Front. Comput. Sci., 2016, 10(6): 1082–1102
DOI 10.1007/s11704-016-5203-5

RESEARCH ARTICLE

Impact of preprocessing on medical data classification

Sarab ALMUHAIDEB (✉), Mohamed El Bachir MENAI

Computer Science Department, College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia

Abstract The significance of the preprocessing stage in any data mining task is well known. Before attempting medical data classification, characteristics of medical datasets, including noise, incompleteness, and the existence of multiple and possibly irrelevant features, need to be addressed. In this paper, we show that selecting the right combination of preprocessing methods has a considerable impact on the classification potential of a dataset. The preprocessing operations considered include the discretization of numeric attributes, the selection of attribute subset(s), and the handling of missing values. The classification is performed by an ant colony optimization algorithm as a case study. Experimental results on 25 real-world medical datasets show that a significant relative improvement in predictive accuracy, exceeding 60% in some cases, is obtained.

Keywords classification, ant colony optimization, medical data classification, preprocessing, feature subset selection, discretization

1 Introduction

Modern clinical information systems store an extensive amount of data in medical databases. This encourages the extraction of useful knowledge from medical databases, providing valuable insight for medical decision support. A branch of data mining, known as medical data mining, is currently considered one of the most popular research subjects in the data mining community [1]. This, in part, is due to the societal significance of the subject and also to the computational challenges it presents [2].

Normally, there exists a dataset of historic data describing a particular medical disorder. Such datasets consist of patients' records relating to demographic, clinical and pathological data, along with the results of medical investigations that have been collected for the diagnosis and prognosis of a particular medical disorder. Modern medical screening and diagnostic methods generate a high volume of heterogeneous data. These data are continually accumulating. Thus, mining such data requires intelligent methods [3,4].

Medical data classification (MDC) refers to learning classification models from medical datasets and aims to improve the quality of health care [5]. Medical data classification can be used for diagnosis and prognosis purposes. MDC is different from medical classification, or medical coding, which is the process of assigning internationally endorsed classification codes to each medical diagnosis and procedure (the WHO Family of International Classifications[1]).

Medical data exhibit unique features including noise resulting from human as well as systematic errors, missing values and even sparseness [4]. Table 1 illustrates medical dataset examples. These datasets are obtained from the University of California Irvine (UCI) repository of machine learning datasets[2]. For example, some datasets, like dermatology, consist of different types of attributes. As another example, the high dimensionality is a feature of the cardiac arrhythmia dataset. The thyroid dataset contains more than 7 000 instances. The hepatitis dataset is imbalanced. The

Received May 24, 2015; accepted January 8, 2016

E-mail: smuhaideb@psu.edu.sa

[1] http://www.who.int/classifications/en/
[2] http://archive.ics.uci.edu/ml/datasets.html

Table 1 Example medical datasets and their associated complexity

Dataset	No. instances	No. attributes	No. classes	Missing values	Input data type
Cardiac arrhythmia	452	279	16	0.32%	206 real, 73 nominal
Hungarian heart	294	13	5	20.46%	3 binary, 10 real
Dermatology	366	34	6	0.06%	1 nominal, 1 binary, 32 integer
Hepatitis	155	19	2[a]	5.67%	13 integer, 6 real
Thyroid	7 200	21	3	No	15 binary, 6 real

Note: [a]=Live (79.35)/Die (20.65%)

percentage of missing values in the Hungarian heart dataset exceeds 20%. Due to this nature, Tanwani et al. [4] called for the classification of medical data as a separate domain.

Data preprocessing has a profound effect on the performance of the learner. The classification potential of a dataset can be improved to a large extent by selecting the right combination of preprocessing methods. Preprocessing is especially important for medical datasets due to their characteristics. However, each dataset is different, and there is no preprocessing method that is best across all datasets. Deciding the best combination of preprocessing methods for a specific dataset is not possible without trial and comparisons. Technology is advancing rapidly. The advent of various open-source libraries, like Weka [6] and KEEL [7], hosting an extensive set of off-the-shelf preprocessing methods, combined with the leisure of standard formats like the attribute-relation file format (ARFF)[3] and advances in computer hardware technology, encourages integration of automatic tuning for preprocessing operations into the data mining task for each dataset on an individual basis. The idea is suitable for off-line applications.

