Methodology of selecting the most informative variables for decision-making problems of classification type

Pavel Pudil, Petr Somol*, Rudolf Střítecký
Faculty of Management, Jindřichův Hradec, Prague University of Economics, Czech Republic
pudil@fm.vse.cz, somol@utia.cas.cz, striteru@fm.vse.cz

Abstract: The paper gives an overview of feature selection (abbreviated FS in the sequel) techniques in statistical pattern recognition with particular emphasis to recent knowledge. FS methods constitute the methodology of selecting the most informative variables for decision-making problems of classification type. Besides discussing the advances in methodology it attempts to put them into a taxonomical framework. The methods discussed include the latest variants of the optimal algorithms, enhanced sub-optimal techniques and the simultaneous semi-parametric probability density function modeling and feature space selection method. Some related issues are illustrated on real data with use of Feature Selection Toolbox software.

1 Introduction

A broad class of decision-making problems can be solved by *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. The first and essential step of statistical pattern recognition is to solve the problem of feature selection or more generally dimensionality reduction.

The methodology of feature selection in statistical pattern recognition will be presented in this survey paper in the form of a tutorial. The problem will be introduced in a wider context of dimensionality reduction which can be accomplished either by a linear or nonlinear mapping from the measurement space to a lower dimensional feature space, or by measurement subset selection. The tutorial will focus on the latter. The main aspects of the problem, i.e., criteria for feature selection and the associated optimization techniques will be discussed. The material presented will be structured according the level of prior knowledge available to solve the feature selection problem. The techniques covered will include efficient optimal search algorithms, Floating Search algorithms, and simultaneous probability density function modeling and dimensionality reduction for feature selection involving nonparametrically distributed classes.

The objectives are: to stress the analogy between decision-making in various fields and the usefulness of learning approaches – e.g., in medicine, management, economics and finances, and to demonstrate the necessity of selecting the most informative variables in order to improve the quality of decision-making based on the learning approach. The target audience may include newcomers to the field of pattern recognition as well as practitioners wishing to become more familiar with available dimensionality reduction methods and with their critical analysis with respect to usability in practical tasks.

Pattern recognition can be with certain simplification characterized as a classification problem combined with dimensionality reduction of pattern feature vectors which serve as the input to the classifier. This reduction is achieved by extracting or selecting a feature subset which optimizes an adopted criterion.

2 Dimensionality Reduction

We shall use the term "pattern" to denote the D-dimensional data vector $\mathbf{x} = (x_1, \ldots, x_D)^T$ 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 $\mathbf{x}$ is to be classified into one of a finite set of C different classes $\Omega = \{\omega_1, \omega_2, \cdots, \omega_C\}$. A pattern $\mathbf{x}$ belonging to class ω_i is viewed as an observation of a random vector $\mathbf{X}$ drawn randomly according to the known class-conditional probability density function $p(\mathbf{x}|\omega_i)$ and the respective *a priori* probability $P(\omega_i)$.

One of the fundamental problems in statistical pattern recognition is representing patterns in the reduced number of dimensions. In most of practical cases the pattern descriptor space dimensionality is rather high. It follows from the fact that in the design phase it is too difficult or impossible to evaluate directly the "usefulness" of particular input. Thus it is important to initially in-

*The author is primarily with the Institute of Information Theory and Automation, Prague

Proc. of the 6th International Conference on Information and Management Sciences, July 1-6, 2007, Lhasa, Tibet, China, pp. 212 - 229.

clude all the "reasonable" descriptors the designer can think of and to reduce the set later on. Obviously, information missing in the original measurement set cannot be later substituted. The aim of dimensionality reduction is to find a set of new d features based on the input set of D features (if possible $d \ll D$), so as to maximize (or minimize) an adopted criterion.

- Dimensionality reduction divided according to the adopted strategy:
  1. *feature selection* (FS)
  2. *feature extraction* (FE)

The first strategy (FS) is to select the best possible subset of the input feature set. The second strategy (FE) is to find a transformation to a lower dimensional space. New features are linear or nonlinear combinations of the original features. Technically FS is special case of FE. The choice between FS and FE depends on the application domain and the specific available training data. FS leads to savings in measurements cost since some of the features are discarded and those selected retain their original physical meaning. The fact that FS preserves the interpretability of original data makes it preferable in, e.g., most problems of computer-assisted medical decision-making. On the other hand, features generated by FE may provide better discriminative ability than the best subset of given features, but these new features may not have a clear physical meaning.

- Alternative division according to the aim:
  1. *dim. reduction for optimal data representation*
  2. *dimensionality reduction for classification.*

The first aims to preserve the topological structure of data in a lower-dimensional space as much as possible, the second one aims to enhance the subset discriminatory power. In the sequel we shall concentrate on the FS problem only. For a broader overview of the subject see, e.g., [4], [18], [29], [39], [43].

3 Feature Selection

Given a set of D features, X_D, let us denote $\mathcal{X}_d$ the set of all possible subsets of size d, where d represents the desired number of features. Let J be some criterion function. Without any loss of generality, let us consider a higher value of J to indicate a better feature subset. Then the feature selection problem can be formulated as follows: find the subset $\tilde{X}_d$ for which

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

Assuming that a suitable criterion function has been chosen to evaluate the effectiveness of feature subsets, feature selection is reduced to a search problem that detects an optimal feature subset based on the selected measure. Note that the choice of d may be a complex issue depending on problem characteristics, unless the d value can be optimized as part of the search process.

One particular property of feature selection criterion, the *monotonicity property*, is required specifically in certain optimal FS methods. Given two subsets of the feature set X_D, A and B such that $A \subset B$, the following must hold:

$$A \subset B \Rightarrow J(A) < J(B). \tag{2}$$

That is, evaluating the feature selection criterion on a subset of features of a given set yields a smaller value of the feature selection criterion.

3.1 FS Categorisation With Respect to Optimality

Feature selection methods can be split into basic families:

1. *Optimal methods*: These include, e.g., *exhaustive search* methods which are feasible for only small size problems and accelerated methods, mostly built upon the Branch & Bound principle. All optimal methods can be expected considerably slow for problems of high dimensionality.

2. *Sub-optimal methods*: essentially trade the optimality of the selected subset for computational efficiency. They include, e.g., Best Individual Features, Random (Las Vegas) methods, Sequential Forward and Backward Selection, Plus-l-Take Away-r, their generalized versions, genetic algorithms, and particularly the Floating and Oscillating Search.

Although the exhaustive search guarantees the optimality of a solution, in many realistic problems it is computationally prohibitive. The well known Branch and Bound (B&B) algorithm guarantees to select an optimal feature subset of size d without involving explicit evaluation of all the possible combinations of d measurements. However, the algorithm is applicable only under the assumption that the feature selection criterion used satisfies the monotonicity property (2). This assumption precludes the use of classifier error rate as the criterion (cf. Wrappers [12]). This is an important drawback as the error rate can be considered superior to other criteria [32], [12], [40]. Moreover, all optimal algorithms become computationally prohibitive for problems of high dimensionality. In practice, therefore, one has to rely on computationally feasible procedures which perform the search quickly but may yield sub-optimal results. A comprehensive list of sub-optimal procedures can be found, e.g., in books [3], [6], [43], [39]. A comparative taxonomy can be found, e.g., in [1], [5], [7], [10], [11], [13], [14], [30], [41] or [44]. Our own research and experience with FS has led us to the conclusion that *there exists no unique generally applicable approach* to the problem. Some are

more suitable under certain conditions, others are more appropriate under other conditions, depending on our *knowledge of the problem.* Hence continuing effort is invested in developing new methods to cover the majority of situations which can be encountered in practice.

3.2 FS Categorisation With Respect to Problem Knowledge

From another point of view there are perhaps two basic classes of situations with respect to *a priori* knowledge of the underlying probability structures:

- *Some a priori knowledge is available* – It is at least known that probability density functions (pdfs) are unimodal. In these cases, one of probabilistic distance measures (like Mahalanobis, Bhattacharyya, etc.) may be appropriate as the evaluation criterion. For this type of situations we recommend either the recent prediction-based B&B algorithms for optimal search (see Sect. 4), or sub-optimal Floating and Oscillating methods (Section 5).

- *No a priori knowledge is available* – We cannot even assume that pdfs are unimodal. The only source of available information is the training data. For these situations we have developed two conceptually different alternative methods. They are based on approximating unknown conditional pdfs by finite mixtures of a special type and are discussed in Section 6.

4 Recent Optimal Search Methods

The problem of optimal feature selection (or more generally of subset selection) is difficult especially because of its time complexity. All known optimal search algorithms have an exponential nature. The only alternative to exhaustive search is the *Branch & Bound* (B&B) algorithm [20], [6] and ancestor algorithms based on a similar principle. All B&B algorithms rely on the monotonicity property of the FS criterion (2). By a straightforward application of this property many feature subset evaluations may be omitted.

Before discussing more advanced algorithms, let us briefly summarize the essential B&B principle. The algorithm constructs a search tree where the root represents the set of all D features, X_D, and leaves represent target subsets of d features. While tracking the tree down to leaves the algorithm successively removes single features from the current set of "candidates" ($\bar{\mathcal{X}}_k$ in k-th level). The algorithm keeps the information about both the till-now best subset of cardinality d and the corresponding criterion value, denoted as the *bound*. Anytime the criterion value in some internal node is found to be lower than the current *bound*, due to condition (2) the whole sub-tree may be cut-off and many computations may be omitted. The course of the B&B algorithm can be seen in Fig. 1 (symbols `C`, `P` and A_i relate to the more advanced B&B version to be discussed in Section 4.2). The described scheme in its simplest form is known as the "Basic B&B" algorithm. For details see [3], [6].

4.1 Branch & Bound Properties

When compared to the exhaustive search, every B&B algorithm requires additional computations. Not only the target subsets of d features $\bar{X}_{D-d}$, but also their supersets $\bar{X}_{D-d-j}$, $j = 1, \cdots, D-d$ have to be evaluated. The B&B principle does not guarantee enough sub-tree cut-offs to keep the total number of criterion computations lower than in exhaustive search. To reduce the amount of criterion computations an additional node-ordering heuristic has been introduced in the more powerful "Improved B&B" (IBB) algorithm [3], [6]. IBB optimizes the order of features to be assigned to tree edges so that the *bound* value can increase as fast as possible and thus enables more effective branch cutting in later stages. Although IBB usually outperforms all simpler B&B algorithms, the computational cost of the additional heuristic can become a strong deteriorating factor. For detailed discussion of B&B drawbacks see [37]. In the following we present a more efficient framework for B&B acceleration.

