Identifying the most informative variables for decision-making problems – a survey of recent approaches and accompanying problems[#]

Pavel Pudil[*] *– Petr Somol*[**]

In the following we give an overview of problems related to variable selection (also known as feature selection) techniques in decision-making problems based on machine learning with particular emphasis to recent knowledge. Several popular methods will be reviewed and assigned to a taxonomical context. Issues related to the generalization versus performance trade-off inherent to currently used variable selection approaches will be addressed and illustrated on real-world examples.

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 variable (feature) selection or more generally of dimensionality reduction, which can be accomplished either by a linear or nonlinear mapping from the measurement space to a lower dimensional feature space. The main aspects of the problem, i.e., criteria for evaluating variables and the associated optimization techniques will be discussed.

The objective of the paper is to demonstrate the necessity of selecting the most informative variables in order to improve the quality of decision-making based on the learning approach. We will examine some of the most popular tools under various settings to point out several pitfalls often omitted in current literature. Note: in the following we will prefer the term *feature selection* to variable selection in accordance with *statistical pattern recognition* conventions.

Common research issues in management and medicine

Though managers, economists and physicians have different priorities in research issues, there exist issues common to both the fields. Such an issue is the problem of selecting only that information which is necessary (and if possible also sufficient) for decision making. Since not only mathematicians and physicians „speak different language“, but often also even managers, economists and physicians, the same problem appears different for all of them. Somebody from outside of both the communities, capable of abstraction and involved directly neither in management or medicine, is needed to find formal similarities of the problems. We

[#] Paper is prepared as one of the outputs of research project „ZIMOLEZ“ supported by the grant of MŠMT No. 2C06019 and the project „DAR“ supported by the grant MŠMT No. 1M0572.

[*] Prof. Ing. Pavel Pudil, Dr.Sc., FM VŠE, ÚTIA AV ČR.

[**] RNDr. Petr Somol, Ph.D., ÚTIA AV ČR, FM VŠE.

strongly believe that statistical pattern recognition is the discipline capable to provide a common methodology.

A typical problem which both managers and physicians often encounter is the problem of too many potential inputs into their respective decision-making problems. This phenomenon has been extensively studied in mathematics and in artificial intelligence. The "curse of dimensionality" problem, as called by a famous American mathematician Richard Bellman, can be found perhaps in all the fields of science and application areas including economics, management and even medicine and medical care. Without going into the details of this phenomenon, it can be stated that in order to make reliable decisions (or more exactly to learn to make them based on the past experience and the available data) the need for the amount of data dramatically grows with the number of inputs. Mathematically it means that the sample size required grows exponentially with the data dimensionality. This problem is very relevant particularly to the field of medicine and economics as the process of medical or economic data acquisition is usually both time consuming and costly. Consequently, the data sets acquired are usually too small with respect to their dimensionality.

Though managers and clinical physicians consider their respective problems to be of a very different nature (perhaps justly from their professional point of view), from the point of view of mathematics a formally the same problem exists. It is just the question of different terminology and abstraction needed to find a unified look at the problem. Let us just give an example what we mean by considering which sets of inputs managers and physicians use for their decision-making:

1. *Managers for managerial decision-making*: economic indices, financial data, time series, prediction estimates, etc.
2. *Physicians for clinical decision-making*: symptoms, anamnestic data, results of biochemical analyses, etc.

For a mathematician, however, both the sets can be looked upon as the set of variables, forming the input vector (in pattern recognition which deals with this problem the term feature vector is used). In the majority of practical cases, the dimensionality (the number of inputs) of the original input space can be rather high. It is just natural consequence of the well known fact that in the design phase of any system for the support of decision-making it is extremely difficult or practically impossible to evaluate directly the "usefulness" of particular input variables.

Both managers and physicians certainly faced this problem many times, even though not when designing any support system, but just when having to make the decision. A manager could have for instance a large number of economic variables to potentially consider for performing a multiple regression analysis (too many potential regressors). On the other hand, a physician could have (or arrange to have) a large number of analyses results to base the decision on. Yet, in both the cases, owing to often very complex relations and dependencies (sometimes rather strong) among all the respective inputs, there exist either economic variables or clinical analyses results (let us speak generally just about variables) which can be left out from decision-making without a great loss of information content. The theory of information, a special mathematical discipline, defines this as the existence of „redundancy“ in the set of variables. However, even if physicians or managers would be aware of this phenomenon, the problem of solving the task to find the redundant variables for complex problems with many potential inputs is beyond human capabilities.

The reasons for trying to reduce the set of our inputs into the decision-making process by eliminating redundancies have both practical and theoretical foundations. The practical ones perhaps need not be discussed - the cost reduction in the data acquisition is sufficient to substantiate the reduction. On the theoretical front we should like to mention a famous theorem by another American mathematician S. Watanabe. He has founded the mathematical theory of cognition and by paraphrasing a world-known fairytale of Norwegian author Hans Andersen, he formulated the „Theorem of ugly ducking". It states, roughly said, that no cognition is possible unless our perceptions (input variables) are weighted and, consequently, many of them are given a null weight and thus eliminated from the cognition process.

Each of the considered fields (managerial and clinical decision-making) has its own specificity and accordingly different ways of treating the problem. On the other hand, with the ever increasing specialization and diversification of scientific disciplines, it is not uncommon fact that similar problems are being tackled in other branches of science, usually without awareness of respective research and application communities. Yet the results and methods from one scientific discipline can be applied not only to solve problems in another quite different discipline, but they can also often enrich its methodology.

It is our belief that the novel methods developed recently in the field of statistical pattern recognition to solve the problem of feature selection can enrich the methodology of selecting the most useful information used in other application areas.

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 **x** is to be classified into one of a finite set of *C* different classes $\Omega = \{\omega_1, \omega_2,\ldots, \omega_C\}$. A pattern **x** belonging to class ω_i is viewed as an observation of a random vector **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 include 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 $d<<D$), so as to maximize (or minimize) an adopted criterion.

