Flexible-Hybrid Sequential Floating Search in Statistical Feature Selection

Petr Somol[1,2], Jana Novovičová[1,2], and Pavel Pudil[1,2]

[1] Dept. of Pattern Recognition, Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, 182 08 Prague, Czech Republic
{somol,novovic}@utia.cas.cz
http://www.utia.cas.cz/user_data/PR_dept
[2] Faculty of Management, Prague University of Economics, Czech Republic
pudil@fm.vse.cz
http://www.fm.vse.cz

Abstract. Among recent topics studied in context of feature selection the hybrid algorithms seem to receive particular attention. In this paper we propose a new hybrid algorithm, the flexible hybrid floating sequential search algorithm, that combines both the filter and wrapper search principles. The main benefit of the proposed algorithm is its ability to deal flexibly with the quality-of-result versus computational time trade-off and to enable wrapper based feature selection in problems of higher dimensionality than before. We show that it is possible to trade significant reduction of search time for negligible decrease of the classification accuracy. Experimental results are reported on two data sets, WAVEFORM data from the UCI repository and SPEECH data from British Telecom.

1 Introduction

Feature selection, as a pre-processing step to machine learning and pattern recognition applications, has been effective in reducing dimensionality. It is sometimes the case that such tasks as classification or approximation of the data represented by so called feature vectors, can be carried out in the reduced space more accurately than in the original space. Liu and Yu [1] provide a comprehensive overview of various aspects of feature selection. Their paper surveys existing feature selection algorithms for classification and clustering, evaluation criteria and data mining tasks and outlines some trends in research and development of feature selection.

Many existing feature selection algorithms designed with different evaluation criteria can be categorized into *Filter* [2], [3] *Wrapper* [4] and *Hybrid* [5], [6]. Filter methods rely on general characteristics of the training data to select some features independently of the subsequent learning algorithm. Therefore they do not inherit any bias of a learning algorithm. The wrapper methods require one predetermined learning algorithm in feature selection and use its performance to

D.-Y. Yeung et al. (Eds.): SSPR&SPR 2006, LNCS 4109, pp. 632–639, 2006.

evaluate and determine which features are selected. These methods tend to give superior performance as they find features better suited to the predetermined learning algorithm, but they also tend to be more computationally expensive. When the number of features becomes very large, the filter methods are usually to be chosen due to computational efficiency. To combine the advantages of both methods, algorithms in a hybrid approach have recently been proposed to deal with high dimensional data.

In this paper we introduce a flexible hybrid version of the floating search, the *hybrid sequential forward floating selection* (hSFFS) as well as its backward counterpart (hSBFS) that cross the boundary between filters and wrappers. We show that it is possible to trade significant reduction of search time for negligible decrease of the classification accuracy.

2 Motivation for Hybrid Algorithms

Filter methods for feature selection are general preprocessing algorithms that do not rely on any knowledge of the learning algorithm to be used. They are distinguished by specific evaluation criteria including distance, information, dependency. Since the filter methods apply independent evaluation criteria without involving any learning algorithm they are computationally efficient. Wrapper methods require a predetermined learning algorithm instead of an independent criterion for subset evaluation. They search through the space of feature subsets using a learning algorithm, calculate the estimated accuracy of the learning algorithm for each feature before it can be added to or removed from the feature subset. It means, that learning algorithms are used to control the selection of feature subsets which are consequently better suited to the predetermined learning algorithm. Due to the necessity to train and evaluate the learning algorithm within the feature selection process, the wrapper methods are more computationally expensive than the filter methods.

The main advantage of filter methods is their speed and ability to scale to large data sets. A good argument for wrapper methods is that they tend to give superior performance. Because of the success of the *sequential floating search* methods of filter type introduced by Pudil et al. [7] on many datasets and our focus on real-world datasets with potentially large number of features and small training sets, we have developed a *hybrid floating selection* algorithm that crosses the boundary between filter and wrapper methods and emphasizes some of the advantages of wrapper methods.

3 Hybrid Floating Sequential Search

Floating search methods [7], [8], sequential forward floating selection (SFFS) and sequential backward floating selection (SBFS), are now considered to be standard feature selection tools, providing good performance and close-to-optimum or optimum results in most tasks [9], [10]. In the following we will focus on the

sequential *forward* floating selection because it has proven appropriate for most real-world datasets. The definition of the backward algorithm is analogous.

