ISSN 0005–1144
ATKAAF 43(1–2), 39–46 (2002)

Davor Antonić, Mario Žagar

Heuristic Algorithms for Extracting Relevant Features in Signal Analysis

UDK 621.391:004.93
IFAC IA 5.8.6

Preliminary communication

Extraction of relevant features is essential stage in a pattern recognition and classification system. Goal of the feature extraction algorithm is to find feature subset where relevant information for recognition is contained in minimal number of features. Proposed algorithms are based on the assumption that features with better individual discrimination ability will also be better in combination with other features. Features are first extracted from the initial set, then sorted according to their individual fitness. Sorted set is used to form the search tree. Two heuristic algorithms are proposed: the first one performs the depth first search, bounded with required increase of fitness function and the second one is based on genetic algorithm. Their performances are compared with complete search and sequential search (FSS, BSS) algorithms.

Key words: feature extraction, pattern recognition, signal analysis

1 INTRODUCTION

Efficiency of the pattern recognition system depends on the quality of the set of features representing object properties relevant for classification and recognition. According to the type of information they contain, features can be divided into two main groups. Features from the first group are related to human perception of the world. They describe some observable attribute of the examined object, e.g. medical symptoms for determination of disease. Second group relies on abstract information provided by certain sensor, e.g. signal energy in some part of the spectrum. In this case it is much harder to extract relevant features, because signal from the sensor contains almost indefinite number of possible features. This type of features is important in problems requiring machine interface to the real world, including man-machine interfaces, robotics and various sensing and detection tasks.

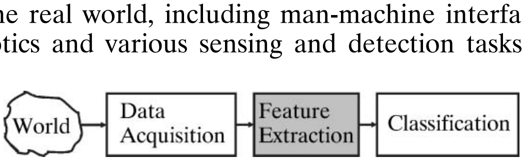

Fig. 1 Parts of the signal analysis system

Typical signal analysis system is shown in Figure 1. Data acquisition subsystem reads real world data through the sensor and appropriate hardware. Feature extraction subsystem reduces raw signal to information relevant for classification and classification subsystem assigns the analyzed sample to the associated class.

The task of the feature extraction subsystem is to automatically generate initial feature set directly from the training samples and reduce it to the minimal subset containing information relevant for classification and recognition.

Proposed method automatically generates initial feature set directly from the acquired signal. Initial set contains large number of highly redundant features (typically more than 10^4). Since complexity of typical feature extraction algorithms varies between polynomial $O(d^2)$ and exponential $O(2^d)$, where d is number of features in a set, it is not feasible to search any significant part of the feature space.

Although possible number of features is very large, number of non-redundant features is significantly smaller. If features are generated e.g. from the spectrum of certain signal, number of non-overlapping frequency ranges is limited. For such problems viable solution is to use a hybrid approach, like in [6]. In the first stage, initially large number of features is reduced by selecting individually best features and removing all features overlapping with the selected one. In the second stage, on such reduced set of some 10^2 features it is possible to apply some of the classical feature extraction algorithms, to extract features containing information relevant for classification.

Proposed heuristic feature extraction algorithms uses knowledge about the quality of individual features, collected during reduction of initial feature set. Feature space search is guided by an assumption that the feature that is better individually is also likely to produce better overall result in combination with other features. This is achieved by appropriate construction of the search tree and definition of the pruning criteria, and by adjusting parameters of the genetic algorithm [12] respectively. Proposed algorithms are compared with some of the well known feature extraction algorithms [5]: Complete search, and standard versions of Forward and Backward sequential selection (FSS and BSS) [1, 7, 8, 10] according to the quality of extracted feature sets and efficiency.

2 AUTOMATIC GENERATION OF THE INITIAL FEATURE SET

Initial feature set may contain various data collected from the signal. It should be created by consistently applying appropriate transformation to different parts of the signal.

In the case of spectral analysis, feature is defined as an average energy in frequency window of variable width and position, represented by frequency interval $[f_0, f_0+\Delta f_i]$, where f_0 takes values from interval $[f_{\min}, f_{\max}-\Delta f_i]$. Window width, Δf_i, takes discrete values from the interval $[f_s/N, f_{\max}]$, where f_s is sampling frequency, and N is the number of samples, and also number of frequency channels in the spectrum. It is appropriate to express parameters $f_{\min}$, $f_{\max}$, and Δf in channel numbers, where the channel width is defined as f_s/N.