In this research, we investigate the influence of individualized preprocessing on the classification of medical datasets, including the removal of missing values and a variety of discretization and attribute selection methods. Experiments were conducted on 25 real-world medical datasets from the UCI machine learning repository. Datasets are then classified by means of the AntMiner$^+$ algorithm [8]. Numerical results show that there is a significant improvement in classification performance, as measured by predictive accuracy, with relative improvement exceeding 60% in some cases, obtained in the majority of datasets in the benchmark, through the individualized tuning of the preprocessing operations. Moreover, given a certain classification algorithm, the design of the preprocessing stage can mean the difference between complete failure and the achievement of good results on the same datasets.

The rest of the paper is organized as follows. Section 2 highlights related work in the area. A discussion about AntMiner$^+$ as a classification algorithm from the family of ant colony optimization (ACO) algorithms is presented in Section 3. Next, Section 4 describes the individualized tuning procedure. Experimental results are presented in Section 5 and discussed in Section 6. The paper is concluded in Section 7.

2 Related work

Tanwani and Farooq [9–11] performed an extensive study to present the challenges associated with biomedical data and approximate the classification potential of a biomedical dataset using a qualitative measure of this complexity. The study concludes that the classification accuracy is found to be dependent on the complexity of the biomedical dataset, not on the classifier choice. The number and type of attributes have no noticeable effect on the classification accuracy, as compared to the quality of the attributes. It is shown that biomedical datasets are noisy and that noise is the dominant factor that affects the resulting classification accuracy. Only high percentages of missing values severely degrade the classification accuracy. The study also shows that evolutionary algorithms tend to overfit for small-sized datasets and are not much affected by the class imbalance problem.

The quality of data has a large implication for the quality of the mining results. It is necessary to perform a preprocessing step in order to remove or at least alleviate some of the problems associated with medical data. Depending on the characteristics of the data themselves, many preprocessing techniques are pertinent. In this section, methodologies for dimensionality reduction (Section 2.1), discretization algorithms (Section 2.2), and handling missing values (Section 2.3) are described.

2.1 Dimensionality reduction

As medical datasets are generally characterized as having high dimensionality, this section introduces techniques com-

[3] http://weka.wikispaces.com/ARFF

mon for dealing with this problem. There are two common dimensionality reduction techniques: feature construction (FC) and feature subset selection (FSS). FC is a transformation technique that constructs a new, reduced feature space with a strong discriminative power but also loses the original feature characteristics, which leads to difficulties in the interpretation of the resulting models. FSS finds the minimum subset of features that are useful for the classification process. Although several features are discarded, this method preserves the original physical interpretation of features. Further, in medical diagnosis, it is desirable to select the clinical tests that have the least cost and risk and that are significantly important for determining the class of the disease.

The first step is the subset search step. A search engine is used to generate candidate feature subsets for evaluation based on a certain search strategy. An exhaustive search strategy is practically prohibitive unless the number of variables is small. Greedy and heuristic methods are more efficient in this respect [12] and include sequential forward selection (SFS) [13] and sequential backward selection (SBS) [14]. Hybrid forms of SFS and SBS have been devised to reduce the nesting effect, such as sequential forward floating selection (SFFS) and sequential backward floating selection (SBFS) [15]. However, all of these methods provide a suboptimal solution [16].

Due to their well-known efficiency in combinatorial optimization problems, optimization-based search methods are also applied to FSS. The use of metaheuristics along with their hybrids is well established in the FSS research [16–20].

Several feature subset selection techniques do not employ any search strategy. Individual features are ranked (weighted) according to their relevance, independently of others' context. Feature ranking is not used to build predictors but is a simple, scalable approach, with good empirical success [21], which can be used as a baseline method to construct nested subsets for building predictors. Similar to filters, correlation and information theoretic measures can be used in feature weighing and ranking. Another approach is to use single-variable classifiers, which rank variables according to their discriminative power when used individually to build a classifier.

