



ELSEVIER

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

SCIENCE @ DIRECT®

Applied Soft Computing 6 (2006) 337–347

Applied Soft  
Computing

[www.elsevier.com/locate/asoc](http://www.elsevier.com/locate/asoc)

# Genetic algorithms in classifier fusion

Bogdan Gabrys<sup>a,\*</sup>, Dymitr Ruta<sup>b</sup>

<sup>a</sup> *Computational Intelligence Research Group, School of Design, Engineering and Computing,  
Bournemouth University, Talbot Campus, Fern Barrow, Poole BH12 5BB, UK*

<sup>b</sup> *British Telecom, Research and Venturing, Adastral Park MLB1, pp12, Martlesham Heath, Ipswich IP5 3RE, UK*

## Abstract

An intense research around classifier fusion in recent years revealed that combining performance strongly depends on careful selection of classifiers to be combined. Classifier performance depends, in turn, on careful selection of features, which could be further restricted by the subspaces of the data domain. On the other hand, there is already a number of classifier fusion techniques available and the choice of the most suitable method depends back on the selections made within classifier, features and data spaces. In all these multidimensional selection tasks genetic algorithms (GA) appear to be one of the most suitable techniques providing reasonable balance between searching complexity and the performance of the solutions found. In this work, an attempt is made to revise the capability of genetic algorithms to be applied to selection across many dimensions of the classifier fusion process including data, features, classifiers and even classifier combiners. In the first of the discussed models the potential for combined classification improvement by GA-selected weights for the soft combining of classifier outputs has been investigated. The second of the proposed models describes a more general system where the specifically designed GA is applied to selection carried out simultaneously along many dimensions of the classifier fusion process. Both, the weighted soft combiners and the prototype of the three-dimensional fusion–classifier–feature selection model have been developed and tested using typical benchmark datasets and some comparative experimental results are also presented.

© 2005 Elsevier B.V. All rights reserved.

*Keywords:* Genetic algorithms; Classification; Classifier fusion; Feature selection; Classifier selection

## 1. Introduction

Over a number of recent years an increasing scientific effort has been dedicated to the development and studies of multiple classifier systems (MCS) [17,18]. It has been frequently demonstrated that combining classifiers can offer significant classification

performance improvement for a number of non-trivial pattern recognition problems [8,16,23]. There are many tools with implementations of many classification algorithms from statistics, machine learning, neural networks, fuzzy systems, and many other fields [6] which can be conveniently used and evaluated. As part of the experimentations using such tools offering many alternative classifiers one of the early approaches to building MCS was to “gather them all and combine” [17,24]. Various studies [23,17,11] have shown that just gathering as many classifiers as

\* Corresponding author. Fax: +44 1202 595314.

*E-mail addresses:* [bgabrys@bournemouth.ac.uk](mailto:bgabrys@bournemouth.ac.uk) (B. Gabrys),  
[dymitr.ruta@bt.com](mailto:dymitr.ruta@bt.com) (D. Ruta).

possible is expensive and very rarely optimal. These findings have prompted investigations into what makes some combination of classifiers work better than others [22]. One of the most vigorously pursued research areas attempting to explain the property which decides about the team strength is known in the literature under the name of classifier diversity [20–22]. Though it has been observed that a well performing MCS has to consist of diverse and well performing individual classifiers various measures of diversity on their own have been of limited use in practical applications due to their limited correlation with MCS performance. Like in evaluation of individual classifiers the MCS performance (i.e. misclassification rate) estimated within one of the statistical method independent learning schemes (i.e. various cross-validation approaches) has been used much more effectively when constructing MCS.

In any case, it is now clear that MCS performance strongly depends on careful selection of classifiers to be combined. The performance of the individual classifiers depends in turn on careful selection of features, the use of training data and subspaces of data domains. The effectiveness of various classifier fusion methods depends again on the selections made within classifiers, features and data spaces. Commonly the active selection/optimisation is carried out along one dimension (i.e. either classifiers, or features, or data, etc.) while other selections are made arbitrarily. While clearly a multidimensional selection method in which data, classifiers and fusion methods are selected simultaneously and cooperatively in order to maximise the MCS performance would be of large potential benefit it is quite a daunting task.

In this work we attempt to tackle this problem using genetic algorithms (GA) [10], which appear to be one of the most suitable techniques for such multidimensional selection task, providing reasonable compromise between the search complexity and the quality of the found solutions. The remaining of the paper starts with further discussion of different potential selection dimensions in the classification process. The basic description of GA is then given in Section 3. The use of GA for finding weights for combining soft classifier outputs is presented in Section 4. The details of a novel multidimensional selection model (MSM) are given in Section 5. Experimental results and analysis of all the methods

are presented in Section 6 which is followed by conclusions and indications of future work inspired by this work and MSM model in particular.

