feasibility we give a simple example of combining the estimated accuracy of k-nearest neighbor classifiers for various k. We perform the experiments on a number of datasets. The potential to improve is clearly seen on improved classifier performance on independent test data as well as on improved feature selection stability.

1 Introduction

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 overfitting 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 [2]. By choosing a small subset of "informative" features we try to reduce the risk of overfitting and to 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.

In most cases a natural way to choose the optimal subset of features would be to minimize the probability of classification error. As the exact evaluation of error probability is usually not viable, we have to minimize some estimates of classification error (wrapper methods) or at least some estimates of its upper bound, or even some intuitive probabilistic criteria like entropy, model-based class distances, distribution divergences, etc. (filter methods) [7]. In order to avoid biased solutions the chosen criterion has to be evaluated on an independent validation

J.A. Benediktsson, J. Kittler, and F. Roli (Eds.): MCS 2009, LNCS 5519, pp. 304–313, 2009.

set. Nevertheless, the problem of overfitting applies to FS criteria and FS algorithms as well [11] and cannot be fully avoided by means of validation. It is well known that different optimality criteria may choose different feature subsets [2]. The resulting feature subsets may differ even if one and the same criterion is applied to differently chosen training data. In this respect the "stability" of the resulting feature subsets becomes a relevant viewpoint [8] [14].

It has been shown repeatedly in literature that classification system performance may be considerably improved in some cases by means of classifier combination [6]. In multiple-classifier systems FS is often applied separately to yield different subsets for each classifier in the system [5] [4]. Another approach is to select one feature subset to be used in all co-operating classifiers [10] [3].

In contrary to such approaches we utilize the idea of combination to eventually produce one feature subset to be used with one classifier. We propose to combine FS criteria with the aim to obtain a feature subset that has better generalization properties than subsets obtained using single criteria. In the course of 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 most cases.

1.1 Notation

Let Y denote the set of all $D = |\mathrm{Y}|$ features. Further let $X_d \subset \mathrm{Y}$ denote the current subset of d features, f_i denote the i-th feature in the set of all features, $i = 1, \ldots, D$ and $J(\cdot)$ denote a FS criterion. Without loss of generality we will assume that higher $J(\cdot)$ value indicates better feature subset.

2 Decomposing Sequential Search Methods

To simplify the discussion of the criterion combination scheme 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 to/from a working subset. The respective algorithm steps can be described as follows (for the sake of simplicity we consider only non-generalized algorithms that process one feature at a time only):

Definition 1. *Let ADD() be the operation of adding feature f^+ to the working set X_d to obtain X_{d+1}:*

$$X_{d+1} = X_d \cup \{f^+\} = ADD(X_d), \qquad X_d, X_{d+1} \subset Y \tag{1}$$

where

$$f^+ = \arg \max_{f \in Y \setminus X_d} \mathcal{J}^+(X_d, f) \tag{2}$$

with $\mathcal{J}^+(X_d, f)$ denoting the criterion function used to evaluate the subset obtained by adding f, where $f \in Y \setminus X_d$, to X_d.

Definition 2. *Let REMOVE() be the operation of removing feature f^- from the working set X_d to obtain set X_{d-1}:*

$$X_{d-1} = X_d \setminus \{f^-\} = REMOVE(X_d), \qquad X_d, X_{d-1} \subset Y \tag{3}$$

where

$$f^- = \arg\max_{f \in X_d} \mathcal{J}^-(X_d, f) \tag{4}$$

with $\mathcal{J}^-(X_d, f)$ denoting the criterion function used to evaluate the subset obtained by removing f, where $f \in X_d$, from X_d.

In standard sequential FS methods the impact of feature adding (resp. removal) in one algorithm step is evaluated simply as follows:

$$\mathcal{J}^+(X_d, f) = J(X_d \cup \{f\}), \qquad \mathcal{J}^-(X_d, f) = J(X_d \setminus \{f\}) \,, \tag{5}$$

where $J(\cdot)$ is either a filter- or wrapper-based criterion [7] to be evaluated on the subspace defined by the tested feature subset.