Starting from empty feature set, the SFFS procedure consists of applying after each forward (feature adding) step a number of backward (feature removing) steps as long as the resulting subsets are better than previously evaluated ones at that level. Consequently, there are no backward steps at all if the performance cannot be improved. The algorithm allows a 'self-controlled backtracking' so it can eventually find good solutions by adjusting the trade-off between forward and backward steps dynamically. It is possible to say that, in a certain way, it computes only what it needs without any parameter setting. In this way it overcomes effectively the so-called *nesting problem* inherent to older methods [11].

The same scheme can be used both in filter and wrapper context, as the floating algorithms put no restrictions on the behavior of criterion functions (unlike, e.g., Branch & Bound, which requires monotonic criteria). Here we introduce a flexible hybrid version of the floating search, hybrid sequential forward floating selection (hSFFS) that crosses the boundary between filters and wrappers. We show, that only a fraction of full wrapper computational time is sufficient to obtain results not too different from the full wrapper ones. This is accomplished by taking use of filter criteria to avoid less promising subsets in wrapper computation. The proportion of subsets to be passed to wrapper-based evaluation can be specified by the user. In this way one can decide the trade-off between the length of computation and criterion maximization effectiveness.

3.1 Formal Description of hSFFS

For the purpose of formal hSFFS description we use the following notion and abbreviations: Let the number of all features be D and the full feature set be $X_D = \{x_i, i = 1, \ldots, D\}$. Due to the hybrid nature of the algorithm to be defined we will distinguish two criterion functions. $J_F(.)$ denotes the faster but possibly less appropriate *filter* criterion, $J_W(.)$ denotes the slower *wrapper* criterion. The *hybridization coefficient*, defining the proportion of feature subset evaluations to be verified by wrapper means, is denoted by $\lambda \in \langle 0, 1\rangle$. Here $\lfloor\cdot\rfloor$ denotes value rounding. Let SFS, SBS denote sequential forward selection and sequential backward selection [11], respectively.

It is required that at each stage k all the so-far best subsets X_i and corresponding criterion values $J_i = J(X_i)$ are known for $i = 1, \ldots, \tilde{k}$ with $\tilde{k}$ denoting the largest subset size tested so-far ($k < \tilde{k}$ while backtracking).

Hybrid SFFS Algorithm

Initialization: The algorithm is initialized by setting $k = 0$ and $X_0 = \emptyset$. Then, Step 1 is called twice to obtain feature sets X_1 and X_2; to conclude the initialization let $J_1 = J_W(X_1), J_2 = J_W(X_2)$ and $k = 2$.

STEP 1: (*Adding)* By analogy to the SFS method, select from the set of available features, $X_D \setminus X_k$ the best feature with respect to the set X_k, say x^+, and add it to the current set X_k to form new feature set X_{k+1}; to achieve this,

first pre-select c_k^+ most promising candidate features by maximizing $J_F(\cdot)$, then decide according to the best $J_W(\cdot)$ value, i.e.:

$$c_k^+ = \max\{1, \lfloor \lambda(D-k) \rfloor\} \tag{1}$$

$$C_k^+ = \{x_{i_t}, t = 1, \ldots, c_k^+ : J_F(X_k \cup \{x_{i_t}\}) \geq J_F(X_k \cup \{x_j\})\ \forall j \neq i_t\} \tag{2}$$

$$x^+ = \arg\max_{x \in C_k^+} J_W(X_k \cup \{x\}), \quad X_{k+1} = X_k \cup \{x^+\}. \tag{3}$$

STEP 2: (*Inferior search path cancellation)* If J_{k+1} is known from before and $J(X_{k+1}) < J_{k+1}$, set $k = k+1$ and go to Step 1.

STEP 3: (*Conditional removal)* By analogy to the SBS method find the worst feature in the set X_{k+1}, say x^-; to achieve this, first pre-select c_k^- most promising candidate features by maximizing $J_F(\cdot)$, then decide according to the best $J_W(\cdot)$ value, i.e.:

$$c_k^- = \max\{1, \lfloor \lambda k \rfloor\} \tag{4}$$

$$C_k^- = \{x_{i_t}, t = 1, \ldots, c_k^- : J_F(X_k \setminus \{x_{i_t}\}) \geq J_F(X_k \setminus \{x_j\})\ \forall j \neq i_t\} \tag{5}$$

$$x^- = \arg\max_{x \in C_k^-} J_W(X_{k+1} \setminus \{x\}). \tag{6}$$

If $J_W(X_{k+1} \setminus \{x^-\}) = J_k$, i.e., no better solution has been found, set $J_{k+1} = J(X_{k+1})$, $k = k+1$ and go to Step 1; otherwise remove this feature from the set X_{k+1} to form a new feature set $X_k^{'}$, i.e.

$$X_k^{'} = X_{k+1} \setminus \{x^-\}. \tag{7}$$

Note that now $J(X_k^{'}) > J(X_k) = J_k$. If $k = 2$, then set $X_k = X_k^{'}$ and $J_k = J(X_k^{'})$ and go to Step 1, otherwise set $k = k\ -1$ and repeat Step 3.