Dimensionality reduction can have different forms 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 non-linear combinations of the original features. Technically FS is a 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., problems of automated credit-scoring or 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.

Dimensionality reduction may follow different aims:

1. *dimensionality reduction for optimal data representation*
2. *dimensionality reduction for classification*

The first aim is to preserve the topological structure of data in a lower-dimensional space as much as possible, the second aim is to enhance the subset discriminatory power. In the sequel we shall concentrate on the *feature selection* problem aimed at *classification* problems only. For a broader overview of the subject see, e.g., [5], [20], [28], [37], [40].

Feature selection

Given a set of D features, X_D, let us denote Ξ_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 $X_d^\bullet$ for which $J(X_d^\bullet)=\max_{\{X_d \in \Xi_d\}} J(X_d)$. 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.

Feature selection criterion functions

The ideal criterion of feature set effectiveness is classification error. Kohavi [13] introduced a practically important distinction between two principally different approaches to FS according to the feature subset evaluation approach (cf. also [38]), differing in the way how the classification error is estimated:

Wrappers – In *wrappers* the features are selected with respect to a chosen decision rule (classifier). Classification performance is then used directly to evaluate feature subset and to direct the feature selection process further. *Wrappers* are often the preferred approach of choice due to their often higher achieved classification accuracy, although the feature selection process can be very slow. Features selected using *wrappers* are unsuitable for any other classifier, than the one used in the FS process. Also, *wrappers* are often worse than *filters* regarding the ability to generalize (*wrappers* may over-train, i.e., select features too specific for training data classification while the performance on independent data deteriorates).

Filters – In *filters* the criterion function used for feature subset evaluation aims at characterizing more general properties of features. Features are selected without respect to the concrete context in which they would be later used. Typically features are evaluated using probabilistic distance functions, probabilistic dependence functions or functions evaluating entropy of the system etc. Often some assumptions are accepted to enable the use of such

functions – many of them are defined for normal distributions only. They often exhibit monotonic behavior. *Filters* are fast, but often yield unsatisfactory results in concrete machine learning applications, as shown later in this paper. On the other hand, they may be less prone to over-fitting.

Currently it is generally agreed that if practical circumstances allow the use of *wrappers*, they should be preferred. However, *wrappers* often cannot be used in practice due to computational complexity or other problems and we have to resort to other measures of feature set goodness, i.e., to *filters*. In the following we focus on the reasoning behind *filter* feature selection criteria.

The ability to classify patterns by machine learning relies on the implied assumption that classes occupy distinct regions in the pattern (feature) space. Intuitively, the more distant the classes are from each other, the better the chances of successful recognition of class membership of patterns. It is reasonable, therefore, to select as the feature space that subspace of the pattern representation space, in which the classes are maximally separated. Various measures have been defined to express in some way this quality of feature subsets. With many such measures simplifying assumptions have to be made about the data to enable measure evaluation. This can be illustrated well on one of the basic measures – the Mahalanobis distance. Mahalanobis distance can be easily evaluated in case the data is normal – therefore the normality assumption is usually implied on the data to enable the use of this distance measure. The principle of the measure can be simply described as follows – it increases with increasing the distance of class mean values and with decreasing the class variance. Many feature selection criteria to evaluate inherent data properties without the need to evaluate classification error have been defined in a similar way – a simplified overview of their framework can be given as follows (for details see [4][5][8][12]):

- *probabilistic distance measures* – Mahalanobis distance, generalized Mahalanobis distance, Bhattacharyya distance, the divergence, Patrick-Fischer distance etc.
- *probabilistic dependence measures* – Mahalanobis dependence, Bhattacharyya dependence, Joshi dependence, Patrick-Fischer dependence, mutual information etc.
- *entropy measures* – Shannon entropy, quadratic entropy etc.
- *interclass distance measures* – linear, non-linear, based on various metrics

In recent years due to the rapidly increasing speed of computers it becomes more feasible to evaluate the classification accuracy directly as a part of the feature selection process. Although the probabilistic criteria mentioned above still retain some advantages (speed, possibly better generalization ability / lower tendency to over-fit), it has been generally agreed that using concrete classifier accuracy in place of FS criterion function, i.e., using *wrappers*, makes it possible to achieve better performance in most practical applications. Yet this approach suffers drawbacks as well – the tendency to over-train is among the most problematic. Accordingly, one of the open problems of current research is the question of how closely the probabilistic measures for use in *filters* actually relate to the expected classification error of the pattern recognition system. Finding better criteria with better properties is probably one of the most difficult and urgent, yet unsolved problems of the field.

The problem of FS method choice – which methods are suitable for what ?

We have shown that there exists a wide variety of criteria that can be used to assess feature subsets. With subset search algorithms the situation is similar. There exists a broad framework of various algorithms capable of accomplishing the same feature selection task.

Question can be raised why do we need many various methods, why don't we just choose one universal method for all problems? Unfortunately, this is a fundamental problem – it is generally agreed that *there exists no unique generally applicable approach* to the feature selection problem. Some search methods may show to be better than others, but only in specific data and classifier set-ups. Under different circumstances the opposite may be true. The suitability of particular FS method for particular task may depend on problem dimensionality, number of classes, training set size, missing values, type of values (nominal, continuous) and other inherent data properties as well as on properties of the classifier for which the features are selected.

Due to this broadness of FS method framework we can only provide in the following an overview of some of the most popular methods and method families and to discuss briefly their basic properties. Some of the methods will be examined on real world data in the experimental section. Before focusing on concrete methods, let us first categorize FS methods into basic families:

FS method categorization with respect to optimality

Optimal methods – methods yielding solutions optimal with respect to the chosen criterion. These include *exhaustive search* (feasible for only small size problems) and accelerated methods, mostly built upon the *branch & bound* principle. The well known *branch & bound* [22] [35] 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 used feature selection criterion J(.) satisfies the *monotonicity* property - given two subsets of the feature set X_D, A and B such that $A \subseteq B$, the following must hold: $A \subseteq B \Rightarrow J(A) \leq J(B)$. That is, evaluating the feature selection criterion on a subset of features of a given set yields a smaller or equal value of the feature selection criterion. Note: This assumption precludes the use of classifier error rate as the criterion (cf. *wrappers*, see above). This is an important drawback as the error rate can be considered superior to other criteria [30], [13], [38]. Moreover, all optimal algorithms including B&B become computationally prohibitive for problems of mid- and high dimensionality. The exponential nature of all optimal algorithms can not be suppressed sufficiently enough for most of real-world problems. The problem of optimal feature selection (or more generally of subset selection) is thus difficult especially because of its time complexity. Therefore the preferred approach is to trade optimality for speed and resort to sub-optimal methods only.