## 2. Selection dimensionality in classification

The selection process is quite simply a process of validation of some elements while rejecting the other elements from a given set of objects. In relation to classification models, selection could take as many different forms as many selection decisions one would have to make in order to reach the final classification output. Starting from the main selection areas of the training data, features, classifiers or fusion techniques, one could also look into more detailed selection processes that could be applied to data domains, classes of data, classifier outputs or even model parameters. All of the aforementioned objects constitute the multiple degrees of freedom along which the classifier fusion design process could be carried out and contribute to the very high complexity of this process.

### 2.1. Data

Until recently there was a common belief that all the available data should be used to build a classification model. This belief, although theoretically true when the training data is accurate, has been gradually relaxed by extensive experimental findings related to the feature selection and real datasets [13,2,15]. It was found that many realistic learning systems cannot fully distinguish between good or representative data and bad data due to their lack of robust mechanisms for dealing for instance with noisy, missing or mislabelled data. In practice, one of the most common approaches in order to avoid performance losses and obtain good generalisation performance, is to filter out bad data and use only good data to build a classification model. Finding the most suitable training data breaks down into a variety of ways these data can be selected. Direct selection of the optimal data points is usually referred to as data editing [15,7] where the aim is to attain the compact data sample that retains maximum representativeness of the original data structure and allows for building accurate classification models. The selection restric-

tions can be specified in many other ways beyond just direct selection of samples. As the data is mapped onto the input space, the selection rules can be attributed to the space rather than to the data forming it. The input space can be simply segmented into many differently shaped subspaces. The shapes of subspaces may be formed in various generic forms, or can be dictated by the classification methodology. For instance, in dynamic classifier selection methodology [11], the shape of the subspace is dictated by the  $k$ -nearest neighbour rule, while in error correcting codes (ECOC) method [5] the shape of the sub-space is fully determined by the structure of classes in the data. In the most common scenario, the input space is divided along parallel or perpendicular space boundaries, which means that selection applies to features and some particular ranges of their variability, respectively. Labelled character of the data for classification adds an additional dimension for potential selection.

#### 2.1.1. Domain

All the features have typically open domains allowing for unlimited variability in the range of  $(-\infty, +\infty)$ . However there could be many reasons for limiting these domains by selecting the narrow range of valid feature variability. One of such reasons could be the need for filtering out the outliers—samples laying far from the areas with high data concentrations. Frequently, to accommodate outliers, the classification model has to stretch model parameters such that a single distant data point has much greater influence on the model than many points within dense regions of the input space. The domain can be limited by a single or multiple ranges or valid variability for each feature. In the special case the domain range can be reduced to none which is equivalent to the exclusion of such feature.

#### 2.1.2. Features

As mentioned above feature selection is a special case of domain selection but due to its simplicity deserves separate treatment. Feature selection has two attractive aspects to consider. First of all selecting some instead of all features significantly reduces computational costs of classification algorithms which are typically at least quadratically complex with respect to the number of features. Secondly, in practice

many features are non-contributory to the classifier performance and sometimes due to imperfect learning algorithms can even cause the performance deterioration. Features can be selected along with their variability range limits. Such scenario is equivalent to selection of particular clusters or subspaces in the input data, such as selection of classes of data.

#### 2.1.3. Classes

The emergence of classes of data adds another degree of freedom in selection process related to data. However rather than another dimension of selection it appears to be a form of restriction on how the domains in each features should be restricted. Selection of classes of data is used in error correcting output coding (ECOC) [5] where the  $N$ -class problem is converted into a large number of 2-class problems. Selection with respect to classes is particularly attractive if there are expert classifiers which specialise in recognising particular class or classes but are very weak in recognising other classes, in which case it makes sense to decompose the problem rather than aggregate performance over all classes.

### 2.2. Classifiers

Classifier selection is probably the most intuitive form of selection with respect to classifier fusion. There are generally two approaches to classifier selection leading to further combining by a fusion method. According to one approach, combiner is first picked arbitrarily and then classifiers are selected in such a way that the combiner results in the maximum performance. Alternatively, it is the combiner that is adjusted so as to fuse given classifiers in a best possible way. A simple example of this approach is given in Section 4 where a set of optimal weights is found by GA for a weighted soft classifier outputs combiner.

#### 2.2.1. Models

Due to the widespread and pervasive nature of classification problems there is a plethora of classification models in use [6]. They have often been researched and developed independently and modern data analysis toolboxes commonly contain classification methods from statistics, machine learning, artificial neural networks, fuzzy and expert systems,

evolutionary computing and hybrid systems domains. With regard to the combination systems all of these models represent an additional selection dimension. And questions like: Which models should be used? or Is combining models from different domains of benefit for the resulting combination system? have been frequently asked. The combination could also be applied at the model level itself as illustrated in [8,9] rather than combining outputs from different models.