4.2 Fast Branch & Bound

The Fast Branch & Bound (FBB) [37] algorithm aims to reduce the number of criterion function computations in internal search tree nodes. A simplified algorithm description is as follows: FBB attempts to utilize the knowledge of past feature-dependent *criterion value decreases* (difference between criterion values before and after feature removal) for future prediction of criterion values without the need of real computation. Prediction is allowed under certain conditions only, e.g., not in leaves. Both the really computed and predicted criterion values are treated as equal while imitating the full IBB functionality, i.e., in ordering node descendants in the tree construction phase. If the predicted criterion value remains significantly higher than the current *bound*, we may expect that even the actual value would not be lower and the corresponding sub-tree could not be cut-off. In this situation the algorithm continues to construct the consecutive tree level. However, if the predicted value is equal or lower than the *bound* (and therefore there arises a chance that the real value is lower than the *bound*), the real criterion value must be computed. Only if real criterion values are lower than the current *bound*, sub-trees may be cut-off. Note that this prediction scheme does

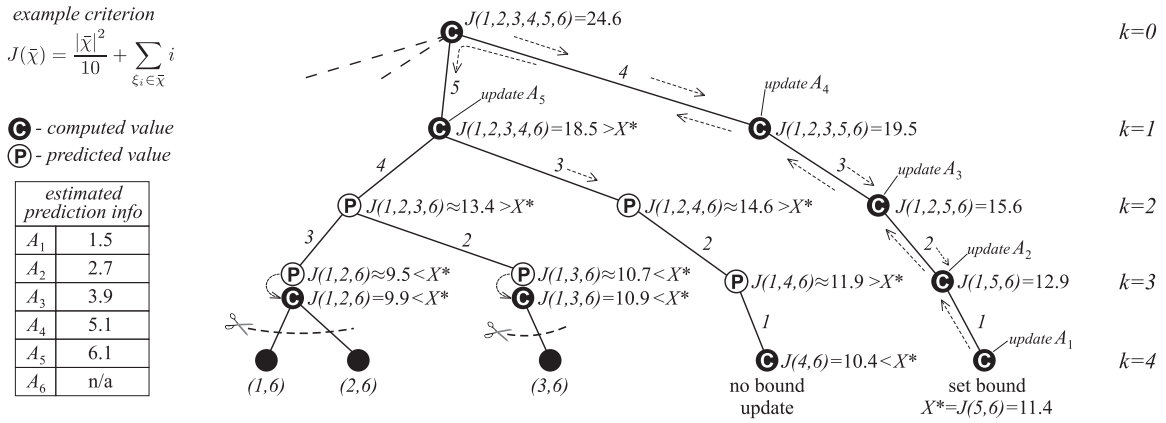

Figure 1: *Example of a "Fast Branch & Bound" problem solution, where $d = 2$ features are to be selected from a set of $D = 5$ features. Dashed arrows illustrate the way of tracking the search tree.*

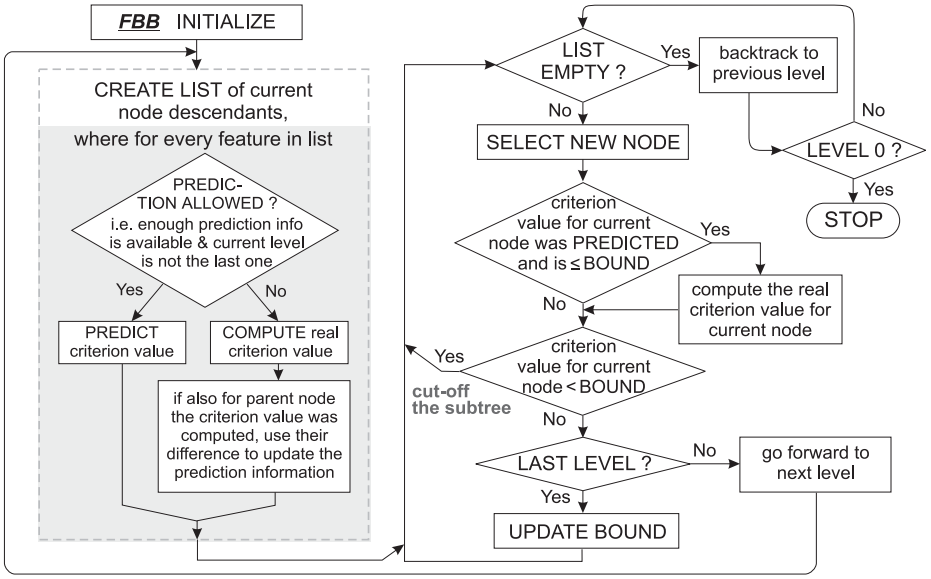

Figure 2: *Simplified diagram of the Fast Branch & Bound algorithm*

not affect the optimality of obtained results. The FBB algorithm course remains similar to that of the IBB, possible sub-tree cut-offs are allowed according to real criterion values only. Possible inaccurate predictions may result in nothing worse than constructing sub-trees, which would have been pruned out by means of classical B&B algorithms. However, this situation is usually strongly outweighed by criterion computation savings in other internal nodes, especially near the root, where criterion computation tends to be slower. The prediction mechanism processes the information about the averaged *criterion value decrease* separately for each feature. The idea is illustrated in Fig. 1. For a detailed and formal description of this rather complex procedure and other B&B related topics see [37].

4.3 Improving the "Improved" Algorithm

The FBB operates mostly the fastest among all B&B algorithms. However, it exhibits some drawbacks: it cannot be used with recursive criterion forms and there is no theoretical guarantee that extensive prediction failures won't hinder the overall speed, despite the fact that such faulty behaviour has not been observed with real data. The B&B with Partial Prediction (BBPP) [37] constitutes a slightly less effective but more robust alternative. While learning similarly to FBB, it does not use predictions to substitute true criterion values inside the tree. Predicted values are used only for ordering before features get assigned to tree edges. In this sense BBPP can be looked upon as a slightly modified IBB with the only difference in node ordering heuristics. The perfor-

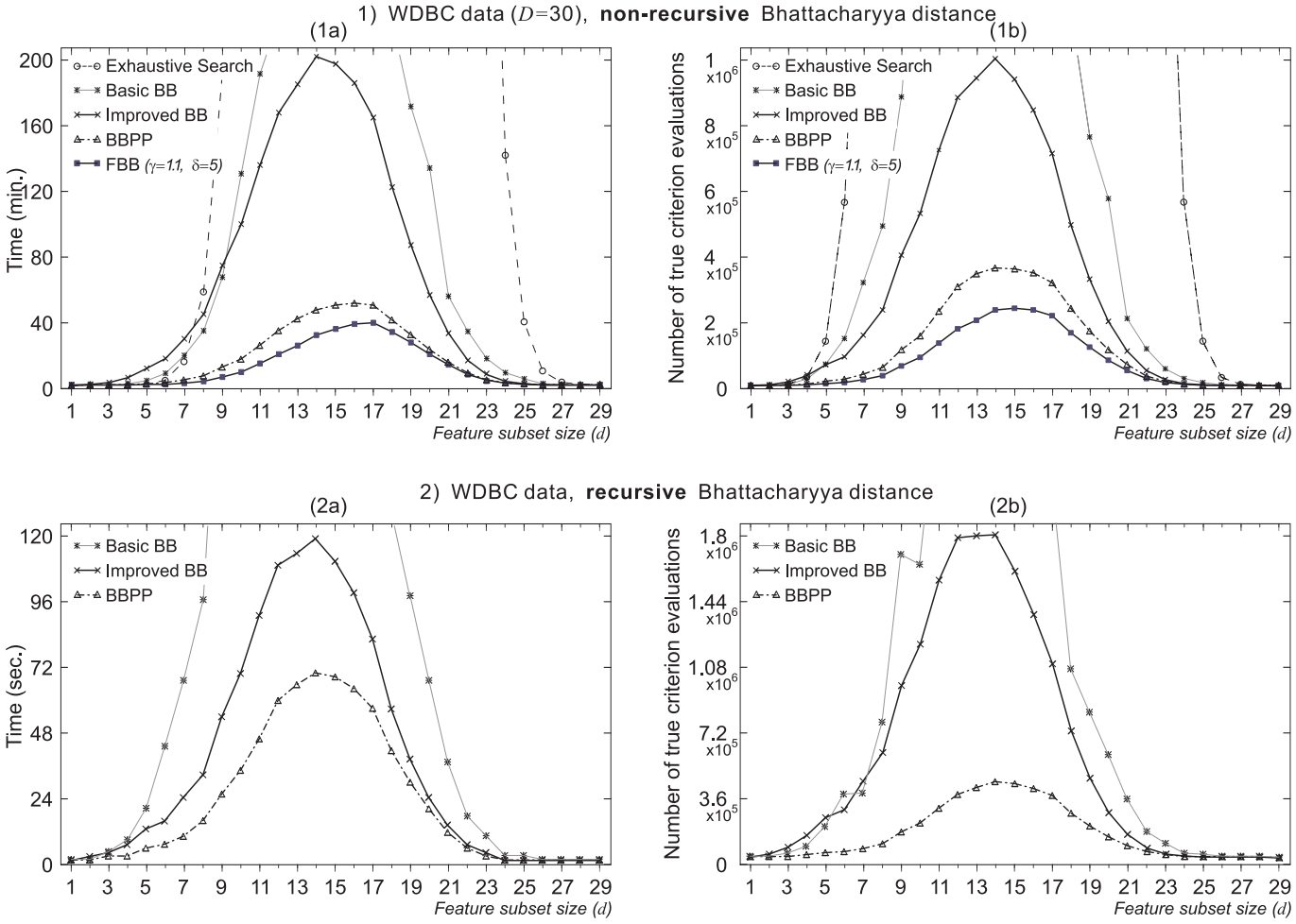

Figure 3: *Optimal subset search methods performance when maximizing the Bhattacharyya distance on 30-dimensional* mammogram *data (Wisconsin Diagnostic Breast Center).*

mance gain follows from the fact, that the original IBB ordering heuristics always evaluates more criterion values than is the number of features finally used. For detailed analysis of BBPP see [37]. Among other recent B&B related ideas the "trading space for speed" approach [9] deserves attention as an alternative that may operate exceptionally fast under certain circumstances. The BBPP and FBB algorithms are further investigated in [38], [42].

4.4 Predictive B&B Properties and Experimental Results

When compared to classical B&B algorithms the predictive algorithms always spend additional time for maintaining the prediction mechanism. However, this additional time showed not to be a factor, especially when compared to time savings arising from the pruned criterion computations. The algorithms have been thoroughly tested on a number of different data sets. Here we show representative results on 2-class 30-dimensional *mammogram* data (for dataset details see Section 7). We used both the recursive (where applicable) and non-recursive Bhattacharyya distance as the criterion function. Performance of different methods is illustrated in Fig. 3. We compare all results especially against the IBB algorithm [3], [6], as this algorithm has been long accepted to be the most effective optimal subset search method. Remark: Where applicable, we implement all algorithms to support the "minimum solution tree" [45].

For discussion about the applicability of optimal mathods in comparison with sub-optimal methods and the impact of optimization on classifier performance see also Section 7.

