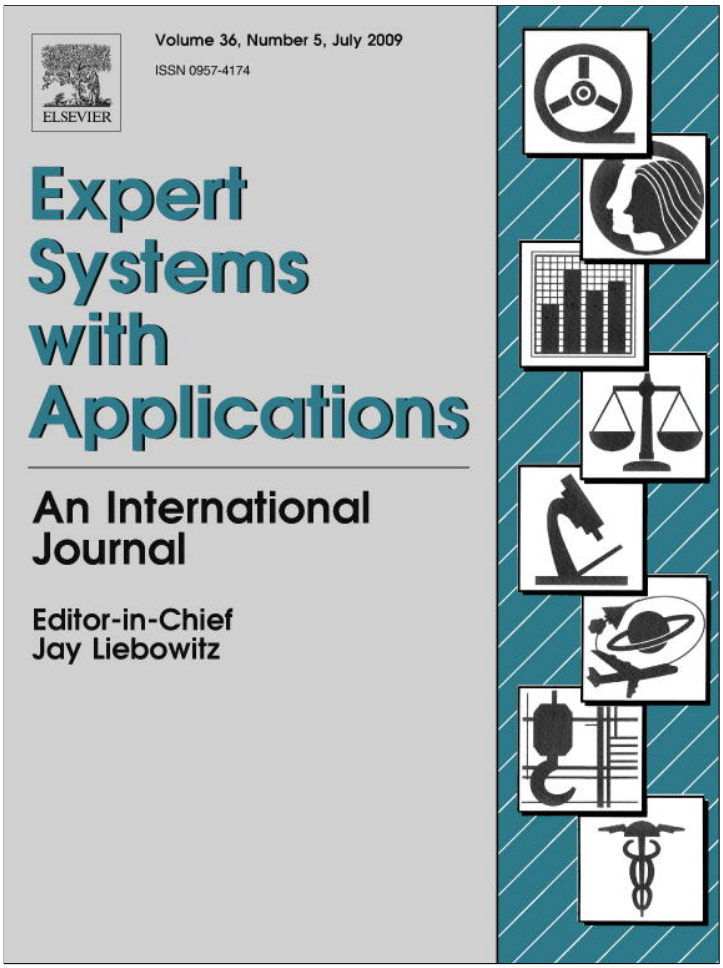

Expert Systems with Applications 36 (2009) 9240–9249

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An application of a new meta-heuristic for optimizing the classification accuracy when analyzing some medical datasets

Huy Nguyen Anh Pham, Evangelos Triantaphyllou *

Department of Computer Science, Louisiana State University, 298 Coates Hall, Baton Rouge, LA 70803, United States

ARTICLE INFO

Keywords:
Optimization
Medical data mining
HBA
Genetic algorithms
Classification errors

ABSTRACT

Medical data mining has recently become one of the most popular topics in the data mining community. This is due to the societal importance of the field and also the particular computational challenges posed in this domain of data mining. However, current medical data mining approaches oftentimes use identical costs or just ignore them for the different cases of classification errors. Thus, their outcome may be unexpected. This paper applies a new meta-heuristic approach, called the Homogeneity-Based Algorithm (or HBA), for optimizing the classification accuracy when analyzing some medical datasets. The HBA first expresses the objective as an optimization problem in terms of the error rates and the associated penalty costs. These costs may be dramatically different in medical applications as the implications of having a false-positive and a false-negative case may be tremendously different. When the HBA is combined with traditional classification algorithms, it enhances their prediction accuracy. It does so by using the concept of homogenous sets. Five medical datasets, obtained from the machine learning data repository at the University of California, Irvine (UCI), USA, were tested. Some computational results indicate that the HBA, when it is combined with traditional methods, can significantly outperform current stand-alone data mining approaches.

1. Introduction

Increasing powerful mechanisms for storing data has made available lots of datasets related to medicine in recent decades. A motivation for extracting useful knowledge from such datasets and thus discovering decision-making insights for the diagnosis and treatment of diseases, is also increasingly recognized. In the typical setting a dataset of historic data, which describe some type of disease or a medical disorder, is assumed to be available. Such datasets consist of records of patients describing physical and laboratory examinations related to that type of disease or medical disorder. Then, the computational challenge is how to develop a diagnostic system, which could assist in diagnosing this type of ailment based on the knowledge extracted from the historic dataset. At this point, human analysts need special computational tools to process and comprehend such large and complex datasets.

Medical data mining can assist in addressing such challenges. Data mining analysts can extract decision regions from a given historic dataset related to a medical condition or disease. Usually, such decision regions consist of medical indicators, which could be used to diagnose the condition or disease. In medical diagnosis (as in most other domains), usually there are three different cases of possible errors:

- The false-negative case in which a patient, who in reality has the disease, is diagnosed as disease free.
- The false-positive case in which a patient, who in reality does not have the disease, is diagnosed as having the disease.
- The unclassifiable case in which the prediction system cannot diagnose a given case. This happens due to insufficient knowledge extracted from the historic data.

Under the above considerations, current medical data mining approaches oftentimes assign identical penalty costs for the false-positive and the false-negative cases or just ignore the penalty cost for the unclassifiable cases. Such approaches will be discussed in Section 2. Thus, their outcome may be unexpected or even unacceptable.

The two penalty costs for the false-positive and the false-negative cases could be dramatically different in a medical application. For instance, in the case of a life threatening condition where time is of essence, if one diagnoses a given case as false-negative, then his/her medical condition goes untreated or is treated inadequately. Thus precious time may be wasted and the situation may turn out to be eventually fatal to the patient. On the other hand, for the same situation, a false-positive diagnosis may just

* Corresponding author.
E-mail addresses: hpham15@lsu.edu (H.N.A. Pham), trianta@lsu.edu (E. Triantaphyllou).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.12.007

add some financial costs and anxiety to the patient but not result in a life threatening condition.

A penalty cost for unclassifiable cases in medical data mining is needed as well. A diagnosis of a patient as an unclassifiable case may require additional medical examinations and involve some costs. However, that particular case may not necessarily result in a wrong diagnosis.

For the above reasons, this paper applies a new meta-heuristic approach, called the Homogeneity-Based Algorithm (or HBA) as developed by Pham and Triantaphyllou (2007, part 4, chap. 5) and Pham and Triantaphyllou (2008, chap. 2), on some well-known medical datasets. The HBA first defines the total misclassification cost of models extracted from classification algorithms as an optimization problem in terms of the false-positive, the false-negative, and the unclassifiable rates along with their penalty costs. The HBA then organizes the extracted models as mutually exclusive decision regions represented by homogeneous sets. These decision regions are refined based on their density by employing a genetic algorithm (GA) approach. This is done in order to minimize the total misclassification cost. The HBA is motivated by the large discrepancy in the previous three penalty costs.

The next section provides a literature review of some related developments. The third section has a brief description of the HBA as adopted from Pham and Triantaphyllou (2007) and Pham and Triantaphyllou (2008). That section shows how the HBA can yield an optimal or near optimal misclassification total cost. The fourth section discusses some computational results from the medical domain. These results give an indication of how this methodology may improve the prediction accuracy in computerized medical diagnosis. The paper ends with some conclusions and an appendix, which describes the key algorithmic aspects of the HBA.

2. Previous work

This paper studies five medical datasets, which current data mining approaches have often used for their analyses. The main characteristics of these datasets are depicted in Table 1. These datasets were selected because the number of attributes were in the range of values that the HBA can handle easily (i.e. approximately less than 9 or 10). Other reasons for selecting these datasets were that traditional approaches have analyzed them with variable success and these datasets represent a variety of important medical diseases and disorders.