### 2.2.2. Outputs

There are essentially three types of outputs [18] that can be generated as a result of the classification process:

- (1) crisp labels of the winning class where a single class is selected as a winner and no additional information is provided by the classifier;
- (2) ranking of classes where the classifier returns a list of alternatives in an order from the most likely to the least likely class;
- (3) soft outputs where a degree of belonging is calculated for all the classes like in the case of class conditional probabilities or fuzzy membership values.

Each of these types of outputs also provide another selection dimension in our optimisation process. There is also a potential level of difficulty associated with combination of different types of outputs. We can notice that both rankings and soft outputs can be easily converted into crisp labels and soft outputs into ranking of classes but the conversion in the other direction is much more difficult. Intuitively it is also clear that the soft outputs carry potentially more useful information which may be exploited by the combination system than the crisp labels and rankings of classes. Having said that the most commonly used combination technique is simple majority voting working with crisp labels. The use of genetic algorithms and comparison of potential benefits of combining linearly weighted soft outputs versus simple majority voting will be discussed in later sections of this paper.

### 2.3. Combiners

The selection process does not have to end on classifiers. Given a set of optimised classifiers it is

reasonable to test a number of available fusion methods and then to select the best performing one. On the other hand if classifier selection is applied for each combiner separately, the best system may turn out to be different. Such top-down decomposition could be further exploited and it may show that normally inferior pair of classifier–combiner suddenly shows best results if built on slightly different subset of features. Such doubts are imminent in any multiple classifier system unless the selection is carried out simultaneously on multiple dimensions and becomes a part of the system design. Fig. 1 depicts different potential dimensions of selection throughout the classification cycle.

### 3. Genetic algorithm

The genetic algorithm was developed in 1970s by Holland [12] as an effective evolutionary optimisation method. Since that time, intensive research has been dedicated to GAs [10], bringing lots of applications in machine learning domain [4,3,19,14] including

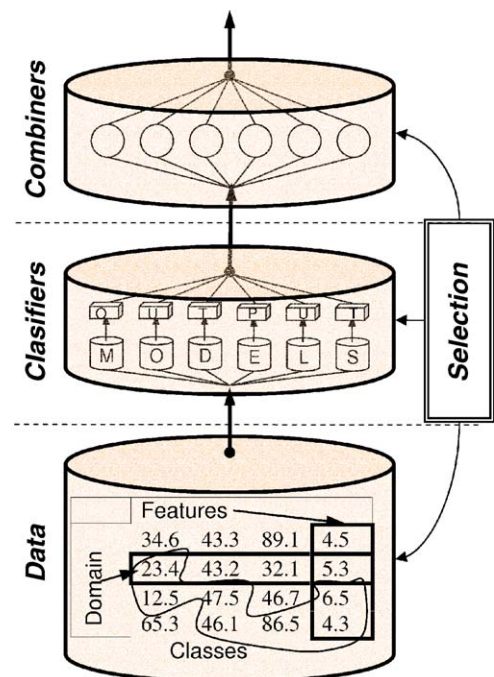


Fig. 1. Selection degrees of freedom within the pattern classification methodology.

classifier selection. Despite many varieties of GAs, its underlying principles remain unchanged. Chromosomes represent binary encoded solutions to the optimisation problem. A randomly initialised population of chromosomes is then evaluated according to the required fitness function and assigned a probability of survival proportional to their fitness. The best chromosomes are most likely to survive and are allowed to reproduce themselves by recombining their genotype and passing it on to the next generation. This is followed by a random mutation of some bits, which was designed to avoid premature convergence and enables the search to access different regions of a search space. The whole process is repeated until the population converges to a satisfactory solution or after a fixed number of generations. The GA is inspired by and takes strength from an explicit imitation of biological life, in which the strongest (fittest) units survive and reproduce further constantly adjusting to the variable conditions of living.

#### 4. GA in weighted combination of soft classifier outputs

In our recent work the GAs [19] and other population-based search techniques have been used for classifier selection for majority voting combiner. However, as many classifiers that are combined produce soft outputs in a form of probabilities or fuzzy membership values it would be interesting to see whether the combination of such soft outputs has any benefits in comparison to the simple majority voting scheme. As part of a natural extension to our previous research utilising evolutionary search techniques and leading to the development of the multidimensional selection model described in the next section the GA has been applied to finding the optimal weights for linear weighted combination of the soft outputs of the classifiers.

In the first of the examined models, referred to as soft linear combiner (SLC), a single weight  $\{w_1, \dots, w_c\}$  is assigned to each of the  $c$  classifiers with the optimisation problem defined as finding the  $c$  optimal weights.