4.5 Summary of Recent Optimal Methods

The only optimal subset search method usable with non-monotonic criteria is the exhaustive (full) search. However, because of exponential nature of the search problem, alternative methods are often needed. Several re-

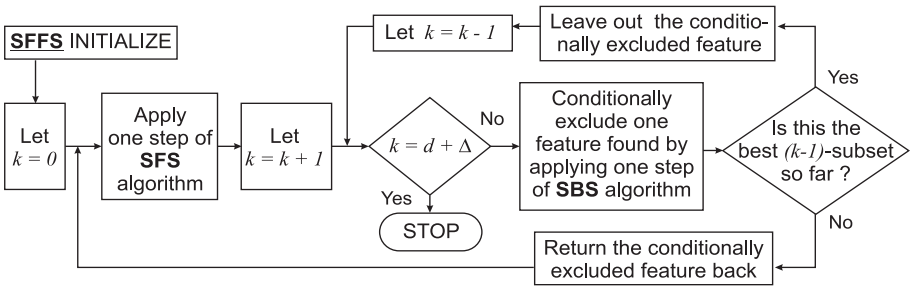

Figure 4: *Sequential Forward Floating Selection Algorithm*

cent improvements of the B&B idea especially in the form of prediction based FBB and BBPP resulted in a speed-up factor of 10 to 100 over the simplest B&B form, depending on particular data and criterion used.

It should be stressed that despite the shown advances all optimal methods remain exponential in nature. If there is no need to insist on optimality of results (note that this optimality may be only indirectly related to classifier performance), sub-optimal search methods offer greater flexibility and acceptable speed even for high-dimensional problems, while the solutions found are not necessarily much worse than optimal.

5 Recent Sub-optimal Search Methods

Despite the advances in optimal search, for larger than moderate-sized problems we have to resort still to sub-optimal methods. The basic feature selection approach is to build up a subset of required number of features incrementally starting with the empty set (*bottom-up* approach) or to start with the complete set of features and remove redundant features until d features retain (*top-down* approach). The simplest (among recommendable choices) yet widely used *sequential forward (or backward) selection* methods [3], SFS (SBS), iteratively add (remove) one feature at a time so as to maximize the intermediate criterion value until the required dimensionality is achieved. Among the more interesting recent approaches the following two families of methods can be pointed out for general applicability and performance reasons:

1. *sequential Floating Search methods [24], [33]*
2. *Oscillating Search methods [34]*

Earlier sequential methods suffered from the so-called nesting of feature subsets which significantly deteriorated the performance. The first attempt to overcome this problem was to employ either the Plus-l-Take away-r [also known as (l, r)] or generalized (l, r) algorithms [3] which involve successive augmentation and depletion process. The same idea in a principally extended and refined form constitutes the basis of Floating Search.

5.1 Sequential Floating Search

The Sequential Forward Floating Selection (SFFS) procedure consists of applying after each forward step a number of backward 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 intermediate result at actual level (of corresponding dimensionality) cannot be improved. The same applies for the backward version of the procedure. Both algorithms allow a 'self-controlled backtracking' so they can eventually find good solutions by adjusting the trade-off between forward and backward steps dynamically. In a certain way, they compute only what they need without any parameter setting.

Formal description of this now classical procedure can be found in [24]. Neverheless, the idea behind is simple enough and can be illustrated sufficiently in Fig. 4. (Condition $k = d + \delta$ terminates the algorithm after the target subset of d features has been found and possibly refined by means of backtracking from dimensionalities greater than d.) The backward counterpart to SFFS is the Sequential Backward Floating Selection (SBFS). Its principle is analogous.

Floating search algorithms can be considered universal tools not only outperforming all predecessors, but also keeping advantages not met by more sophisticated algoritms. They find good solutions in all problem dimensions in one run. The overall search speed is high enough for most of practical problems.

5.2 Adaptive Floating Search

As the Floating Search algorithms have been found successful and generally accepted to be an efficient universal tool, their idea was further investigated. The so-called Adaptive Floating Search has been proposed in [33]. The ASFFS and ASBFS algorithms are able to outperform the classical SFFS and SBFS algorithms in certain cases, but at a cost of considerabe increase of search time and the necessity to deal with unclear parameters. Our experience shows that AFS is usually inferior to newer algorithms, which we focus on in the following.

5.3 Oscillating Search

The recent Oscillating Search (OS) [34] can be considered a "higher level" procedure, that takes use of other feature selection methods as sub-procedures in its own search. The concept is highly flexible and enables modifications for different purposes. It has shown to be very powerful and capable of over-performing standard sequential procedures, including (Adaptive) Floating Search. Unlike other methods, the OS is based on repeated modification of the current subset X_d of d features. In this sense the OS is independent on predominant search direction. This is achieved by alternating so-called *down-* and *up-swings*. Both *swing*s attempt to improve the current set X_d by replacing some of the features by better ones. The *down-swing* first removes, then adds back, while the *up-swing* first adds, then removes. Two successive opposite swings form an *oscillation cycle*. The OS can thus be looked upon as a controlled sequence of oscillation cycles. The value of o (denoted *oscillation cycle depth*) determines the number of features to be replaced in one swing. o is increased after unsuccessful oscillation cycles and reset to 1 after each X_d improvement. The algorithm terminates when o exceeds a user-specified *limit* Δ. The course of Oscillating Search is illustrated in comparison with SFFS in Fig. 5.

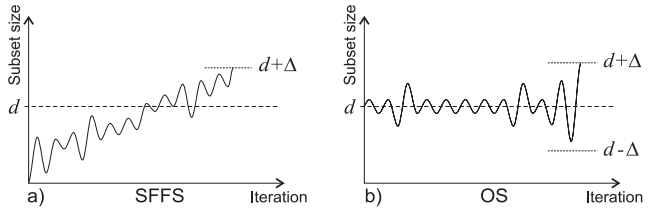

Figure 5: *Graphs demonstrate the course of search algorithms: a) Sequential Floating Forward Selection, b) Oscillating Search.*

Every OS algorithm requires some initial set of d features. The initial set may be obtained randomly or in any other way, e.g., using some of the traditional sequential selection procedures. Furthermore, almost any feature selection procedure can be used in *up-* and *down-swing*s to accomplish the replacements of feature o-tuples. Therefore, for the sake of generality in the following descriptions let us denote the adding / removing of a feature o-tuple by ADD(o) / REMOVE(o). For OS flow-chart see Fig. 6.

5.3.1 Oscillating Search – Formal Algorithm Description

Step 1: (*Initialization*) By means of any feature selection procedure (or randomly) determine the initial set X_d of d features. Let $c = 0$. Let $o = 1$.

Step 2: (*Down-swing*) By means of REMOVE(o) remove such o-tuple from X_d to get new set X_{d-o} so that $J(X_{d-o})$ is maximal. By means of ADD(o) add such o-tuple from $X_D \setminus X_{d-o}$ to X_{d-o} to get new set $X_d^{'}$ so that $J(X_d^{'})$ is maximal. If $J(X_d^{'}) > J(X_d)$, let $X_d = X_d^{'}$, $c = 0$, $o = 1$ and go to Step 4.

Step 3: (*Last swing has not improved the solution*) Let $c = c + 1$. If $c = 2$, then nor the last *up-* nor *down-swing* led to a better solution. Extend the search by letting $o = o + 1$. If $o > \Delta$, stop the algorithm, otherwise let $c = 0$.

Step 4: (*Up-swing*) By means of ADD(o) add such o-tuple from $X_D \setminus X_d$ to X_d to get new set X_{d+o} so that $J(X_{d+o})$ is maximal. By means of REMOVE(o) remove such o-tuple from X_{d+o} to get new set $X_d^{'}$ so that $J(X_d^{'})$ is maximal. If $J(X_d^{'}) > J(X_d)$, let $X_d = X_d^{'}$, $c = 0$, $o = 1$ and go to Step 2.

Step 5: (*Last swing has not improved the solution*) Let $c = c + 1$. If $c = 2$, then nor the last *up-* nor *down-swing* led to a better solution. Extend the search by letting $o = o + 1$. If $o > \Delta$, stop the algorithm, otherwise let $c = 0$ and go to Step 2.

5.4 Oscillating Search Properties

The generality of OS search concept allows to adjust the search for better speed or better accuracy (lower Δ and simpler ADD / REMOVE vs. higher Δ and more complex ADD / REMOVE). In this sense let us denote *sequential OS* the simplest possible OS version which uses a sequence of SFS steps in place of ADD() and a sequence od SBS steps in place of REMOVE(). As opposed to all sequential search procedures, OS does not waste time evaluating subsets of cardinalities too different from the target one. This "focus" improves the OS ability to find good solutions for subsets of given cardinality. The fastest improvement of the target subset may be expected in initial phases of the algorithm, because of the low initial cycle depth. Later, when the current feature subset evolves closer to optimum, low-depth cycles fail to improve and therefore the algorithm broadens the search ($o = o + 1$). Though this improves the chance to get closer to the optimum, the trade-off between finding a better solution and computational time becomes more apparent. Consequently, OS tends to improve the solution most considerably during the fastest initial search stages. This behavior is advantageous, because it gives the option of stopping the search after a while without serious result-degrading consequences. Let us summarize the key OS advantages:

- It may be looked upon as a universal tuning mechanism, being able to improve solutions obtained in other way.

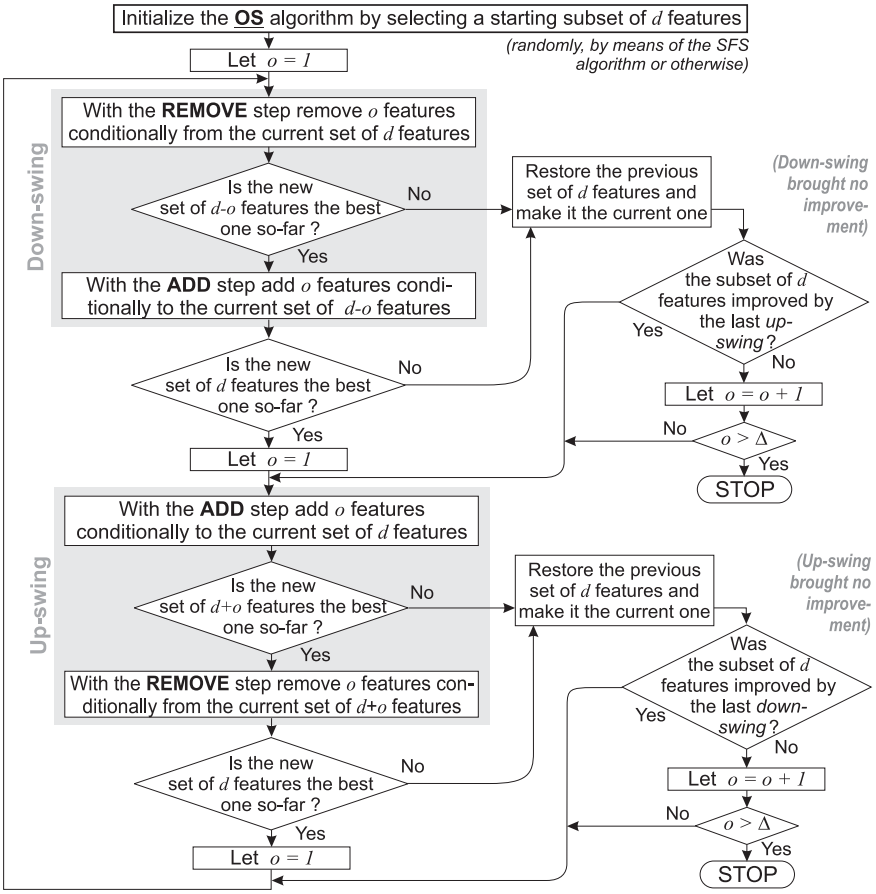

Figure 6: *Simplified Oscillating Search algorithm flowchart.*

- The randomly initialized OS is very fast, in case of very high-dimensional problems may become the only applicable procedure. E.g., in document analysis for search of the best 1000 words out of a vocabulary of 50000 even the simplest alternative methods may prove to be too slow.

