PERGAMON

Pattern Recognition 35 (2002) 2749–2759

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Feature selection toolbox ☆

P. Somol, P. Pudil *

Department of Pattern Recognition, Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, 182 08 Prague 8, Czech Republic

Received 31 October 2001; accepted 31 October 2001

Abstract

A software package developed for the purpose of feature selection in statistical pattern recognition is presented. The software tool includes both several classical and new methods suitable for dimensionality reduction, classification and data representation. Examples of solved problems are given, as well as observations regarding the behavior of criterion functions.

Keywords: Pattern recognition; Feature selection; Subset search; Search methods; Software toolbox

1. Introduction

The most common task in Pattern Recognition is the classification of patterns "data records" into a proper class. Problems like this one can be considered as a part of recognition tasks, computer-aided decision tasks and other applications as well. One of the most important tasks is the problem of dimensionality reduction. In order to reduce the problem dimensionality we often use "feature selection" methods because of their relative simplicity and meaningful interpretability of results.

Undoubtedly, many similarities can be found between Pattern Recognition and Data Mining. Selecting features can be viewed as selection of relational database columns (and thus the information they hold) by maximizing some criterion which was defined upon the database content. The usual goal is to find such a part of the data that holds most of the information (suitable for classification, approximation or other purpose). Storing the rest of the data may then be considered as wasting the computer memory. Dimensionality reduction may result not only in improving the speed of data manipulation, but even in improving the classification rate by reducing the influence of noise.

When combined with data approximation methods, the dimensionality reduction process may result in substantial data compression, while the overall statistical properties remain preserved. Moreover, different ways of data manipulations and queries become possible without need of access to original data (which may thus become redundant).

The paper is organized as follows. In the next section the feature selection toolbox (FST) is briefly described. Then the search strategies implemented in FST are outlined with references to particular papers where they are discussed in more detail as this paper is focused more to the software issues description. The reason is that to discuss here all the feature selection methods is impossible as to each of them a full size paper has been devoted. Conceptually, very different search strategies based on the approximation model are just listed in Section 4, while more examples of FST applications to real world problems are treated more thoroughly in Section 5. The paper is concluded with discussion of implementation issues and directions of further work.

2. Feature selection toolbox

The FST software has been serving as a platform for data testing, feature selection, approximation-based modeling of data, classification and mostly testing newly

☆ Both the authors are also with the Joint Laboratory of Faculty of Management—University of Economics, Prague and Institute of Information Theory and Automation, Czech Academy of Sciences, and EuroMISE Kardiocentrum.

* Corresponding author. Fax: +420-2-688-3031.
E-mail address: pudil@utia.cas.cz (P. Pudil).

PII: S0031-3203(01)00245-X

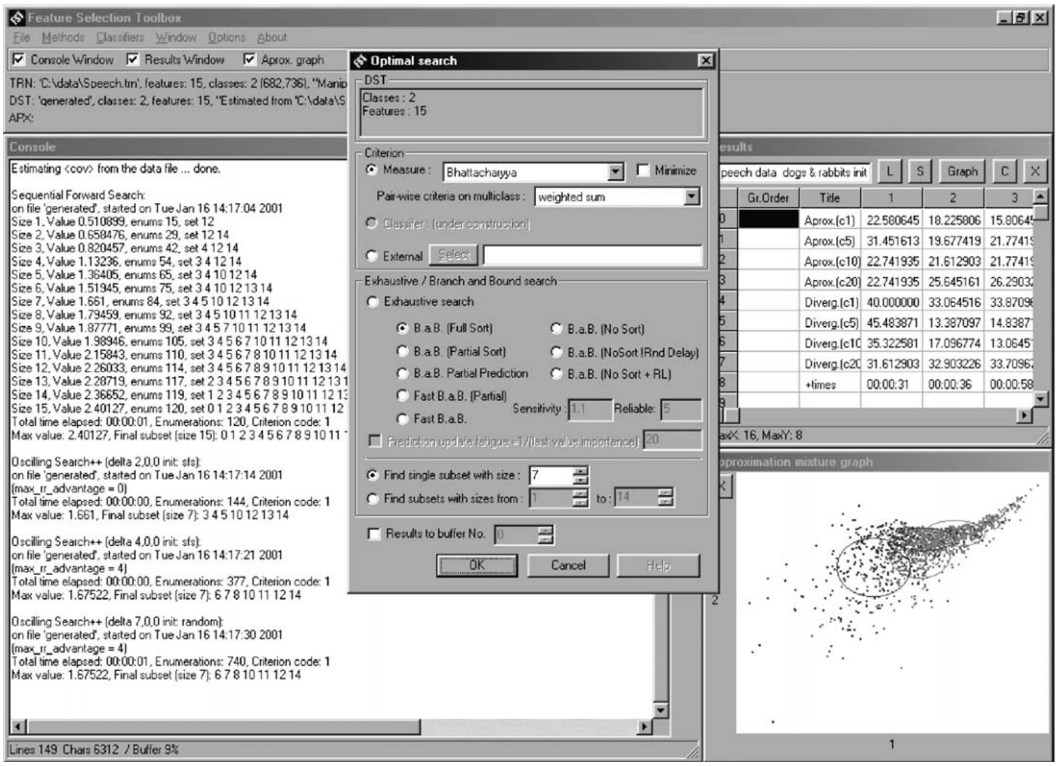


Fig. 1. Feature selection toolbox—windows GUI workplace.

developed methods. It is used basically for pattern recognition purposes, however, we used it for solving different problems related to decision making in economics and other branches as well.