In the second of the examined models, referred to as class independent soft linear combiner (CISLC), each of the  $c$  classifiers to be combined are assigned

weights for each of the  $m$  classes  $W_i = \{w_{i1}, \dots, w_{im}\}$  for  $i = 1, \dots, c$  resulting in the optimisation problem defined as finding  $m \times c$  optimal weights.

We can now write the soft output for the  $j$ th class of the combined linearly weighted classifier based on the soft support values  $y_{ij}$  given by each of the  $c$  individual classifiers as

$$y_j^{\text{comb}} = \sum_{i=1}^c w_{ij} y_{ij} \quad \text{for } j = 1, \dots, m \quad (1)$$

The winning class is selected on the basis of finding the maximum value of  $y_j^{\text{comb}}$ :

$$d = \text{maxind}(y_j^{\text{comb}}) \quad (2)$$

The optimisation criterion that has also been used as a fitness function driving the genetic algorithm's search for optimal combination weights is given as

$$J = \sum_{i=1}^n (d_i - t_i) \quad (3)$$

where  $d_i$  are the labels of the winning class as given by Eq. (2) and  $t_i$  are the known target class labels from the validation set of  $n$  samples used for the estimation of the weights.

In our previous examination of using GAs for the classifier selection [19] the weights in Eq. (1) could only take values of 0: classifier excluded or 1: classifier included in the final pool of classifiers (this is also illustrated as the classifier selection dimension in Fig. 2 and the actual implemented model is referred to as MCSS-1D in Section 6). The values that were combined were also simplified as they only included the binary (correct/incorrect) classifier output required for our study of majority vote (MV) combiner.

By introducing the additional levels of flexibility (i.e. weights that can take values between 0 and 1) and using potentially more useful information given by individual classifiers (i.e. support values for each of the classes rather than just the binary outputs) we would hope that the overall performance of the combined system would be improved.

The weights in our models have been represented as binary strings and standard GA has been applied.

While the optimisation itself (i.e. finding the weights) was smooth and GA has worked very well as in our previous studies, we have found that though

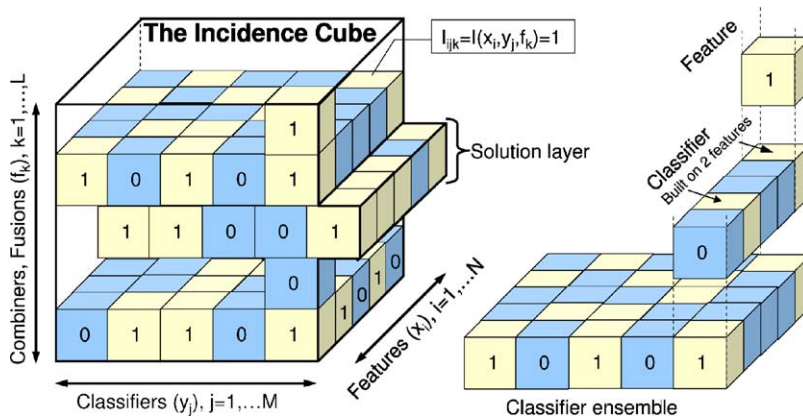


Fig. 2. Incidence cube representation of the three-dimensional fusion-classifier-feature selection model. Small cubes correspond to the triplet combiner-classifier-fusion and take values of 1 (light colour) if the corresponding feature is included in a classifier, which is then included in the corresponding combiner or 0 (dark colour) if the particular triplet is not selected.

there were some improvements in comparison to simple MV scheme the use of linearly weighted soft classifier outputs has not been as successful as we would have hoped. This prompted our investigations towards the multidimensional selection model described in the next section.

## 5. Multidimensional selection model (MSM)

The weaknesses of traditional selection models related to classification stem from the fact that selection is typically carried out along only single dimension. The choices related to other dimensions are made arbitrarily, based on some heuristic optimality measures. A challenging alternative is a multidimensional selection method in which data, classifiers and fusion methods are selected simultaneously and cooperatively to maximise classification performance of the system. The common doubt of processes operating along multiple degrees of freedom is the exploding computational complexity. To realise the significance of this problem let us consider a system with  $f$  features,  $c$  classifiers and  $b$  combiners. Let further assumption be that the combiners are selected as singletons only, as we cannot combine them at this stage. In such case the number of different systems to examine is

$$N = (2^f - 1)(2^c - 1)b \sim b2^{f+c} \quad (4)$$

Such high complexity means that for a simple system with 10 features, 10 classifiers and 10 combiners one needs to make more than  $10^7$  evaluations to exhaustively search for the best design.