The first dataset is the Pima Indian diabetes (PID) as described in Asuncion and Newman (2007). Attributes of 768 female patients of Pima Indian heritage were recorded in this dataset. The class variable denotes whether a person has diabetes or not. Smith, Everhart, Dickson, Knowler, and Johannes (1998) achieved 76% accuracy by using an Early Neural Network (ENN). Jankowski and Kadirkamanathan (1997) obtained 77.6% accuracy by using a radial basis function network suite, called IncNet. Au and Chan (2001) improved the correct classification percentage by using a fuzzy approach. Their approach achieved 77.6% accuracy. Rutkowski and Cpalka (2003) obtained 78.6% accuracy by introducing a new neural-fuzzy structure, called a flexible neural-fuzzy inference system (FLEXNFIS). Leon (2006) obtained 81.8% accuracy by using a Fuzzy Neural Network (FNN) associated with the BK-Square products. Different classification algorithms in the StatLog project in Michie, Spiegelhalter, and Taylor (1994, chap. 9) obtained less than 78% accuracy. Pham and Triantaphyllou (2008) applied the HBA in conjunction with some Support Vector Machine (SVM), Artificial Neural Network (ANN), and Decision Tree (DT) algorithms. Their accuracy reached about 93.8%.

The second medical dataset is the Haberman Surgery Survival (HSS) as described in Asuncion and Newman (2007). This is one of the most difficult datasets for classification algorithms. The dataset contains records which describe 306 patients who have undergone surgery for breast cancer. Kecman and Arthanari (2001) proposed an SVM approach using linear terms in the objective function for analyzing the HSS dataset. Their approach yielded 71.2% accuracy. Fung and Mangasarian (2001) reformulated Kecman's approach to decrease its complexity. Their approach is called the Proximal Support Vector Machine (PSVM) classifier and it uses a purely quadratic objective function with equality constraints. Their approach yielded 72.5% accuracy. Domm, Engel, Louis, and Goldberg (2005) proposed the Integer Support Vector Machine (ISVM) classifier, which used binary indicator error variables in order to directly minimize the number of potential errors. Their accuracy was 62.7%. Shevked and Dakovski (2007) represented sets of positive and negative training points as logical functions. These logical functions were then minimized in order to find the target functions, which were prime implicants. Their approach yielded 66.2% accuracy.

Some classification approaches used the breast cancer (BC) dataset as described in Asuncion and Newman (2007). This dataset contains records which describe 286 patients who had either breast cancer or no cancer. One of the tested algorithms on this dataset was C4.5 as developed by Quinlan (1996). Quinlan's approach reached 94.7% accuracy by using 10-fold cross-validation. Hamilton, Shan, and Cercone (1996) used the Rule Induction (RI) approach based on approximation of classification to enhance the accuracy. Their approach obtained 96% accuracy. Similarly, Ster and Dobnikar (1996) achieved 96.8% accuracy with the Linear Discriminant Analysis (LDA) approach. Bennet and Blue (1997) used an SVM approach. Their accuracy was 97.2%. In the following two years, Nauck and Kruse (1999) achieved 95.1% accuracy by using a Neuro-Fuzzy approach. At the same time, Pena-Reyes and Sipper (1999) developed a Fuzzy-GA approach, which yielded 97.5% accuracy. Furthermore, Setiono's approach (2000) reached 98.1% accuracy by using a Neuro-Rule approach. Abonyi and Szeifert (2003) applied the Supervised Fuzzy Clustering (SFC) approach and achieved 95.6% accuracy. Polat, Sahan, Kodaz, and Gunes (2007) applied the Fuzzy Artificial Immune Recognition System (FAIRS) to form fuzzy-logic rules. Their approach reached 98.5% accuracy.

The fourth medical dataset is the Liver Disorders (LD) as described in Asuncion and Newman (2007) that many classification approaches have used for their analyses in recent years. This dataset contains records which describe 345 patients who had con-

Table 1
Characteristics of the five medical datasets.

Dataset	No. attributes	No. records	No. positive records	No. negative records	No. records in the training dataset T_1	No. records in the testing dataset
Pima Indian diabetes (PID)	8	768	268	500	576	192
Haberman Surgery Survival (HSS)	3	306	225	81	230	76
Breast cancer (BC)	9	286	85	201	214	72
Liver disorders (LD)	6	345	145	200	276	69
Appendicitis (AP)	7	106	85	21	85	21

firmed either liver disorders or no disorders. Pham, Dimov, and Salem (2000) used the RULES-4 algorithm on the LD dataset. Their approach yielded 55.9% accuracy. Cheung (2001) used different classification algorithms and his analysis showed that C4.5, Naive Bayes classifier, Bayesian Network with Naive Dependence (BNND) classifier, and a combination of a Bayesian Network with Naive Dependence and Feature Selection (BNNF) classifier obtained 65.5, 63.4, 61.8, and 61.4% classification accuracies, respectively. Lee and Mangasarian (2001a) and Lee and Mangasarian (2001b) used the following two SVM approaches: Smooth Support Vector Machine (SSVM) and Reduced Support Vector Machine (RSVM) in the same year. Their SVM approaches yielded 70.3 and 74.9% accuracies, respectively. Similarly, Van et al. (2002) with a Support Vector Machine approach reached 69.2% accuracy. Çomaka, Polatb, Güneşb, and Arslana (2007) combined a Least Squares Support Vector Machine (LSSVM) with Fuzzy Weighting Pre-processing for analyzing the LD dataset. Their approach yielded 94.3% accuracy. Also in the same year Polat et al. (2007) used the FAIRS to form fuzzy-logic rules. Their approach reached 83.7% accuracy.

The last dataset is the Appendicitis (AP) donated by Weiss and Kapouleas (1989). This dataset consists of seven examinations of 106 records which describe patients who had confirmed acute appendicitis. Weiss and Kapouleas then used the Predictive Value Maximization (PVM) approach and achieved 89.6% accuracy. Nakashima, Nakai, and Ishibuchi (2003) proposed the use of a fuzzy classification system for mapping the input space to a fuzzy rule-based classification system. Their approach yielded 84% accuracy. Blachnik and Duch (2006) used different classification algorithms. Their analysis showed that C4.5, Decision Table, Nefclass, Heterogeneous Decision Tree (HDT), and Prototype Threshold Decision List (PTDL) obtained 85.8, 82, 87.7, 85.8, and 83.8% accuracy, respectively. Next, the following section provides a brief description of the HBA, which was used in this paper.

3. The meta-heuristic approach – HBA

3.1. Some key issues for the HBA

The HBA defines the total misclassification cost, denoted as *TC*, in an optimization formulation as follows. Assume that C_{FP}, C_{FN}, and C_{UC} are the unit penalty costs for the false-positive, the false-negative, and the unclassifiable cases, respectively. The notations *RateFP*, *RateFN*, and *RateUC* denote the false-positive, the false-negative, and the unclassifiable rates, respectively. Then, the desired goal of the HBA is to minimize, or at least to significantly reduce, the *TC* defined as follows:

$$TC = \min\left(C_{FP} \times RateFP + C_{FN} \times RateFN + C_{UC} \times RateUC\right) \quad (1)$$

As adopted from Pham and Triantaphyllou (2007), Pham and Triantaphyllou (2008) there are two key assumptions in HBA's development. In order to explain these two assumptions, we will use a simple hypothetical example. For simplicity of the demonstration, consider a hypothetical medical training dataset regarding some type of cancer. The dataset is assumed to be defined on two attributes only, as depicted in Fig. 1a. The *X* and *Y* values indicate values for two laboratory examinations. These two values are assumed in this demonstration to be adequate to derive whether a patient has that type of cancer or not. Suppose that each circle depicted in Fig. 1a represents a patient who has cancer (also called a positive point), while a rectangle shows a patient who is disease free (also called a negative point).

We assume that a data mining approach has been applied on this dataset and derived the positive and negative decision regions (i.e. ovals) depicted in Fig. 1a. Decision regions A and B in Fig. 1a define the positive decision regions, while region C defines the negative decision region.