- Because the OS processes subsets of target cardinality from the very beginning, it may find solutions even in cases, where the sequential procedures fail due to numerical problems.

- Because the solution improves gradually after each oscillation cycle, with the most notable improvements at the beginning, it is possible to terminate the algorithm prematurely after a specified amount of time to obtain a usable solution. The OS is thus suitable for use in real-time systems.

- In some cases the sequential search methods tend to uniformly get caught in certain local extremes. Running the OS from several different random initial points gives better chances to avoid that local extreme.

5.5 Experimental Results of Sub-optimal Search Methods

All described sub-optimal sequential search methods have been tested on a large number of different problems. Here we demonstrate their performance on 2-class 30-dimensional *mammogram* data (for dataset details see Section 7). The graphs in Figure 7 show the OS ability to outperform other methods even in the simplest *sequential* form (here with $\Delta = d$ in only one randomly initialized run). ASFFS behavior is well illustrated here showing better performance than SFFS at a cost of uncontrollably increased time. SFFS and SFS need one run only to get all solutions. SFFS performance is uniformly better than that of SFS. Note that results in Fig. 7 have been obtained only to compare various search methods among themselves – using data resubstitution (all data used both for training and testing). Resubstitution should not be used to assess the resulting classifier performance because it yields optimistically biased estimates. Figure 7 illustrates this effect when compared to Figures 8 and 10 where the accuracy of gaussian classifier under similar setup is estimated using cross-validation –

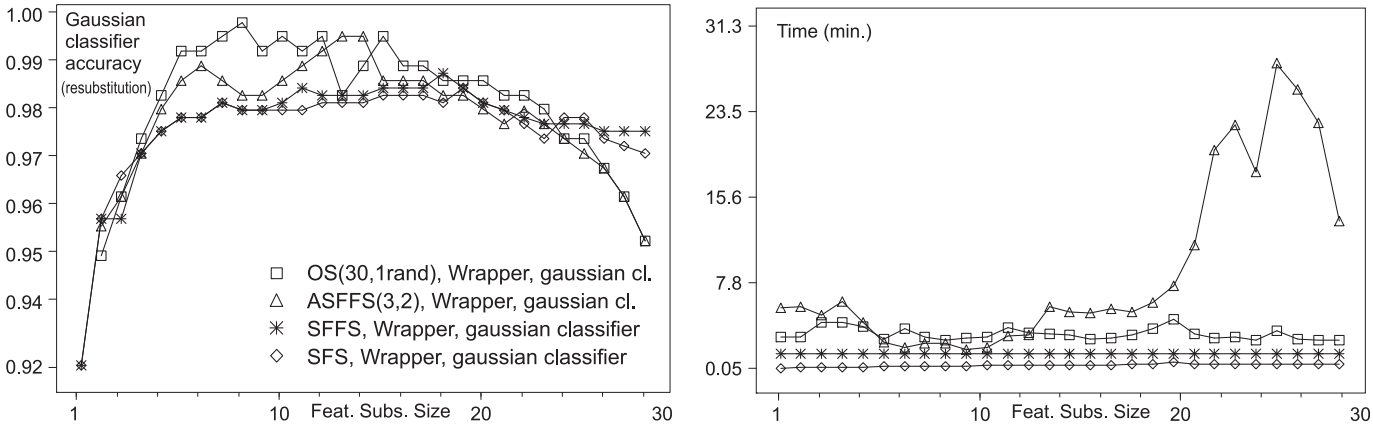

Figure 7: *Comparison of sub-optimal methods on Wrapper-based search task (to maximize gaussian classifier accuracy)*

the estimate is then notably lower by ca. 1%. For further discussion on the applicability of sub-optimal feature selection methods see also Section 7.

5.6 Summary of Recent Sub-optimal Methods

Concerning our current experience, we can give the following recommendations. Floating Search can be considered the first tool to try. It is reasonably fast and yields generally very good results in all dimensions at once, often succeeding in finding global optimum. The Oscillating Search becomes better choice whenever: 1) the highest quality of solution must be achieved but optimal methods are not applicable, or 2) a reasonable solution is to be found as quickly as possible, or 3) numerical problems hinder the use of sequential methods, or 4) extreme problem dimensionality prevents any use of sequential methods, or 5) the search is to be performed in real-time systems. Especially when repeated with different random initial sets the Oscillating Search shows outstanding potential to overcome local extremes in favor of global optimum.

It should be stressed that, as opposed to B&B, the Floating Search and Oscillating Search methods are tolerant to deviations from monotonic behaviour of feature selection criteria. It makes them particularly useful in conjunction with non-monotonic FS criteria like the error rate of a classifier (cf. Wrappers [12]), which according to a number of researchers seem to be the only legitimate criterion for feature subset evaluation.

Note: Floating and Oscillating Search source codes can be found at `http://ro.utia.cas.cz/dem.html`.

6 Mixture Based Methods

For the cases when no simplifying assumptions can be made about the underlying class distributions we developed a new approach based on approximating the unknown class conditional distributions by finite mixtures of parametrized densities of a special type. In terms of the required computer storage this pdf estimation is considerably more efficient than nonparametric pdf estimation methods.

Denote the ωth class training set by $\mathbf{X}_\omega$ and let the cardinality of set $\mathbf{X}_\omega$ be N_ω. The modeling approach to feature selection taken here is to approximate the class densities by dividing each class $\omega \in \Omega$ into M_ω artificial subclasses. The model assumes that each subclass m has a multivariate distribution $p_m(\mathbf{x}|\omega)$ with its own parameters. Let α_m^ω be the mixing probability for the mth subclass, $\sum_{m=1}^{M\omega} \alpha_m^\omega = 1$.The following model for ωth class pdf of $\mathbf{x}$ is adopted [26], [21]:

$$\begin{aligned} p(\mathbf{x}|\omega) &= \textstyle\sum_{m=1}^{M\omega} \alpha_m^\omega p_m(\mathbf{x}|\omega) = \\ &= \textstyle\sum_{m=1}^{M\omega} \alpha_m^\omega g_0(\mathbf{x}|\mathbf{b}_0) g(\mathbf{x}|\mathbf{b}_m^\omega, \mathbf{b}_0, \Phi) \end{aligned} \tag{3}$$

Each component density $p_m(\mathbf{x}|\omega)$ includes a nonzero "background" pdf g_0, common to all classes:

$$g_0(\mathbf{x}|\mathbf{b}_0) = \prod_{i=1}^{D} f_i(x_i|b_{0i}), \quad \mathbf{b}_0 = (b_{01}, b_{02}, \cdots, b_{0D}), \tag{4}$$

and a function g specific for each class of the form:

$$g(\mathbf{x}|\mathbf{b}_m^\omega, \mathbf{b}_0, \Phi) = \prod_{i=1}^{D} \left[\frac{f_i(x_i|b_{mi}^\omega)}{f_i(x_i|b_{0i})} \right]^{\phi_i}, \quad \phi_i = \{0, 1\} \tag{5}$$

$$\mathbf{b}_m^\omega = (b_{m1}^\omega, b_{m2}^\omega, \cdots b_{mD}^\omega),$$
$$\Phi = (\phi_1, \phi_2, \cdots, \phi_D) \in \{0, 1\}^D.$$

The univariate function f_i is assumed to be from a family of normal densities. The model is based on the idea to identify a common "background" density for all the classes and to express each class density as a mixture of

the product of this "background" density with a class-specific modulating function defined on a subspace of the feature vector space. This subspace is chosen by means of the nonzero binary parameters ϕ_i and the same subspace of $\mathcal{X}$ for each component density is used in all the classes. Any specific univariate function $f_i(x_i|b^{\omega}_{mi})$ is substituted by the "background" density $f_i(x_i|b_{0i})$ whenever ϕ_i is zero. In this way the binary parameters ϕ_i can be looked upon as *control variables* as the complexity and the structure of the mixture (3) can be controlled by means of these parameters. For any choice of ϕ_i the finite mixture (3) can be rewritten by using (4) and (5) as

$$\begin{aligned} &p(\mathbf{x}|\alpha_\omega, \mathbf{b}_\omega, \mathbf{b}_0, \Phi) = \\ &= \textstyle\sum_{m=1}^{M\omega} \alpha^{\omega}_m \prod_{i=1}^{D} [f_i(x_i|b_{0i})^{1-\phi_i} f_i(x_i|b^{\omega}_{mi})^{\phi_i}] \\ &\alpha_\omega = (\alpha^{\omega}_1, \alpha^{\omega}_2, \cdots, \alpha^{\omega}_{M\omega}), \\ &\mathbf{b}_\omega = (\mathbf{b}^{\omega}_1, \mathbf{b}^{\omega}_2, \cdots, \mathbf{b}^{\omega}_{M\omega}). \end{aligned} \tag{6}$$

The EM ("Expectation-Maximization") algorithm can be extended to allow a mixture of the form (6) to be fitted to the data. It should be emphasized that although the model looks rather *unfriendly*, its form leads to a tremendous simplification [26] when we use normal densities for functions f. The use of this model (6) makes the process of feature selection a much simpler task.

So as to select those features that are most useful in describing differences between two classes, the Kullback's J-divergence defined in terms of the a posteriori probabilities has been adopted as a criterion of discriminatory content. The goal of the method is to maximize the divergence discrimination, hence the name "Divergence" method (see [21]). Sample mixture-based FS results are given in Section 7.

7 Applicational Examples

Taking use of our Feature Selection Toolbox (FST) [36] and related software we have collected a set of examples to illustrate the expectable behaviour of optimal vs. sub-optimal and Wrapper [12] vs. Filter [12] feature selection methods. We investigated the following real data-sets:

- 2-class, 15-dimensional *speech* data representing words "yes" and "no" obtained from the British Telecom; classes are separable with great difficulty.
- 2-class, 30-dimensional *mammogram* data representing 357 benign and 212 malignant patient samples, obtained from the Wisconsin Diagnostic Breast Center via the UCI repository [19]. The same dataset has been used in experiments in Sections 5.5 and 4.4; classes difficult to separate.
- 3-class, 20-dimensional *marble* data representing different brands of marble stone; classes well separable.