### 5.1. Representation

In response to such huge computational demands the presented system employs efficient adjustment of a genetic algorithm [12]. To handle this algorithm along many dimensions the chromosomes are designed as incidence cubes dimensions of which correspond to the selection dimensions of features, classifiers and combiners as shown in Fig. 2. The meaning of “1” (“0”) in each small cube is that the corresponding feature is (not) included in the corresponding classifier, and combiner of the system. The cube matches the hierarchical structure of the dimensions in which combiners are built using many classifiers which in turn are built using many features. Note that such hierarchical structure means that the classifier can only be dropped if it does not have any features selected. Likewise combiner can be excluded only if it corresponds to a whole layer of zeros, corresponding to the lack of any classifiers selected. Another important aspect of the incidence cube is that it is not fully operational along the combiners dimension. The reason for that is the inability to further combine classifier fusion methods at this stage and hence the chromosome is a

collection of selection solutions associated with layers of the incidence cube.

### 5.2. Multidimensional GA (MGA)

The cube representation of the proposed multidimensional selection model for classification is particularly suitable for selection by means of GA, yet requires few adjustments to its standard form. Due to the three-dimensional cube corresponding to a chromosome and the fact that each chromosome contains many solution layers, there are few changes required in mutation and particularly in cross-over operation of the standard GA. Mutation is quite straightforward as it only involves sampling from a mutation probability applied to all genes (small cubes). The cross-over operation is more complicated as there are many degrees of freedom by which the chromosomes can be recombined. The presented model uses a two-stage cross-over operation. First the chromosomes recombine internally by exchanging subsets of classifiers and features among randomly selected pairs of combinars as shown in Fig. 3. Then the whole chromosomes recombine among each other by swapping parts split by the randomly oriented plane

cutting through the incidence cube. Due to multiplicity of solutions within single chromosome the evaluation process is carried out on the basis of an average from the classification performances from each solution layer of the incidence cube. The algorithm uses also the elitism operation but realised through the natural selection process in which the best from both parents and offsprings are passed on to the next generation. The algorithm can be summarised in the following form:

- (1) Collect and fix the selection space with  $f$  features  $c$  classifiers and  $b$  combinars.
- (2) Initialise a random population of  $n$  chromosomes ( $f \times c \times b$  binary incidence cubes).
- (3) Perform mutation and two-stage cross-over.
- (4) Pool offspring and parents together and calculate the fitness for all potential solutions.
- (5) Select  $n$  best chromosomes for the next generation.
- (6) If convergence then finish, else go to step 3.

Note that this particular implementation of GA represents a hill-climbing algorithm, as it guarantees that the average performance will not decrease in the

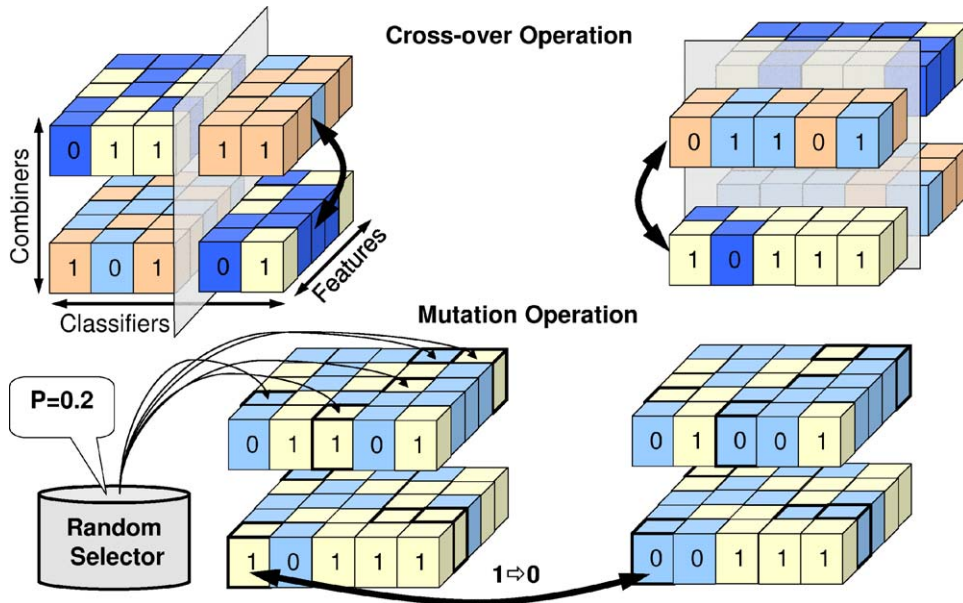


Fig. 3. Visualisation of genetic algorithms operations of mutation and cross-over carried out on the incidence cube representation of three-dimensional selection for classification.

subsequent generations. Mutation along with two-stage cross-over ensure sufficient exploration ability of the algorithm. The convergence condition can be associated with the case when no change in the average fitness is observed for an arbitrarily large number of generations. Previous comparative experiments with real classification datasets confirmed the superiority of the presented version of the GA to its standard definition [1,19].