A rather simple user interface (Fig. 1) was constructed upon a strong functional kernel. Most of the results are generated in the form of textual protocol into the Console window. Numerical results may be collected in tables and used for generating graphs. Data may be displayed in a 2D projection.

2.1. What data can the FST process and how?

A typical use of the FST consists of the following steps: after opening the proper data file, the user has the option to choose some feature selection method from the menu. Each method displays a specific dialogue allowing setting of different parameters specific for the chosen method. Then computation follows with a thorough listing of performed steps logged into the Console window. Both optimal and sub-optimal methods find the resultant feature subset and the corresponding criterion value. On the other hand, approximation methods generate data model which may be further used. Beside basic use for feature selection, the software package may also be used for basic classification purposes and different manipulations of data files.

As the feature selection process may be time consuming (particularly for high dimensional data), a thorough information about the current computation state, current dimensionality, current best criterion value, current direction of search and numbers of performed and expected computational steps are displayed during the algorithm run.

Because of strong diversification of data formats used for scientific purposes, we adopted a most general way of data storage—standard text files containing numerical values in ANSI C format. Files must begin with a simple textual header containing information about the number of classes, class members, dimensionality, etc. Correct files may be processed using the file manipulation tool to change the number of classes, join or cut files, delete features, etc. To extend the usability of FST we plan to equip it with a special Data Import Filter allowing conversion (user-controlled, if necessary) of virtually any text file into a usable form. This relates especially to files with strange ordering, strange formatting, etc. The current version of FST supports three types of text data files:

- Data *files containing samples* (identified by .trn extension)—data have the form of number vectors representing individual samples (patterns). They should be ordered according to classes, then according to samples (vectors—points in pattern space) inside classes, then according to features inside samples. Such a data ordering may be viewed as a file of relational databases (classes) with numerical values. It is possible to construct an approximation model upon such data (.apx file type), also data structure in the form of mean vectors and covariance matrices may be estimated from .trn file (generates a .dst type file). Sample files may be used for feature selection, classification of unknown data or estimating the classification error rate.

- Data *structure files* containing mean vectors and covariance matrices (identified by .dst extension). This data form is suitable for use with optimal and sub-optimal feature selection methods based on feature set evaluation criteria like Bhattacharyya distance, etc.
- Approximation *model files* (identified by .apx extension)—are generated by approximation or divergence method. These files may serve for classification of sample files (.trn) with pseudo-Bayes classifier.

3. Search strategies for feature selection

Feature selection can be viewed as a special case of a more general subset selection problem. The goal is to find a subset maximizing some adopted criterion. In case of feature selection we usually use some probabilistic inter-class distance measures or better directly the classifier correct classification rate.

The list of implemented criterion functions is as follows (for details see, e.g., Ref. [1]): *Bhattacharyya distance*, *Mahalanobis distance*, *Generalized Mahalanobis*, *Patrick–Fischer distance*, *Divergence*. It is also possible to maximize functions programmed in external executables. In this way, we are able to minimize error rates of different classifiers, etc.

Feature selection methods can be divided into optimal and sub-optimal ones. The only universal method for finding optimal solution (feature subset yielding maximum value of criterion function) is exhaustive search [2]. However, this method is unusable for problems of higher dimensions because of its exponential time complexity. The practical problem of dimension limitation given by the current state of computer hardware is approximately 40 for exhaustive search (if selecting 20 features). This limit will remain prohibitive in the future.

The only alternative to exhaustive search, yielding the optimal solution, is the branch and bound (BB) based algorithms (e.g. Refs. [1,3]). These algorithms are limited to monotonic criteria only. Their speed strongly depends on the data (classical BB may run several times faster than exhaustive search but on the other hand it may be even slower). The FST implements the exhaustive search as well as several versions of BB algorithms including the currently fastest prediction-based ones:

- The *Basic* BB algorithm as described in Ref. [4]. This is the slowest BB algorithm, implemented here for comparison purposes.
- The *Enhanced* BB algorithm described, e.g., in Ref. [3]. This is the most widely used algorithm version, having been accepted as the fastest optimal algorithm so far. It utilises a heuristics for effective reduction of the number of candidate subsets. Our implementation further improves its performance by means of generating the *minimum solution tree* [5]. This algorithm has served as a reference for evaluation of the following prediction-based algorithms.
- The *fast branch & bound* (FBB) algorithm described in Ref. [6] investigates differences between criterion values before and after individual feature removal. This information is later used (under certain circumstances) to quickly predict criterion values in certain search-tree nodes instead of slowly computing the true value. For more details about how to preserve optimality using this scheme, see the cited paper.
- The *branch & bound with partial prediction* (BBPP) by Somol et al. [7] addresses the problem of recursive criterion computation which is not possible using the FBB. In contrast to FBB, the BBPP cannot skip criterion computations in search-tree nodes. However, it uses a prediction-based heuristics for effective ordering of tree nodes which makes it still faster than classical BB algorithms.