Remark 1: Definitions (1) and (4) ensure that for any $\lambda \in \langle 0, 1\rangle$ at least one evaluation of $J_W(\cdot)$ is done in each algorithm step for each tested subset size.

Remark 2: Algorithm Step 2 can be considered optional. It is defined to prevent possible criterion decrease that may occur when the algorithm returns to higher dimensionality after backtracking. Keeping intermediate criterion values as high as possible is certainly desirable, yet as such cannot guarantee a better result.

3.2 Simplified Flowchart of the hSFFS

A simplified flowchart of the hSFFS algorithm is given in Fig. 1. The alternative terminating condition $k = d + \delta$ in the flowchart allows premature termination of the search process, should there be no reason to evaluate cardinalities greater than d. In such a case it is good to let the algorithm reach a little higher dimensionality $(d+\delta)$ to allow possible find of a better solution for d by means of backtracking. The value of δ can be selected arbitrarily, or estimated heuristically, e.g., as the longest backtracking sequence performed so-far. Nevertheless, letting the algorithm finish (reach dimensionality D) is to be recommended. The fact that floating search finds solutions for all cardinalities in one run is one of its key advantages.

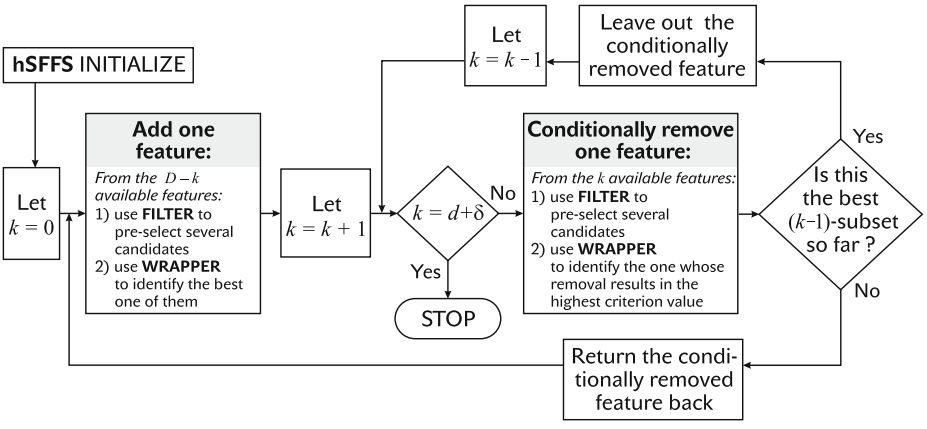

Fig. 1. Simplified diagram of the *hybrid* SFFS

4 Experiments

4.1 Datasets

The performance of the proposed algorithm is illustrated on two datasets. We used WAVEFORM data (40 features, 1692 samples from class 1 and 1653 samples from class 2) obtained via the UCI repository [12] and SPEECH data originating from British Telecom (15 features, 682 word "yes" and 736 word "no" samples), obtained from the Centre for Vision, Speech, and Signal Processing of the University of Surrey, UK.

4.2 Feature Subset Selection Criteria

We suppose, that the class-conditional densities are multivariate Gaussian, but the parameters of the densities (i.e. mean vectors and covariance matrices) are unknown and are replaced by their maximum likelihood estimates.

In the case of the filter model we used estimation of Bhattacharyya distance as the independent criterion $J_F(\cdot)$. A dependent criterion $J_W(\cdot)$ used in the wrapper model is the classification rate of the Bayes Gaussian plug-in classifier. All classification rates have been verified by a 25-fold cross-validation.

4.3 Experimental Results

For each dataset the results are presented in two graphs. The first graph (Figures 2 and 3) shows the Gaussian classifier correct classification rate on best feature subsets selected by the *hybrid* SFFS for different values of the hybridization coefficient λ as well as results of the filter SFFS and the wrapper SFFS. The second graph (Figure 4) shows the times of complete hSFFS(λ) runs for each λ. Note that floating search yields all subsets in one run, thus the graph of time contains just a single line.