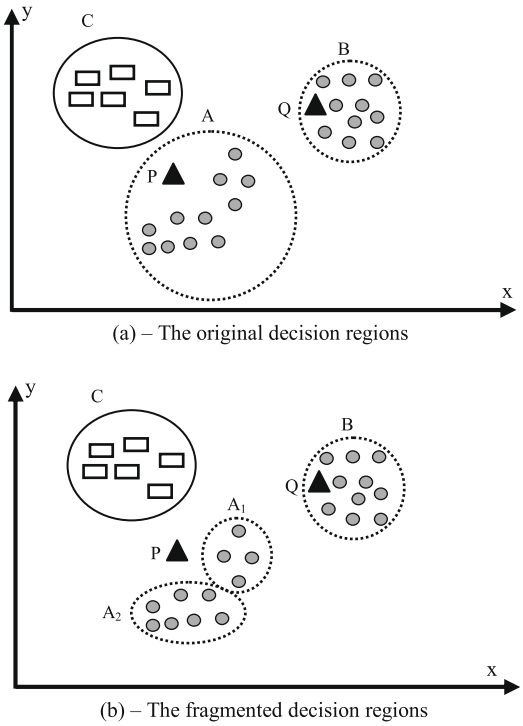

Fig. 1. Region B is a homogenous set while region A is a non-homogenous set. Region A can be fragmented into the two homogenous sets A_1 and A_2 as shown in part (b).

Furthermore, assume that in Fig. 1a there are two new patients P and Q, shown as small triangles. At this point, the system has not diagnosed these two new patients. We would like to use the inferred decision regions to diagnose these two patients. Because patients P and Q are covered by decision regions A and B, respectively, both of these patients may be assumed to have cancer.

A closer examination of Fig. 1 reveals that in decision region A, there are some sub-regions of the state space that are not sufficiently filled up by positive training points. We can see such sub-regions in Fig. 1a at the upper left corner and the lower part of region A. Thus, unclassified points covered by decision region A and also inside such sub-regions in region A may erroneously be assumed to be positive points. One may now observe that patient P is inside one of these sparely covered sub-regions in decision region A. Hence, the assumption that patient P has cancer may not be very accurate.

In contrast, sparely covered sub-regions do not exist in decision region B (see also Fig. 1a). Thus, it may be more likely that unclassified points covered by region B can more accurately be assumed to be positive points. Consequently, the assumption that patient Q has cancer may be more accurate than P's assumption. The above simple observations lead to surmise the following key assumption:

Assumption 1: The more compact and homogenous the decision regions are, the more accurate the inferred models are.

Next, let us consider a decision region D of size *N* (that is, it covers *N* training data points). Region D is first partitioned into smaller bins of the same size *h*. Then, region D is a homogenous set if the density of these bins is equal or almost equal to each other. How to choose an appropriate value for *h* is discussed in Heuristic Rule 1 (in the Appendix). As it was mentioned earlier, from Fig. 1a it looks like region A is a non-homogenous set, while region B is a homogenous set.

At this point, it is assumed that somehow decision region A is fragmented into two more homogenous sets denoted as A_1 and

A_2 as in Fig. 1b. These fragments are now more homogenous than the original region A. Under this consideration, patient Q is inferred to have cancer, while patient P is assumed to be of an unclassifiable case. Clearly, the homogenous property of decision regions may be used to affect the number of misclassification cases of the inferred models.

Furthermore, the number of training points covered by a homogenous set may be another factor that affects the accuracy of the overall inferred models. In fact, suppose that all decision regions A_1, A_2 and B in Fig. 2 correspond to homogenous sets and a new patient S (indicated as a triangle) is covered by region A_1. A closer examination of this figure shows that region B has many more training points than A_1. Although both patients Q and S are inside homogenous sets, the assumption that patient S has cancer may be less accurate than the assumption that patient Q has cancer. This is because there seems to be less support in sub-region A_1 when compared with that for region B (as B has more training points than A_1). This simple observation leads to surmise the second key assumption:

Assumption 2: The denser the decision regions are, the more accurate the inferred models are.

A density measure for a homogenous set is called the *homogeneity degree* and will be denoted as *HD*. This measure can be defined as the number of training points in a given homogenous set per unit of its area or volume. The Appendix shows an appropriate definition for the homogeneity degree of a homogeneous set. The next section shows how the HBA is implemented and it is adopted from Pham and Triantaphyllou (2007), Pham and Triantaphyllou (2008).

3.2. Details of the HBA

As mentioned in the introduction section, the HBA is used in combination with other classification approaches in order to enhance their classification accuracy. Fig. 3 shows how the HBA can be used in the above manner.

Assume that a medical training dataset is given. We first apply a traditional classification approach (such as a DT, ANN, and SVM) on the training dataset to infer the first pair of classification models. That is, the positive model, with decision areas defined on positive training data points, and the negative model with decision areas defined on negative training data points. Next, the HBA adjusts the two inferred models by using the two assumptions discussed in Section 3.1. In this way it is hoped that the accuracy of the traditional classification approach will be enhanced.

A detail description of the HBA algorithm is depicted in Fig. 4. There are five controlling parameters in the HBA:

- α^+ and α^- to be used for expanding the positive and the negative homogenous sets, respectively.

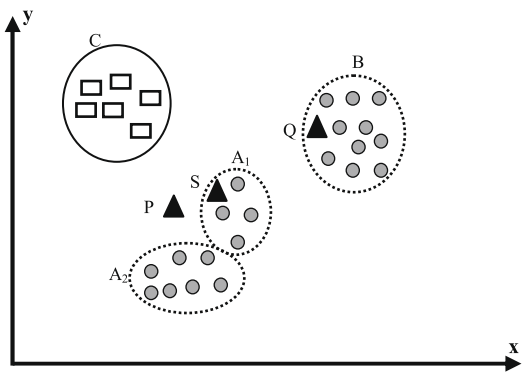

Fig. 2. An illustrative example of homogenous sets.

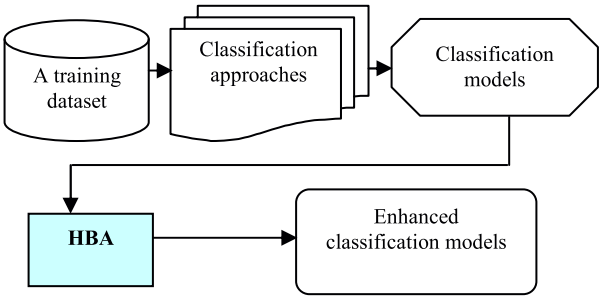

Fig. 3. The role of the HBA.

Input:
- The training dataset T with the positive and the negative data points.
- A given classification algorithm (such an SVM, ANN, or DT approach).
- The density threshold value γ.

1. Divide T into the two random sub-datasets: T_1 the actual training set whose size is equal to, say 90%, of T 's size and T_2 the calibration set whose size is equal to the rest, i.e., 10% of T 's size.
2. Randomly initialize the values of the control parameters α^+, α^-, β^+, and β^-.
3. Apply the input data mining approach (such as the SVM, ANN, or DT approach) on the training dataset T_1 to infer the two classification systems (i.e., the positive and the negative classification systems).
4. Fragment the inferred decision regions into hyperspheres.
5. For each hypersphere, denoted as C, do:
   Determine whether C is a homogenous set or not.
   If so, then its homogeneity degree is estimated with the help of γ.
   Else, fragment C into smaller hyperspheres and go to Step 5.
6. Sort the homogeneity degrees in decreasing order.
7. For each homogenous set C, do:
   If $HD(C) \geq \beta^+$ (for positive sets) or $HD(C) \geq \beta^-$ (for negative sets), then
   Expand C by using $HD(C)$ and the corresponding expansion threshold value α^+ or α^-, respectively.
   Else, fragment C into smaller homogenous sets.
8. Apply a GA approach on Steps 5 to 7 by using Equation (1) as the fitness function and T_2 as a calibration dataset to find the classification model S_1 and the optimal threshold values $(\alpha^+_*, \alpha^-_*, \beta^+_*, \beta^-_*)$.
9. For the unclassifiable points in T_2 under model S_1, we use Steps 3 to 7 with the optimal threshold values $(\alpha^+_*, \alpha^-_*, \beta^+_*, \beta^-_*)$ to infer the additional classification model S_2.
10. Let $S = S_1 \cup S_2$.