7.1 Data, classifiers and feature selectors all determine classification performance

To illustrate the complexity of problems related to classification system design we have collected a series of experimental results in Figures 8 to 11 – in all cases the *mammogram* data have been used. We compare standard feature selection methods in both Wrapper and Filter settings. In case of Oscillating Search we compare various example method setups (many others remain possible) - from fast deterministic `OS(5,IB)` (deterministic sequential Oscillating Search with $\Delta = 5$ initialized by Individually Best features) to slow `OS(5,rand15)` (sequential Oscillating Search with $\Delta = 5$ called repeatedly with random initialization as long as at least 15 times no better solution has been found). Individually Best (IB) denotes the subset of features obtained simply by ranking according to individual criterion values.

Whenever a classifier has been trained, standalone or as a Wrapper, its classification rate was determined using 10-fold cross-validation (90% of data used for training, 10% for testing – repeated 10× to cover all test combinations).

Figure 8 illustrates mainly two points: 1) for this dataset the gaussian classifier is notably inferior to 1-Nearest Neighbour. This suggests, that the data distribution does not exhibit normal behaviour. 2) Better results can be obtained by investing more search time (this is made possible here by the flexibility of the Oscillating Search procedure). However, the trade-off between achieved classification rate and search time is clearly visible. From certain thoroughness of OS setting any improvement becomes negligible while the search time penalty increases beyond acceptable limits. Moreover, pushing the search procedure to its limits may have negative consequences in form of unwantedly biased result [15], [28].

In Figure 8(b) the speed difference between deterministic and randomized search can be clearly seen. Deterministic procedures (`OS(5,IB)`) tend to be significantly faster than those randomized, with more stable time complexity across various subset sizes. However, randomness may be the key property needed to avoid local extremes (see the problem, e.g., in Figure 8(a) for `OS(5,IB)` gaussian Wrapper with subset sizes 5, 7, and 9). Our experience shows that all deterministic sequential methods are prone to getting caught in local extremes. As there is no procedure available to guarantee optimal Wrapper based feature selection result, the best results we could get come from the sub-optimal randomized Oscillating Search.

The well known "peaking phenomenon" is clearly visible in Figures 8(a) to 10(a). Figure 8 shows that with the *mammogram* dataset the 1-NN classifier performs best on ca. 13-dimensional subspace, while gaussian clas-

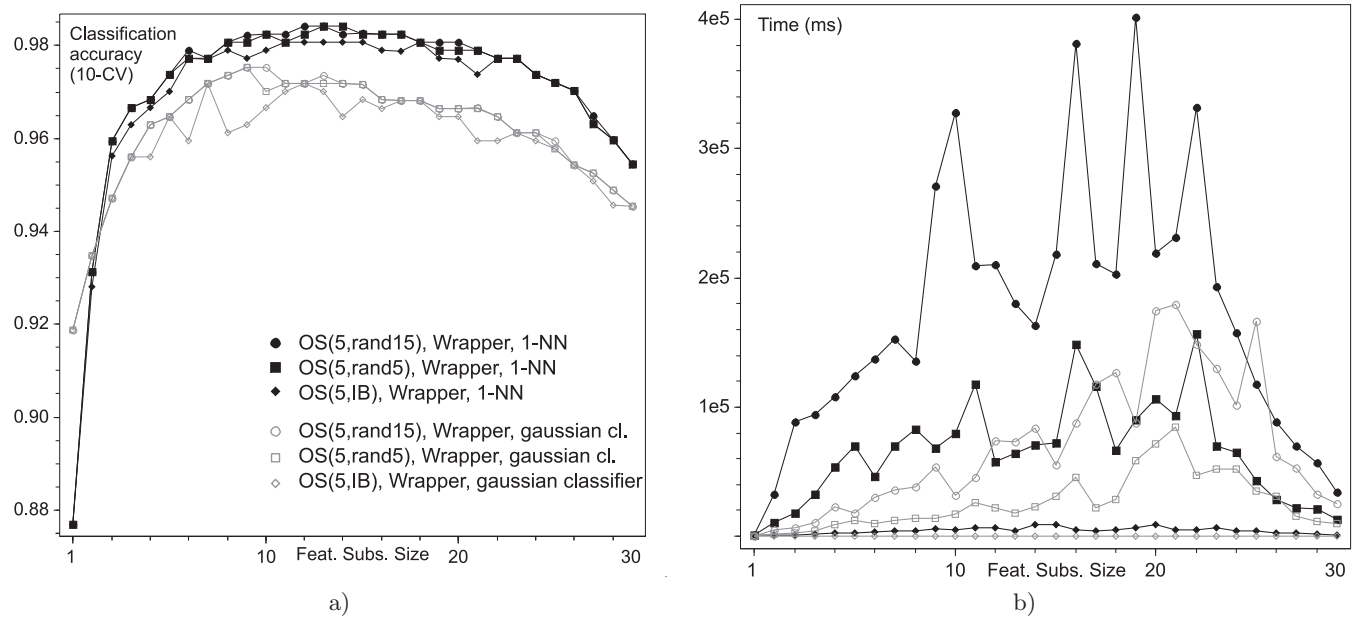

Figure 8: *Subset search methods performance in optimizing classifier accuracy on mammogram data*

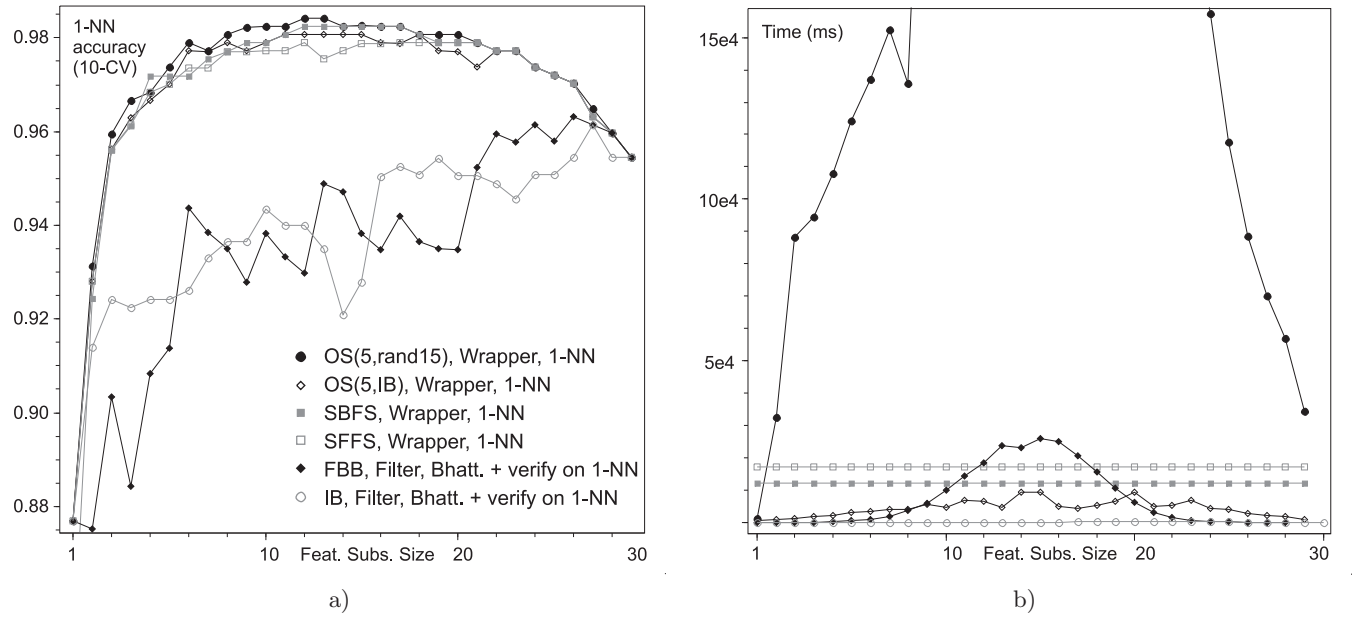

Figure 9: *1-Nearest Neighbour classifier performance optimized on mammogram data by means of Wrappers and Filters*

sifier performs best on ca. 9-dimensional subspace.

Figures 9 and 10 share one representative set of feature selection procedures used to optimize two different classifiers – 1-Nearest Neighbour in Figure 9 and gaussian classifier in Figure 10. The main observable points are in both cases: 1) very good performance/time-cost ratio of Floating Search in Wrapper setting is confirmed here, 2) the problem of often indirect (and thus insufficient) relation between probabilistic distance criteria and concrete classifiers is clearly visible – Filter based results tend to be inferior to those of Wrappers when assessed using concrete classifier accuracy.

In Figure 9 the Filters exhibit mediocre performance. Bhattacharyya distance clearly has very weak relation to 1-NN performance on this dataset. This is emphasised even more by the fact that Bhattacharyya optimization (optimal result yielded by Fast Branch & Bound vs. mere Individually Best feature ranking) does not lead to any observable improvement of 1-NN accuracy; moreover, its impact seems to be of almost random nature. Another

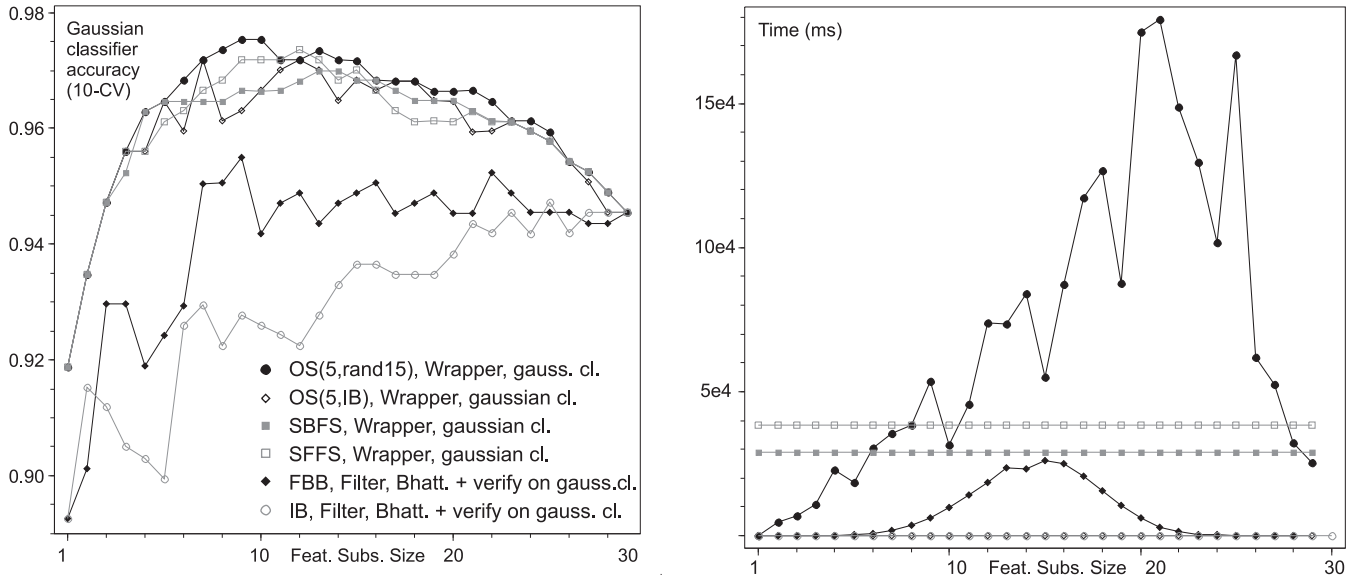

Figure 10: *Gaussian classifier performance optimized on mammogram data by means of Wrappers and Filters*

important observation is the Filter and Wrapper time cost. Wrappers are often considered unusable due to high time complexity. Here we can see that in many setups sub-optimal Wrappers are faster than optimal Filters. Certainly for the presented type of data the problem of Wrapper time complexity does not play a role.