Next, the candidate feature subset is evaluated in the subset evaluation step. The evaluation criterion is essential in determining the best candidate subset. FSS methods can be classified into two approaches based on their dependency on the learning model: model-free and model-based approaches.

• Model-free approach In the model-free (filter) approach, features are selected independently of the classification algorithm, as a pre-processing step. The selection employs measures that utilize intrinsic characteristics of the data to determine the relevance between the features and the class. These measures evaluate the class separability. Among the popular measures applied are distance [22], consistency [23] and correlation measures [24,25]. All these measures rely on the actual values of the training dataset and are thus sensitive to noise and outlier values [12]. Information measures [26–28] are popular as well.

• Model-based approach The model-based approach for FSS applies a specific learning algorithm and uses its predictive accuracy as a measure of the subset effectiveness. Model-based methods fall into two types [29]. Wrappers [30,31] use the learning machine as a black box. Embedded methods incorporate the feature subset selection process as a part of the training process, such as that in genetic programming [32] and decision tree algorithms like CART [33].

Hybrid methods exist as well. For example, as an efficient method, filters can be used initially to eliminate definitely redundant features, thus reducing the search space, and then a wrapper can be used for the search in the second phase [34].

2.2 Discretization algorithms

Some learning algorithms cannot deal directly with numeric attributes. Discretization is thus an essential step to transform these attributes into a form that can be handled. The numeric domain is partitioned into a finite number of non-overlapping intervals. An association is then established between each numeric value that belongs to the attribute and the interval to which it belongs. As a side effect, discretization works as a data reduction and simplification method because it converts the huge numeric domain into a much smaller subset of nominal intervals [35]. Finding the optimal discretization of a numeric attribute is NP-hard [36]. A recent survey [37] lists over 80 discretization methods that belong to 33 different categories of the discretization taxonomy.

If the class information is considered along with the evaluation measure during discretization, then it is called a supervised discretization method. Most available discretization methods belong to the supervised discretization type. The class label is not considered in unsupervised methods. EqualWidth and EqualFrequency [38] discretizers are examples of unsupervised discretization methods.

Simple discretization methods do not use any evaluation measure to decide cut points. A predefined number of bins n is established and the domain is partitioned into n equal-width intervals (EqualWidth) or into n intervals that contain

an equal number of values per attribute (EqualFrequency) [38]. Other discretization methods rely on some metric to decide the next cut (or merge) point. Common evaluation criteria include information measures like entropy and its derived measures, as in ID3 [28] and the Fayyad and Irani discretizer [26]. Statistical measures use correlation and consistency measures to determine cut (or merge) points, as in ChiMerge [24]. Alternatively, a simple classifier is run for each evaluation, and the classification error is the metric used for the decision for cut (or merge) points.

2.3 Handling missing values

When dealing with real-life medical data, missing or unknown values are unavoidable. For example, in a diagnosis problem, the results of some tests may not be available because a health care institution lacks these tests. A physician may decide that some tests need not be performed, as their results are evident or they are unrelated. A test may also be avoided due to the high hazard or cost associated. The model learning process must deal with such missing values.

The most popular methods for handling missing values [39] are based either on the removal of missing values or on imputation methods. The latter aim to substitute missing values with new values as close as possible to the real missing ones. Imputation methods includes statistical methods such as mean (and mode) imputation, regression imputation, hot and cold deck imputation (which replace missing values with the corresponding values from a similar case to the one with missing values), and multiple imputation (where multiple datasets are generated, each with a possible value for the missing ones). Hot deck analysis might require several runs to replace missing values and may end up not completely eliminating missing values when the percentage of missing values is large [40]. In imputation based on machine learning methods, a model is built to predict the missing values of a certain feature based on available other features, such as employing the k-nearest neighbor (k-NN) algorithm [41], support vector machines (SVMs) [42], and artificial neural networks (ANNs) [43]. A separate model has to be built for each attribute having missing values. These methods are generally robust; however, the main drawback is their high computational cost and sensitivity to some parameters. Model-based imputation methods [44–46] are based on assumptions about the statistical probability distribution of the variables in the model, such as the expectation-maximization (EM) algorithm.