## 6. Experiments

A number of experiments have been carried out to test the performance of the three-dimensional classifier selection model and the weighted linear combiners for soft classifier outputs presented in previous sections. Due to its novel character and interesting characteristics our main analysis in this section will concentrate on the three-dimensional classifier selection model. Short comparative results of all investigated models will be given towards the end of this section. Throughout the experiments a fixed sets of 10 different classifiers and 5 combiners were being applied to 2 known datasets from UCI repository.<sup>1</sup> Details of datasets, classifiers and combiners are shown in Table 1.

To limit the computational complexity for each dataset the selection algorithm used a population of only 10 incidence cubes. The mutation rate was set to  $p = 0.1$  while specific selection technique described in previous section ensured non-decreasing convergence of the GA in the average classification performance. The chromosomes were built along three dimensions capturing features, classifiers and combiners incidence. They have been evaluated by the average misclassification rate obtained for all layers (combiners) separately. To preserve generalisation abilities of the system, the classifiers and hence the combiners were built on the separate training sets and tested on parts of the dataset which have not been used during training. Then the training and testing sets were swapped such that an equivalent of two-fold cross-validation rule has been used for chromosome evaluation. For simplicity the GA

Table 1  
Datasets, classifiers and combiners used in experiments

No.	Classifier	Description
1	klclc	Linear with KL expansion
2	Loglc	Logistic linear classifier
3	ldc	Linear discriminant classifier
4	qdc	Quadratic with normal density
5	pfsvc	Pseudo-Fisher SVM classifier
6	lmnc	Levenberg–Marquardt neural net

No.	Combiner	Description
1	meanc	Mean combiner
2	minc	Minimum rule combiner
3	maxc	Maximum rule combiner
4	prodc	Product rule combiner
5	majorc	Majority voting combiner

Name	Classes	Samples × features
Iris	3	150 × 4
Liver	2	345 × 6

was stopped after 100 generations for all datasets, despite the fact that for some cases convergence was achieved earlier. Fig. 4 illustrates the dynamics of the testing performance characteristics during selection process carried out by the GA algorithm. The typical observation is that the algorithm relatively quickly finds the best performing system and then in subsequent generations it keeps improving other solutions in the population. The algorithm showed the capacity to get out of local minima which effectively means discovery of significantly better solution spreading swiftly in many variations during subsequent generations. Fig. 5 depicts the evaluation of a final population of chromosomes for both datasets. For *Iris* dataset the Min combiner showed the best average performance including the absolute best performing system with only 1.33% misclassification rate. Majority voting showed on average best performance for *Liver* dataset, including the absolute best performing system with 27.8% error rate. The best systems for both datasets were then further uncovered by illustrating the structure of the classifiers and features selected as shown in Fig. 6. Interestingly, for each selected classifier the algorithm selected at least two features. One classifier for both datasets was excluded. Other than that there is nothing significant about the selection structures shown in Fig. 6. This could only prove that it is very difficult to find the best performing systems as they do not exhibit any visible

<sup>1</sup> University of California Repository of Machine Learning Databases and Domain Theories, available free at: <http://ftp.ics.uci.edu/pub/machine-learning-databases>.

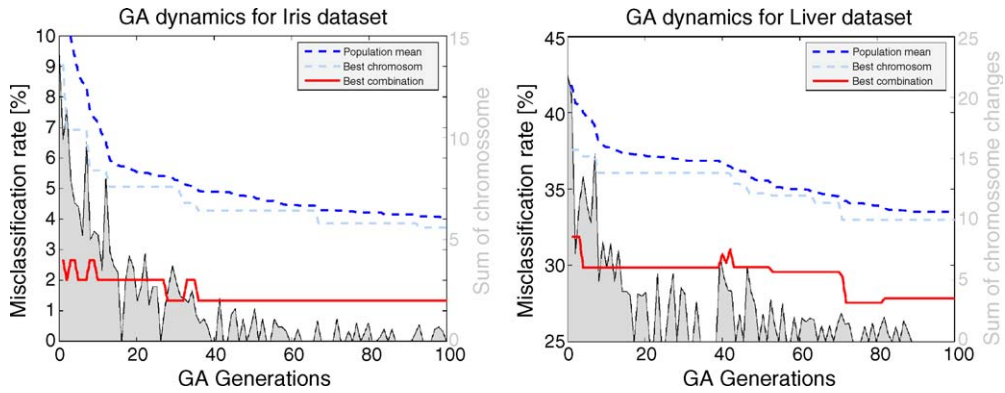


Fig. 4. Performance characteristics evolution during GA selection.