However, all the optimal methods are practically unusable for problems involving hundreds or thousands of dimensions. A lot of time has been therefore invested into development of sub-optimal methods.

Sub-optimal methods cannot guarantee optimal solutions. However, they can yield optimal or near-optimal results in most cases. The speed of sub-optimal methods is generally significantly higher than the speed of optimal methods. The trade off between the quality of feature selection results and computation time may be often altered by setting user parameters. The FST includes both sub-optimal methods known from literature (for overview see, e.g., Ref. [1] or [8]) and methods developed recently in our department:

- The *sequential forward search* (SFS) and its backward counterpart SBS—basic methods known for their simplicity and speed. They yield worse results than other listed methods [1].
- The *plus-L-minus-R*—this method is the first one handling the nesting-effect problem [1].
- Generalized forms of previous methods—based on group-wise feature testing, they may find better solutions but at the cost of increased time complexity [1].
- The *floating search* methods (SFFS, SFBS)—fast and powerful methods, most suitable for general use [9]. They have been evaluated as the currently best sub-optimal methods for feature selection [8].

- The *adaptive floating search* methods (ASFFS, ASFBS). While requiring more computational time, these methods allow finding better solutions than floating search, if floating search fails to find the optimum [10].
- The *oscillating search* method (OS)—a method featuring wide possibilities of being altered through user parameters. It allows both very fast and very thorough search. Because of its different search principle, this method may become an interesting alternative to the methods described above, because it yields the best solutions in isolated cases. It can also be used to refine the solution found by other methods. The search may be limited also by the time constraint [11].

4. Approximation model based methods

Approximation model based methods represent a different but powerful approach to dimensionality reduction and classification especially in cases of multimodal and non-Gaussian data. The approach is based on approximating unknown conditional pdfs by finite mixtures of a special product type.

Two different methods are available: the "approximation" method is suitable mainly for data representation [12], the "divergence" method is based on maximizing the Kullback's J-divergence and is more suitable for discrimination of classes [13].

Both the methods encapsulate the feature selection process into the statistical model construction. The importance of these methods follows from their independence on a priori knowledge related to the data. Generic data may be processed without preparation.

The definition of approximation model based methods is followed by defining the "pseudo-Bayes" classifier. The title "pseudo-Bayes" is used since the probabilities in Bayes formula are replaced by their approximations and also because the decision is made in a lower-dimensional subspace.

Those readers who would like to learn more details of this approach are referred to the papers cited above, however, to get an idea, we provide here a brief description of the approach.

For the cases when we cannot even assume that class-conditional pdfs are unimodal and the only available source of information is the training data, a new approach has been developed based on approximating the unknown class conditional distributions by finite mixtures of parametrized densities of a special type.

The following modified model with latent structure for ωth class-conditional pdf of $\mathbf{x}$ has been suggested in the considered approach presented in Pudil et al. [12]:

$$p(\mathbf{x}|\Theta^{\omega})=\sum_{m=1}^{M_\omega}\alpha_m^{\omega}p_m(\mathbf{x}|\omega)=\sum_{m=1}^{M_\omega}\alpha_m^{\omega}g_0(\mathbf{x}|\theta_0)g(\mathbf{x}|\theta_m^{\omega},\theta_0,\Phi), \tag{1}$$

where $\Theta^{\omega}=\{\alpha_m^{\omega},\theta_m^{\omega},\theta_0,\Phi;m=1,\dots,M_\omega\}$ is the complete set of unknown parameters of the finite mixture (1), M_ω is the number of artificial subclasses in the class ω, α_m^{ω} is the mixing probability for the mth subclass in class ω, $\leqslant \alpha_m^{\omega} \leqslant 1$, $\sum_{m=1}^{M_\omega}\alpha_m^{\omega}=1$. Each component density $p_m(\mathbf{x}|\omega)$ includes a nonzero "background" probability density function g_0, common to all classes:

$$g_0(\mathbf{x}|\theta_0)=\prod_{i=1}^{D}f(x_i|\theta_{0i}),\quad \theta_0=(\theta_{01},\theta_{02},\dots,\theta_{0D}) \tag{2}$$

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

$$g(\mathbf{x}|\theta_m^{\omega},\theta_0,\Phi)=\prod_{i=1}^{D}\left[\frac{f(x_i|\theta_{mi}^{\omega})}{f(x_i|\theta_{0i})}\right]^{\phi_i},\quad \phi_i=\{0,1\}, \tag{3}$$

$$\theta_m^{\omega}=(\theta_{m1}^{\omega},\theta_{m2}^{\omega},\dots\theta_{mD}^{\omega}),\ \ \Phi=(\phi_1,\phi_2,\dots,\phi_D)\in\{0,1\}^D.$$

The proposed model is based on the idea to posit a common "background" density for all classes and to express each class-conditional pdf as a mixture of a 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 parameters ϕ_i and the same subspace of feature vector space for each component density is used in all classes. Any specific univariate function $f(x_i|\theta_{mi}^{\omega})$ is substituted by the "background" density $f(x_i|\theta_{0i})$ whenever ϕ_i is zero. In this way, the binary parameters ϕ_i can be looked upon as *control variables* since the complexity and the structure of the mixture (1) can be controlled by means of these parameters.