Sub-optimal methods – essentially trade the guarantee of optimality of the selected subset for computational efficiency. A comprehensive list of sub-optimal procedures can be found, e.g., in books [4], [8], [37], [40]. A comparative taxonomy can be found, e.g., in [2], [6], [9], [11], [12], [15], [16], [29], [39] or [42]. Because sub-optimal methods have been found considerably more useful for practical purposes, we focus on them in more detail in the following.

FS method categorization 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 (see above) may be appropriate as the feature subset evaluation criterion. For this type of situations one of the optimal or sub-optimal sequential search methods is appropriate. Many alternative or task-specific methods also exist for use with particular types of problems [7][10][14].

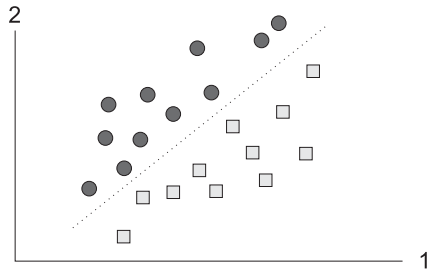
No a priori knowledge is available – we cannot even assume that pdfs are unimodal. The only source of available information is the training data. For these situations we have developed two alternative methods. Both are based on approximating unknown conditional pdfs by finite mixtures of a special type with feature selection being inherited in the mixture parameters estimation process [23] [24].

Evolution of sub-optimal search methods

Despite recent advances in optimal search, for larger than moderate-sized problems we have to resort to sub-optimal FS methods. A broad range of sub-optimal feature selection methods is currently available, based on various assumptions and adopting various search algorithms.

The simplest choice is to evaluate each feature individually and eventually choose the features yielding the highest criterion value. This so-called *individually best* (IB) search is widely used in problems of prohibiting dimensionality (e.g., in text categorization or gene analysis with thousands of features) due to search time constraints. However, it completely ignores inter-feature relations and as such can not reveal solutions, where combinations of features are needed (see Figure 1).

Figure 1: In this 2D case neither feature 1 nor 2 is sufficient to distinguish patterns from classes of rectangles and circles. Only when information from both features is combined, classes can be separated (dotted line)



When feature relations are to be taken into account, 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 widely used choice, the *Sequential Forward* [41] (or *Backward* [18]) *Selection* methods – SFS (SBS) – iteratively add (remove) one feature at a time so as to maximize the intermediate criterion value until the required dimensionality is achieved. 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*-Minus-*r,* also known as "(l,r)" or "+L-R" [36] which involves successive augmentation and depletion process, or generalized algorithms [4]. Among the more recent approaches the following two families of methods can

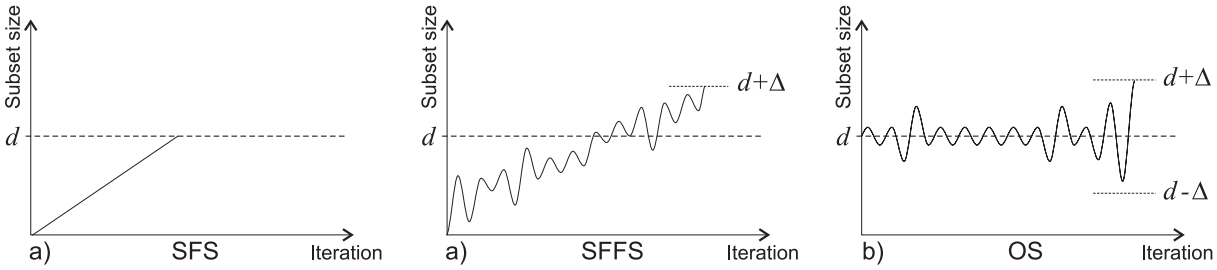
be pointed out for general applicability and performance reasons: *Sequential Forward* (or *Backward*) *Floating Search* methods SFFS, SBFS, [25], and *Oscillating Search* (OS) methods [33]. An overview of the evolution of sequential search methods is given in Table 1.

Tab. 1: Evolution of sequential search methods

Method (Simplest first)	Properties / Improvement over previous method
Individually Best (IB)	Evaluate each variable separately, completely ignore variable relations
SFS / SBS (Sequential Selection)	Sequentially build subset, in each step with respect to the currently included features
GSFS / GSBS (Generalized Seq. Sel.)	As SFS/SBS, but in each step evaluate groups of features instead of single features to reveal more complicated dependencies
Plus-*l*-Minus-*r*	Prevent "nesting": alternate the adding and removing of one feature based of parameters L and R
GPlus-*l*-Minus-*r* (Generalized P-*l*-M-*r*)	Same as Plus-*l*-Minus-*r*, but in each step evaluate groups of features instead of single features to reveal more complicated dependencies
SFFS / SBFS (Floating Search)	Automatically determine the sequence of additions and removals – to avoid user parameters and improve search effectivity
GSFFS / GSBFS (Generalized Float. S.)	As SFFS/SBSF, but in each step evaluate groups of features instead of single features to reveal more complicated dependencies
ASFFS / ASBFS (Adaptive Float. Search)	Automatically adjust the size of feature groups evaluated in each step to better focus on desired dimensionality
OS (Oscillating Search)	Focus straight on the desired dimensionality + enable greater flexibility: optional randomized search, result tuning, time-constrained search etc.