3 Classification with ant colony optimization algorithms

A wide spectrum of algorithms for classification model learning has been proposed, each with associated strengths and limitations. Metaheuristic methods stand as interesting techniques, because of their good performance and low computational requirements. Metaheuristics require little or no background knowledge of the problem at hand. Although finding the optimal solution is not guaranteed, metaheuristics obtain reasonable solutions in acceptable time. Among metaheuristics, algorithms from the family of swarm intelligence are particularly interesting when dealing with biological systems [47–52]. This is because these algorithms are themselves inspired by biology. In ant colony optimization algorithms, artificial ants use pheromone trails and heuristic information to guide solution construction for finding the shortest path from food sources to their nest. The pheromone is a special chemical released by ants during their search. The amount of pheromone laid on a path increases with the number of ants passing along that path and, thus, attracts more ants to follow. With time, the pheromone evaporates, causing old and expired paths to be forgotten [53]. Pheromone evaporation facilitates the adaptation and fast finding of new, suboptimal paths in a robust and reactive way when sources of food change dynamically. Ant-Miner [54] is the first ACO algorithm for classification tasks. Among the different variants of Ant-Miner, AntMiner$^+$ [8] has been chosen as the classification algorithm in this research. AntMiner$^+$ is based on the MAX–MIN ant system [55], which is recognized as one of the best-performing algorithms in the ant colony optimization family [56]. The classification model is constructed using the sequential covering strategy. The results reported show that AntMiner$^+$, on average, obtained the highest rank among all rule-based classifiers included [8], such as the C4.5 algorithm [28], RIPPER [57], and Ant-Miner [54], as well as other classification methods like logistic regression, 1-nearest neighbor and RBF-SVM (Vapnik 1995). A study by Minnaert et al. [58] examines several performance measures for sequential covering rule-induction algorithms. AntMiner$^+$ has been successfully hybridized with other nature-inspired metaheuristics for the task of medical data classification in the literature [5,59].

4 Individualized preprocessing procedure

The AntMiner$^+$ is based on a sequential-covering strategy

and a default rule related to the majority class. In effect, rule induction focuses on classes other than the majority class. This particular strategy is advantageous in MDC because the majority of class instances are normally the negative cases of which we care less. The sequential-covering strategy helps in handling large-sized datasets; due to the removal of instances already covered by induced rules, the progressive reduction of the training set size is thus achieved.

AntMiner$^+$ algorithm cannot handle instances containing missing values. Thus, these instances are removed from the dataset in the first step. To reduce the size of the solution space, the number of attributes is limited to no more than a default value of 10. If the dataset contains a larger number of attributes, then attribute selection takes place prior to induction.

Various attribute types can be handled by the AntMiner$^+$ algorithm. These include nominal and ordinal values, as well as numeric values, including integer and continuous attributes that are discretized. In effect, numeric values are encoded as discrete intervals defined by [$lower\_bound - upper\_bound$].

The order of preprocessing steps in the concerned AntMiner$^+$ implementation is as follows:

1) removal of instances with missing values;
2) discretization;
3) attribute selection.

The AntMiner$^+$ configuration is tuned for the following aspects:

1) timing of removing instances having missing values (Section 4.1);
2) discretization algorithm (Section 4.2);
3) feature subset selection algorithm (Section 4.3);
4) rule evaluation function (Section 4.4).

4.1 Timing of removing instances having missing values

In the context of the AntMiner$^+$ algorithm, all instances having missing values are removed in the first step of preprocessing. The next steps in the preprocessing consist of the application of the discretization algorithm and attribute selection algorithm (if necessary). This procedure might not be the best in some cases. For example, consider datasets with a large number of predictive attributes. If the removal of instances having missing values is delayed after the attribute selection step, then this would allow more instances to be available for training and testing subsets, thus perhaps improving the results. Otherwise, some instances would be removed because they include missing values in attributes that will be next removed by the attribute selection step. Thus, the removal of these instances is no longer rationalized. We hypothesize that if the removal of instances with missing values were delayed until after the attribute selection step, then better results would be obtained.

4.2 Discretization method

Different discretization methods exist, but none can prove to be the best across all problems and learners [37]. When dealing with a specific problem or dataset, the choice of the discretization method has a considerable effect on the classification results in terms of both predictive accuracy and model simplicity.