For any choice of ϕ_i, the finite mixture (1) can be rewritten by using Eqs. (2) and (3) as

$$p(\mathbf{x}|\Theta_\omega)=\sum_{m=1}^{M_\omega}\alpha_m^{\omega}\prod_{i=1}^{D}[f(x_i|\theta_{0i})^{1-\phi_i}f(x_i|\theta_{mi}^{\omega})^{\phi_i}]. \tag{4}$$

Setting some $\phi_i=1$, we replace the function $f(x_i|\theta_{0i})$ in the product in Eq. (4) by $f(x_i|\theta_{mi}^{\omega})$ and introduce a new independent parameter θ_{mi}^{ω} in the mixture (4). The actual number of involved parameters is specified by the condition $\sum_{i=1}^{D}\phi_i=\gamma, 1\leqslant\gamma\leqslant D$.

The proposed approach to feature selection based on the finite mixture model (4) is somewhat more realistic than the other parametric approaches. It is particularly useful for the case of multimodal distributions when other feature selection methods based on distance measures (e.g., Mahalanobis distance, Bhattacharyya distance), would totally fail to provide reasonable results as has been shown in Refs. [12,13]. An important characteristic of our approach is that it effectively partitions the set X_D of all D features into two disjunct subsets X_d and X_D-X_d, where the joint distribution of the features from X_D-X_d is common to all the classes and constitutes the background distribution, as opposed to features forming X_d, which are significant for discriminating the classes. The joint distribution of these features constitutes the "specific" distribution defined in Eq. (3).

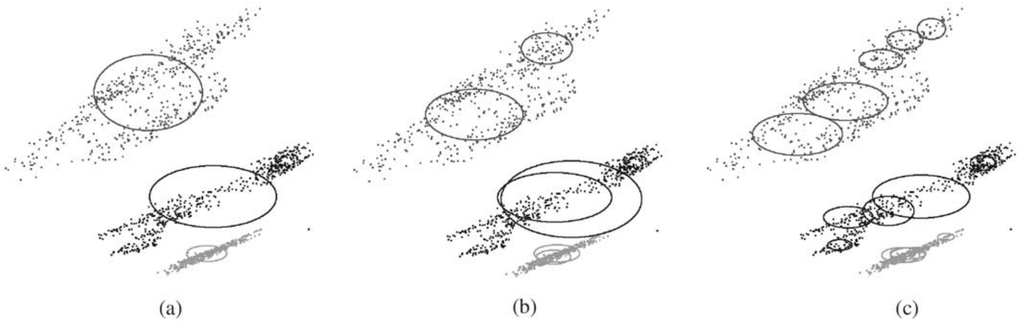


Fig. 2. Visual comparison of 2D projections of approximation models estimated by means of the approximation method on marble data (see in text): (a) single mixture component, (b) 2 mixture components, and (c) 5 mixture components. Ellipses illustrate the equipotential component planes, component weights are not displayed.

Table 1
Error rates (%) of different classifiers with different parameters

		Gauss	Approx. 1c	Approx. 5c	Approx. 10c	Approx. 20c
Speech	(random init.)	8.39	21.61	7.58	9.19	9.03
Data	(dogs & rabbits init.)	—	21.61	7.42	6.45	8.39
Mammo	(random init.)	5.96	5.26	5.26	5.96	4.56
Data	(dogs & rabbits init.)	—	5.26	5.26	5.96	5.96

The ‘Gauss’ column contains results of a Gaussian classifier. Other columns contain results obtained using the ‘approximation’ method (in this case the ‘divergence’ method yielded the same results). Results in second row for each data have been obtained after preliminary cluster-detection used to initialise the ‘approximation’ method. 5c means 5 components of mixture, etc.

According to these features alone, a new pattern **x** is classified into one of C classes and under this partition of the feature set X_D either the Kullback–Leibler distance is minimised (so called “approximation method”, see Fig. 2) or the Kullback J-divergence is maximised (so called “divergence method”). Two proposed methods yield the feature subset of required size without involving any search procedure. Furthermore, in the inequality for the sample Bayes decision rule assuming model with latent structure (4), the “background” density g_0 is reduced and therefore, the new approach provides a pseudo-Bayes decision plug-in rule employing the selected features. Consequently, the problems of feature selection and classifier design are solved simultaneously.

It should be emphasized that although the model looks rather unfriendly, its form leads to a tremendous simplification [12]) when the univariate density f is from the family of Gaussian densities. The use of this model (4) makes the feature selection process a simple task.

5. Application examples

Perhaps the best way of introducing the FST software scope is demonstration on task examples. We used 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 benign and malignant patients, obtained from the Wisconsin Diagnostic Breast Center via the UCI repository—ftp.ics.uci.edu.
- 3-class, 20-dimensional *marble* data representing different brands of marble stone; data are well separable.