If the relevant part of the spectrum contains 1 024 channels, and above parameters are defined as:

$$f_{\min}=1\cdot f_s/N,\ f_{\max}=1\,024\ f_s/N,$$
$$\Delta f_i \in \{8,\ 12,\ ...,80\}\cdot f_s/N,$$

initial set will contain $19\cdot 1\,024-4\cdot(2+3+\ldots+20) = = 18\,620$ features. Size of the corresponding feature space is about $10^{5\,600}$, which is unmanageable for any feature extraction algorithm. Therefore, it is necessary to reduce the number of the initial features to the manageable size.

3 REDUCTION OF THE INITIAL FEATURE SET

Initial feature set is highly redundant, due to many features containing information from the same channel. In the above example minimal width of the frequency window is eight channels, which implies that reduced feature set may contain at most 128 non-overlapping features, covering frequency range of 1024 channels. Initial feature set is reduced by selecting individually best features. Fitness function for each feature is calculated individually, and the best f features are chosen. Fitness function used for feature evaluation is a ratio between average Euclidean distance between instances from different classes, and average distance between instances belonging to the same class (1), [4, 9].

$$D_1 = \frac{\sum_{i=1}^{C}\sum_{j=1}^{n_i-1}\sum_{k=j+1}^{n_1} d(x_{ij},x_{ik})}{\sum_{i=1}^{C}\binom{n_i}{2}}$$

$$D_0 = \frac{\sum_{i=1}^{C-1}\sum_{j=i+1}^{C}\sum_{k=1}^{n_i}\sum_{l=1}^{n_j} d(x_{ij},x_{jl})}{\binom{n}{2}-\sum_{i=1}^{C}\binom{n_i}{2}}, \tag{1}$$

where:

C – number of classes
n – total number of instances
n_i – number of instances in i-th class
x_{ij} – j-th instance from i-th class
$d(x_{ij}, x_{kl})$ – Euclidean distance between x_{ij} and x_{kl}

Because that algorithm does not take into consideration interactions between features, if applied alone usually gives poor results. Pseudo code of the algorithm for extracting individually best features from the initial feature set is shown in Figure 2.

```
sort InitialSet according to Fitness
extract best feature from InitialSet
remove overlapping features form InitialSet
```

Fig. 2 Algorithm for extracting individually best features

Array *InitialSet* contains the value of the fitness function for each feature from the initial set, as well as starting and ending frequency of the frequency range covered by the particular feature. In each iteration, initial set is sorted according to the fitness and the current best feature is extracted. All features whose frequency range overlap with the range of the extracted feature (i.e. they are partially redundant) are removed from the set. Described procedure is repeated until all features are either extracted or removed from the initial set. Features from the reduced set cover entire frequency range, and contain features whose frequency ranges do not overlap.

The first few features will usually have significantly higher individual discrimination ability. Therefore, it is justified to organize search in such a way that will favor them. Feature extraction algorithm should also take into consideration other features, which may also contain significant information that should be included into the extracted feature subset.

4 GENETIC ALGORITHM

Genetic algorithms use principle of operation similar to the selection process known from the evolution. Population is usually formed from the constant number of individuals, representing samples from the search space. In this case, individuals are feature subsets containing different features.

New individuals for the next generation are formed by applying two genetic operators: *crossover* and *mutation*, to the individuals from the current generation. Crossover operation randomly selects one or more points in two selected individuals and exchange their segments to form new individuals. Mutation randomly changes certain number of components within selected individuals. It is used to in-troduce new information into the population (e.g. add certain feature), thus avoiding the search to be trapped into the local minimum.

Parents that will produce new individuals are chosen according to their fitness, so better individuals are more likely to pass their genes to the next generation. Therefore each generation will have better overall fitness. It is appropriate to pass certain number of best individuals directly to the next generation, which is called *elitism*.

Proper representation and adequate evaluation function are the key issues in applying genetic algorithm to the particular problem. For the problem of feature extraction, representation is straightforward. Feature subset is represented by a binary string of length N, where each bit represents the presence or absence of corresponding feature [12]. Crossover and mutation can be directly applied to such defined population.