Four discretization methods are selected for discretization tuning. All of them are classified as static, univariate, and splitting methods. A brief description of each, along with its acronym used, is presented next.

- Fayyad and Irani discretizer (fay)
  The Fayyad and Irani discretizer [26] is one of the most popular discretizers that obtains a reasonable balance between the number of intervals and the accuracy obtained [37]. It is based on a supervised method that uses an entropy measure to decide split points. Stopping is based on a minimum description length (MDL) [60] criteria that explains the attractive balance between model complexity (number of intervals in this case) and performance (accuracy).
- Kononenko's MDL discretizer (kon)
  This is similar to the Fayyad and Irani discretizer, but it uses the Kononenko's MDL criterion [61]. The Kononenko's MDL criterion has a lower bias in handling multi-valued attributes and multi-class problems.
- EqualWidth discretizer (eib)
  EqualWidth [38], or equal interval binning (eib), partitions the continuous domain into a predefined number of equal-width bins. For each dataset, a number of 5, 10, 15, and 20 intervals are examined. The resulting models are referred to as eib5, eib10, eib15, and eib20, respectively. EqualWidth is an unsupervised discretization method.
- EqualFrequency discretizer (efb)
  EqualFrequency [38], or equal frequency binning (efb), partitions the continuous domain into a predefined number of intervals such that the intervals have an equal

number of values. Similar to eib, for each dataset, a number of 5, 10, 15, and 20 intervals are examined. The resulting models are referred to as efb5, efb10, efb15, and efb20, respectively. Also similar to eib, efb is an unsupervised discretization method.

4.3 Feature subset selection method

FSS (or attribute selection) process is described relatively to four aspects: subset search, subset evaluation, halting criteria, and result validation.

Following is a list of the considered FSS methods. Next, for each method we explain the strategy used for subset search, subset evaluation, halting criteria, and result validation.

1) ReliefF attribute evaluation (rel) [62,63].
2) Correlation-based feature subset selection (cfs) [64].
3) Consistency subset evaluation (con) [65].
4) Chi-squared attribute evaluation (chi).
5) Gain ration attribute evaluation (gai).
6) Information gain attribute evaluation (inf).
7) OneR attribute evaluation (1R).
8) Symmetrical uncertainty attribute evaluation (sym) [66].
9) No attribute selection employed (0AS).

• Subset search

The subset search methods described in Section 2.1 can be grossly classified into exhaustive, exact, greedy, and heuristic methods. In addition, there is the simple feature-ranking method. Exhaustive methods are computationally intractable. Heuristic methods require high computational time, which is not suitable in combination with AntMiner$^+$. For (cfs) and (con), the best first search is used with a backward search direction. The rest of the methods use the simple ranking of attributes according to the attribute evaluation used [21]. Ranking is chosen due to its scalability, simplicity, and performance [29].

• Subset evaluation

The methods selected include both model-free and model-based subset evaluation. As for the model-free evaluation methods, the selection includes distance-based (rel), consistency-based (con), correlation-based (cfs), and information theoretic-based evaluation measures (chi, gai, and inf). Attributes are individually evaluated using ReliefF (rel). The (gai) method measures gain ratio with respect to class. The (inf) method evaluates each attribute individually according to its measured information gain with respect to class. The (chi) method uses the chi-squared statistical measure to assess the degree of independence between the attribute and the associated class. The (chi) method works on categorical attributes.

A model-based evaluation is also included (1R). Each attribute is evaluated individually by using the simple OneR classifier [67]. The (1R) method generates a single rule for each attribute and ranks attributes according to the error rate associated with these rules.

• Halting criterion

Datasets are examined as follows. For model-free methods, we use a default number of attributes (10) to retain, as recommended by Minnaert et al. [58]. The best-first search in cfs and con terminates when a default number (5) is reached for non-improving consecutive nodes. Finally, (1R) requires no halting criteria as it generates a single rule per attribute and then performs the ranking.

• Result validation

The (1R) model-based method uses a 10-fold cross validation procedure. The inclusion of the no attribute selection method is applied for baseline comparisons.