5.1. Classification task example

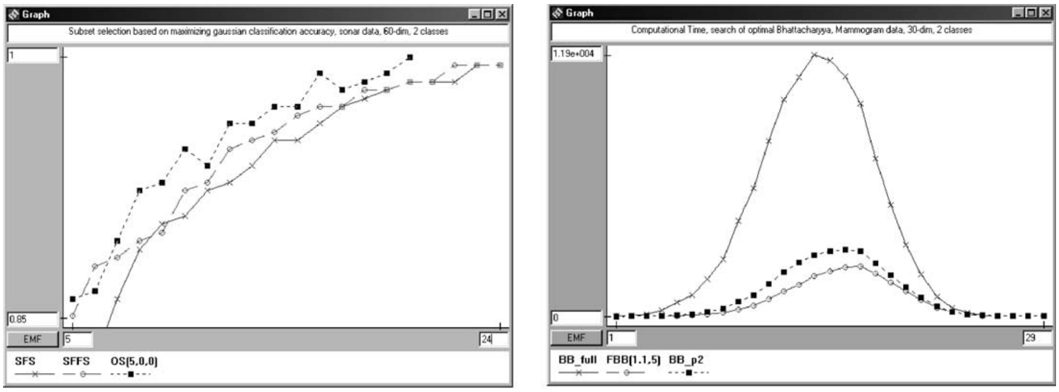
Using the FST we compared the performance of Gaussian classifier to the pseudo-Bayes classifier, defined especially for use with multimodal data, and defined in relation to “approximation” and “divergence” methods (cf. Section 4). Table 1 illustrates the potential of the approximation model based classifiers. However, it also illustrates the necessity of experimenting to find a suitable number of components (the issue is discussed, e.g., in Ref. [14]).

Results were computed on the full set of features. In case of approximation and divergence methods the algorithms were initialized randomly (1st row) by means of the “dogs & rabbits” cluster analysis [15] pre-processor (2nd row). Classifiers were trained on the first half of the dataset and tested on the second half.

Speech data, dogs & rabbits init, error rates

	Title	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	Aprox.(c1)	22.58	18.22	15.80	16.77	17.25	18.38	19.35	20.32	20.80	21.12	21.12	21.77	20.80	21.45	21.61
1	Aprox.(c5)	31.45	19.67	21.77	19.51	21.29	23.54	22.41	14.83	15.32	9.51	8.70	8.22	7.58	6.93	7.09
2	Aprox.(c10)	22.74	21.61	21.77	18.87	23.22	22.41	21.77	18.70	9.67	7.25	7.58	7.25	6.93	6.93	7.58
3	Aprox.(c20)	22.74	25.64	26.29	23.54	23.06	20.80	21.45	17.25	10.16	11.77	10.16	9.03	8.54	7.90	8.38
4	Diverg.(c1)	40.00	33.06	33.87	23.70	22.25	23.38	24.51	23.87	23.06	23.70	23.87	21.61	20.80	20.96	21.61
5	Diverg.(c5)	45.48	13.38	14.83	17.09	15.48	13.22	10.16	9.35	8.06	8.22	6.93	6.61	6.61	6.93	7.09
6	Diverg.(c10)	35.32	17.09	13.06	8.38	7.25	13.54	16.45	14.83	12.58	7.90	8.70	7.25	8.38	7.41	7.58
7	Diverg.(c20)	31.61	32.90	33.70	25.64	15.32	12.09	11.12	11.45	8.06	10.32	10.48	6.93	7.58	7.41	8.38

Fig. 3. Approximation model based methods performance on the speech data. The screenshot shows the way FST stores numerical results. Different lines may be selected for graph display using specified colors and/or line thickness and shapes, as shown on Fig. 4.



(a) (b)

Fig. 4. Subset search methods performance as shown by the FST graphic output. The left picture demonstrates sub-optimal methods performance comparison, i.e. maximal achieved criterion values for subsets of 5 to 24 features. The right picture demonstrates optimal methods performance comparison, i.e. computational time needed to find optimal subsets of 1 to 29 features.

Table 1 demonstrates a potential of mixture approximation methods—with 5 mixture components (see column approx.5c) for the speech data and 1, 5 or 20 components for mammo data. The underlying data structure has been modeled precisely enough to achieve a better classification rate when compared to the Gaussian classifier. Second row for each data contains approximation and divergence method results after preliminary initialization by means of the so-called "dogs and rabbits" clustering method [15]. The method is inspired by the self-organizing-map principle. Single training set samples are processed sequentially in order to slightly attract the closest cluster candidate center. In this way the "dogs and rabbits" method effectively identifies cluster centers and its results may be used for setting initial component mean parameters. However, component sizes (variance parameters) have to be specified otherwise, e.g., randomly.

5.2. Dimensionality reduction task example

The table screen-shot in Fig. 3 stores error rate values achieved by the approximation and divergence methods with different number of components. Columns represent selected subset sizes. From this table it is possible to guess that single component modeling is not sufficient, best results have been achieved with approximately five or more components and twelve or more features. It is fair to say that this somewhat "guessing" way of specifying the suitable number of selected features is often the only applicable one.

Fig. 4(a) demonstrates the performance of sub-optimal feature selection methods being used for maximizing Gaussian classifier performance. Fig. 4(b) demonstrates speeds of different optimal feature selection methods on the mammogram data set. All the optimal methods (except the exhaustive search) are based on the BB idea and are restricted for use with monotonic criterion functions only.

5.3. 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 we usually lack enough information about the real underlying probabilistic structures and analytical evaluation may become computationally too expensive. Therefore, many alternative evaluation approaches were introduced.