Fitness function defined for the tree-searching algorithm cannot be applied directly. It is monotonic, which will guide the algorithm toward producing feature subsets containing all features. Therefore, fitness function is modified according to equation (2):

$$f' = \frac{f}{1.1^{N_f}}, \qquad (2)$$

where f is value of the original fitness function, and N_f is number of features in the particular feature set. That modification emphasizes feature sets containing smaller number of features. Currently there is no method to automatically adjust fitness function. Therefore, denominator in equation (2) is chosen experimentally. Chosen value will prefer feature subsets containing between three and eight features.

Parameters of the genetic algorithm are set as follows. Population consists of 30 individuals, and each individual is represented by binary string whose length corresponds to the number of features in the reduced set. Two best individuals are transferred directly to the next generation. They are allowed to produce new individuals through crossover, and they are not subject of mutation. Crossover rate is set to 100 % (i.e. 15 crossovers are performed to produce 30 new individuals). Mutation rate is one percent of all bits in population, i.e. about one mutation per individual is expected.

The main problem is that reduced set still contains large number of features, with only a few relevant. Generic genetic algorithm performs poorly, primarily because all features have equal probability to be included in the population. Also, if initial population is initialized as usual, with 50 % bits set, feature sets will contain on the average half of the number of features from the reduced set. Number of relevant features in signal spectrums depends on application, but it is usually bellow ten features. Larger number of features is rarely justified, because it complicates recognition algorithm. Number of features in a set will slowly decrease mainly due to the mutation. Therefore two modifications to the generic algorithm are proposed and tested: initialize population to average smaller number of features, and increase the probability of crossover point being among individually better features. Initial population is generated with 10 % probability for each feature to be included in particular individual (feature set). Moreover, probability distribution for crossover point is adjusted to prefer crossover of individually better features. Probability density function is shown in Figure 3.

The probability that crossover point will be among of the best ten features is almost 40 %. Analysis of the population has shown that irrelevant features decay from the population being removed by mutation.

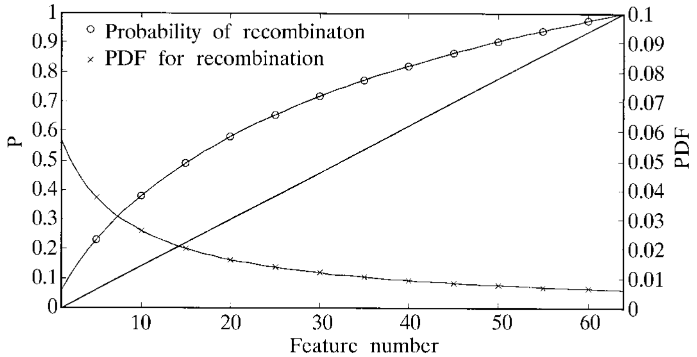

Fig. 3 PDF for determining the crossover point

5 HEURISTIC PRUNING OF THE SEARCH TREE

Feature space of some 10^2 features is still too large to be searched exhaustively. Evaluating all possible combinations of e.g. 100 features will require 10^{30} iterations. Knowledge about the feature space can guide the search, significantly reducing the size of the search space. In proposed algorithm information about the quality of individual features, attained during the reduction of the initial set, will be used.

Algorithm performs the depth first search, bounded with required increase of fitness function. Features are sorted according to their fitness, directing the search to earlier evaluate combinations of individually better features. Example of the search tree for the set of five features is presented in Figure 4.

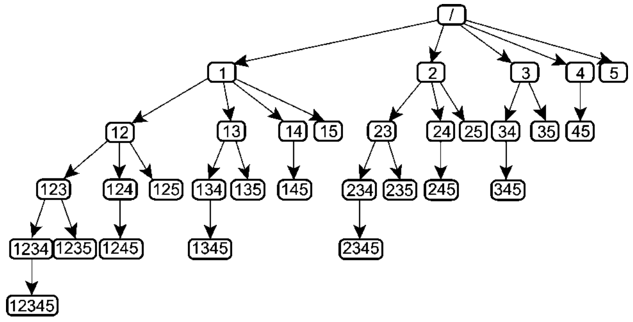

Fig. 4 Search tree

Figure 5 presents sequence of visiting nodes.

```
1 2        | 1 3      | 2 3      | 3 4
1 2 3      | 1 3 4    | 2 3 4    | 3 4 5
1 2 3 4    | 1 3 4 5  | 2 3 4 5  | 3 5
1 2 3 4 5  | 1 3 5    | 2 3 5    | 4
1 2 3 5    | 1 4      | 2 4      | 4 5
1 2 4      | 1 4 5    | 2 4 5    | 5
1 2 4 5    | 1 5      | 2 5      |
```