Note: if the methods in Table 1 are used as *filters*, the hierarchy corresponds reasonably well with the ability to find solutions closer to the optimum with respect to the chosen evaluation function (IB – often the worst, OS – often the best). However, this is often insufficient to enable construction of good decision rules, because better criterion values often coincide with worse generalization, i.e., resulting classification performance on new, previously unseen data. Tests on independent data show, that any method regardless its principle may become the best for some problem settings. The negative implication is that it is difficult to give any universal recommendation about which method to choose. According to long-term experiments and independent studies (see, e.g., [15]) we can state that it is the *floating search* that most often offers the best compromise between performance and generalization.

Figure 2: Comparing the course of search (current subset size depending on time) in standard sequential search methods



Floating search methods

The *Sequential Forward Floating Selection* (SFFS) [25] procedure consists of applying after each forward step (adding the feature that maximizes the criterion the most) a number of backward steps (removing the feature, that causes the least criterion decrease) 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 (unlike Plus-*l*-Minus-*r*). Formal description of this now classical procedure can be found in [25]. The *floating* course of search is illustrated and compared to other approaches in Figure 2.

Floating search algorithms have been critically acclaimed as universal tools not only outperforming all predecessors, but also keeping advantages not met by more sophisticated algorithms (e.g., cf. [15]). They find good solutions in all problem dimensions in one run. The overall search speed is high enough for most of practical problems. Recent experiments show that the floating search principle overcomes exceptionally well the optimization performance vs. generalization trade-off problem (see experiments below). The idea of floating search was later extended in the *adaptive floating search* algorithms [32].

Oscillating search method

The *Oscillating Search* (OS) [33] can be considered a „higher-level" procedure, that takes use of other feature selection methods as sub-procedures within the main course of search. The concept is highly flexible and enables modifications for different purposes. 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 pre-dominant search direction. This is achieved by alternating so-called *down*- and *up*-swings. Both swings attempt to improve the current set X_d by replacing some of the features by better ones. The *down*-swing first removes worst feature(s), then adds back best ones, while *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 of specified depth (number of features to be replaced in one swing). For details see [33]. The course of OS search is compared to SFFS and SFS in Figure 2.

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 to accomplish the *up*- and *down*-swings. This makes the OS more a framework than a single procedure. The OS can thus be adjusted for extremely fast search (low cycle depth limit, random initialization) in problems of high dimensionality or very intensive search aimed at achieving highest possible criterion values (high cycle depth limit, repeated runs from different random starting points to avoid local extremes, or other complex initialization). The OS can be used to tune solutions obtained elsewhere.

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 run. This behavior is advantageous, because it gives the option of stopping the search after a while without serious result-degrading consequences (OS is thus usable in real-time systems). Moreover, because the OS processes subsets of target cardinality from the very beginning, it may find solutions even in cases, where standard sequential procedures fail due to numerical problems.

Non-sequential and alternative methods

In addition to sequential search methods a broad range of alternative approaches to feature selection is available, often having properties targeted at particular problems. Other alternatives aim at making the best of existing methods (not necessarily FS methods only) by means of combinations. Many methods inherit both the search procedure and subset evaluation criteria in one indivisible unit.

Randomized methods. Sub-optimal sequential methods are prone to get stuck in local extremes. Randomizing may help to overcome this problem. It may also help to find solutions in significantly shorter time, although this is not guaranteed. The optimization power of purely randomized procedures like *genetic algorithms* [10] [19] [39] has been found slightly inferior to sequential methods. Extending sequential methods to include limited randomization may be a good compromise, as is the case with repeatedly randomly initialized *oscillating search* [33]. A well-known procedure performing the search semi-randomly with an inherited evaluation criterion is the *relief* algorithm [14].

Hybrid methods. The motivation to take the best of various approaches led to development of the so-called hybrid methods. These usually attempt to take use of the better properties of several existing methods while suppressing their drawbacks; the search then often consists of steps performed by means of various sub-methods. Attempts have been made to achieve Wrapper-like performance in Filter-like search time, etc. The idea of hybridization is studied in detail in [16].

Mixture-modeling based methods. Mixture-modeling approaches are suitable especially when the data is large and suspected to be multi-modal or otherwise complex in structure. Mixture modeling methods enable simultaneous construction of decision rules and feature selection [23] [24] [26].

Problem-specific methods. In many fields the standard methods can be used only with difficulties or not at all, often due to extreme dimensionality and small number of samples in the input data. This is the case in genetics [1] or text categorization [7], where the *individually best* feature selection is often the only applicable procedure. The deficiency of the IB search is compensated for by defining highly-specialized criteria suitable for the particular tasks.

Examining FS methods on real-world data

The differences between standard FS methods and practical consequences of their application can be best illustrated on real world data. We have collected a set of examples to investigate the expectable behavior of *optimal* versus *sub-optimal* FS methods and *wrapper* versus *filter* [13] methods. We investigated two standard datasets available at the UCI Repository [21]: the standard benchmark 30-dimensional continuous *mammogram* data representing 357 benign and 212 malignant patient samples (classes not completely separable), collected at the Wisconsin diagnostic breast center and the 14-dimensional (6 continuous, 8 nominal) *australian credit scoring* data representing 4 classess of 204, 256, 103 and 127 bank clients (not well separable) divided by their credibility (cf. also [31]).

The methods and functions we intend to employ and evaluate are currently defined for continuous data only. For this reason, we had to transform the *australian* dataset to eliminate the nominal features while preserving the information contained in them. We have adopted the simplest way to do that – we replaced each nominal feature by a number of binary features representing every possible nominal value that the original feature can take. It should be noted that such a transformation has negative consequences. First, the overall dimensionality of the problem increases with the unchanged number of data samples, in case of the *australian* data to 38 dimensions. Second, the new binary features are highly correlated. However, it will be shown that despite these drawbacks the use of feature selection methods leads to reasonable results.