Table 2
Criterion functions comparison on 2-class speech data

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
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

Single features have been ordered increasingly according to individual criterion values, i.e. the "individual discriminative power".

Table 3
Criterion functions comparison on 2-class speech data

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
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	—	—	—	—	—	—

The table shows subsets of seven features selected to maximize different criteria. In contrast the last line shows a criterion-minimizing subset.

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) some classifier-independent criteria—probabilistic distance measures—have been defined. For a good overview and discussion of their properties, see Ref. [1]. The "approximation" and "divergence" methods (cf. Section 4) also incorporate a feature evaluation function, which is closely related to the purpose of these respective methods.

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 on 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 only. Traditional distance measures (first four rows) gave similar 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 a 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 seven features out of 15 so as to maximize particular criteria to compare the differences between detected "optimal" subsets.

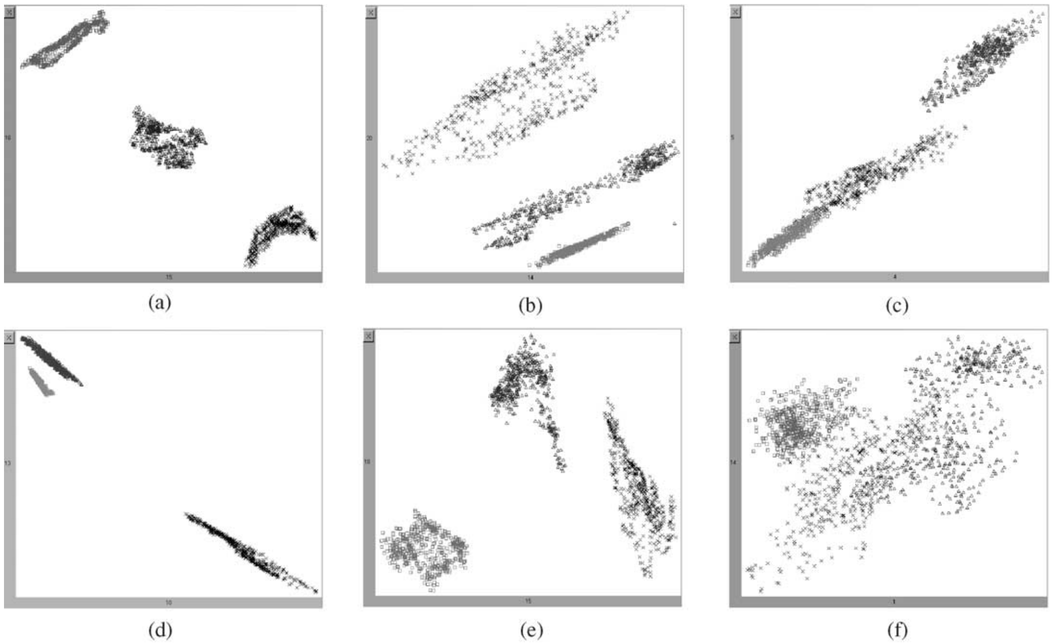


Fig. 5. Visual comparison of 2D subspaces found on 20-dimensional marble data by maximizing: (a) Bhattacharyya (the same was found by Generalized Mahalanobis), (b) Divergence, (c) Patrick–Fischer distances. Mixture model methods using 5 components results: approximation method—(d), and divergence method—(e). Picture (f) demonstrates a subspace unsuitable for discrimination (found by minimizing the Bhattacharyya distance).