Output: The new classification system S.

Fig. 4. The HBA decision process.

- β^+ and β^- to be used for fragmenting the positive and the negative decision regions, respectively.
- γ to be used for determining whether a decision region is approximately a homogenous set.

The algorithms and the illustrative examples for steps 4, 5, 7, and 8 are described in more detail in the appendix of this paper and in Pham and Triantaphyllou (2007), Pham and Triantaphyllou (2008). The following section presents some computational results. Furthermore, the following website gives an overview of the HBA

and some computational tools for its implementation: www.csc.lsu.edu/~huypham/HBA_guide.html.

4. A computational study

4.1. The experimental methodology

The procedure for conducting the experiments in this study is described next. The HBA divided the datasets into the training and the testing datasets as depicted in Table 1. For a fair comparison, the number of records in each training dataset was the same as in the studies described in Section 2. However, we had no information on which exact records were used for training. The stand-alone algorithms used in the experiments were SVMs, ANNs, and DTs. Let us consider a certain 3-tuple of the unit penalty costs (C_{FP}, C_{FN}, C_{UC}). The experiments were done as follows:

Step 1: The value for TC_1 was obtained by applying the training dataset T on the stand-alone algorithms and then testing the models by using the testing dataset.

Step 2: The value for TC_2 was obtained by applying the training dataset T on the HBA and then by using the testing dataset as in step 1. In the experiments, we assumed that β^+ and β^- were in the interval [0,2] while α^+ and α^- were in the interval [0, 20].

Step 3: Finally, the two values for TC_1 and TC_2 were compared with each other and all results were recorded.

In other words, for a given 3-tuple (C_{FP}, C_{FN}, C_{UC}), we expect that the values for TC after applying the HBA would be less than or at most equal to the one achieved by the stand-alone algorithms.

4.2. The experimental results

We ran the experiments on a PC with 2.8 GHz speed and 3 GB RAM under the Windows XP operating system. There were more than 300 experiments done on the PID, HSS, BC, LD, and AP datasets with different values for the 3-tuple (C_{FP}, C_{FN}, C_{UC}). The libraries in Neural Network Toolbox 6.0, Genetic Algorithm and Direct Search Toolbox 2.1, and Statistics Toolbox 6.0 (Artificial Neural Network) were used for HBA's implementation. The experimental details are follows:

Test 1: This test would set identical costs, say all costs were equal to one unit, for the false-positive and the false-positive cases, while it would not penalize at all for the unclassifiable cases (i.e., that cost was set equal to 0). This scenario is the same as with most current classification approaches (see Tables 3–7). Thus, the objective function under this test could be:

$$TC = min(1 \times RateFP + 1 \times RateFN)$$

The results for this test are presented in Table 2. This table shows the three failure rates and the values for TC as obtained under the various algorithms. Please notice that the notation "SVM-HBA" means that the HBA was used in conjunction with the SVM algorithm. Similar interpretation holds for DT-HBA (the Decision Tree algorithm and the HBA) and ANN-HBA (the Artificial Neural Network algorithm and the HBA).

Table 2 shows that after 100 generations of the GA approach, the SVM-HBA, DT-HBA, and ANN-HBA approaches when were applied on the medical datasets found the optimal values of the TC. By "optimal" we mean in the GA sense as the actual global optimum value for TC may not have been determined yet. The average values of the TC obtained from the HBA based approaches on the PID, HSS, BC, LD, and AP datasets were 6.2, 9.6, 1.9, 1.9, and 0.0 units, respectively. These values for TC were less (i.e., superior) than the average values of the TC achieved by the stand-alone algorithms based approaches on the PID, HSS, BC, LD, and AP datasets by about 81.6%, 68.5%, 93.2%, 95.2%, and 100.0%, respectively. Fur-

Table 2
Results in minimizing $TC = 1 \times RateFP + 1 \times RateFN$.

Dataset	Algorithm	*RateFP* (%)	*RateFN* (%)	*RateUC* (%)	*TC*	Improvement (%)
PID	SVM	0.0	38.5	56.8	38.5	
	DT	14.1	18.8	61.5	32.8	
	ANN	11.5	20.3	61.5	31.8	
	SVM-HBA	0.0	5.2	74.5	5.2	86.5
	DT-HBA	0.0	8.3	58.9	8.3	74.6
	ANN-HBA	0.0	5.2	74.5	5.2	83.6
HSS	SVM	7.9	22.4	35.5	30.3	
	DT	21.1	14.5	32.9	35.5	
	ANN	11.8	15.8	36.8	27.6	
	SVM-HBA	1.3	7.9	27.6	9.2	69.6
	DT-HBA	1.3	7.9	27.6	9.2	74.1
	ANN-HBA	1.3	9.2	18.4	10.5	61.9
BC	SVM	12.5	15.3	52.8	27.8	
	DT	16.7	12.5	51.4	29.2	
	ANN	18.1	5.6	56.9	23.6	
	SVM-HBA	1.4	0.0	29.2	1.4	95.0
	DT-HBA	2.8	0.0	12.5	2.8	90.5
	ANN-HBA	1.4	0.0	15.3	1.4	94.1
LD	SVM	43.5	0.0	55.1	43.5	
	DT	23.2	17.4	52.2	40.6	
	ANN	23.2	17.4	50.7	40.6	
	SVM-HBA	0.0	0.0	97.1	0.0	100.0
	DT-HBA	1.4	0.0	89.9	1.4	96.4
	ANN-HBA	4.3	0.0	87.0	4.3	89.3
AP	SVM	0.0	4.8	95.2	4.8	
	DT	19.0	4.8	76.2	23.8	
	ANN	0.0	4.8	95.2	4.8	
	SVM-HBA	0.0	0.0	100.0	0.0	100.0
	DT-HBA	0.0	0.0	100.0	0.0	100.0
	ANN-HBA	0.0	0.0	100.0	0.0	100.0

Table 3
Results in the PID dataset.

Algorithm	Accuracy (%)	Avg. improvement (%)
ENN in Smith et al. (1998)	76.0	
IncNet in Jankowski and Kadirkamanathan (1997)	77.6	
Fuzzy approach in Au and Chan (2001)	77.6	
FLEXNFIS in Rutkowski and Cpalka (2003)	78.6	
FNN in Leon (2006)	81.8	
Different approaches in Michie et al. (1994)	77.7	
SVM-HBA	94.8	16.5
ANN-HBA	94.8	16.5
DT-HBA	91.7	13.5

Table 4
Results in the HSS dataset.

Algorithm	Accuracy (%)	Avg. improvement (%)
SVM using linear terms in Kecman and Arthanari (2001)	71.2	
PSVM in Fung and Mangasarian (2001)	72.5	
ISVM in Domm et al. (2005)	62.7	
Logical functions in Shevked and Dakovski (2007)	66.2	
SVM-HBA	90.8	22.6
ANN-HBA	90.8	22.6
DT-HBA	89.5	21.6

Table 5
Results in the AP dataset.

Algorithm	Accuracy (%)	Avg. improvement (%)
PVM in Weiss and Kapouleas (1989)	89.6	
Fuzzy classification system in Nakashima et al. (2003)	84	
C4.5 in Blachnik and Duch (2006)	85.8	
Decision table in Blachnik and Duch (2006)	82	
Nefclass in Blachnik and Duch (2006)	87.7	
HDT in Blachnik and Duch (2006)	85.8	
PTDL in Blachnik and Duch (2006)	83.8	
SVM-HBA	100	14.5
ANN-HBA	96.4	10.9
DT-HBA	89.3	3.7

Table 6
Results in the BC dataset.