Data, classifiers and feature selectors all determine classification performance

To illustrate the complexity of problems related to a classification system design we have collected a series of experimental results in Figures 3 to 5. We compare standard feature selection methods in both the *wrapper* and *filter* settings. In some cases we use different setups of the same method – e.g., in case of the *oscillating search* we compare the fast `OS(5,IB)` setup (denoting the deterministic sequential OS with maximum swing depth 5, initialized by means of the *individually best* feature search) to the slow, more thorough `OS(5,rand15)` setup (denoting the sequential OS with the maximum swing depth 5, called repeatedly with random initialization as long as at least 15 consecutive runs do not lead to a better solution).

We conducted the experiments with various standard classifiers. For detailed explanation of the principle of the considered classifiers – *1-Nearest Neighbor* (1-NN) classifier, *Gaussian classifier*, *Support Vector Machine* (SVM) – as tools that automatically decide on the assignment of samples to classes see, e.g., [3][5][37][40].

Note: whenever a classifier had been trained, standalone or as a *wrapper*, its classification rate was determined using 10-fold *cross-validation*. In 10-fold cross-validation the data is split to 10 equally sized intervals; the experiment is repeated 10 times with 9 data intervals used for training and 1 for testing (each interval used once for testing during the 10 experiments). The 10 obtained classification rates are eventually averaged to get the final result. This technique helps to reduce the problem of small sample size – most of the data is used for training in this way to obtain better classification rate estimates.

Figure 3: 1-Nearest Neighbor classifier performance optimized on *mammogram* data by means of *wrapper*- and *filter*-based feature selection methods

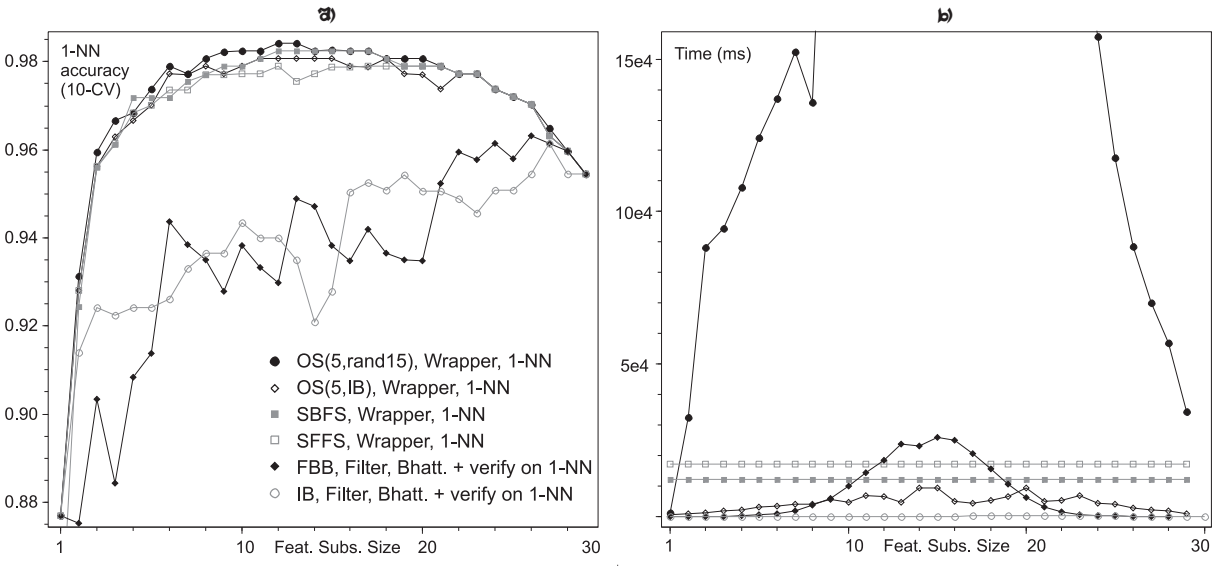
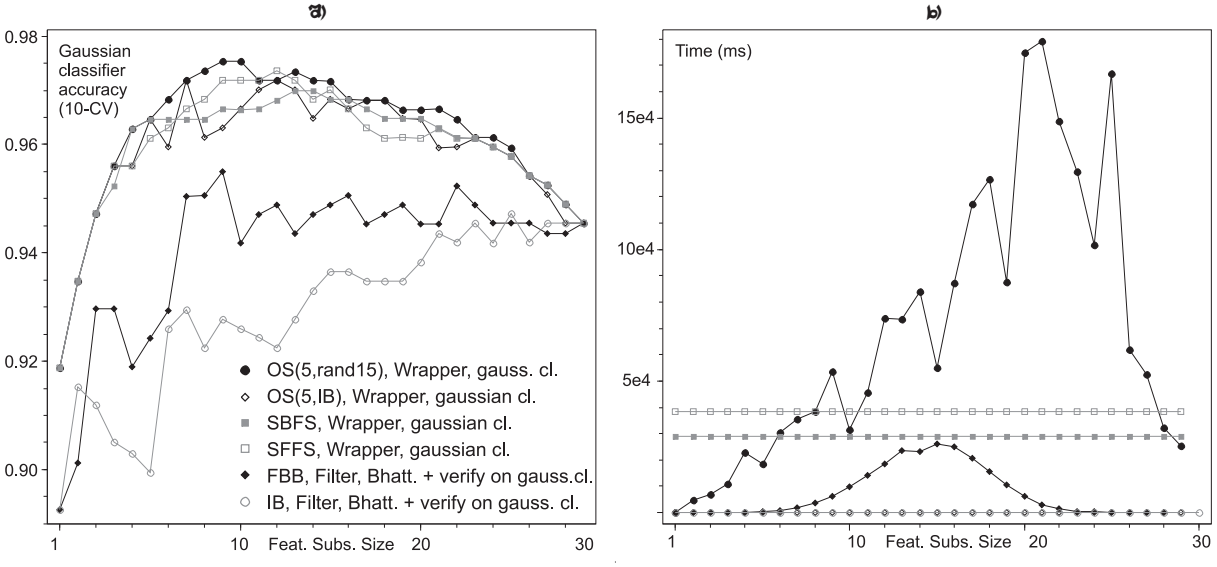


Figure 4: Gaussian classifier performance optimized on *mammogram* data by means of *wrappers* and *filters*



Figures 3 and 4 share one representative set of feature selection procedures used to optimize two different classifiers –1-NN in Figure 3 and the Gaussian classifier in Figure 4. 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 3 the *filters* exhibit mediocre performance. Bhattacharyya distance clearly has very weak relation to 1-NN performance on this dataset. This is emphasized even more by the fact that Bhattacharyya optimization (optimal result yielded by the *Fast Branch & Bound* (FBB)[35] vs. the mere *individually best* (IB) feature ranking) does not lead to any observable

improvement of the 1-NN accuracy; moreover, its impact seems to be of almost random nature. Another 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 matter.