It can be observed that especially for lower subset sizes the increase of λ quickly improves the classification rate. The improvement of the classification rate does not depend linearly on increased computational time. For the values of λ less than roughly 0.5 the classification rate tends to increase considerably faster than the time (an exception being, e.g., the 11 features case in Fig. 2). This is quite important. It suggests that investing some additional time into hybrid search with $\lambda \leq 0.5$ brings relatively more benefit than investing all the time needed for full wrapper based feature selection. The results for $\lambda \approx 0.5$ tend to be closer to those of wrappers than those of filters. This positive effect can be understood as an illustration of the ability of Bhattacharyya distance to pre-select reasonable feature candidates for further evaluation in the Gaussian wrapper. However, it also shows the limits of this Bhattacharyya ability. A hypothetically perfect filter criterion would cause the hSFFS yield for each λ the same best solution. The lack of such perfect criteria is the reason for using wrapper based search.

Remark: This is not to say that the time complexity of the proposed hybrid search is negligible. Obviously, it is to be expected considerably slower than the time complexity of filter search, yet only a fraction of the time complexity of wrapper search.

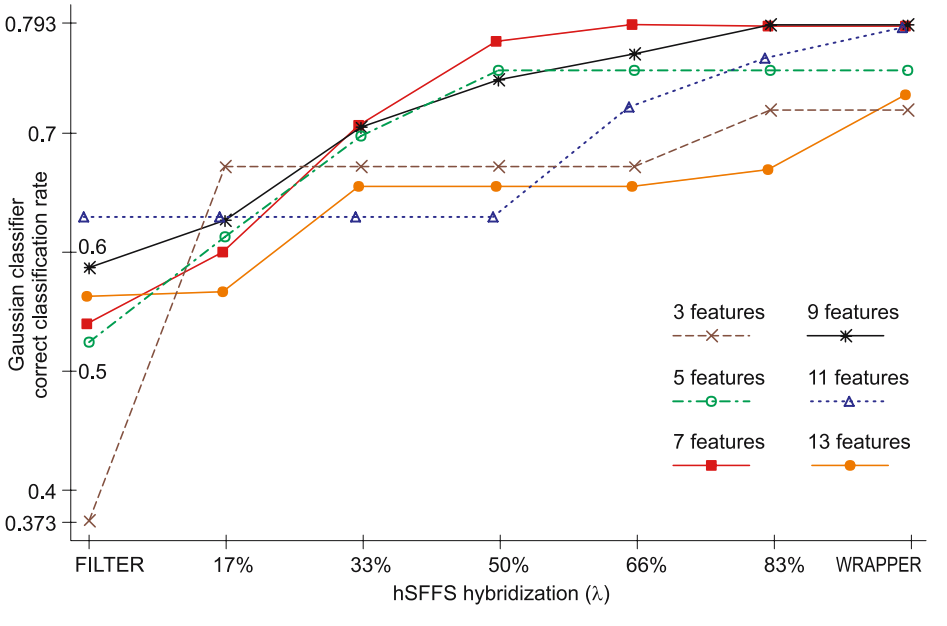

Fig. 2. SPEECH dataset: Comparison of classifier performance on feature subsets selected by the hSFFS for different λ, the *filter* SFFS and the *wrapper* SFFS

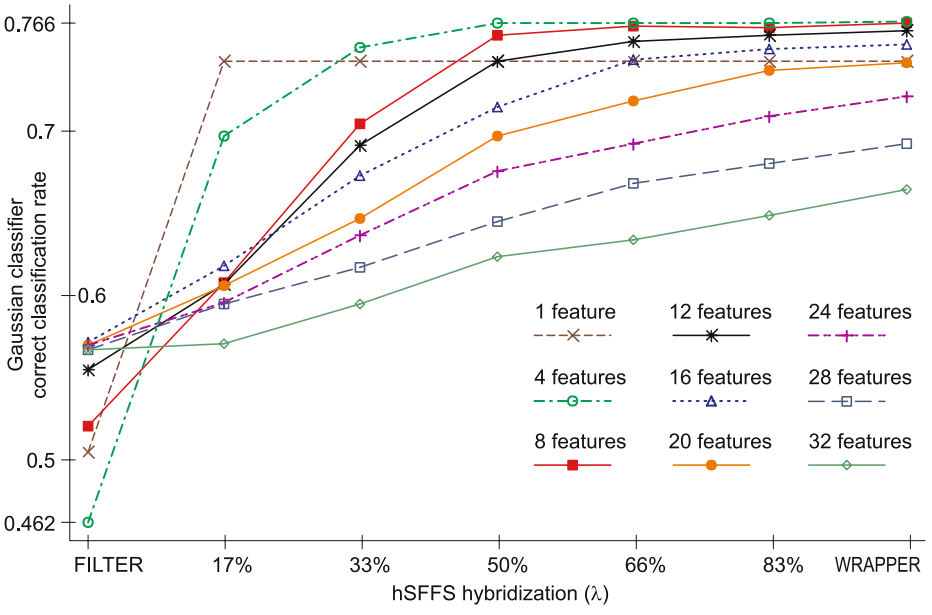

Fig. 3. WAVEFORM dataset: Comparison of classifier performance on feature subsets selected by the hSFFS for different λ, the *filter* SFFS and the *wrapper* SFFS