2.1 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

$$\begin{aligned} X_{d+2} &= ADD(X_{d+1}) = ADD(ADD(X_d)) = ADD^2(X_d) \,, \\ X_{d-2} &= REMOVE(REMOVE(X_d)) = REMOVE^2(X_d) \,, \end{aligned} \tag{6}$$

and more generally

$$X_{d+\delta} = ADD^\delta(X_d), \qquad X_{d-\delta} = REMOVE^\delta(X_d) \tag{7}$$

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

SFS (*Sequential Forward Selection* [16] yielding a subset of t features):

1. $X_t = ADD^t(\emptyset)$.

SFFS (*Sequential Forward Floating Selection* [9] yielding a subset of t features, with optional search-restricting parameter $\Delta \in [0, D-t]$):

1. Start with $X_0 = \emptyset$, $d = 0$.
2. $X_{d+1} = ADD(X_d)$, $d = d + 1$.
3. Repeat $X_{d-1} = REMOVE(X_d)$, $d = d - 1$ as long as it improves solutions already known for the lower d.
4. If $d < t + \Delta$ go to 2.

OS (*Oscillating Search* [13] yielding a subset of t features, with optional search-restricting parameter $\Delta \geq 1$):

1. Start with initial set X_t of t features. Set cycle depth to $\delta = 1$.
2. Let $X_t^{\downarrow} = ADD^{\delta}(REMOVE^{\delta}(X_t))$.
3. If $X_t^{\downarrow}$ better than X_t, let $X_t = X_t^{\downarrow}$, let $\delta = 1$ and go to 2.
4. Let $X_t^{\uparrow} = REMOVE^{\delta}(ADD^{\delta}(X_t))$.
5. If $X_t^{\uparrow}$ better than X_t, let $X_t = X_t^{\uparrow}$, let $\delta = 1$ and go to 2.
6. If $\delta < \Delta$ let $\delta = \delta + 1$ and go to 2.

DOS (*Dynamic Oscillating Search* [15] yielding a subset of optimized size p, with optional search-restricting parameter $\Delta \geq 1$):

1. Start with $X_p = ADD(ADD(\emptyset))$, p=2. Set cycle depth to $\delta = 1$.
2. Compute $ADD^{\delta}(REMOVE^{\delta}(X_t))$; if any intermediate subset X_i, $i \in [p - \delta, p]$ is found better than X_p, let it become the new X_p with $p = i$, let $\delta = 1$ and restart step 2.
3. Compute $REMOVE^{\delta}(ADD^{\delta}(X_t))$; if any intermediate subset X_j, $j \in [p, p + \delta]$ 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 $\delta < \Delta$ let $\delta = \delta + 1$ and go to 2.

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

3 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. 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 Definitions 1 and 2. We will consider ensembles of arbitrary feature selection criteria $J^{(k)}$, $k = 1, \ldots, K$. In Section 4 concrete examples will be given for $J^{(k)}$, $k = 1, \ldots, 4$ standing for the accuracy of $(2k - 1)$-Nearest Neighbor classifier.

3.1 Simplest Criterion Combination

First let us discuss the simplest combination option. To realize a simple criterion ensemble consisting of criteria $J^{(k)}$, $k = 1, \ldots, K$, consider modifying Definitions 1 and 2 as follows

$$\mathcal{J}^{+}_{\text{avg}}(X_d, f) = \frac{1}{K} \sum_{k=1}^{K} J^{(k)}(X_d \cup \{f\}) \tag{8}$$

$$\mathcal{J}^{-}_{\text{avg}}(X_d, f) = \frac{1}{K} \sum_{k=1}^{K} J^{(k)}(X_d \setminus \{f\}) \,,$$

or, to put more preference on features that generally "fail the least" with respect to all of the considered criteria, modify Definitions 1 and 2 as follows

$$\mathcal{J}^{+}_{\min}(X_d, f) = \min_{k=1,\ldots,K} J^{(k)}(X_d \cup \{f\}) \tag{9}$$
$$\mathcal{J}^{-}_{\min}(X_d, f) = \min_{k=1,\ldots,K} J^{(k)}(X_d \setminus \{f\}) \,.$$

Remark: Maximizing would meaninglessly emphasize feature over-selection.

Clearly, none of the approaches (8) and (9) is applicable unless all $J^{(k)}$, $k = 1, \ldots, K$ yield equally bounded values. This should apparently be no problem with wrappers, where the estimated classification accuracy can be easily normalized to $[0, 1]$. However, both (8) and (9) produce feature preferences that are hard to interpret, especially if the used criteria $J^{(k)}$, $k = 1, \ldots, K$ tend to yield values of differing size (albeit equally bounded). Accordingly, no consistent advantage over single-criterion FS has been observed throughout the numerous experiments we have performed. Therefore, the simple criterion value combination as described in this Section *is to be considered unsatisfactory* and unable to bring reliable improvement over the traditional single-criterion FS methods.