In Figure 4 the superiority of *wrappers* is confirmed. However, unlike in the preceding case here *filter* optimization brings notable improvement (compare FBB to IB). This is most likely due to the fact that the Gaussian classifier and Bhattacharyya distance criterion (here in the 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 implies mediocre overall Gaussian classifier performance.

A comparison of Figures 3 and 4 illustrates two additional points: 1) for this dataset the Gaussian classifier is notably inferior to 1-NN. This implies that the data distribution does not exhibit normal behavior. 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 OS thoroughness-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 undesirably biased result [17] [27], what would lead to deteriorated classification performance on new, independent data (outside the training sets).

In Figures 3b and 4b the speed difference between deterministic and randomized search can be clearly seen. Deterministic procedures [`OS(5,IB)`] tend to be significantly faster than the randomized [`OS(5,rand15)`], with more consistent 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 3a where the best overall result comes from `OS(5,IB)` for subsets of 13 and 14 features). 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 demonstrated in Figures 3a and 4a. Figure 3a shows that with the *mammogram* dataset the 1-NN classifier performs best on 13-dimensional subspace, while the Gaussian classifier performs best on 9-dimensional subspace. Although theoretically it should be possible to achieve better classification accuracy at any moment by adding features, in practice this is not the case due to finite and often too small number of samples in the training set.

It should be noted that for both SFFS and SBFS the speed advantage over other methods is considerably higher than it may seem from Figures 3b and 4b – note that unlike other presented methods the SFFS and SBFS need only one run to obtain results for all subset sizes (denoted by respective horizontal lines in Figures 3b and 4b).

Figure 5: Classifier performance optimized on Austrian credit scoring data by means of Wrapper- and Filter-based feature selection methods

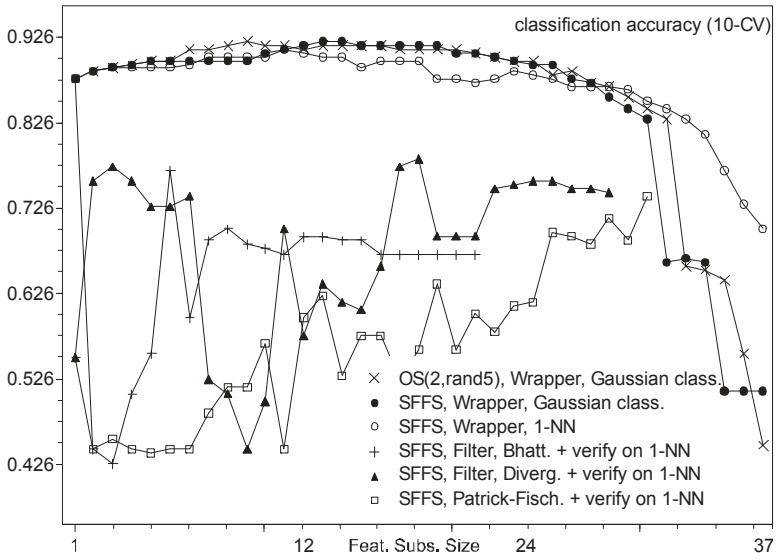


Figure 5 illustrates comparable results as Figures 3 and 4, obtained for the credit scoring data. The superiority of *wrappers* over *filters* is confirmed. The graph confirms the weak relation of normality-assuming probabilistic distance measures (Bhattacharyya, Divergence, Patrick-Fischer distances) to the performance of the 1-NN classifier, which does not assume normality of the data. This situation is to be expected in general, as most of the probabilistic feature selection criteria impose assumptions on data that are most often not met.

As in Figures 3 and 4 the graph illustrates well the so-called *peaking phenomenon*, or the fact that from a certain number of features the overall classification performance drops. In case of this credit scoring dataset the best results have been obtained with the Gaussian classifier *wrapper* optimized using the randomized *oscillating search* (for subset size ~10) or using the *sequential forward floating search* (for subset size ~14). The shape of the graph coincides with the fact that the features were highly correlated – the maximum achieved classification rate does not change much for most subset sizes roughly up to 30. Note: the Gaussian classifier is shown to be equally good as 1-NN for this data.

Note: missing values in Fig. 5 follow from numerical problems. This also illustrates the advantage of *forward* methods over *backward* ones – here the *sequential forward floating search* proved capable of yielding results for subset sizes up to 70% of the full dimensionality, in the same setting the *backward* methods fail to yield any result at all.

Pitfalls of feature subset evaluation – experimental comparison of criterion functions

As stated before, in certain type of tasks it is important to judge the importance of individual features. 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. In fact, even more differences can be observed between feature evaluation made using *wrappers* and *filters*.