Fig. 5 Search order

Without restriction, algorithm will perform the exhaustive search of the feature space. Therefore it is necessary to bind the search tree. In the proposed algorithm, bound is defined by minimum increase in fitness, Δf, between the two successive nodes. In the above example, if the increase of fitness between the feature subsets **12** and **123** is less than required minimum, whole subtree from the node **123** is pruned, and algorithm immediately proceeds to the node **13**, discarding nodes **124**, **125**, and all their successors. Such a policy is too restrictive, because it gives too much importance to the fitness of individual features.

On the other hand, examining all nodes at the certain level will significantly increase the number of iterations. Testing the algorithm on real data shows that even with such restrictive policy, algorithm produces comparable results in fraction of time required by other algorithms.

Pseudo code of the algorithm is shown in Figure 6.

```
// Return from recursion if the bottom of the
// tree is eventualy reached
if Level>MaxDepth
  return
end

if Level==1
  StartingNode = 0
else
  StartingNode = FeatureSet(Level-1)
end

// Visiting child nodes
for j = StartingNode+1 to MaxDepth
  FeatureSet(Level) = j
  PF = Fitness (X,f)
  // Testing the prunning condition
  if Level>1 & PF - PF0 < $\Delta f$
    return
  end
  store FeatureSet and PF
  // Enter the subtree
  HeurTree (Level+1,FeatureSet,MaxDepth,PF)
end
```

Fig. 6 Algorithm for heuristic pruning

Core of the algorithm is the recursive function *HeurTree* that generates the search tree. At the beginning, the function checks whether the bottom of the tree is reached. The pruning condition will usually bound the search much earlier. Then the starting number of the added features is determined. Except for the first level, features are added from the number of the last feature in the parent node upwards. In the main loop new feature is added to form the subset for the current node. Subset is evaluated using *Fitness* function. Array X contains feature values for all samples. Pruning condition is tested on each but the first level. If the fitness gain is below predefined minimum Δf, the subtree is pruned. If the condition is satisfied, evaluated feature subset is stored together with corresponding fitness. Recursive call of the same function enters the subtree.

Fitness function needs some explanation. In this stage different fitness function is used. Since features in reduced feature set are selected according to their discrimination ability, they already have the intrinsic property of grouping together instances from the same class. Fitness function is defined as minimal distance between neighboring classes. In this way feature subsets resulting in balanced distribution of classes will prevail subsets that well separate one class, leaving the others close together. It could be proved that such defined function satisfies the *monotonic property*, meaning that by adding new feature to any subset, fitness value will remain the same or increase.

Due to the monotonic property, the »best« subset will be the one containing all features, because it will have the highest fitness. But from the classifier point of view, it is better to work with subset containing smaller number of features and having sufficient discrimination ability. For this reason, described algorithms classify extracted feature subsets according to the number of features in a set. Therefore it is possible to determine the best subset containing appropriate number of features.

6 EXPERIMENTAL RESULTS

The sensor used for testing and evaluation of proposed algorithms is simple demining prodder equipped with microphone [2, 3, 4], shown in Figure 7.

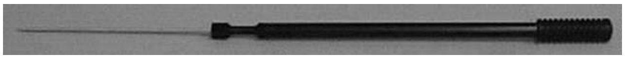

Fig. 7 Demining prodder

Microphone placed inside the prodder handle registers vibrations generated by touching the buried object with the prodder tip. Objective of the described sensor and the signal analysis subsystem is to recognise the material of the buried object based on acoustic signal generated from the prodder. Experiments were restricted to four different materials: wood, plastic, iron, and stone.

Signals are recorded using PC sound card, at the sample rate of 48 kHz. For each signal 8192 samples were collected, corresponding to the interval of 170 ms. Feature analysis is performed on signal spectrums by analyzing average signal energy across different frequency windows. Squared fast Fourier transform (FFT) is performed on normalized signals to determine signal energy in different frequency ranges. Analysis is performed on first 1024 frequency channels, covering the frequency range from 0 to 6 kHz. Figure 8 presents energy spectrums for given samples.

For each material, spectrums for 20 samples are shown at the same plot. Parts of the spectrum that are different for different samples and similar for the same sample are good feature candidates.