In Figure 10 the superiority of Wrappers is confirmed. However, unlike in the preceeding case here Filter optimization brings visible improvement (compare FBB to IB). This is most likely due to the fact that gaussian classifier and Bhattacharyya distance criterion (here in normal form) are both based on the same assumption of the normality of data. The fact that the assumption is not true for this particular dataset results in mediocre overall gaussian classifier performance.

The graph of Filter results in Figure 11 illustrates mainly the power of advanced sub-optimal feature selection methods. In this case both the Floating Search and Oscillating Search methods in various settings yielded solutions equal or negligibly different from the optimum (verified by means of Fast Branch & Bound). However, it also illustrates the limits of Individual Best feature ranking. It is a well-known fact that two best features may not be equal to the best pair; this is well illustrated here. Note also the monotonicity of the evaluated Bhattacharyya distance criterion. On one hand it enables finding optimum by means of FBB, on the other hand it makes impossible to identify the best feature subset cardinality. Figure 11(b) shows the principal difference between optimal and sub-optimal methods regarding time complexity.

It should be stressed that for both SFFS and SBFS the speed advantage is considerably higher than it may seem from Figures 9(b), 10(b) and 11(b) – note that unlike other presented methods the SFFS and SBFS need only one run to obtain results for all subset sizes (its one time cost denoted by respective horizontal lines).

7.2 Mixture-based classification task example

Table 1 shows classification error rates achieved by the "Approximation" and "Divergence" mixture-based methods (see Section 6) with different number of components in comparison to gaussian classifier. All results were computed on the full set of features. In case of the "Approximation" and "Divergence" methods the algorithms were tested with two different initializations: random and "dogs & rabbits" cluster analysis [17] inspired by the self-organizing-map principle. Classifiers were trained on the first half of the dataset and tested on the second half (holdout method). Both mixture-based methods with the respective pseudo-Bayes classifiers were defined especially for use with multimodal data. Correspondingly, from table 1 it is possible to see that single component modeling may not be sufficient for real data, best results have been achieved with more than one component – with 5 mixture components (see column approx.5c) for *speech* data and 1, 5 or 20 components for *mammogram* data. The underlying data structure has been modeled precisely enough to achieve better classification rate when compared to the gaussian classifier.

Note that Table 1 also illustrates the problem of finding the suitable number of components (the issue is dis-

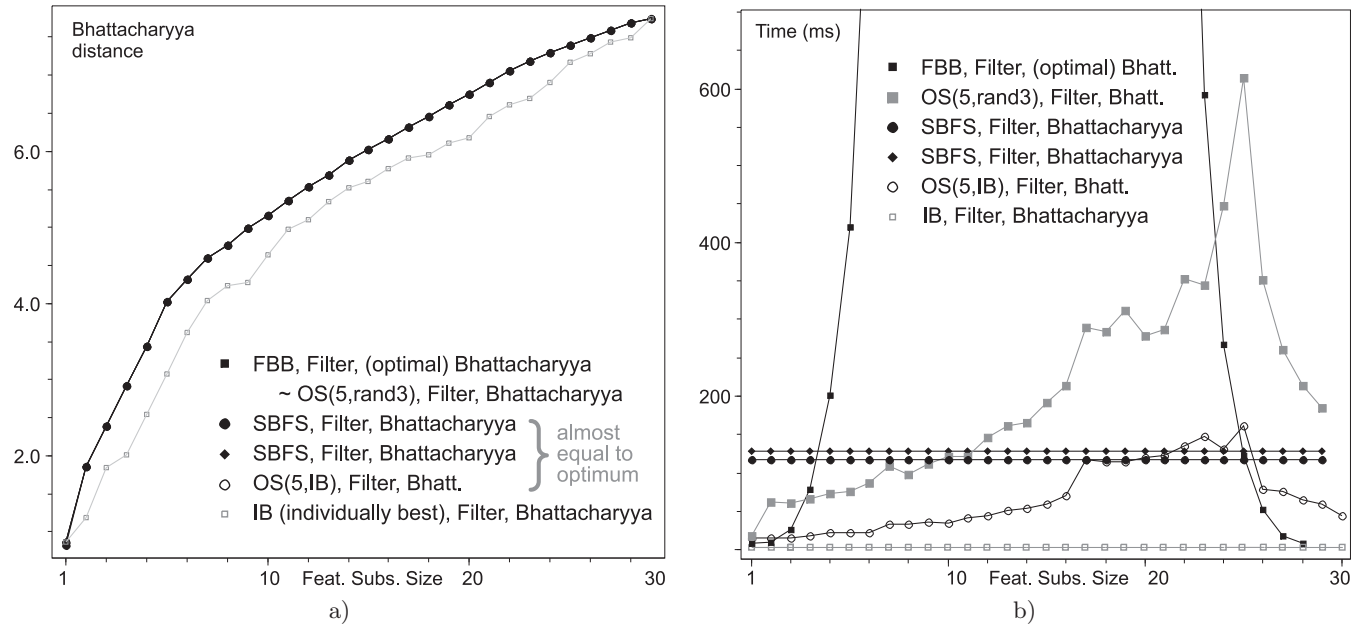

Figure 11: *Performance of recent optimal and sub-optimal Filter methods when maximizing Bhattacharyya distance on mammogram data*

		Gaussian classifier	Approx. 1 component	Approx. 5 components	Approx. 10 comp.	Approx. 20 comp.
speech data	(random initialization)	8.39	21.61	7.58	9.19	9.03
	(dogs & rabbits init.)	-	21.61	7.42	6.45	8.39
mammo data	(random initialization)	5.96	5.26	5.26	5.96	4.56
	(dogs & rabbits init.)	-	5.26	5.26	5.96	5.96

Table 1: *Error rates [%] of mixture-based classifiers with different parameters. Results were obtained using the "Approximation" mixture modeling method (in this case the alternative "Divergence" method yielded identical results).*

cussed, e.g., in Sardo [31]). Note that with the *mammogram* data about 20 components is needed to achieve notable improvement of classification performance. Compare the achieved classification rates to those in Figures 8, 9 and 10.

7.3 A different view to criterion functions – experimental comparison

An interesting problem may be to judge the importance of individual features in real classification tasks. Although in decision theory the importance of every feature may be evaluated, in practice 1) we usually lack enough information about the real underlying probabilistic structures and 2) analytical evaluation may become computationally too expensive. Therefore, many alternative evaluation approaches were introduced.

It is generally accepted that in order to obtain reasonable results, the particular feature evaluation criterion should relate to a particular classifier. From this point of view, we may expect at least slightly different behavior of the same features with different classifiers.

However, because of different reasons (performance and simplicity among others) classifier-independent criteria – typically probabilistic distance measures like Bhattacharyya etc. – have been defined to substitute for classifier accuracy evaluation (cf. Filters [12]). For a good overview and discussion of their properties, see Devijver and Kittler [3]. The "Approximation" and "Divergence" methods (cf. Section 6) also incorporate a feature evaluation function, which is closely related to their purpose.

In our example (Table 2) we demonstrate the differences of criterion functions implemented in the FST. We evaluated single features using different criteria and ordered them increasingly according to the obtained criterion values. In this way "more distinctive" features appear in the right part of the table, while the "noisy" ones should remain in the left.

A detailed discussion about the differences between different criteria behavior is beyond the scope of this paper. Let us point out some particular observations

Bhattacharyya	7	1	4	2	5	0	3	6	10	8	13	9	11	14	12
Divergence	7	1	4	2	0	5	6	3	10	8	13	9	11	12	14
G.Mahalanobis	7	1	4	5	2	3	6	8	0	13	10	11	9	14	12
Patrick Fisher	7	1	4	3	2	0	6	5	10	9	8	13	12	11	14
Gauss. cl. (10-f. CV)	12	14	11	9	13	0	2	8	6	1	3	4	7	10	5
1-NN (10-f. CV)	7	1	4	2	5	6	0	3	8	13	10	11	9	14	12
SVM RBF (10-f. CV)	7	1	2	5	4	6	3	8	0	13	10	9	11	14	12
approx.1c	7	1	4	2	0	5	6	3	10	8	13	9	11	12	14
approx.5c	0	13	1	4	12	7	10	3	2	5	9	14	11	6	8
approx.10c	0	13	1	12	4	7	2	10	3	14	5	9	6	8	11
approx.20c	0	12	13	1	4	7	10	2	3	14	9	5	11	6	8
diverg.1c	10	7	4	12	1	0	9	2	11	6	13	3	5	8	14
diverg.5c	5	12	8	1	0	7	6	2	4	9	10	13	3	11	14
diverg.10c	5	8	6	7	1	4	10	0	2	9	12	13	3	11	14
diverg.20c	1	6	5	8	2	10	7	3	11	9	12	0	14	13	4

Table 2: *Single features ordered increasingly according to individual criterion values (i.e., "individual discriminative power"), 2-class speech data*

only. Traditional distance measures (first four rows) gave comparable results, e.g. feature 14 has been evaluated as important, 7 or 1 as less important. Results of the "Divergence" method based evaluation remain relatively comparable, even if the result depends on the number of mixture components. More dissimilarities occurred in the "Approximation" method based evaluation which is caused by the different nature of approximation criterion which ranks the features not according to their suitability for classification, but for data representation in subspace only.

Our second example (Table 3) demonstrates criteria differences in another way. We selected subsets of 7 features out of 15 so as to maximize particular criteria to compare the differences between detected "optimal" subsets. Again, results given by traditional distance measures are comparable. Differences between subsets found by means of "Approximation" and "Divergence" methods illustrate their different purpose, although still many particular features are included in almost every found subset. Additionally, the "worst" subset, found to minimize the Bhattacharyya distance, is shown for illustration only.

Nevertheless, it should be stressed that by employing classifier-independent criteria one accepts certain simplification and possibly misleading assumption about data (e.g., most of probabilistic criteria are defined for unimodal normal distributions only).