3.2 Multiple Criterion Voting

A better way to realize the idea of criterion ensemble is to implement a form of voting. The intention is to reveal stability in feature 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 ordering preferences instead of directly combining criterion values.

In the following we define $\mathcal{J}^{+}_{order}$ as replacement of $\mathcal{J}^{+}$ in Definition 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 $J^{(k)}(X_d \cup \{f_i\})$ for $i = 1, \ldots, D - d$, where $f_i \in Y \setminus X_d$. Next, order these values descending with possible ties resolved arbitrarily at this stage and encode the ordering using indexes $i_j, j = 1, \ldots, D - d, i_j \in [1, D - d]$ where $i_m \neq i_n$ for $m \neq n$:

$$J^{(k)}(X_d \cup \{f_{i_1}\}) \geq J^{(k)}(X_d \cup \{f_{i_2}\}) \geq \cdots \geq J^{(k)}(X_d \cup \{f_{i_{D-d}}\}) \,. \tag{10}$$

Next, express feature preferences using coefficient $\alpha^{(k)}_j$, $j = 1, \ldots, D - d$, defined to take into account possible feature preference ties as follows:

$$\alpha^{(k)}_{i_1} = 1 \tag{11}$$
$$\alpha^{(k)}_{i_j} = \begin{cases} \alpha^{(k)}_{i_{j-1}} & \text{if } J^{(k)}(X_d \cup \{f_{i_{(j-1)}}\}) = J^{(k)}(X_d \cup \{f_{i_j}\}) \\ \alpha^{(k)}_{i_{j-1}} + 1 & \text{if } J^{(k)}(X_d \cup \{f_{i_{(j-1)}}\}) > J^{(k)}(X_d \cup \{f_{i_j}\}) \end{cases} \quad \text{for } j \geq 2 \,.$$

Now, having collected the values $\alpha^{(k)}_j$ 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 by defining:

$$\mathcal{J}^{+}_{order}(X_d, f_i) = -\frac{1}{K} \sum_{k=1}^{K} \alpha^{(k)}_i \,. \tag{12}$$

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

3.3 Multiple Criterion Weighted Voting

Suppose we introduce an additional restriction to 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 methods 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 preferences of criterion k. In the following we define $\mathcal{J}^{+}_{weigh}$ as replacement of $\mathcal{J}^{+}$ in Def. 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 $J^{(k)}(X_d \cup \{f_i\})$ for fixed k and $i = 1, \ldots, D - d$, where $f_i \in Y \setminus X_d$. Next, order the values descending with possible ties resolved arbitrarily at this stage and encode the ordering using indexes $i_j, j = 1, \ldots, D - d$ in the same way as shown in (10). Now, express feature preferences using coefficient $\beta^{(k)}_j$, $j = 1, \ldots, D - d$ defined to take into account the differences between the impact the various features from $Y \setminus X_d$ have on the criterion value:

$$\beta^{(k)}_{i_j} = J^{(k)}(X_d \cup \{f_{i_1}\}) - J^{(k)}(X_d \cup \{f_{i_j}\}) \text{ for } j = 1, \ldots, D - d \,. \tag{13}$$

Now, having collected the values $\beta^{(k)}_j$ 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 by defining:

$$\mathcal{J}^{+}_{weigh}(X_d, f_i) = -\frac{1}{K} \sum_{k=1}^{K} \beta^{(k)}_i \,. \tag{14}$$

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

3.4 Resolving Voting Ties

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

$$\mathcal{J}^{+}_{order}(X_d, f_i) = \mathcal{J}^{+}_{order}(X_d, f_j) \quad \text{for} \quad i \neq j \,. \tag{15}$$

(The same can happen for $\mathcal{J}^{-}_{order}$, $\mathcal{J}^{+}_{weigh}$, $\mathcal{J}^{-}_{weigh}$.) 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 f_i, $i = 1, \ldots, D$ the information about all values (12) evaluated so far. In case of $\mathcal{J}^{+}$ ties the feature with higher average over previous values (12) is preferred. (Tie resolution for J^{-}_{order}, J^{+}_{weigh}, J^{-}_{weigh} is analogous.)