Forty measurements were performed for each of the samples of four different materials, making total of 160 measurements. Half of the measurements were used for training, and the other half for testing extracted feature sets.

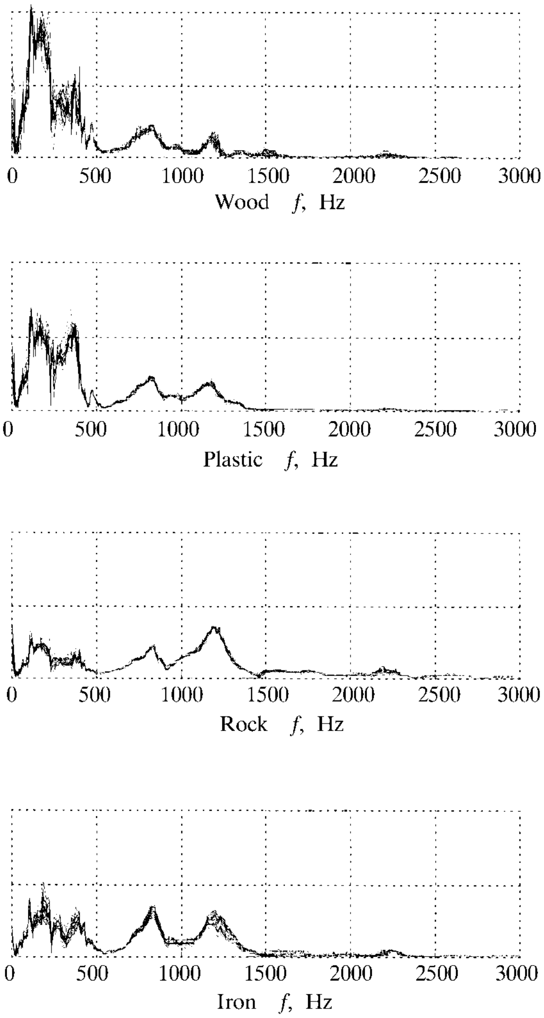

Fig. 8 Spectrums

As described, feature is defined as an average energy in frequency interval $[f_0, f_0 + \Delta f]$, where f_0 takes values from interval $[f_{\min}, f_{\max} - \Delta f]$, $f_{\min} = 5.9$ Hz, $f_{\max} = 6$ kHz. Δf is the window width and takes discrete values from the set $\{8, 12, ..., 80\}$ channels, that correspond to frequencies from approximate 47 Hz to 470 Hz. This gives initial set of 19 247 features.

Values of fitness function for all features in initial set are shown in Figure 9. Mean value of the corresponding frequency range is used as a reference point where fitness value is plotted.

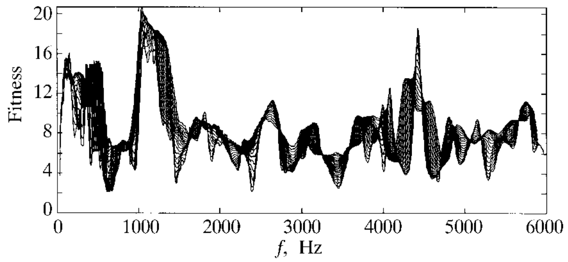

Fig. 9 Values of fitness function for the initial feature set

Values of fitness function for extracted individually best features are shown in Figure 10. Dots present values of fitness function calculated for single channels.