4.4 Rule evaluation function

A rule evaluation function maps a rule r into a fitness value $Q^+(r)$ that quantifies the quality of r. The higher $Q^+(r)$ is, the better quality of r is. In the AntMiner$^+$ algorithm, pheromone is reinforced on the best ant's path proportionally to the associated quality of the resulting rule $Q^+(r)$.

Instances that belong to the current target class are referred to as positive instances. Those that belong to other classes are referred to as negative instances. Let the number of correctly classified positive instances be denoted TP, the number of positive instances incorrectly classified into negative FN. Similarly, the number of negative instances correctly classified as negative TN and those falsely classified into positive as FP. A tradeoff has to be established between TP and FP so that the coverage is maximized. Also, let D^+ denote the total number of positive instances remaining in the training dataset. Similarity, let D^- denote the total number of negative instances remaining in the training dataset. Let $D^+ + D^- \neq 0$.

- Klösgen (K) measure

  The Klösgen measure [68] balances the tradeoff between precision and coverage. The parameter ω controls the weight assigned to coverage Eq. (1), (TP +

$FP) \neq 0$.

$$Q_K^+ = (\frac{TP + FP}{D^+ + D^-})^\omega \times (\frac{TP}{TP + FP} - \frac{D^+}{D^+ + D^-}). \quad (1)$$

- m-estimate (M)
  Equation 2 defines the m-estimate measure with parameter m. Setting $m = 0$ leads to the precision.

$$Q_M^+ = \frac{TP + m \times \frac{D^+}{D^+ + D^-}}{TP + FP + m}, \ (TP + FP + m) \neq 0. \quad (2)$$

- F-measure (F)
  F-measure balances coverage and precision using the parameter β (Eq. (3)).

$$Q_F^+ = \frac{(\beta^2 + 1) \times \frac{TP}{TP+FP} \times \frac{TP}{D^+}}{\beta^2 \times \frac{TP}{TP+FP} + \frac{TP}{D^+}}, \ (TP + FP, D^+) \neq 0. \quad (3)$$

- The relative cost measure (RCM)
  The relative cost measure [69] balances the TP and FP rates through a parameter c (Eq. (4)). Setting $c = 1$ would reward TP, thus leading to a low precision and high coverage rules. On the other hand, if $c = 0$, then FP would be punished, thus leading to a high precision and low coverage rules.

$$Q_{RCM}^+ = c \times \frac{TP}{D^+} - (1 - c)\frac{FP}{D^-}. \quad (4)$$

- The sum of confidence and coverage (A^+)
  Confidence measures the fraction of remaining instances covered by r that are correctly classified. Coverage measures the fraction of remaining instances correctly covered and classified by r [35]. This is the rule evaluation function used in AntMiner$^+$ [8] (Eq. (5)).

$$Q_{A^+}^+ = \frac{TP}{TP + FP} + \frac{TP}{D^+ + D^-}. \quad (5)$$

- The product of sensitivity and specificity (SS)
  This measure has been used in Ref. [54]. It balances the sensitivity and specificity often used in the medical domain (Eq. (6)). A recent survey by Martens et al. [70] shows that most ACO-based data mining algorithms adopt this function.

$$Q_{SS}^+ = \frac{TP}{TP + FN} \times \frac{TN}{TN + FP}. \quad (6)$$

A large-scale empirical study of rule evaluation functions on multiple metaheuristic-based, sequential-covering algorithms was made by Minnaert et al. [58]. The default rule evaluation function for AntMiner$^+$ recommended in this study is the function K with parameter $\omega = 0.44$.

5 Experimental results

The implementation of the AntMiner$^+$ algorithm from the AntMiner$^+$ website[4] [58] is adopted with the same recommended settings as shown in Table 2.

Table 2 Default parameters for the AntMiner$^+$ algorithm

Parameter	Default value
Maximum iterations per rule (limit)	200
Fraction of nonmajority class data (stop)	0.01
Number of ant (ants)	1 000
evaporation factor (rho)	0.85
MAX-MIN AS parameter (p)	0.1
Sensitivity of the convergence check (epsilon)	0.05

The above-described implementation uses a reasonable set of methods from the open-source machine learning software Weka[5]. The methods consist of the implementation of different discretization and attribute selection algorithms, along with filters used for the removal of instances containing missing values and other miscellaneous methods for dataset partitioning and randomization. In addition, the AntMiner$^+$ implementation addresses a variety of rule evaluation measures, which are used to evaluate a rule within the rule list. These methods/measures are used for the tuning process of the preprocessing stage in this study and are described in Section 4.4.