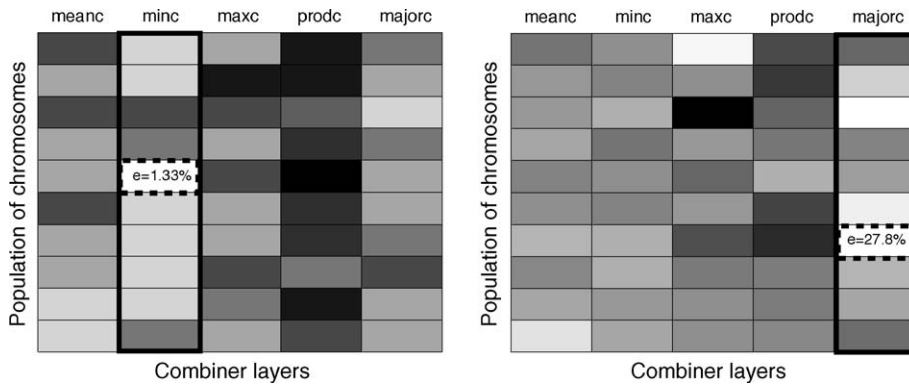


Fig. 5. Diagrams showing the misclassification rates of the final population of chromosomes returned by GA. The lighter the field the lower error rate of the corresponding classifier fusion system. Thick frame indicates the best combiner for particular dataset while dashed line shows the performance of the overall best system found.

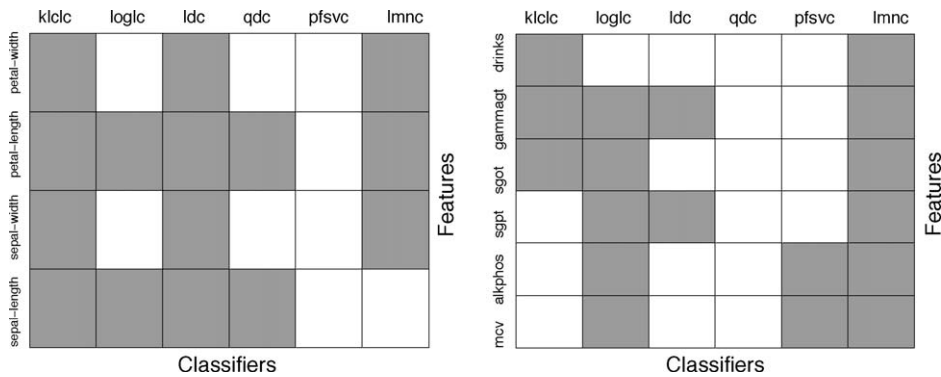


Fig. 6. Diagrams showing the subsets of features and classifiers selected by the GA for the best performing combinations fused by *Min* combiner for Iris dataset and *Majority Voting* for the Liver dataset. Dark field indicate inclusion of the corresponding pairs feature–classifier in the final system.

Table 2

Comparison of the error rates obtained for the best systems using: SB, SLC, CISLC, MCS-1D, MCS-3D

Dataset	SB (classifier)	SLC	CISLC	MCSS-1D	MCSS-3D
Iris	2.47 (klclc)	2.06	2.06	2.13	1.33
Liver	32.35 (loglc)	28.75	28.86	29.06	27.78

distinctiveness but are simply lost among large number of system designs embodying huge selection complexity as shown in Eq. (4).

Finally the last experiment intends to compare the performances of systems designed by means of three-dimensional selection process (MCSS-3D) with other more traditional systems of single best classifiers (SB), multiple classifier systems with: GA-based classifier selection only (MCSS-1D), GA-optimised weighted soft linear combiner (SLC) and GA-optimised class independent soft linear combiner (CISLC). Table 2 shows the error rates of the best system found in the aforementioned design groups. The presented MCSS-3D clearly outperformed all the other systems. The performances of the weighted soft output combiners (SLC and CISLC) while being marginally better than that of the simpler version investigated in MCSS-1D have been disappointing and did not justify the additional computational effort required for finding the weights and processing of the soft outputs. However, the benefits of the selection carried out along many dimensions of the classification process at the same time have been confirmed.

## 7. Conclusions and future work

In this paper we have focused on the challenge of many selection dimensions and decisions that need to be made when constructing well performing multiple classifier systems. Genetic algorithms have been identified as a very good technique for carrying out various types of parameter optimisation and multi-dimensional selection offering a good compromise between searching complexity and the quality of the found solutions. Among the discussed techniques, a new three-dimensional fusion–classifier–feature selection model and a corresponding multidimensional GA based on the incidence cube representation have been proposed and discussed in detail. The comparison with other GA-based multiple classifier

systems including two versions of weighted soft linear combination systems have illustrated the importance of such multidimensional optimisation as the proposed three-dimensional selection model resulted in a better performing classification systems automatically optimised for a given data set. The new multidimensional representation of the solutions opens an opportunity for future development of a general framework with evolutionary mechanisms continuously applied to the working systems along many dimensions in order to accommodate the dynamically changing data sets and environments. The inclusion and exclusion of new inputs as well as classifiers while combining them in a flexible manner as required will be one of the foreseen investigated topics.