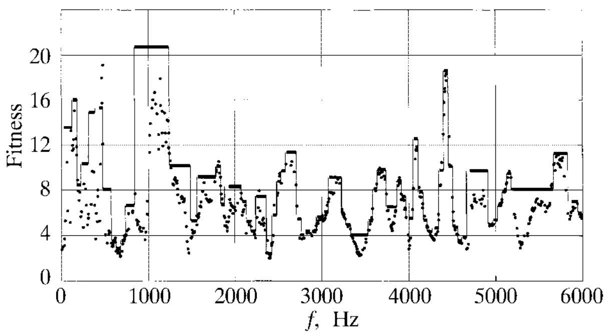

Fig. 10 Values of fitness function for the reduced feature set

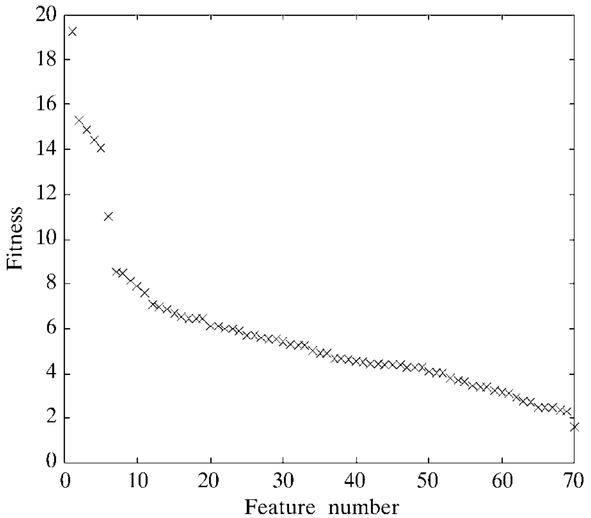

Fig. 11 Values of fitness function for extracted features

Figure 11 presents values of the fitness function for extracted features, sorted by individual fitness.

Performance of two proposed algorithms is compared with Complete search, and Forward and Backward sequential selection algorithms. Complete search performs exhaustive search of all subsets containing up to 10 features, from the set of individually best 15 features. It will always find an optimal solution but with cost of computational time. Even performed on such a reduced set, it requires 30.826 iterations to evaluate all feature subsets.

Forward and backward sequential selection [1, 7, 8, 10] are the most common sequential search algorithms. FSS begins with zero features, evaluates all subsets with exactly one feature and select the one with largest fitness function. It evaluates all subsets with previously selected feature and one of the remaining features, then again selects one with largest fitness function. This cycle repeats while improvement by adding the new feature is above the predefined level. BSS instead begins with all features and repeatedly removes a feature whose removal causes the least decrease of the fitness function value.

Figure 12 presents comparison of algorithms according to the quality of the extracted feature set. Results of the Complete search presents the upper bound, restricted to evaluated subset of the search space. FSS and BSS perform poorly, mainly because they are too restrictive in selecting feature sets to be evaluated. FSS selects feature that produces highest gain in the limited context of previously selected features. Once selected, feature cannot be removed even if spoiling the set. BSS removes the feature whose removal results in lowest decrease of overall fitness. Starting with all features, where contribution of particular features is not so obvious, it is likely to remove the good feature early in the process.

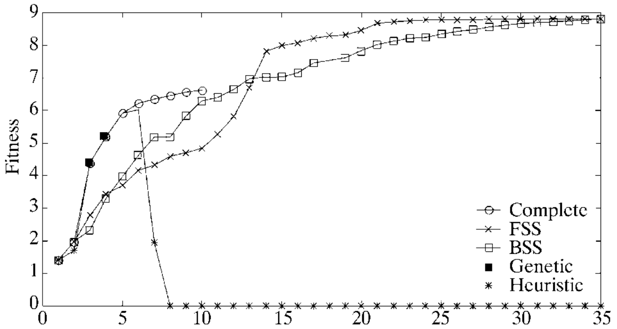

Fig. 12 Comparison of algorithms according to the extracted feature sets

Algorithm with heuristic tree pruning, even with restrictive search policy gives good results. With Δf of 0.1, it finds the same subsets containing three to five features as Complete search, requiring only 385 iterations. Decreasing the value of Δf will widen the search space, making possible finding of better subsets containing more features, but at the cost of number of iterations. The main drawback of proposed algorithm is sensitivity to the value of Δf. Figure 13 presents relationship between value of Δf and required number of iterations for described samples.

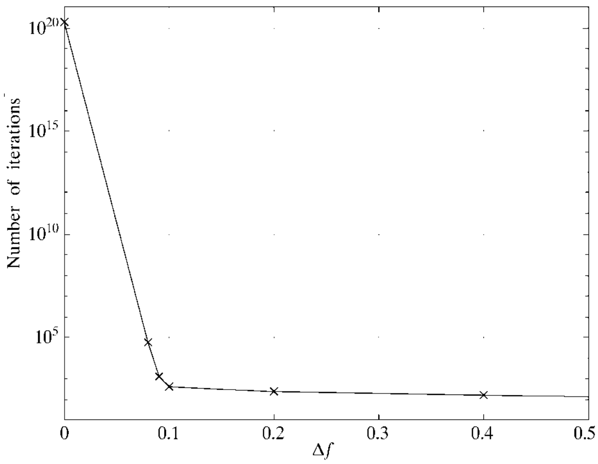

Fig. 13 Required number of iterations for different Δf

For $\Delta f = 0$ algorithm turns into the Complete search, requiring 10^{21} iterations for the space containing 70 features. The optimal value of Δf is at the curve knee. Currently, the proper value of Δf is determined experimentally, by starting with some larger value (e.g. 10 % of the maximum value of the fitness function of the subset containing all features), examining the results and decreasing the value of Δf.