Tab. 2: Single features in descending order, first best 7 then last worst 7, according to individual criterion values (i.e., „individual discriminative power"), 4-class, 38-dimensional *australian credit scoring* data

Bhattacharyya	**37**	**29**	**31**	**19**	**6**	**28**	**20**	**…**	**14**	**24**	**23**	**13**	**32**	**0**	**12**
Divergence	**37**	**31**	**29**	**6**	**19**	**28**	**20**	**…**	**14**	**24**	**23**	**13**	**32**	**0**	**12**
G.Mahalanobis	**29**	**30**	**31**	**28**	**37**	**2**	**26**	**…**	**0**	**25**	**17**	**13**	**24**	**35**	**12**
Patrick-Fisher	**29**	**6**	**19**	**10**	**25**	**20**	**18**	**…**	**12**	**13**	**32**	**0**	**1**	**37**	**36**
Gauss. cl. (10-f. CV)	**6**	**10**	**35**	**25**	**29**	**19**	**2**	**…**	**34**	**8**	**9**	**20**	**33**	**31**	**37**
1-NN (10-fold CV)	**29**	**30**	**31**	**28**	**19**	**16**	**26**	**…**	**35**	**37**	**12**	**14**	**36**	**1**	**2**
SVM lin (10-f. CV)	**29**	**3**	**16**	**18**	**14**	**31**	**4**	**…**	**20**	**27**	**22**	**15**	**6**	**7**	**24**

In the example in Table 2 we demonstrate the differences between some standard criterion functions – both the probabilistic measures (*filter* setting: Bhattacharyya, Divergence, generalized Mahalanobis, Patrick-Fisher distances) and the classification accuracy (*wrapper* setting: *Gaussian classifier*, *1-Nearest Neighbor*, *Support Vector Machine* with linear kernel [3], classification accuracy evaluated by means of *10-fold cross-validation*). We evaluated single features of the *australian credit scoring* data using each of the criteria and ordered them descending according to the respective criterion values. In this way the more distinctive features should appear in the left part of the table, while the noisy and less important should appear in the right. The differences in feature ordering illustrate the importance and also the possible pitfalls of the choice of suitable criterion. Although some features are evaluated as good by most of the criteria (29, 31) and some as bad (12), with many others the results vary considerably and may show conflicting evidence (6, 14). This is an undesired effect illustrating how difficult it may be to draw general conclusions about which features are generally best to select – it can also be taken as an argument in favor of using *wrappers* instead of *filters*, to identify features with more certainty with respect to the given decision rule.

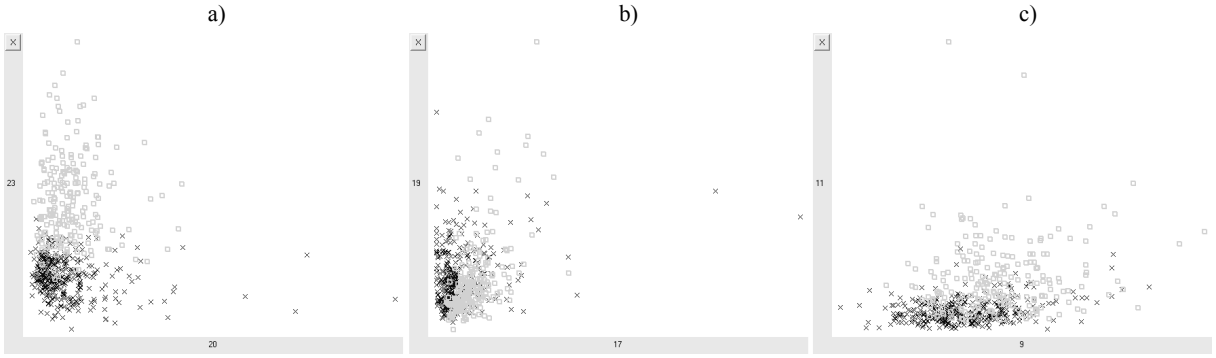
Following the examples above it can be concluded that by employing classifier-independent criteria one accepts certain simplification and possibly misleading assumption about data (note that most of probabilistic criteria are defined for unimodal normal distributions only). Nevertheless, classifier-independent criteria may prove advantageous to prevent over-fitting in cases when *wrapper* based feature selection fails to identify feature subsets that generalize well.

A different view of criterion functions – visual subspace comparison

The examples in Figure 5 illustrate spatial properties of different feature subsets (pairs, for easy viewing) identified in the *mammogram* data. Figure 6a shows the best pair of features identified by maximizing the Bhattacharyya distance (the same pair has been found using Divergence). Figure 6b shows the best pair identified by maximizing the Patrick-Fisher distance. Figure 6c illustrates an example of „bad" feature pairs unsuitable for discrimination, obtained for this purpose by means of minimizing the Bhattacharyya distance. Note the difference between the "good' feature pairs (Figs. 6a and 6b) and the "bad" pair (Fig. 6c). In case of the "bad" pair the classes overlap considerably, what makes their distinction considerably harder. This example also illustrates the common situation with real world data – often there exists no subspace in which the classes are completely separable. In most cases

some non-zero classification error is to be expected. Note: The images for Figure 6 were obtained using the Feature selection toolbox software [34].

Figure 6: Visual comparison of 2D subspaces found on *mammogram* data by maximizing: a) Bhattacharyya distance [the same was found by the Divergence], b) Patrick-Fischer distance. Figure c) shows a subspace unsuitable for discrimination due to high overlap of classes, found by minimizing the Bhattacharyya distance.



Summary of recent sub-optimal feature selection methods and recommendations

Our own research and experience has led us to the conclusion that *there exists no unique generally applicable approach* to the feature selection problem. Some feature selection approaches and methods are more suitable under certain conditions, others are more appropriate under other conditions, depending on the properties and our knowledge of the given problem. Hence continuing effort is invested in developing new methods to cover the majority of situations which can be encountered in practice.

A number of recent feature subset search strategies have been reviewed and compared. Following the experimental analysis of their respective advantages and shortcomings, the conditions under which certain strategies are more pertinent than others have been suggested.

Recent developments of algorithms for optimal search led to considerable improvements of the speed of search. Nevertheless, the exponential nature of optimal search remains and will remain one of key factors motivating the development of principally faster sub-optimal strategies. Among the family of sequential search algorithms the *floating search* and *oscillating search* methods deserve particular attention as a practically useful compromise between speed and optimization performance.