Algorithm	Accuracy (%)	Avg. improvement (%)
C4.5 in Quinlan (1996)	94.7	
RI in Hamilton et al. (1996)	96.0	
LDA in Ster and Dobnikar (1996)	96.8	
SVM in Bennet and Blue (1997)	97.2	
Neuro-Fuzzy in Nauck and Kruse (1999)	95.1	
Fuzzy-GA in Pena-Reyes and Sipper (1999)	97.5	
Neuro-Rule in Setiono and Diagnosis (2000)	98.1	
SFC in Abonyi et al. (2003)	95.6	
FAIRS in Polat et al. (2007)	98.5	
SVM-HBA	98.6	2.0
ANN-HBA	98.6	2.0
DT-HBA	97.2	0.6

Table 7
Results in the LD dataset.

Algorithm	Accuracy (%)	Avg. improvement (%)
RULES-4 in Pham and Dimov (2000)	55.9	
C4.5 in Cheung (2001)	65.5	
Naïve Bayes in Cheung (2001)	63.4	
BNND in Cheung (2001)	61.4	
BNNF in Cheung (2001)	61.8	
SSVM in Lee and Mangasarian (2001a)	70.3	
RSVM in Lee and Mangasarian (2001b)	74.9	
SVM in Van et al. (2002)	69.2	
LSSVM in Çomaka et al. (2007)	94.3	
FAIRS in Polat et al. (2007)	83.7	
SVM-HBA	100	30.1
ANN-HBA	98.6	28.6
DT-HBA	95.7	25.7

thermore, the number of false-negative cases obtained from the HBA on these datasets was on the average less than the one achieved by the stand-alone algorithms by about 84.6%.

A comparison between the accurate percentages achieved by the different classification algorithms is presented in Tables 3–7. In all the derived results, the HBA based approaches were significantly more accurate than the stand-alone approaches.

Test 2: Now we consider the test in which the penalty costs for all three error types is assigned to an identical value, say equal to three units (actually the value of 3 makes no difference as long all three costs are identical with each other). Although this test could somehow not be realistic, we would like to consider it to better understand HBA's performance. Thus, the objective function under this test is as follows:

$$TC = min(3 \times RateFP + 3 \times RateFN + 3 \times RateUC)$$

Table 8
Results in minimizing $TC = 3 \times RateFP + 3 \times RateFN + 3 \times RateUC$.

Dataset	Algorithm	*RateFP* (%)	*RateFN* (%)	*RateUC* (%)	*TC*	Improvement (%)
PID	SVM	0.0	38.5	56.8	285.9	
	DT	14.1	18.8	61.5	282.8	
	ANN	11.5	20.3	61.5	279.7	
	SVM-HBA	1.0	20.8	28.1	150.0	47.5
	DT-HBA	0.5	31.8	12.5	134.4	52.5
	ANN-HBA	0.5	29.7	15.1	135.9	51.4
HSS	SVM	7.9	22.4	35.5	197.4	
	DT	21.1	14.5	32.9	205.3	
	ANN	11.8	15.8	36.8	193.4	
	SVM-HBA	1.3	14.5	10.5	78.9	60.0
	DT-HBA	1.3	14.5	9.2	75.0	63.5
	ANN-HBA	1.3	13.2	10.5	75.0	61.2
BC	SVM	12.5	15.3	52.8	241.7	
	DT	16.7	12.5	51.4	241.7	
	ANN	18.1	5.6	56.9	241.7	
	SVM-HBA	15.3	0.0	12.5	83.3	65.5
	DT-HBA	15.3	1.4	8.3	75.0	69.0
	ANN-HBA	15.3	0.0	12.5	83.3	65.5
LD	SVM	43.5	0.0	55.1	295.7	
	DT	23.2	17.4	52.2	278.3	
	ANN	23.2	17.4	50.7	273.9	
	SVM-HBA	29.0	1.4	31.9	187.0	36.8
	DT-HBA	13.0	4.3	53.6	213.0	23.4
	ANN-HBA	14.5	2.9	50.7	204.3	25.4
AP	SVM	0.0	4.8	95.2	300.0	
	DT	19.0	4.8	76.2	300.0	
	ANN	0.0	4.8	95.2	300.0	
	SVM-HBA	0.0	0.0	100.0	300.0	No improvement
	DT-HBA	0.0	0.0	100.0	300.0	No improvement
	ANN-HBA	0.0	0.0	100.0	300.0	No improvement

The results for this test are presented in Table 8. After 100 generations (again, in the GA sense), the SVM-HBA, DT-HBA, and ANN-HBA approaches found the optimal values of *TC*. The average values of *TC* obtained from the HBA based approaches on the PID, HSS, BC, LD, and AP datasets were 140.1, 76.3, 80.5, 201.4, and 300.0 units, respectively. These values for *TC* were less than the average values of *TC* achieved by the stand-alone algorithms based approaches on the PID, HSS, BC, and LD datasets by about 50.5%, 61.6%, 66.7%, and 28.5%, respectively. Table 8 also shows that the optimal values for *TC* when running the HBA based approaches on the AP dataset are the same as the values of *TC* achieved by the stand-alone algorithms. A reason for achieving the identical values for *TC* is that the stand-alone algorithms may have reached the global optimal values (or close to that) for *TC*. Furthermore, the HBA meta-heuristic may have also found the same values. The number of false-negative cases obtained from the HBA on these datasets was on the average less than the one achieved by the stand-alone algorithms by about 55.3%.

An analysis of the impact of the false-positive, the false-negative, and the unclassifiable rates is driven by the corresponding penalty costs. A higher penalty cost placed on one compared to the others implies that the system has more to lose from that type of inaccurate performance than from the others. The following type of experiments show such cases.

Test 3: Now we consider the test in which the application would penalize considerably more for the false-negative case than for the other two cases. This scenario is the most realistic one in situations which deal with life threatening conditions in medical data mining. It is hoped that the higher the penalty cost for the false-negative case is, the fewer cases of the false-negative will be found. In these tests we assumed that the penalty cost for the false-negative case is 20 times higher, while for the false-positive and the

Table 9
Results in minimizing $TC = 1 \times RateFP + 20 \times RateFN + 3 \times RateUC$.

Dataset	Algorithm	*RateFP* (%)	*RateFN* (%)	*RateUC* (%)	*TC*	Improvement (%)
PID	SVM	0.0	38.5	56.8	941.1	
	DT	14.1	18.8	61.5	573.4	
	ANN	11.5	20.3	61.5	602.1	
	SVM-HBA	0.0	8.3	54.7	330.7	64.9
	DT-HBA	2.6	5.2	70.8	319.3	44.3
	ANN-HBA	0.0	5.2	74.5	327.6	45.6
HSS	SVM	7.9	22.4	35.5	561.8	
	DT	21.1	14.5	32.9	409.2	
	ANN	11.8	15.8	36.8	438.2	
	SVM-HBA	1.3	13.2	11.8	300.0	46.6
	DT-HBA	1.3	9.2	18.4	240.8	41.2
	ANN-HBA	1.3	13.2	10.5	296.1	32.4
BC	SVM	12.5	15.3	52.8	476.4	
	DT	16.7	12.5	51.4	420.8	
	ANN	18.1	5.6	56.9	300.0	
	SVM-HBA	15.3	0.0	11.1	48.6	89.8
	DT-HBA	15.3	0.0	12.5	52.8	87.5
	ANN-HBA	15.3	0.0	11.1	48.6	83.8
LD	SVM	43.5	0.0	55.1	1034.8	
	DT	23.2	17.4	52.2	527.5	
	ANN	23.2	17.4	50.7	523.2	
	SVM-HBA	1.4	0.0	89.9	271.0	No improvement
	DT-HBA	0.0	8.7	84.1	426.1	19.2
	ANN-HBA	0.0	0.0	94.2	282.6	45.9
AP	SVM	0.0	4.8	95.2	381.0	
	DT	19.0	4.8	76.2	342.9	
	ANN	0.0	4.8	95.2	381.0	
	SVM-HBA	0.0	0.0	100.0	300.0	21.3
	DT-HBA	0.0	0.0	100.0	300.0	12.5
	ANN-HBA	0.0	0.0	100.0	300.0	21.3