4 Experimental Results

We performed a series of FS experiments on various data-sets from UCI repository [1] 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 problems any improvement of generalization properties plays crucial role. To put the robustness of the proposed criterion voting schemes on test we used in all experiments the *Dynamic Oscillating Search* algorithm [15] as one of the strongest available subset optimizers, with high risk of over-fitting.

To illustrate the concept we have resorted in all experiments to combining classification accuracy of four simple wrappers – *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%

Table 1. 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. sample size	FS Wrapper(s)	Classsif. rate Mean	Classsif. rate S.Dv.	Subset size d Mean	Subset size d S.Dv.	FS Stability C	FS Stability CW	FS Stability CW_{rel}	FS Stability ATI (GK)	FS time h:m:s
derm	36	6	1.657	3-NN	.970	*.023*	9.6	*0.917*	.481	.664	.597	.510	00:03:24
				1,3,5,7-NN	**.978**	*.027*	10.7	*1.676*	.406	.636	.534	.486	00:15:50
hous	14	5	7.229	3-NN	.707	*.088*	4.9	*1.513*	.308	.617	.456	.478	00:01:19
				1,3,5,7-NN	.689	*.101*	5.4	*1.744*	**.389**	**.650**	**.497**	**.509**	00:04:48
iono	34	2	5.162	3-NN	.871	*.078*	5.6	*1.500*	.200	.349	.303	.216	00:02:10
				1,3,5,7-NN	**.882**	*.066*	4.7	*1.269*	**.262**	**.454**	**.441**	**.325**	00:06:09
mammo	65	2	0.662	3-NN	.821	*.124*	4.2	*1.833*	.248	.476	.497	.343	00:00:30
				1,3,5,7-NN	**.846**	*.153*	3	*1.483*	**.306**	**.519**	**.519**	**.420**	00:01:23
opt38	64	2	8.773	3-NN	.987	*.012*	9	*1.414*	.192	.449	.412	.297	01:34:14
				1,3,5,7-NN	.987	*.012*	9.5	*1.360*	**.219**	**.512**	**.490**	**.362**	06:22:00
sati	36	6	20.532	3-NN	.854	*.031*	14.2	*3.156*	.367	.557	.347	.392	32:59:47
				1,3,5,7-NN	**.856**	*.037*	14.5	*3.801*	**.392**	**.567**	**.357**	**.399**	116:26:
segm	19	7	17.368	3-NN	.953	*.026*	4.7	*1.735*	.324	.648	.610	.550	00:35:13
				1,3,5,7-NN	**.959**	*.019*	4.6	*2.245*	.282	**.652**	**.625**	**.601**	02:02:40
sonar	60	2	1.733	3-NN	.651	*.173*	12.8	*4.895*	.244	.411	.327	.260	00:07:15
				1,3,5,7-NN	**.676**	*.130*	8.8	*4.020*	.185	.389	**.350**	.260	00:16:02
specf	44	2	3.034	3-NN	.719	*.081*	9.5	*4.522*	.160	.281	.174	.157	00:03:56
				1,3,5,7-NN	**.780**	*.111*	9.8	*3.092*	**.210**	**.358**	**.255**	**.237**	00:15:36
wave	40	3	41.667	3-NN	.814	*.014*	17.2	*2.561*	.486	.792	.680	.657	62:36:30
				1,3,5,7-NN	**.817**	*.011*	16.4	*1.356*	.477	**.826**	**.753**	**.709**	70:27:36
wdbc	30	2	9.483	3-NN	.965	*.023*	10.3	*1.676*	.329	.507	.327	.345	00:12:18
				1,3,5,7-NN	**.967**	*.020*	10.1	*3.176*	**.338**	**.530**	**.360**	**.375**	00:41:07
wine	13	3	4.564	3-NN	.966	*.039*	5.9	*0.831*	.544	.731	.568	.594	00:00:15
				1,3,5,7-NN	.960	*.037*	6	*1.000*	**.556**	**.748**	**.575**	**.606**	00:00:54
wpbc	31	2	3.194	3-NN	.727	*.068*	9.1	*3.048*	.226	.347	.168	.211	00:01:53
				1,3,5,7-NN	.727	*.056*	7.2	*2.600*	.197	.312	**.189**	.188	00:04:41
xpxi	57	7	4.313	3-NN	.895	*.067*	10.8	*1.939*	.434	.648	.618	.489	05:07:06
				1,3,5,7-NN	.894	*.069*	11.5	*3.233*	.421	**.657**	**.630**	**.495**	21:19:43