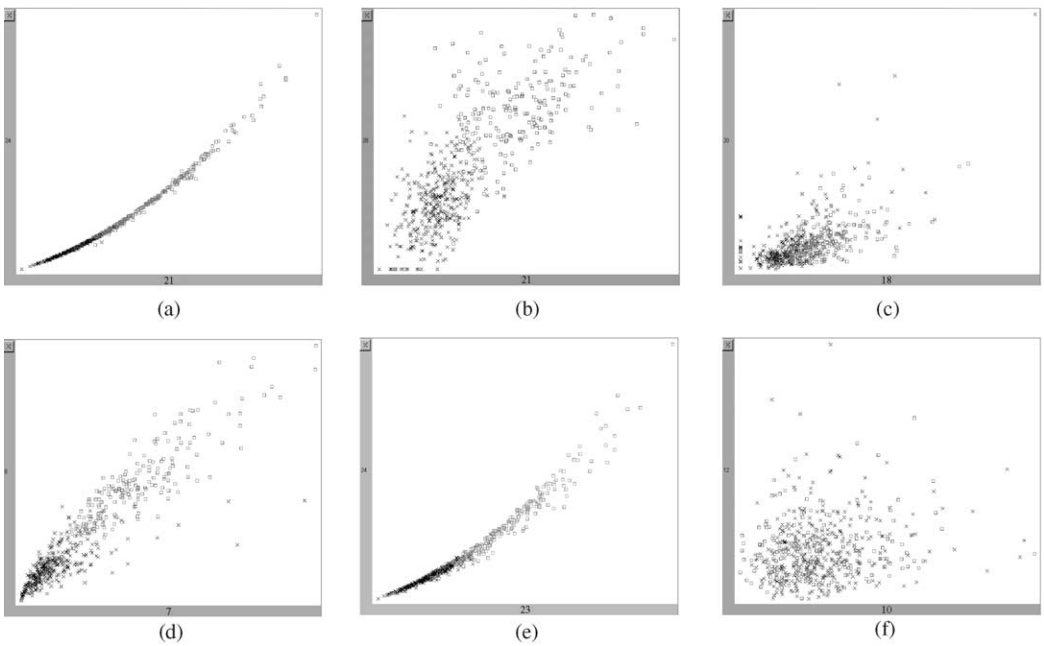


Fig. 6. Visual comparison of 2D subspaces found on less separable 30-dimensional mammogram data by maximizing: (a) Bhattacharyya (the same was found by Divergence), (b) Generalized Mahalanobis, (c) Patrick–Fischer distances. Mixture model methods using 5 components results: approximation method—(d), divergence method—(e). Picture (f) demonstrates a subspace unsuitable for discrimination (found by minimizing the Bhattacharyya distance).

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 minimizing the Bhattacharyya distance that has been found is shown for illustration only.

5.4. A different view of criterion functions—visual subspace comparison

The FST may be used to obtain a visual illustration of selected feature subsets. Our examples illustrate spatial properties of different data sets (easily separable 3-class marble set in Fig. 5, a poorly separable 2-class mammogram set in Fig. 6 and the speech set). We selected feature pairs yielding optimal values of different criteria. Figs. 5(a)–(c) illustrate subsets obtained by means of optimizing different probabilistic distance measures, Fig. 5(d) illustrates the Approximation method (5 components), and Fig. 5(e) the Divergence method (5 components). As opposed to subsets selected for class discrimination the picture (f) illustrates an example of "bad" feature pairs being not suitable for discrimination. Fig. 5(f) was obtained by means of minimizing the Bhattacharyya distance.

6. Implementation issues

FST software has been developed over a period of three years. It has a form of 32-bit Windows application. The kernel incorporates all the procedures written in ANSI C language. This kernel is connected to a user interface which has been developed in (Sybase) Powersoft Optima++ 1.5 RAD compiler (today known as Power++). Most of the programming work was done by 1–2 programmers; theoretical questions, definitions and specifications are consulted within a team of 4–5 programmers and researchers [16]. The programming work was focused on keeping high quality of kernel functions. As describing all kernel code properties would go beyond the scope of this paper, we will mention their general properties only.

Dimensionality reduction algorithms may often be very time-consuming. The kernel code is therefore optimized for speed (especially when accessing complicated multidimensional memory structures). The speed of criterion function value computations is the most important issue when programming such enumeration algorithms, where the criterion value is repeatedly calculated. Even if speed was the main goal, we did not omit mechanism for error recovery, etc. (e.g. incorrect properties of data in file).

Most subset search algorithms are defined in two forms according to the prevailing direction of search: forward and backward. The forward search starts with an empty feature subset. Features are then added to it stepwise. The backward search starts with the full set from which features are removed in a stepwise manner. However, adding and removing steps may be combined in the course of one algorithm. Single steps may process not only single features, but also groups of features. In order to be able to implement even complex variants of algorithms like, e.g., oscillating search, it was necessary to develop some fast and flexible way of working with features in such complicated algorithms. For this purpose, we use a special vector (having the same size as the full feature set) representing states of every single feature. In general, positive values represent currently selected features, other values represent excluded features. Different values denote features in different states of processing (definitely selected feature, conditionally selected feature, etc.). We mention the existence of this vector because of its following advantage: by a relatively simple exchange of values of several variables, we are able to switch the search direction as well as other algorithm properties. As a result, the coding is simplified since just one code is needed for either search direction (forward or backward) only; switching to the opposite one is then simple. It should be noted, that such a "compact" code does not reduce the algorithm speed in comparison to separate algorithm versions. Moreover, the code has become more lucid and the debugging time decreased, too. Our coding approach allows also a relatively straightforward implementation of very complicated versions of combined algorithms.

7. Future work and applications

Results obtained using the FST 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 Ref. [17]). FST 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 into using the method most convenient for the problem at hand.

We believe that not only pattern recognition community but also researchers from various other branches of science may benefit from our work.

8. Concluding summary

The most common task in Pattern Recognition is classification of patterns into a proper class. Problems like this can be considered as a part of recognition tasks, computer-aided decision tasks and other applications as well. One of the most important intermediate steps in these tasks is the problem of dimensionality reduction. In order to reduce the problem dimensionality we often use “feature selection” methods because of their relative simplicity and meaningful interpretability of results. When combined with data approximation methods, the dimensionality reduction process may result in substantial data compression, while the overall statistical properties remain preserved. Moreover, different ways of data manipulations and queries become possible without need of access to original data (which may thus become redundant).

A software package developed for the purpose of feature selection in statistical pattern recognition is presented in the paper. The software tool includes both several classical and new methods suitable for dimensionality reduction, classification and data representation. The software is meant to be used both for practical and educational purposes. Therefore it implements many of currently known feature selection methods to enable their comparison.

Three particular method groups are implemented: (a) sub-optimal sequential search methods like SFS, SBS, SFFS, SBFS, Plus-L-Minus-R, generalised methods, A SFFS, several OS versions, etc., (b) optimal methods like Exhaustive Search, classical BB and its extended versions, predictive BB etc., and (c) normal mixture approximation model based methods for combined process of data representation and feature selection. Both (a) and (b) type methods are based on normal distribution estimates and can be used to maximize one of 6 implemented probabilistic distance measures (Bhattacharyya distance, Mahalanobis distance etc.).