The criterion for entering the subtree is the weakest point of the proposed algorithm. Numbers of modifications are possible, and the most promising one is to incorporate the global measure of the quality of currently evaluating feature subset. Since algorithm keeps records of all evaluated feature subsets, it is possible to compare current feature subset to the current best subset having the same number of features and use that information as a subtree entering criterion.

Genetic algorithm finds best subsets containing three and four features, requiring 22 generations (average for ten runs). Modification of the fitness function (e.g. denominator in equation (1)) will make possible finding of better subsets containing more features. Figure 14 presents comparison of described algorithms according to the execution time.

Figure 15 presents distribution of samples for the best subset containing two features, and Figure 16 presents distribution for the best subset containing three features. Subsets containing more than three features are hard to visualize.

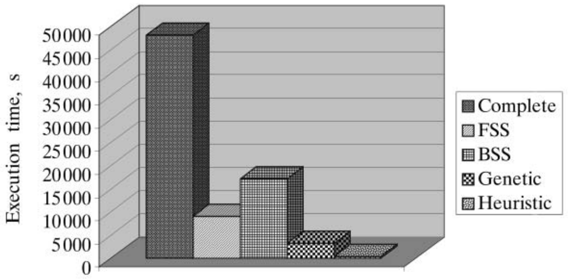

Fig. 14 Comparison of algorithms according to the execution time

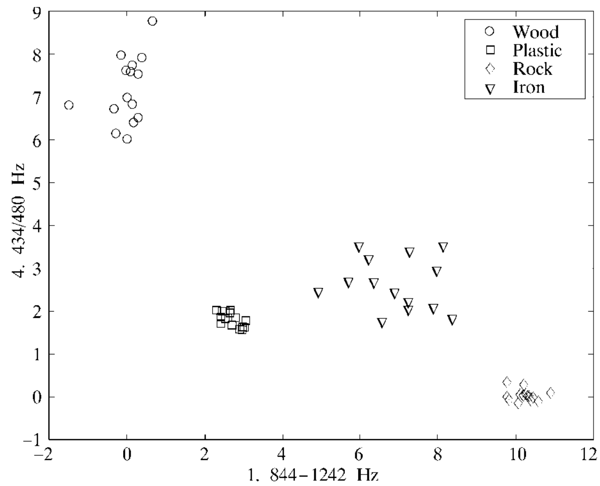

Fig. 15 Distribution of samples for different materials for two best features

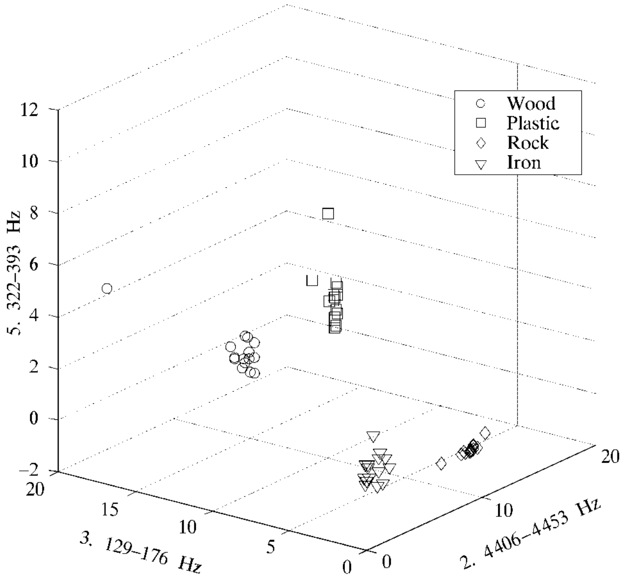

Fig. 16 Distribution of samples for three best features

7 CONCLUSION

For problems where domain knowledge is costly to exploit or unavailable, which is often the case in signal analysis, viable approach is to automatically generate large number of candidate features, and gradually reduce them toward the more condensed representation, using knowledge acquired during the process. Currently the process of tuning various algorithm parameters is iterative, containing human inside the loop. Further research will include automation of parameter adjustments according to the requirements of the problem.