Table 2. 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. sample size	FS Wrapper(s)	Classsif. rate Mean	Classsif. rate *S.Dv.*	Subset size d Mean	Subset size d *S.Dv.*	FS Stability C	FS Stability CW	FS Stability CW ^{rel}	FS Stability ATI (GK)	FS time h:m:s
derm	36	6	1.657	3-NN	.970	*.023*	9.6	*0.917*	.481	.664	.597	.510	00:03:24
				1,3,5,7-NN	**.978**	*.017*	10.3	*1.552*	**.491**	**.721**	**.658**	**.573**	00:17:42
hous	14	5	7.229	3-NN	.707	*.088*	4.9	*1.513*	.308	.617	.456	.478	00:01:19
				1,3,5,7-NN	**.716**	*.099*	5.6	*2.29*	**.455**	**.639**	**.459**	**.495**	00:03:33
iono	34	2	5.162	3-NN	.871	*.078*	5.6	*1.500*	.200	.349	.303	.216	00:02:10
				1,3,5,7-NN	**.897**	*.059*	4.9	*1.758*	**.278**	**.426**	**.393**	**.345**	00:07:40
mammo	65	2	0.662	3-NN	.821	*.124*	4.2	*1.833*	.248	.476	.497	.343	00:00:30
				1,3,5,7-NN	.813	*.153*	2.6.	*1.428*	.210	**.487**	**.542**	**.390**	00:00:43
opt38	64	2	8.773	3-NN	.987	*.012*	9	*1.414*	.192	.449	.412	.297	01:34:14
				1,3,5,7-NN	**.988**	*.011*	8.6	*1.020*	**.304**	**.576**	**.569**	**.423**	07:39:33
sati	36	6	20.532	3-NN	.854	*.031*	14.2	*3.156*	.367	.557	.347	.392	32:59:47
				1,3,5,7-NN	**.856**	*.038*	13.8	*2.182*	**.400**	**.618**	**.448**	**.456**	99:30:44
segm	19	7	17.368	3-NN	.953	*.026*	4.7	*1.735*	.324	.648	.610	.550	00:35:13
				1,3,5,7-NN	**.959**	*.019*	4.6	*2.245*	**.354**	**.667**	**.644**	**.610**	02:26:29
sonar	60	2	1.733	3-NN	.651	*.173*	12.8	*4.895*	.244	.411	.327	.260	00:07:15
				1,3,5,7-NN	.614	*.131*	10.1	*3.015*	.192	.361	.301	.224	00:20:32
specf	44	2	3.034	3-NN	.719	*.081*	9.5	*4.522*	.160	.281	.174	.157	00:03:56
				1,3,5,7-NN	**.787**	*.121*	9.1	*3.590*	**.205**	**.369**	**.285**	**.229**	00:17:54
wave	40	3	41.667	3-NN	.814	*.014*	17.2	*2.561*	.486	.792	.680	.657	62:36:30
				1,3,5,7-NN	.814	*.016*	16.9	*1.700*	**.560**	**.822**	**.727**	**.700**	287:06:
wdbc	30	2	9.483	3-NN	.965	*.023*	10.3	*1.676*	.329	.507	.327	.345	00:12:18
				1,3,5,7-NN	**.967**	*.020*	10.3	*4.267*	**.347**	**.524**	**.352**	**.346**	00:55:08
wine	13	3	4.564	3-NN	.966	*.039*	5.9	*0.831*	.544	.731	.568	.594	00:00:15
				1,3,5,7-NN	.960	*.037*	6.6	*1.200*	**.606**	**.741**	.567	**.606**	00:00:28
wpbc	31	2	3.194	3-NN	.727	*.068*	9.1	*3.048*	.226	.347	.168	.211	00:01:53
				1,3,5,7-NN	.686	*.126*	6.9	*2.508*	.196	.322	**.211**	.192	00:04:24
xpxi	57	7	4.313	3-NN	.895	*.067*	10.8	*1.939*	.434	.648	.618	.489	05:07:06
				1,3,5,7-NN	.895	*.071*	11	*2.683*	**.444**	.638	.595	.475	38:35:53

testing part. FS was done on the training part. Because we used *wrapper* setup, each criterion evaluation involved training and testing classifier(s). To utilize the training data better, it was processed by means of "inner" 10-fold cross-validation, i.e., was repeatedly split to 90% part used for classifier training and 10% 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 resulting classification accuracy, averaged over "outer" data splits, is reported in Tables 1 and 2.