unclassifiable cases these penalty costs are equal to one and three units, respectively. Thus, the objective function under this type of testing is as follows:

$$TC = min(1 \times RateFP + 20 \times RateFN + 3 \times RateUC)$$

These results are presented in Table 9. After 100 generations (in the GA sense), the SVM-HBA, DT-HBA, and ANN-HBA approaches found the optimal values for *TC*. The average values for *TC* obtained from the HBA based approaches on the PID, HSS, BC, LD, and AP datasets were 325.9, 279.0, 50.0, 326.6, and 300.0 units, respectively. These values for *TC* were less than the average values of *TC* achieved by the stand-alone algorithms based approaches on the PID, HSS, BC, LD, and AP datasets by about 51.6%, 40.1%, 87.0%, 11.8%, and 18.3%, respectively. Furthermore, the number of false-negative cases obtained from the HBA based approaches on these datasets was on the average less than the one achieved by the stand-alone algorithms by about 81.3%. Next, we also assumed that the false-negative cases could be penalized much more. Thus, the next objective function was as follows:

$$TC = min(1 \times RateFP + 100 \times RateFN + 3 \times RateUC)$$

These results are presented in Table 10. After 100 generations (in the GA sense), the SVM-HBA, DT-HBA, and ANN-HBA approaches found the optimal values for *TC*. The average values for *TC* obtained from the HBA based approaches on the PID, HSS, BC, LD, and AP datasets were 110.1, 50.0, 1250.4, 263.3, and 300.0 units, respectively. These values for *TC* were less than the average values of *TC* achieved by the stand-alone algorithms based approaches on the PID, HSS, LD, and AP datasets by about 95.7%, 97.3%, 56.1%, and 59.9%, respectively. Please observe that only the SVM-HBA approach was it was applied on the BC dataset yielded a better value for the *TC* than the one achieved by the stand-alone algorithms by about 47.3%. A reason for reaching the higher values (and thus inferior) for *TC* under the DT-HBA and the ANN-HBA approaches on the BC dataset is that the stand-alone algorithms may have reached optimal (or near optimal) values for *TC*. The HBA meta-heuristic could not reach these values. Furthermore, the number of false-negative cases obtained under the HBA on these datasets was on the average less than the one achieved by the stand-alone algorithms by about 82.1%. By comparing the average percentages of the number of false-negative cases achieved un-

Table 10
Results in minimizing $TC = 1 \times RateFP + 100 + RateFN + 3 \times RateUC$.

Dataset	Algorithm	*RateFP* (%)	*RateFN* (%)	*RateUC* (%)	*TC*	Improvement (%)
PID	SVM	0	38.5	56.8	4024.5	
	DT	14.1	18.8	61.5	2073.4	
	ANN	11.5	20.3	61.5	2227.1	
	SVM-HBA	31.8	0.5	12.5	121.4	96.9
	DT-HBA	29.2	0.5	16.7	131.3	93.8
	ANN-HBA	30.7	0	15.6	77.6	96.5
HSS	SVM	7.9	22.4	35.5	2351.3	
	DT	21.1	14.5	32.9	1567.1	
	ANN	11.8	15.8	36.8	1701.3	
	SVM-HBA	14.5	0	13.2	53.9	97.7
	DT-HBA	14.5	0	9.2	42.1	97.3
	ANN-HBA	14.5	0	13.2	53.9	96.8
BC	SVM	12.5	15.3	52.8	1698.6	
	DT	16.7	12.5	51.4	1420.8	
	ANN	18.1	5.6	56.9	744.4	
	SVM-HBA	2.8	8.3	19.4	894.4	47.3
	DT-HBA	0	13.9	15.3	1434.7	No improvement
	ANN-HBA	0	13.9	11.1	1422.2	No improvement
LD	SVM	43.5	0	55.1	208.7	
	DT	23.2	17.4	52.2	1918.8	
	ANN	23.2	17.4	50.7	1914.5	
	SVM-HBA	15.9	0	65.2	211.6	No improvement
	DT-HBA	0	0	97.1	291.3	84.8
	ANN-HBA	0	0	95.7	287	85.1
AP	SVM	0	4.8	95.2	761.9	
	DT	19	4.8	76.2	723.8	
	ANN	0	4.8	95.2	761.9	
	SVM-HBA	0	0	100	300	60.6
	DT-HBA	0	0	100	300	58.6
	ANN-HBA	0	0	100	300	60.6

der the HBA from Tables 9 and 10, one can see that the higher the penalty cost for the false-negative is, the fewer cases of the false-negative type can be found. Clearly, this is a highly desirable property in many medical situations in which these costs may be highly different (as when one deals with life threatening conditions).

5. Conclusions

Medical datasets may possess large amounts of useful information about patients and their medical conditions which may still be unknown to the medical community. Relationships among key attributes of the data and decision regions within these datasets could unveil new and important medical knowledge by using medical data mining approaches. However, current medical data mining approaches oftentimes use identical costs or just ignore the costs for the three different types of classification errors. Thus, the performance of such data mining approaches may be coincidental.

This paper applied a meta-heuristic approach, called the Homogeneity-Based Algorithm (HBA). That is, the HBA first defined the main objective as an optimization problem in terms of the false-positive, false-negative, and unclassifiable rates along with their associated penalty costs. When the HBA is combined with traditional classification algorithms (such as SVMs, DTs, ANNs) then it may significantly enhance their prediction accuracy by using the concept of homogenous sets. The HBA was analyzed on the following well-known medical datasets: the one for the Pima Indian diabetes, the one known as the Haberman Surgery Survival dataset, the Breast Cancer dataset, the Liver Disorders dataset, and the Appendicitis dataset. Each dataset was analyzed under some representative different penalty costs. The derived results clearly show that the total misclassification costs (*TC*s) obtained under the HBA approach are less than the *TC*s achieved by the traditional stand-alone approaches. This appears to have important implications for the computerized diagnosis and treatment of these diseases.

Regarding the penalty costs for classification errors, a theoretical model proposed by Thomas and Hofer (1999) can be used to find their optimal values. Furthermore, analyses on the HBA show that medical datasets which have higher numbers of attributes (i.e., greater than 9 or 10) cannot be tested because of HBA's high complexity. An appropriate solution to decrease HBA's complexity might be to use certain distance based approaches for determining homogenous sets as described in Turner (1989). Current work by the authors of this paper now focuses on developing such alternative approaches, which could also be used in conjunction with traditional data mining methods.

Appendix A

A.1. An algorithm for fragmenting regions into hyperspheres

Let us consider a decision region C of size n_C. Fig. 5 shows a heuristic algorithm to find the minimum number of hyperspheres that can cover C. At first, the densities of the n_C points in C are estimated by using Eq. (11) as described in Appendix A.5. Next, the algorithm sets the values for K from 1 to n_C. Each iteration will pick K points in C with the highest densities and use them as centroids in the K-means clustering approach to form hyperspheres. The loop will stop if the K hyperspheres cover C. Otherwise, we do the same with the next value for K.

A.2. An algorithm for determining homogenous sets

Let us consider a hypersphere C of size n_C. At first, hypersphere C is divided into a number of small bins of the same size h and then the density at the center x of each bin is estimated. If the densities at all centers are approximately equal to each other, then C is a homogenous set. The algorithm is summarized in Fig. 6.

Input: Decision region C of size n_C.

1. Estimate the densities of the n_C points by using Equation (11) mentioned in Section 6.5.
2. For K=1 to n_C do
   Pick K points in C with the highest densities.
   Use the K-means clustering approach to find K hyperspheres.
   If the K hyperspheres cover C, then STOP.
   Else, $K = K + 1$.

Output: K hyperspheres.

Fig. 5. The algorithm for fragmenting regions into hyperspheres.