## References

- [1] S. Baluja, Population-based Incremental Learning: A Method for Integrating Genetic Search based Function Optimization and Competitive Learning, vol. 163, Carnegie Mellon University, Pittsburgh, PA, 1994.
- [2] K. Chen, L. Wang, H. Chi, Methods of combining multiple classifiers with different features and their applications to text independent speaker identification, *Int. J. Pattern Recogn. Artif. Intell.* 11 (3) (1997) 417–445.
- [3] S.-B. Cho, Pattern recognition with neural networks combined by genetic algorithms, *Fuzzy Sets Syst.* 103 (1999) 339–347.
- [4] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.
- [5] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, *J. Artif. Intell. Res.* 2 (1995) 263–286.
- [6] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification*, John Wiley and Sons Inc., 2000.
- [7] B. Gabrys, Data Editing for Neuro-Fuzzy Classifiers, Proceedings of the SOCO/ISFI'2001 Conference, Abstract page 77, Paper no. 1824-036, Paisley, UK, 2001, ISBN 3-906454-27-4.
- [8] B. Gabrys, Combining neuro-fuzzy classifiers for improved generalisation and reliability, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN2002), A Part of the WCCI2002 Congress, Honolulu, USA, (2002), pp. 2410–2415, ISBN 0-7803-7278-6.

- [9] B. Gabrys, Learning hybrid neuro-fuzzy classifier models from data: to combine or not to combine? *Fuzzy Sets Syst.* 147 (2004) 39–56.
- [10] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
- [11] G. Giacinto, F. Roli, Methods for dynamic classifier selection, in: *Proceedings of the 10th International Conference on Image Analysis and Processing*, Venice, Italy, (1999), pp. 659–664.
- [12] J.H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Michigan, 1975.
- [13] H. Ishibuchi, T. Nakashima, M. Nii, Genetic-algorithm-based instance and feature selection, in: H. Liu, H. Motoda (Eds.), *Instance Selection and Construction for Data Mining*, Kluwer Academic Publishers, 2001, pp. 95–112.
- [14] L.I. Kuncheva, L.C. Jain, Designing classifier fusion systems by genetic algorithms, *IEEE Trans. Evol. Comput.* 4 (4) (2000) 327–336.
- [15] L.I. Kuncheva, L.C. Jain, Nearest neighbor classifier: simultaneous editing and feature selection, *Pattern Recogn. Lett.* 20 (4) (1999) 1149–1156.
- [16] L.I. Kuncheva, *Fuzzy Classifier Design*, Physica-Verlag, Heidelberg, 2000.
- [17] L.I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, John Wiley and Sons Inc., 2004.
- [18] D. Ruta, B. Gabrys, An overview of classifier fusion methods, in: M. Crowe (Ed.), *Computing and Information Systems*, vol. 7, no. 1, University of Paisley, 2000, pp. 1–10.
- [19] D. Ruta, B. Gabrys, Application of the evolutionary algorithms for classifier selection in multiple classifier systems with majority voting, in: *Proceedings of the Second International Workshop on Multiple Classifier Systems*, Springer-Verlag, Cambridge, UK, 2001, pp. 399–408.
- [20] D. Ruta, B. Gabrys, Analysis of the correlation between majority voting error and the diversity measures in multiple classifier systems, in: *Proceedings of the SOCO/ISFI2001 Conference*, Abstract page 50, Paper no. 1824-025, Paisley, UK, 2001, ISBN 3-906454-27-4.
- [21] D. Ruta, B. Gabrys, New measure of classifier dependency in multiple classifier systems, in: *Proceedings of the MCS2002 Conference*, Springer-Verlag, Italy, 2002.
- [22] D. Ruta, B. Gabrys, Set analysis of coincident errors and its applications for combining classifiers, in: D. Chen, X. Cheng (Eds.), *Pattern Recognition and String Matching*, Kluwer Academic Publishers, 2002, ISBN 1-4020-0953-4.
- [23] D. Ruta, B. Gabrys, Classifier selection for majority voting, *J. Inf. Fusion*, 6 (1) (2005) 63–81 (special issue on Diversity in Multiple Classifier Systems).
- [24] A.J.C. Sharkey, *Combining Artificial Neural Nets: Ensemble and Modular Multi-net Systems*, Springer-Verlag, Berlin, 1999.