The paper discusses the particular methods briefly regarding their implementation and gives references for further reading. Particular methods behavior and applicability is demonstrated on examples of solved problems.

An alternative approach to criterion function comparison is presented as well. The software enables observing the obtained feature subsets subspaces. A comparison of standard probabilistic distance measures is given showing the differences of obtained equally sized feature subsets. Another example demonstrates the differences of single-feature evaluation based on different measures. The examples show the fact, that probabilistic distance measures have a limited connection to the “ground truth” only and should be therefore used carefully.

Acknowledgements

The work has been supported by grants of the Czech Ministry of Education CEZ:J18/98:31600001, Aktion 29p7, EuroMISE Kardiocentrum LN 00 B 107, and the Grant Agency of the Czech Republic No. 402/01/0981.

References

[1] P.A. Devijver, J. Kittler, Pattern Recognition: a statistical approach, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[2] T.M. Cover, J.M. Van Campenhout, On the possible orderings in the measurement selection problem, IEEE Trans. System Man Cybernet. SMC-7 (1977) 657–661.

[3] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd Edition, Academic Press, New York, 1990.

[4] P.M. Narendra, K. Fukunaga, A branch and bound algorithm for feature subset selection, IEEE Trans. Comput. C-26 (1977) 917–922.

[5] B. Yu, B. Yuan, A more efficient branch and bound algorithm for feature selection, Pattern Recognition 26 (1993) 883–889.

[6] P. Somol, P. Pudil, F.J. Ferri, J. Kittler, Fast branch & bound algorithm in feature selection, Proceedings of the SCI 2000 Conference, Vol. IIV, Orlando, FL, 2000, pp. 646–651.

[7] P. Somol, P. Pudil, J. Grim, Branch & bound algorithm with partial prediction for use with recursive and non-recursive criterion forms, Proceedings of the Second International Conference on Advances in Pattern Recognition ICAPR 2001, Rio de Janeiro, 2001, Lecture Notes in Computer Science, Vol. 2013, pp. 230–239, Springer, Berlin.

[8] A.K. Jain, D. Zongker, Feature selection: evaluation, application and small sample performance, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1997) 153–158.

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

[10] P. Somol, P. Pudil, J. Novovičová, P. Paclík, Adaptive floating search methods in feature selection, Pattern Recognition Lett. 20 (11/13) (1999) 1157–1163.

[11] P. Somol, P. Pudil, Oscillating search algorithms for feature selection, Proceedings of the 15th International Conference on Pattern Recognition, Vol. 2, Barcelona, 2000, pp. 406–409.

[12] P. Pudil, J. Novovičová, N. Choakjarerwanit, J. Kittler, Feature selection based on the approximation of class densities by finite mixtures of special type, Pattern Recognition 28 (9) (1995) 1389–1397.

[13] J. Novovičová, P. Pudil, J. Kittler, Divergence based feature selection for multimodal class densities, IEEE Trans. Pattern Anal. Mach. Intell. 18 (2) (1996) 218–223.

[14] L. Sardo, J. Kittler, Model complexity validation for PDF estimation using Gaussian mixtures, Proceedings of the 14th International Conference on Pattern Recognition, Vol. 2, Brisbane, 1998, pp. 195–197.

[15] P. McKenzie, M. Alder, Initializing the EM algorithm for use in Gaussian mixture modelling, Technical Report, University of Western Australia, 1994.

[16] P. Pudil, J. Novovičová, Novel methods for subset selection with respect to problem knowledge (special issue on feature transformation and subset selection), IEEE Trans. Intell. Systems (1998) 66–74.

[17] H.A. Mayer, P. Somol, R. Huber, P. Pudil, Improving statistical measures of feature subsets by conventional and evolutionary approaches, Proceedings of the 3rd IAPR International Workshop on Statistical Techniques in Pattern Recognition, Alicante, 2000, pp. 77–81.

About the Author—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 is also with the Medical Informatics Unit, IPH, University of Cambridge, United Kingdom. His current activities include development of feature selection techniques, mixture modelling algorithms and decision support systems. He is particularly interested in statistical pattern recognition. His part-time interests include both theoretical and practical computer graphics, design and graphical arts.

About the Author—PAVEL PUDIL is currently the Head of Pattern Recognition Department at the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic and the Vice-Dean for research and science at the Prague University of Economics. In 1996–2000, he was the Chairman of IAPR Technical Committee on "Statistical Techniques in Pattern Recognition". His primary research interests include statistical approach to pattern recognition, particularly the problem of dimensionality reduction and its applications in economics, management and medical diagnostics. He spent as a Research Fellow altogether 5 years at British universities (Cambridge, Surrey). After graduating from the Czech Technical University in Prague in 1964 he received a Ph.D. degree in Technical Cybernetics in 1970 and was appointed Professor in 2001. He is a member of the Czech Society for Cybernetics and Informatics, Czech Pattern Recognition Society, Society for Biomedical Engineering, a member of IEEE and the IAPR Fellow.