One can relax the condition which requires to have identical densities at all centers of the bins. That is, if the standard deviation of the densities at all centers of the bins is approximately less or equal to some threshold γ, say for $\gamma = 0.01$, then the hypersphere C may be considered to be a homogenous set.

As mentioned in Section 3.1, one needs to determine an appropriate definition for *HD(C)*. Pham and Triantaphyllou (2007) and Pham and Triantaphyllou (2008) proposed a way for computing *HD(C)* as follows:

$$HD(C) = \frac{\ln(n_C)}{h} \tag{2}$$

The value for *HD(C)* depends on the value h defined in Heuristic Rule 1 as mentioned next in Appendix A.5 and n_C. In fact, if h increases, then the average distance between pairs of points in C increases. This issue leads *HD(C)* to decrease. Furthermore, if n_C increases, then *HD(C)* would somewhat increase since the volume of C does not change and C has more points. Hence, the value for *HD(C)* is inversely proportional to h, while *HD(C)* is directly proportional to n_C. In Eq. (2), the function $\ln(n_C)$ is used to show the slight effect of n_C on *HD(C)*.

A.3. Expansion algorithms

Let us consider a positive (or negative) hypersphere F with its homogeneity degree *HD(F)*, the fragmenting threshold value β^+, and the expansion threshold value α^+. There are two types of expansion: a radial expansion in which the hypersphere F is expanded in all directions and a linear expansion in which the hypersphere F is expanded in a certain direction. The following section shows the details for these two expansion types.

Input: Hypersphere C and density threshold value γ.

1. Compute the distances between all pairs of points in C.
2. Let h be the distance as mentioned in Heuristic Rule 1.
3. Superimpose C into hypergrid V of unit size h.
4. Approximate the density at the center x of each bin.
5. Compute the standard deviation of the densities at the centers of the bins.
6. If the standard deviation is less than or equal to γ, then
   C is a homogenous set and its homogeneity degree *HD(C)* is computed by using Equation (2).
   Else, C is not a homogenous set.

Output: Decide whether C is a homogenous set.

Fig. 6. The algorithm for determining homogeneous sets.

Input: Hypersphere F with $HD(F)$, R_F, and α^+

1. Set $M = F$ (i.e., $R_F = R_M$).
2. Set hypersphere G covering M with radius $R_G = 2 \times R_M$.
3. Repeat
   Set $E = M$ (i.e., $R_E = R_M$).
   Expand M by using Equation (5).
   Until (R_M satisfies the stopping conditions discussed in Section 6.3.3 or $R_M = R_G$).
4. If R_M satisfies the stopping conditions, then STOP.
   Else, go to Step 2.

Output: An expanded region E.

Fig. 7. The algorithm for the radial expansion.

A.3.1. Radial expansion

In the radial type, let a hypersphere M be a region, which has been formed by expanding the hypersphere F. The notations R_F and R_M stand for the radiuses of F and M, respectively. In the radial expansion algorithm depicted in Fig. 7, the radius R_F is increased by a certain value denoted as T and is called a *step-size increase*:

$$R_M = R_F + T \tag{3}$$

Pham and Triantaphyllou (2007) and Pham and Triantaphyllou (2008) proposed a value for T as follows:

$$T = \frac{R_G - R_F}{2} \times \frac{1}{L \times HD(F)} \tag{4}$$

A threshold value L in Eq. (4) ensures that $HD(F)$ is always greater than one. If we substitute Eq. (4) back into Eq. (3), R_M becomes:

$$R_M = R_F + \frac{R_G - R_F}{2} \times \frac{1}{L \times HD(F)} \tag{5}$$

A.3.2. Linear expansion

In the linear type, a hypersphere F is first expanded to form hypersphere M by using the radial expansion. Then, the hypersphere M is expanded in given directions by using the radial approach until it meets the stopping conditions mentioned next in Appendix A.3.3. The final expanded region is the union of all the expanded regions.

A.3.3. Stopping conditions for radial expansion

The stopping conditions for expanding a hypersphere F size of n_F should:

- Depend on $HD(F)$.
- Stop when F's expanded region meets other decision regions. In the expansion process, the expanded region can accept several noisy data points if the value of $HD(F)$ is high.

In the first stopping condition, the radius R_M of the expanded region M should not be greater than the product of $HD(F)$, α^+, and R_F. The second stopping condition can be determined while expanding. The number of noisy points should be directly proportional to $HD(F)$ and inversely proportional to n_F. The stopping conditions are shown in Eq. (6) (a similar way exists for α^-):

$$R_M \leqslant HD(F) \times R_F \times \alpha^+ \text{ and}$$
$$\text{The number of noisy points} \leqslant \frac{HD(F) \times \alpha^+}{n_F} \tag{6}$$

A.4. The GA approach

As it was described in the previous algorithms, the four threshold values (α^+, α^-, β^+, β^-) are used to control the number of misclassification cases of the final models. If the values for β^+ and β^- are too high, then the regions will be fragmented into hyperspheres of size one. This results in the overfitting problem. Otherwise, too low value for the fragmenting thresholds may not be adequate to deal with overgeneralization problem. The opposite situation is true with α^+ and α^-. Furthermore, the search space for α^+, α^-, β^+, and β^- may be large. Therefore, an exhaustive search would be impractical.

A^+	α^-	β^+	β^-

Fig. 8. A chromosome.

Pham and Triantaphyllou (2007) and Pham and Triantaphyllou (2008) proposed to use a GA approach for finding approximate optimal values for α^+, α^-, β^+, and β^-. The HBA uses Eq. (1) as the fitness function and the dataset T_2 mentioned in Section 3.2 as a calibration dataset. The GA approach has been applied here because Eq. (1) is not unimodal. Each chromosome in the GA approach consists of four genes which correspond to the four threshold values (α^+, α^-, β^+, β^-) as depicted in Fig. 8.

The crossover function creates children by combining pairs of parents in the current population. At each coordinate of the child, the crossover function randomly picks the gene up at the same coordinate from one of the two parents and then assigns it to the child. The mutation function creates a child (g_1, g_2, g_3, g_4) by randomly changing the genes of the parent chromosome (α^+, α^-, β^+, β^-). Let us consider the first two genes α^+ and α^- belonging to the range $[a, b]$, while the last two genes β^+ and β^- are in the range $[c, d]$. The mutation function first randomizes a chromosome (t_1, t_2, t_3, t_4) by using the Gaussian distribution (this random distribution was determined empirically). Next, the genes of the mutation child are created by using one of the following Eqs. (7) and (8):

$$g_1 = ((\alpha^+ \text{or } t_1) \text{ or } a) \text{ and } b, \quad g_2 = ((\alpha^- \text{ or } t_2) \text{ or } a) \text{ and } b \tag{7}$$
$$g_3 = ((\beta^+ \text{ or } t_3) \text{ or } c) \text{ and } d, \quad g_4 = ((\beta^- \text{ or } t_4) \text{ or } c) \text{ and } d \tag{8}$$

The GA approach is terminated when the fitness function during successive iterations results in no improvement.

A.5. Non-parametric density estimation

As seen in the previous sections, the density estimation of a typical bin is used in many algorithms. Pham and Triantaphyllou (2007) and Pham and Triantaphyllou (2008) proposed to use Parzen Windows, a non-parametric density estimation described in Duda and Hart (1973), for the density estimation. A fundamental assumption in the Parzen Windows approach states that a bin R is a D-dimensional hypercube of unit size h. Under this consideration, the Parzen Windows approach defines a kernel function $\varphi(u)$ to find the number of points that fall within this bin as follows:

$$\varphi(u) = \begin{cases} 1, & |u| \leqslant 1/2. \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Eq. (9) shows that the value for $\varphi(\frac{x-x_i}{h})$ is equal to unity if the point x_i is inside the bin of unit size h and centered at x, and zero otherwise. For the extension, the kernel function in the D-dimensional space can be formed as follows:

$$\varphi\left(\frac{x - x_i}{h}\right) = \prod_{m=1}^{D} \varphi\left(\frac{x^m - x_i^m}{h}\right) \tag{10}$$

Let us consider a region C of size N. Point x is in C and $d(x)$ denotes x's density, then:

$$d(x) \approx \frac{1}{N \times h^D} \sum_{i=1}^{N} \prod_{m=1}^{D} \varphi\left(\frac{x^m - x_i^m}{h}\right) \tag{11}$$

As it can be seen in Eq. (11), choosing an appropriate value for h provides a smoother value for $d(x)$. Let us define S as a set of the distances of pairs of training points within C that have the highest frequency. Pham and Triantaphyllou (2007) and Pham and Triantaphyllou (2008) proposed an appropriate value for h to be as follows:

Heuristic Rule 1: If the minimum value in set S is assigned to h and we use h to compute d(x) with Eq. (11), then d(x) approaches to a true density.