7.4 A different view of criterion functions – visual subspace comparison

The FST may be used to obtain a visual projection of selected feature subsets. Our examples illustrate spatial properties of different data sets (easily separable 3-class *marble* set in Figure 12, a poorly separable 2-class *mammogram* set in Figure 13 and the *speech* set). We selected feature pairs yielding optimal values of different criteria. Figures 12(a)–(c) illustrate subsets obtained by means of optimizing different probabilistic distance measures, 12(d) illustrates the "Approximation" method (5 components), and 12(e) the "Divergence" method (5 components). As opposed to subsets selected for class discrimination Figure 12(f) illustrates an example of "bad" feature pairs unsuitable for discrimination, obtained by means of minimizing the Bhattacharyya distance.

8 Future Work and Applications

Results obtained using the F.S. Toolbox have been repeatedly used in our work for several research projects. Feature selection has been performed on different kinds of real world data. The kernel code is being flexibly altered for use in different situations (e.g., for comparison of statistical and genetic approaches to feature selection, see Mayer et al. [16]). F.S. Toolbox serves as a testing platform for development of new methods. Several directions of future development are possible. Undoubtedly, modification of the code to a parallel version would be beneficial. As far as the user interface is concerned, several improvements are possible. The same holds for the whole package which is built as open one with the intention to implement newly developed methods in future. In addition, for the future we plan to build a sort of expert or consulting system which would guide an inexperienced user toward using the method most convenient for the problem at hand.

opt. Bhattacharyya	-	-	-	-	-	-	6	7	8	-	10	11	12	-	14
opt. Divergence	-	-	-	-	-	-	6	7	8	9	10	11	-	-	14
opt. G.Mahalanobis	-	-	-	3	4	-	6	7	-	9	10	-	12	-	-
opt. Patrick Fisher	-	-	-	-	-	-	6	7	8	9	10	11	-	-	14
s-opt. Gauss.cl. 10CV	-	-	-	-	4	-	6	-	8	-	10	11	12	13	-
s-opt. 1-NN 10CV	0	-	-	3	4	-	6	-	-	9	10	-	12	-	-
s-opt. SVM RBF 10CV	-	-	-	-	4	-	-	7	8	9	10	11	12	-	-
approx.1c	-	-	-	-	-	-	-	-	8	9	10	11	12	13	14
approx.5c	-	-	2	-	-	5	6	-	8	9	-	11	-	-	14
approx.10c	-	-	-	3	-	5	6	-	8	9	10	11	-	-	-
approx.20c	-	-	-	3	-	5	6	-	8	9	10	11	-	-	-
diverg.1c	-	-	-	3	-	5	6	-	8	-	-	11	-	13	14
diverg.5c	-	-	-	3	4	-	-	-	-	9	10	11	-	13	14
diverg.10c	-	-	2	3	-	-	-	-	-	9	-	11	12	13	14
diverg.20c	0	-	-	-	4	-	-	-	-	9	-	11	12	13	14
worst Bhattacharyya	0	1	2	3	-	5	6	-	8	-	-	-	-	-	-

Table 3: *Selected subsets of 7 features, 2-class speech data*

9 Summary

The current state of art in feature selection based dimensionality reduction for decision problems of classification type has been overviewed. A number of recent feature subset search strategies has been reviewed and compared. Following the analysis of their respective advantages and shortcomings, the conditions under which certain strategies are more pertinent than others have been suggested.

Recent developments of B&B based algorithms for optimal search led to considerable improvements of the speed of search. Nevertheless, the principal exponential nature of optimal search remains and will remain one of key factors motivating the development of sub-optimal strategies. Among the family of sequential search algorithms the Floating and Oscillating search methods deserve particular attention. Two alternative feature selection methods based on mixture modeling have been presented. They are suitable for cases, when no *a priori* information on underlying probability structures is known. Many of recent feature selection methods have been implemented in Feature Selection Toolbox and discussed here in connection with real classification tasks. The software has been used to demostrate the differences between different criteria and differently selected feature subsets as well as other aspects of classification problems. The importance of feature selection for classification performance has been clearly demonstrated.

10 Acknowledgements

The work has been supported by grants of the Czech Ministry of Education 1M0572 DAR and 2C06019, EC project No. FP6-507752 MUSCLE, Grant Agency of the Academy of Sciences of the Czech Republic No. A2075302 and the Grant Agency of the Czech Republic No. 402/03/1310.

References

[1] Blum A. and Langley P., "Selection of Relevant Features and Examples in Machine Learning", Artificial Intelligence, 97(1-2), 245–271, 1997.

[2] Cover T.M. and Van Campenhout J.M., "On the possible orderings in the measurement selection problem", IEEE Transactions on System, Man and Cybernetics, SMC-**7**:657–661, 1977.

[3] Devijver P.A. and Kittler J., "Pattern Recognition: A Statistical Approach", Prentice-Hall, 1982.

[4] Duda R.O., Hart P.E. and Stork D.G., "Pattern Classification, 2nd Ed.", Wiley-Interscience, 2000.

[5] Ferri F.J., Pudil P., Hatef M., Kittler J., "Comparative Study of Techniques for Large-Scale Feature Selection", Gelsema E.S., Kanal L.N. (eds.) Pattern Recognition in Practice IV, Elsevier Science B.V., 403–413, 1994.

[6] Fukunaga K., "Introduction to Statistical Pattern Recognition", Academic Press, 1990.

[7] Guyon I. and Elisseeff A., "An introduction to variable and feature selection", Journal of Machine Learning Research 3:1157–1182, 2003.

[8] Hastie T. and Tibshirani R., "Discriminant analysis by Gaussian mixtures", Journal Royal Statist. Soc. B, Vol. 58:155–176, 1996.

[9] Chen X., "An Improved Branch and Bound Algorithm for Feature Selection", Pattern Recognition Letters 24(12):1925–1933, 2003.

[10] Jain A.K. and Zongker D., "Feature Selection: Evaluation, Application and Small Sample Performance", IEEE Transactions on Pattern Analysis and Machine Intelligence 19(2):153–158, 1997.

[11] Jain A.K., Duin R.P.W. and Mao J., "Statistical Pattern Recognition: A Review", IEEE Transactions on Pattern Analysis and Machine Intelligence 22(1):4–37, 2000.

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

[13] Kudo M. and Sklansky J., "Comparison of Algorithms that

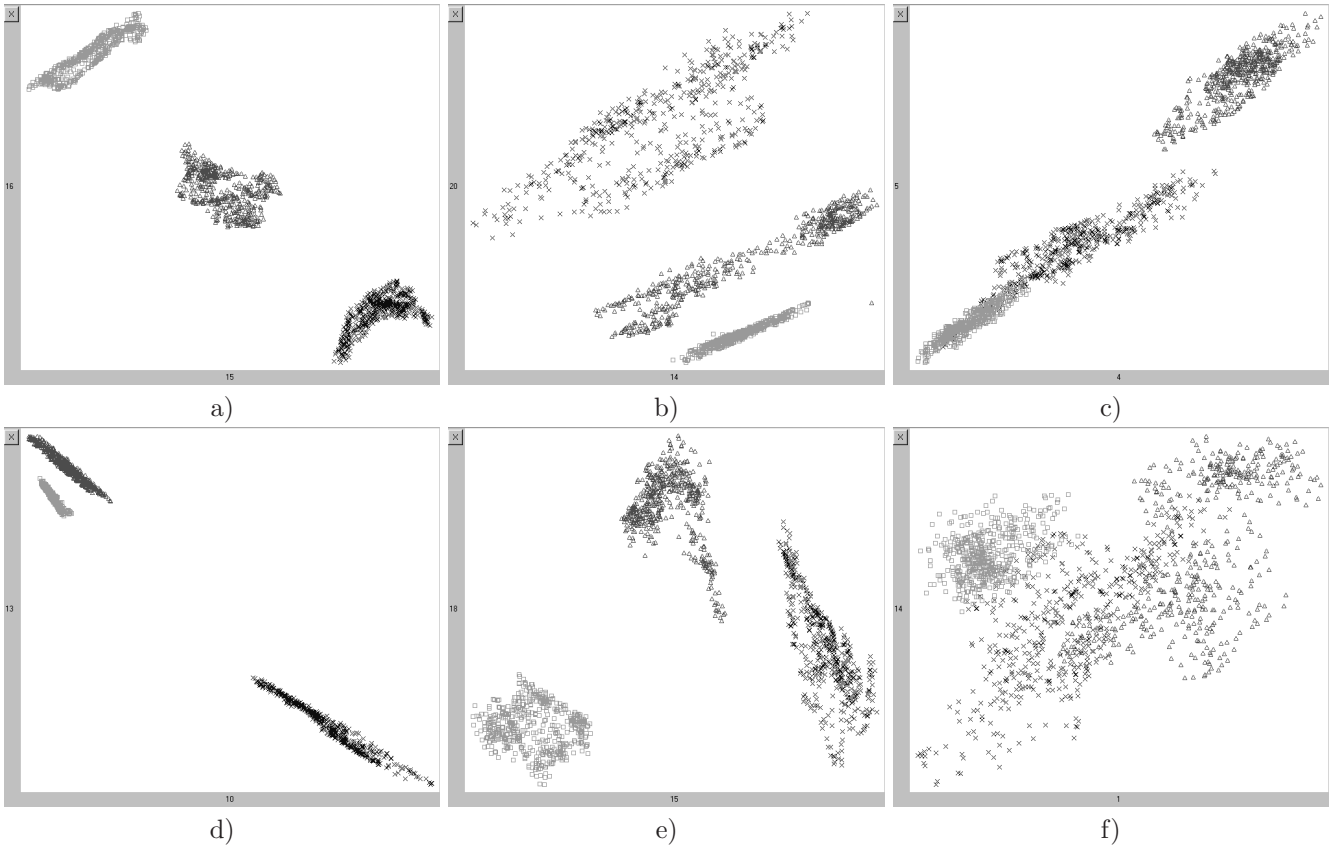

Figure 12: *Visual comparison of 2D subspaces found on marble data by maximizing: a) Bhattacharyya (the same was found by Generalized Mahalanobis), b) Divergence, c) Patrick-Fischer distances. Mixture model methods using 5 components results: "Approximation" method - d), "Divergence" method - e). Picture f) demonstrates a subspace unsuitable for discrimination (found by minimizing the Bhattacharyya distance).*

Select Features for Pattern Classifiers", Pattern Recognition 33(1):25–41, 2000.

[14] Liu H. and Yu L., "Toward Integrating Feature Selection Algorithms for Classification and Clustering", IEEE Transactions on Knowledge and Data Engineering 17(4):491–502, 2005.

[15] Loughrey J. and Cunningham P., "Overfitting in Wrapper-Based Feature Subset Selection: The Harder You Try the Worse it Gets", 24th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, (AI- 2004) Bramer, M., Coenen, F., T. Allen, 33–43, Springer.