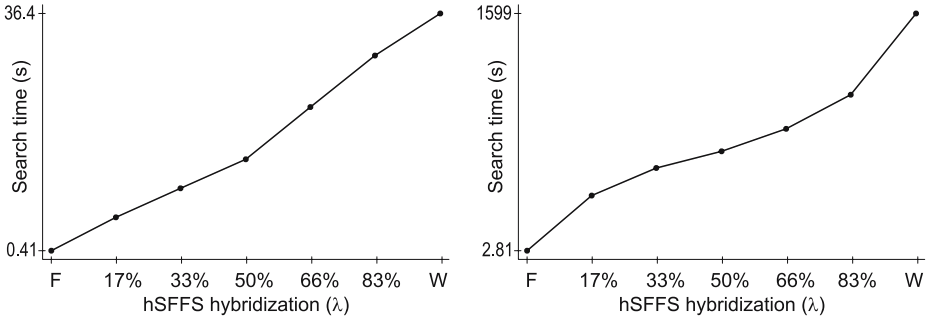

Fig. 4. SPEECH and WAVEFORM datasets: Time complexity of the *filter* SFFS, the hSFFS as a function of λ and the *wrapper* SFFS

5 Conclusions and Future Work

We have defined a flexible hybrid version of floating search methods for feature selection. The main benefit of the proposed floating search hybridization is the possibility to deal flexibly with the quality-of-result vs. computational time trade-off and to enable wrapper based feature selection in problems of higher dimensionality than before. We have shown that it is possible to trade significant reduction of search time for often negligible decrease of the classification accuracy.

In the future we intend to "hybridize" other search methods in a similar way as presented here and to investigate in detail the hybrid behavior of different combinations of various probabilistic measures and learning methods.

Acknowledgement. The work has been supported by the following grants: CR MŠMT grant 1M0572 DAR, EC project FP6-507752 MUSCLE, Grant Agency of the Academy of Sciences of the Czech Republic (CR) No. A2075302, and CR Grant Agency grant No. 402/03/1310.

References

1. Liu, H., Yu, L.: Toward Integrating Feature Selection Algorithms for Classification and Clustering. IEEE Trans. on Knowledge and Data Engineering **17** (2005) 491-502
2. Yu, L., Liu, H.: Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. In: Proc. 20th Intl Conf. Machine Learning (2003) 856-863
3. Dash, M., Choi, K., Scheuermann, P., Liu, H.: Feature Selection for Clustering - a Filter Solution. In: Proc. Second Int. Conf. Data Mining (2002) 15-122
4. Kohavi, R., John, G.H.: Wrappers for Feature Subset Selection. Artificial Intelligence **97** (1997) 273–324
5. Das, S.: Filters, Wrappers and a Boosting-Based Hybrid for Feature Selection. In: Proc. 18th Intl Conf. Machine Learning (2001) 74-81
6. Sebban, M., Nock, R.: A Hybrid Filter/Wrapper Approach of Feature Selection using Information Theory. Pattern Recognition **35** (2002) 835–846
7. Pudil, P., Novovicova, J., Kittler, J.: Floating Search Methods in Feature Selection. Pattern Recognition Letters **15** (1994) 1119-1125
8. Pudil, P., Novovicova, J., Somol, P.: Recent Feature Selection Methods in Statistical Pattern Recognition. In: Pattern Recognition and String Matching, Springer-Verlag, Berlin Heidelberg New York (2003)
9. Jain, A.K., Zongker, D.: Feature selection: evaluation, application and small sample performance. IEEE Trans. PAMI **19** (1997) 153–158
10. Kudo, M., Sklansky, J.: Comparison of algorithms that select features for pattern classifiers. Pattern Recognition **33** (2000) 25–41
11. Devijver, P.A. Kittler, J Pattern Recognition: A Statistical Approach. Prentice-Hall (1982)
12. Murphy, P.M., Aha, D.W.: UCI Repository of Machine Learning Databases [Machine-readable data repository]. University of California, Department of Information and Computer Science Irvine CA (1994)