References

Abonyi, J., & Szeifert, H. (2003). Supervised fuzzy clustering for the identification of fuzzy classifiers. *Pattern Recognition Letters, 24*, 2195–2207.

Artificial Neural Network Toolbox 6.0 and Statistics Toolbox 6.0, Matlab Version 7.0. <http://www.mathworks.com/products/>.

Asuncion, A., Newman, D. J. (2007). *UCI-Machine Learning Repository*. School of Information and Computer Sciences, University of California, Irvine, CA, USA.

Au, W. H., Chan, K. C. C. (2001). Classification with degree of membership: a fuzzy approach. In *Proceedings of the 1st IEEE international conference on data mining* pp. 35–42. San Jose, CA, USA.

Bennet, K. P., Blue, J. A. (1997). *A support vector machine approach to decision trees,*. Math Report, No. 97-100, Rensselaer Polytechnic Institute, Troy, NY, USA.

Blachnik, M., & Duch, W. (2006). *Prototype-based threshold rules. LNCS of neural information processing* (Vol. 4234). Berlin/Heidelberg: Springer. pp. 1028–1037.

Cheung, N. (2001). *Machine learning techniques for medical analysis*. BSc thesis, School of Information Technology and Electrical Engineering, University of Queensland, Australia.

Çomaka, E., Polatb, K., Güneşb, S., & Arslana, A. (2007). A new medical decision making system: Least square support vector machine (LSSVM) with fuzzy weighting pre-processing. *Expert Systems with Applications, 32*(2), 409–414 (February).

Domm, M., Engel, A., Louis, P. P., Goldberg, J. (2005). An integer support vector machine. In *Proceedings of the 6th international conference on software engineering, artificial intelligence, networking and parallel/distributed computing* pp. 144–149. Towson, MD, USA.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. New York, NY, USA: Wiley Publisher. pp. 56–64.

Fung, G., Mangasarian, O. L. (2001). Proximal support vector machine classifiers. In *Proceedings of the 7th ACM SIGKDD international conference on knowledge discovery and data mining* pp. 77–86. San Francisco, CA, USA: ACM Press.

Hamilton, H. J., Shan, N., Cercone, N. (1996). *RIAC: A rule induction algorithm based on approximate classification*. Technical Report, No. CS 96-06, University of Regina, Regina, Canada.

Jankowski, N., Kadirkamanathan, V. (1997). Statistical control of RBF-like networks for classification. In *Proceedings of the 7th international conference on artificial neural networks (ICANN)* pp. 385–390. Lausanne, Switzerland.

Kecman, V., & Arthanari, T. (2001). Comparisons of QP and LP based learning from empirical data. In L. Monostori, J. Ývancza, & M. Ali (Eds.), *LNCS and LNAI* (pp. 326–332). New York, NY, USA: Springer (June).

Lee, Y. J., Mangasarian, O. L. (2001). RSVM: Reduced support vector machines. In *Proceedings of the first SIAM international conference on data mining*, Chicago, IL, USA.

Lee, Y. J., & Mangasarian, O. L. (2001a). SSVM: A smooth support vector machine for classification. *Computational Optimization and Applications, 20*(1), 5–22.

Leon, W. D., IV., (2006). *Enhancing pattern classification with relational fuzzy neural networks and square BK-products*. PhD dissertation in computer science. FL, USA: Springer (pp. 71–74).

Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). Machine learning neural and statistical classification. In *Series artificial intelligence* (pp. 157–160). Englewood Cliffs, NJ, USA: Prentice Hall.

Nakashima, T., Nakai, G., Ishibuchi, H. (2003). Constructing fuzzy ensembles for pattern classification problems. In *Proceedings of the international conference on systems, man and cybernetics* (Vol. 4, pp. 3200–3205). Washington, DC, USA (October).

Nauck, D., & Kruse, R. (1999). Obtaining interpretable fuzzy classification rules from medical data. *Artificial Intelligence in Medicine, 16*, 149–169.

Pena-Reyes, C. A., & Sipper, M. (1999). A fuzzy-genetic approach to breast cancer diagnosis. *Artificial Intelligence in Medicine, 17*, 131–155.

Pham, D. T., Dimov, S. S., Salem, Z. (2000). Technique for selecting examples in inductive learning. In *Proceedings of the European symposium on intelligent techniques (ESIT 2000)* pp. 119–127. Aachen, Germany.

Pham, H. N. A., & Triantaphyllou, E. (2007). The impact of overfitting and overgeneralization on the classification accuracy in data mining. In O. Maimon & L. Rokach (Eds.), *Soft computing for knowledge discovery and data mining, part 4* (pp. 391–431). New York, NY, USA: Springer.

Pham, H. N. A., & Triantaphyllou, E. (2008). Prediction of diabetes by employing a new data mining approach which balances fitting and generalization. In Roger Yin Lee (Ed.). *Studies in computation intelligence* (Vol. 131, pp. 11–26). Germany: Springer.

Polat, K. A., Sahan, S., Kodaz, H., & Gunes, S. (2007). Breast cancer and liver disorders classification using artificial immune recognition system (AIRS) with performance evaluation by fuzzy resource allocation mechanism. *Expert Systems with Applications, 32*, 172–183.

Quinlan, J. R. (1996). Improved use of continuous attributes in C4.5. *Artificial Intelligence Research, 4*, 77–90.

Rutkowski, L., & Cpalka, K. (2003). Flexible neuro-fuzzy systems. *IEEE Transactions on Neural Networks, 14*, 554–574.

Setiono, R. (2000). Generating concise and accurate classification rules for breast cancer diagnosis. *Artificial Intelligence in Medicine, 18*, 205–219.

Shevked, Z., Dakovski, L. (2007). Learning and classification with prime implicants applied to medical data diagnosis. In *Proceedings of the 2007 international conference on computer systems and technologies*, Rousse, Bulgaria, June.

Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., Johannes, R. S. (1998). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the 12th symposium on computer applications and medical care* pp. 261–265. Los Angeles, CA, USA.

Ster, B., Dobnikar, A. (1996). Neural networks in medical diagnosis comparison with other methods. In *Proceedings of the international conference on engineering applications of neural networks (EANN'96)* pp. 427–430. London, UK.

Thomas, J. W., & Hofer, J. P. (1999). Accuracy of risk-adjusted mortality rate as a measure of hospital quality of care. *Medical Care, 37*(1), 83–92.

Turner, M. G. (1989). Landscape ecology: The effects of pattern on process. *Annual Review of Ecology and Systematics, 20*, 171–197.

Van, G. T., Suykens, J. A. K., Lanckriet, G., Lambrechts, A., de Moor, B., & Vandewalle, J. (2002). Bayesian framework for least squares support vector machine classifiers Gaussian processes and Kernel Fisher discriminant analysis. *Neural Computation, 14*, 1115–1147.

Weiss, S. M., Kapouleas, I. (1989). An empirical comparison of pattern recognition, neural nets, and machine learning classification methods. In *Proceedings of the 11th international joint conference on artificial intelligence* pp. 781–787. Detroit, MI, USA.