In both Tables 1 and 2 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 C, CW, CW_{rel} and ATI (a.k.a. GK), all yielding values from $[0, 1]$, which reflect various aspects of the stability problem as described in [14]. 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 1 illustrates the impact of multiple criterion voting (12) as described in Section 3.2. Table 2 illustrates the impact of multiple criterion weighted voting (14) as described in Section 3.3. Improvement is emphasized in bold. The results presented in both Tables 1 and 2 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). The positive effect of either (12) or (14) is not present in all cases (in some cases the performance degraded) 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 better choice. Moreover, care should be taken when applying either of the two, as the criterion ensemble effect may be even counterproductive as was the case of *house* dataset in Table 1 and *sonar* and *wpbc* datasets in Table 2.

5 Concluding Remarks

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.

Acknowledgements. The work has been supported by GAČR grant 102/08/0593 and CR MŠMT grants 2C06019 ZIMOLEZ and 1M0572 DAR.

References

1. Asuncion, A., Newman, D.: UCI machine learning repository (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
2. Devijver, P.A., Kittler, J.: Pattern Recognition: A Statistical Approach. Prentice-Hall International, London (1982)
3. Dutta, D., Guha, R., Wild, D., Chen, T.: Ensemble Feature Selection: Consistent Descriptor Subsets for Multiple QSAR Models. J. Chem. Inf. Model. 47(3), 989–997 (2007)

4. Emmanouilidis, C., Hunter, A., MacIntyre, J., Cox, C.: Multiple-criteria genetic algorithms for feature selection inneuro-fuzzy modeling. In: Proc. Int. Joint Conf. on Neural Networks, vol. (6), pp. 4387–4392 (1999)
5. Günter, S., Bunke, H.: Feature selection algorithms for the generation of multiple classifier systems and their application to handwritten word recognition. Pattern Recogn. Lett. 25(11), 1323–1336 (2004)
6. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On Combining Classifiers. IEEE Trans. Pattern Anal. Mach. Intell. 20(3), 226–239 (1998)
7. Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial Intelligence 97, 273–324 (1997)
8. Kuncheva, L.I.: A stability index for feature selection. In: Proc. 25th IASTED Int. Multi-Conf. Artificial Intelligence and Applications, pp. 421–427 (2007)
9. Pudil, P., Novovičová, J., Kittler, J.: Floating search methods in feature selection. Pattern Recognition Letters 15, 1119–1125 (1994)
10. Raykar, V.C., Krishnapuram, B., Bi, J., Dundar, M., Rao, R.B.: Bayesian multiple instance learning: automatic feature selection and inductive transfer. In: Proc. 25th Int. Conf. on Machine Learning, pp. 808–815 (2008)
11. Raudys, S.: Feature over-selection. In: Yeung, D.-Y., Kwok, J.T., Fred, A., Roli, F., de Ridder, D. (eds.) SSPR 2006 and SPR 2006. LNCS, vol. 4109, pp. 622–631. Springer, Heidelberg (2006)
12. Saeys, Y., Abeel, T., de Peer, Y.V.: Towards robust feature selection techniques. In: Proceedings of Benelearn, pp. 45–46 (2008)
13. Somol, P., Pudil, P.: Oscillating search algorithms for featute selection. In: Proc. 15th IAPR Int. Conference on Pattern Recognition, pp. 406–409 (2000)
14. Somol, P., Novovičová, J.: Evaluating the stability of feature selectors that optimize feature subset cardinality. In: Proc. SSPR/SPR. LNCS, vol. 5342, pp. 956–966. Springer, Heidelberg (2008)
15. Somol, P., Novovičová, J., Pudil, P., Grim, J.: Dynamic oscillating search algorithm for feature selection. In: Proc. 19th IAPR Int. Conf. on Pattern Recognition. IEEE Computer Society Press, Tampa (2008) file: WeAT9.15.pdf
16. Whitney, A.W.: A direct method of nonparametric measurement selection. IEEE Trans. Comput. 20(9), 1100–1103 (1971)