As the cardinal essence of this research is the optimization of decision lists generated using AntMiner$^+$, we perform a tuning of the associated preprocessing configuration to get closer to the individual needs of each dataset included in the benchmark.

Regarding the order of these steps, it is worth noting that there exists no standard order for performing the preprocessing steps in Ref. [35]. The first step is chosen to be the removal of instances having missing value timing. This is because the AntMiner$^+$ fails to run in several datasets due to this timing, as will be shown in Section 5.2. As for the remaining steps, any procedure to carry on these steps other than the full permutation of all possibilities is not optimal. We choose to perform the tuning of these steps in the same order of that used for their processing in AntMiner$^+$ implementation. This allows the tuning for attribute selection to be done when numeric attributes are in the same form that will be used for rule

[4] http://www.antminerplus.com/
[5] http://www.cs.waikato.ac.nz/~ml/weka/

induction. Rule evaluation is not considered a preprocessing step but is in the core of the rule induction process. Its tuning should take place after the preprocessing tuning.

The stratified 10-time, 10-fold cross-validation procedure is used as it is considered the best error estimation strategy [71,72].

Further, to compare the performance of the different models examined on a solid basis, it is necessary to use statistical tests. In recent studies, the use of non-parametric statistical tests is highly recommended when dealing with evolutionary computation algorithms [73]. The Wilcoxon signed-ranks test [74] is used for pairwise model analysis. The Friedman test [75] is used for multiple comparison tests. Datasets having statistically significant difference among their different models are marked with an asterisk (*). According to these tests, the winner with a significance level $\alpha = 0.05$ is stressed in bold typeface. The first model in ranking is selected as (one of) the best learning models. If the aim is only to locate the best model, then the procedure is as follows. If the difference is found statistically significant, then this means that at least one of the models included in the comparison is significantly higher (or lower) than the rest. Nevertheless, the top-ranked model is selected in most cases as it represents the (or one of the) best performing models. In some cases, several models are equally good. That is, the difference among their performance is not considered statistically significant. In this case, other factors may be considered as will be explained in each case.

5.1 Benchmark of the experimentations

In this research, we use 25 medical datasets obtained from the UCI machine learning repository [76]. The list of medical datasets in the benchmark, together with the abbreviations used hereafter, are demonstrated as follows.

1) Cardiac arrhythmia (arr).
2) Breast cancer Wisconsin original wbcd (bcw).
3) Contraceptive method choice (cmc).
4) Dermatology (derma).
5) Echocardiogram (echo).
6) Ecoli (ecoli).
7) Haberman's survival (haber).
8) Heart disease Cleveland dataset (h_c).
9) Heart disease Hungarian dataset (h_h).
10) Statlog heart (h_stat).
11) Heart disease Swiss dataset (h_swiss).
12) Hepatitis (hep).
13) Horse colic (horse).
14) Thyroid disease hypothyroid dataset (hypo).
15) Liver disorders (liver).
16) Breast cancer (ljub).
17) Lymphography (lymph).
18) Mammographic mass (mammo).
19) Thyroid disease new thyroid dataset (new_thy).
20) Parkinson's disease (park).
21) Pima Indian diabetes (pima).
22) Primary tumor (p_tumor).
23) Thyroid disease sick dataset (sick).
24) Breast cancer Wisconsin diagnostic (wdbc).
25) Breast cancer Wisconsin prognostic (wpbc).

The benchmark used hosts a wide variety of the characteristics listed above. A summary of the main characteristics is presented in Table 3. For each dataset, the number of instances (#Inst.), number of attributes (#Attr.) including numeric (#Num.) and nominal (#Nom.) attributes, and number of classes (#Class.) are listed. Also included is the percentage of overall missing values (%MV) computed as $(\frac{\#\text{missing values}}{\#\text{Inst.} \times \#\text{Attr.}} \times 100)$ and the percentage of instances with missing values (%Inst.MV) computed as $(\frac{\#\text{inst. with missing values}}{\#\text{Inst.}} \times 100)$.