ACKNOWLEDGEMENT

Thanks to the reviewers for their comments and suggestions.

REFERENCES

[1] D. W. Aha, R. L. Bankert, **A Comparative Evaluation of Sequential Feature Selection Algorithms.** In. Fisher, D., Lenz, J. H. (Eds.), Artificial Intelligence and Statistics, Springer--Verlag, New York, 1996.

[2] D. Antonić, I. Ratković, **Ground Probing Sensor for Automated Mine Detection.** KoREMA'96 – 41st Annual Conference, Opatija, Croatia, pp. 137–140, September 18–20, 1996.

[3] D. Antonić, **Improving the Process of Manual Probing.** SusDem'97 – International Workshop on Sustainable Humanitarian Demining, Zagreb, Croatia, pp. 4.30–4.32, Sept. 29–Oct. 1, 1997.

[4] D. Antonić, M. Žagar, **Method for Determining Classification Significant Features from Acoustic Signature of Mine--like Buried Objects.** 15th World Conference on Non-Destructive Testing, Rome, Italy, Oct. 15–21, 2000.

[5] M. Dash, H. Liu, **Feature Selection Methods for Classification.** Intelligent Data Analysis, Vol. 1, No. 3, pp. 131–156, 1997.

[6] M. Dash, H. Liu, **Hybrid Search of Feature Subsets.** Pacific Rim 5th International Conference on Artificial Intelligence, pp. 22–27, Singapore, 1998.

[7] P. A. Devijver, J. Kittler, **Pattern Recognition: A Statistical Approach.** Prentice Hall, New York, 1982.

[8] M. A. Hall, L. A. Smith, **Practical Feature Subset Selection for Machine Learning.** Proceedings of Australian Computer Science Conference, pp. 181–191, Perth, 1998.

[9] A. K. Jain, R. C. Dubes, **Algorithms for Clustering Data.** Prentice Hall, New Jersey, 1988.

[10] D. Koller, M. Sahami, **Toward Optimal Feature Selection.** ICML-96: Proceedings of the Thirteenth International Conference on Machine Learning, pp. 284–292, San Francisco, 1996.

[11] ..., **Matlab** – Signal processing toolbox user's guide, MathWorks Inc, Natick, 1999.

[12] H. Vafaie, K. De Jong, **Robust Feature Selection Algorithms.** Proceedings of the 5th IEEE International Conference on Tools for Artificial Intelligence, pp. 356–363, IEEE Press, Boston, 1993.

Heuristički postupci izdvajanja značajki u obradi signala. Izdvajanje relevantnih značajki je ključan korak u sustavu za raspoznavanje uzoraka i klasifikaciju. Cilj postupka izdvajanja značajki je pronalaženje najmanjeg skupa značajki koji sadrži informacije potrebne za raspoznavanje uzorka. Predloženi postupak temeljen je na pretpostavci da će značajke koje pojedinačno bolje razlikuju uzorke iz različitih klasa to svojstvo imati i u kombinaciji s drugim značajkama. Nakon izdvajanja iz početnog skupa, značajke se sortiraju po padajućoj vrijednosti kriterijske funkcije. Iz sortiranog skupa značajki formira se stablo pretraživanja, tako da će skupovi koji sadrže pojedinačno bolje značajke biti pretraženi prije. Predložena su dva postupka izdvajanja značajki: prvi provodi pretraživanje stabla po dubini ograničeno zadanim porastom vrijednosti kriterijske funkcije, a drugi je temeljen na genetskom algoritmu. Postupci su prema kvaliteti izdvojenih skupova značajki i efikasnosti uspoređeni s postupkom potpunog pretraživanja i slijednim postupcima (FSS, BSS).

Ključne riječi: analiza signala, izdvajanje značajki, raspoznavanje uzoraka

AUTHORS' ADDRESSES:

Dr. sc. Davor Antonić, dipl. ing.,
HR-10000 Zagreb, Klaićeva 21, CROATIA

Prof. Dr. sc. Mario Žagar, dipl. ing.
Faculty of Electrical Engineering and Computing,
HR-10000 Zagreb, Unska 3, CROATIA

Received: 2002–10–05