Concerning our current experience, we can give the following recommendations – the *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 with respect to the chosen criterion. The *floating search* also shows to be a good compromise to deal with the *optimization efficiency* versus *generalization* (impact on classifier performance on new data) trade-off. The *oscillating search* may become a better choice when: 1) the highest possible criterion value 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 standard sequential methods, or 4) extreme problem dimensionality prevents any use of standard sequential methods, or 5) the search is to be performed in real-time systems. Especially when repeated with different random initial feature subsets the *oscillating search* shows outstanding ability to avoid local extremes in favor of finding the global optimum.

It should be stressed that, as opposed to the optimal *branch & bound* algorithm, the sub-optimal sequential methods are tolerant to deviations from monotonic behavior 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* [13]), which according to a number of researchers seem to be the only legitimate criterion for feature subset evaluation. Superior performance of *wrappers* over *filters* has been verified experimentally in this paper as well.

The importance of feature selection for classification performance has been clearly demonstrated. Note: selected algorithm source codes are available for download at `http://ro.utia.cas.cz/dem.html`.

References:

[1] Alexe G. – Alexe S. – Hammer P.L. – Vizvari B. (2006), *Pattern-based feature selection in genomics and proteomics*, Annals of Operations Research, vol. 148, no. 1, pp. 189–201, 2006.

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

[3] Chang Ch.-Ch. – Lin Ch.-J. (2001), *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.

[4] Devijver P.A. – Kittler J. (1982), *Pattern Recognition: A StatisticalApproach*, Prentice-Hall, 1982.

[5] Duda R.O. – Hart P.E. – Stork D.G. (2000), *Pattern Classification, 2nd edition*, Wiley-Interscience, 2000.

[6] Ferri F.J. – Pudil P. – Hatef M. – Kittler J. (1994), *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.

[7] Forman G. (2003), *An extensive empirical study of feature selection metrics for text classification*, J. Mach. Learn. Res., vol. 3, pp. 1289–1305, 2003.

[8] Fukunaga K. (1990), *Introduction to Statistical Pattern Recognition*, Academic Press, 1990.

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

[10] Hussein F. – Ward R., and Kharma N. (2001), *Genetic algorithms for feature selection and weighting, a review and study*, icdar, vol. 00, p. 1240, 2001.

[11] Jain A.K. – Zongker D. (1997), *Feature Selection: Evaluation, Application and Small Sample Performance*, IEEE Transactions on PAMI 19(2):153–158, 1997.

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

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

[14] Kononenko I. (1994), *Estimating attributes: Analysis and extensions of Relief*, in ECML-94: Proc. European Conf. on Machine Learning. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1994, pp. 171–182.

[15] Kudo M. – Sklansky J. (2000), *Comparison of Algorithms that Select Features for Pattern Classifiers*, Pattern Recognition33(1):25–41, 2000.

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

[17] Loughrey J. – Cunningham P. (2004), *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, 2004.

[18] Marill T. – Green D. (1963), *On the effectiveness of receptors in recognition systems*, IEEE Trans. on Information Theory 9 (1) : 11-17, 1963.

[19] Mayer H.A. – Somol P. – Huber R. – Pudil P. (2000), *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.

[20] McLachlan G.J. (1992), *Discriminant Analysis and Statistical Pattern Recognition*, John Wiley & Sons, New York, 1992.

[21] Murphy P.M. – Aha D.W. (1994), *UCI Repository of Machine Learning Databases [http://archive.ics.uci.edu/ml/]*, University of California, Department of Information and Computer Science, Irvine, CA, 1994.

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

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

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

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

[26] Pudil P. – Novovičová J. (1998), *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.

[27] Raudys Š (2006), *Feature Over-Selection*, Lecture Notes in Computer Science LNCS 4109, Springer, 622–631, 2006.

[28] Ripley B. (1996), *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, Massachusetts, 1996.

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

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

[31] Somol P. – Baesens B. – Pudil P. – Vanthienen J. (2005) *Filter- versus wrapper-based feature selection for credit scoring*, International Journal of Intelligent Systems 20 (10): 985–999, 2005.

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

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

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

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

[36] Stearns S.D. (1976), *On selecting features for pattern classifiers*, in: Proc. of the 3rd Int. IEEE Joint Conf. on Pattern Recognition, Coronado, CA, USA, 71–75, 1976.

[37] Theodoridis S. – Koutroumbas K. (2003), Pattern Recognition, 2nd Ed.,Academic Press, 2003.

[38] Tsamardinos I. – Aliferis C. (2003), *Towards Principled Feature Selection: Relevancy, Filters, and Wrappers*, Artificial Intelligence and Statistics, 2003.

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

[40] Webb A. (2002), *Statistical Pattern Recognition, 2nd Ed.*, John Wiley &Sons, 2002.

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

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

Identifikace nejinformativnějších proměnných pro problémy rozhodovacího typu – přehled současných postupů a problémů

Pavel Pudil – Petr Somol

Článek podává přehled problémů souvisejících s vyhledáváním proměnných (výběru příznaků) pro rozhodování založené na metodách strojového učení, a to s ohledem na aktuální stav problematiky. Je porovnáno několik populárních metod a zařazeno do taxonomického kontextu. Rovněž je diskutován problém protichůdnosti požadovaných vlastností příslušných metod – na schopnost zobecňovat a schopnost efektivní optimalizace. Problém je ilustrován pomocí experimentů na reálných datech.

Klíčová slova: identifikace proměnných, výběr příznaků, strojové učení, rozhodovací pravidla, klasifikace.

Identifying the most informative variables for decision-making problems – a survey of recent approaches and accompanying problems

ABSTRACT

We give an overview of problems related to variable selection (also known as feature selection) techniques in decision-making problems based on machine learning with particular emphasis to recent knowledge. Several popular methods are reviewed and assigned to a taxonomical context. Issues related to the generalization versus performance trade-off inherent to currently used variable selection approaches are addressed and illustrated on real-world examples.

Key words: variable selection, feature selection, machine learning, decision rules, classification.

JEL classification: C60, C80