[16] Mayer H.A., Somol P., Huber R. and Pudil P., "Improving Statistical Measures of Feature Subsets by Conventional and Evolutionary Approaches", Proc. 3rd IAPR International Workshop on Statistical Techniques in Pattern Recognition (Alicante), 77–81, 2000.

[17] McKenzie P. and Alder M., "Initializing the EM algorithm for use in gaussian mixture modelling", Technical report, University of Western Australia, 1994.

[18] McLachlan G.J., "Discriminant Analysis and Statistical Pattern Recognition", John Wiley & Sons, New York, 1992.

[19] Murphy P.M. and Aha D.W., "UCI Repository of Machine Learning Databases [ftp.ics.uci.edu]", University of California, Department of Information and Computer Science, Irvine, CA, 1994.

[20] Narendra P.M. and Fukunaga K., "A Branch and Bound Algorithm for Feature Subset Selection", IEEE Transactions on Computers 26:917–922, 1977.

[21] Novovičová J., Pudil P. and Kittler J., "Divergence Based Feature Selection for Multimodal Class Densities", IEEE Transactions on Pattern Analysis and Machine Intelligence 18(2):218–223, 1996.

[22] Novovičová J. and Pudil P., "Feature selection and classification by modified model with latent structure", in: Dealing With Complexity: Neural Network Approach, Springer Verlag, 126–140, 1997.

[23] Palm H.Ch., "A new methods for generating statistical classifiers assuming linear mixtures of Gaussian densities", in: Proc. 12th International Conference on Pattern Recognition, Jerusalem, 483–486, 1994.

[24] Pudil P., Novovičová J. and Kittler J. "Floating Search Methods in Feature Selection", Pattern Recognition Letters 15(11):1119–1125, 1994.

[25] Pudil P., Novovičová J. and Kittler J., "Feature selection based on approximation of class densities by finite mixtures of special type", Pattern Recognition, 28:1389–1398, 1995.

[26] Pudil P., Novovičová J., Kittler J., "Simultaneous Learning of Decision Rules and Important Attributes for Classification Problems in Image Analysis", Image and Vision Computing 12:193–198, 1994.

[27] Pudil P. and Novovičová J., "Novel Methods for Subset Selection with Respect to Problem Knowledge", IEEE Transactions on Intelligent Systems – Special Issue on Feature Transformation and Subset Selection, 66–74, 1998.

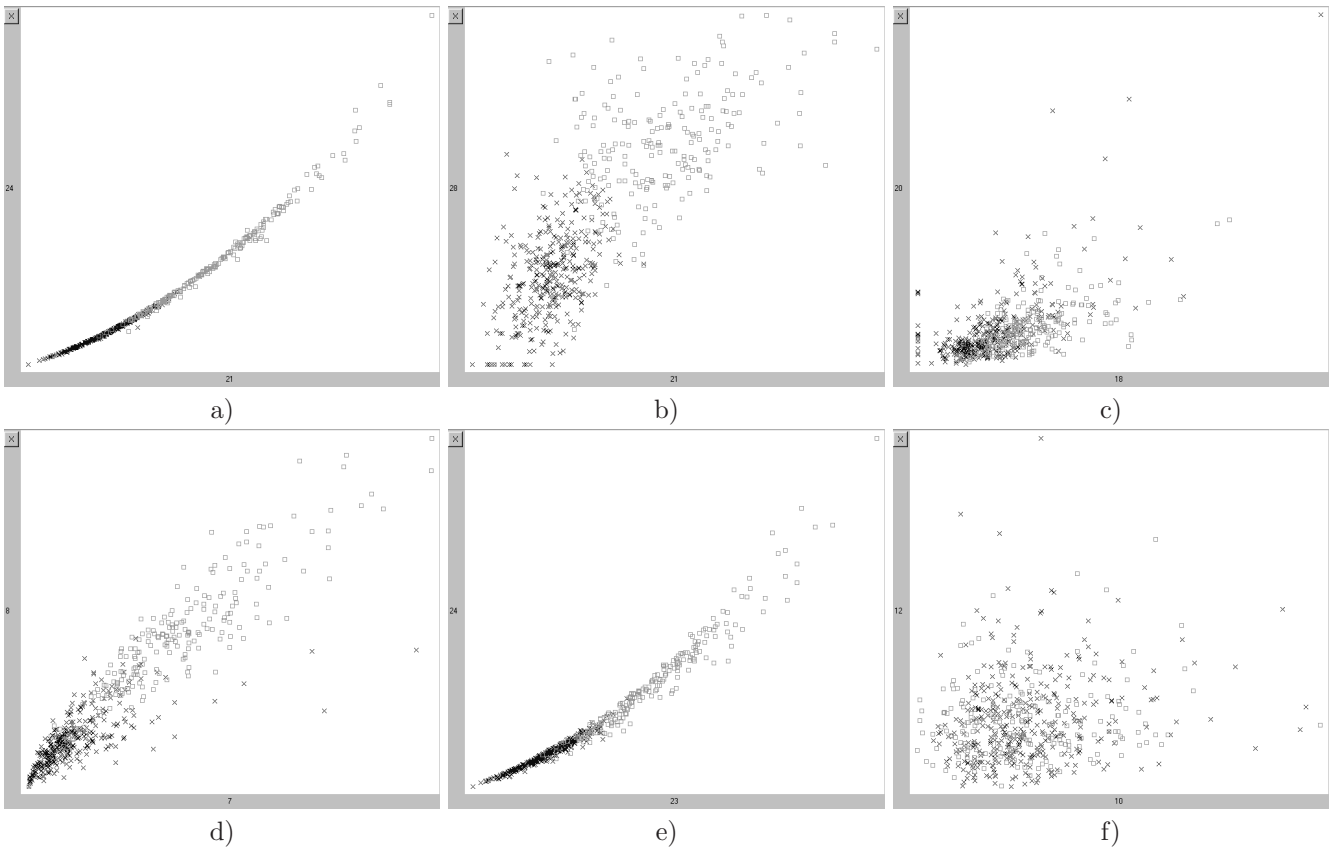

Figure 13: *Visual comparison of 2D subspaces found on less separable* mammogram *data by maximizing: a) Bhattacharyya (the same was found by Divergence), b) Generalized Mahalanobis, c) Patrick-Fischer distances. Mixture model methods using 5 components results: "Approximation" method - d), "Divergence" method - e). Picture f) demonstrates a subspace unsuitable for discrimination (found by minimizing the Bhattacharyya distance).*

[28] Raudys S., "Feature Over-Selection", Lecture Notes in Computer Science LNCS 4109, Springer, 622–631, 2006.

[29] Ripley B., "Pattern Recognition and Neural Networks", Cambridge University Press, Cambridge, Massachusetts, 1996.

[30] Salappa A., Doumpos M. and Zopounidis C., "Feature selection algorithms in classification problems: an experimental evaluation", Optimization Methods and Software 22(1):199-214, 2007.

[31] Sardo L. and Kittler J., "Model Complexity Validation for PDF Estimation Using Gaussian Mixtures", Proc. 14th International Conference on Pattern Recognition, Brisbane, Vol. 2, 195–197, 1998.

[32] Siedlecki W., Sklansky J., "On Automatic Feature Selection", International Journal of Pattern Recognition and Artificial Intelligence 2(2):197–220, 1988.

[33] Somol P., Pudil P., Novovičová J. and Paclík P., "Adaptive Floating Search Methods in Feature Selection", Pattern Recognition Letters 20(11,12,13):1157–1163, 1999.

[34] Somol P. and Pudil P., "Oscillating Search Algorithms For Feature Selection", Proc. 15th IAPR International Conference on Pattern Recognition, Barcelona, Spain, 406–409, 2000.

[35] Somol P., Pudil P. and Grim J., "Branch & bound algorithm with partial prediction for use with recursive and non-recursive criterion forms", Lecture Notes in Computer Science LNCS 2013, Springer, 230–239, 2001.

[36] Somol P. and Pudil P., "Feature Selection Toolbox", Pattern Recognition 35(12):2749–2759, 2002.

[37] Somol P., Pudil P. and Kittler J., "Fast Branch & Bound Algorithms for Optimal Feature Selection", IEEE Transactions on Pattern Analysis and Machine Intelligence 26(7):900–912, 2004.

[38] Somol P., Pudil P. and Grim J., "On Prediction Mechanisms in Fast Branch & Bound Algorithms", Lecture Notes in Computer Science 3138, Springer, Berlin, 716–724, 2004.

[39] Theodoridis S. and Koutroumbas K., "Pattern Recognition, 2nd Ed.", Academic Press, 2003.

[40] Tsamardinos I. and Aliferis C., "Towards Principled Feature Selection: Relevancy, Filters, and Wrappers", Artificial Intelligence and Statistics, 2003.

[41] Vafaie H. and Imam I., "Feature Selection Methods: Genetic Algorithms vs. Greedy-like Search", In: Proceedings of the International Conference on Fuzzy and Intelligent Control Systems, 1994.

[42] Wang Z., Yang J. and Li G., "An Improved Branch & Bound Algorithm in Feature Selection", Lecture Notes in Computer Science LNCS 2639, Springer, 549–556, 2003.

[43] Webb A., "Statistical Pattern Recognition, 2nd Ed.", John Wiley & Sons, 2002.

[44] Yang J. and Honavar V., "Feature Subset Selection Using a Genetic Algorithm", IEEE Intelligent Systems 13:44–49, 1998.

[45] Yu B. and Yuan B., "A More Efficient Branch and Bound Algorithm for Feature Selection", Pattern Recognition 26:883–889, 1993.

Pavel PUDIL is a Professor of Technical Cybernetics. He received the M.Sc. in Control Engineering, 1964, Ph.D. in Cybernetics 1970, and Sc.D. and Professor in 2001 from the Czech Technical University). He is currently the Dean of the Faculty of Management, University of Economics Prague. In 1994 - 2000 he was the Chairman of IAPR (International Association for Pattern Recognition) TC on "Statistical Techniques in Pattern Recognition". His research interests include statistical approach to pattern recognition, particularly the problem of dimensionality reduction, feature selection and its applications in economics, management and medical diagnostics. He is author/co-author of about 150 research papers published in scientific journals and conference proceedings. He is a member of IEEE and the IAPR Fellow.

Petr SOMOL received his B.S., M.S., and Ph.D. degrees in 1993, 1995 and 2000, respectively, from the Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic, all in computer science. He is currently with the Department of Pattern Recognition at the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic. He spent two years at the Medical Informatics Unit, IPH, University of Cambridge, UK. His current activities include development of feature selection techniques for statistical pattern recognition where he is the author/co-author of about 60 research papers published in scientific journals and conference proceedings. He is also the author of an extensive software package Feature Selection Toolbox.

Rudolf STŘÍTECKÝ received his M.D.degree from the Medical Faculty in Brno, Czech Republic. He is the Head of Pediatric Department at the district hospital at Jindrichuv Hradec and also the Head of Institute of Health Care Management at the Faculty of Management, Prague University of Economics. His current activities include management of quality and economics of health care delivery, from which area he published several papers.