As for class imbalance, and as far as we are concerned, there is no consensus in the literature on a quantitative measure for describing a dataset as imbalanced. The majority of studies either limit the domain to binary classification problems, or compute the imbalance ratio as the cardinality of the minority class over the total number of instances, ignoring other classes that may be present in the dataset (for example, Refs. [77,78]). The last two columns in Table 3 report the class noise (Noise) and imbalance ratio (Imb.Ratio) as reported in Ref. [9]. The imbalance ratio recommended by in this study accounts for all classes in the dataset. For those datasets that were not reported in Ref. [9], a dash (—) is placed.

5.2 Timing of removing instances having missing values

Referring to Table 3, among the 25 datasets in the benchmark, 15 datasets contain missing values. To test the hypothesis, we conduct the following experiment. We modify AntMiner$^+$ such that the removal of instances having missing values is delayed after the attribute selection step. We call this version

Table 3 Summary of medical dataset characteristics

No.	Dataset	#Inst.	#Attr.	#Num.	#Nom.	#Class.	%MV	%Inst. MV	Noise	Imb. Ratio
1	arr	452	279	206	73	16	0.32	84.96	11.28	1.57
2	bcw*[a]	699	9	9	0	2	0.25	2.29	2.72	1.21
3	cmc	1 473	9	2	7	3	0	0	31.98	1.04
4	derma	366	34	1	33	6	0.06	2.19	0.82	1.05
5	echo	132	10	8	2	2	7.37	45.26	6.06	1.24
6	ecoli	336	7	7	0	8	0	0	6.55	1.25
7	haber	306	3	2	1	2	0	0	16.67	1.57
8	h_c	303	13	6	7	5	0.18	2.31	17.82	1.37
9	h_h	294	13	6	7	5	27.94	99.7	13.61	1.74
10	h_stat	270	13	7	6	2	0	0	15.19	1.03
11	h_swiss	123	13	6	7	5	17.07	100	32.52	1.14
12	hep	155	19	6	13	2	5.67	48.39	10.97	2.05
13	horse	368	22	7	15	2	23.8	98.9	11.96	1.15
14	hypo*[b]	3 772	29	7	22	4	5.41	100	0.54	9.99
15	liver	345	6	6	0	2	0	0	9.86	1.05
16	ljub	286	9	0	9	2	0.35	3.15	—	2.79
17	lymph	148	18	0	18	4	0	0	10.81	1.46
18	mammo	961	4	1	3	2	30.77	13.53	14.15	1.01
19	new_thy	215	5	5	0	3	0	0	2.79	1.78
20	park	195	22	22	0	2	0	0	—	3.39
21	pima	768	8	8	0	2	0	0	20.18	1.20
22	p_tumor	339	17	0	17	22	3.9	61.06	—	0.90
23	sick*[b]	3 772	29	7	22	2	5.41	100	0.71	7.72
24	wdbc	569	30	30	0	2	0	0	2.11	1.14
25	wpbc	198	33	33	0	2	0.06	2.02	13.64	1.76

Note: *[a]means in this study, attributes for the bcw dataset are encoded as nominal; *[b]means different number of instances stated by Ref. [9]

AntMiner$^+$ with remove missing last (AM^+RML). The aim of this experiment is to decide which version of AntMiner$^+$ ($AM^+Original$ or AM^+RML) is more suitable to each dataset considered. Results are compared and conclusions are based on the following strategy. If $AM^+Original$ has failed, or there is not a notable difference in the average predictive accuracy, then AM^+RML is used.

• Results

The results of this experiment are summarized in Fig. 1 and Table 4. For each AntMiner$^+$ model, only the predictive accuracy is shown in the comparison (Acc. ± STD). In the first observation from Fig. 1, we note that the $AM^+Original$ model fails in five datasets. The reason for failing is that there are not enough instances to generate any folds. In all of these datasets, we note that the percentage of instances containing missing values is very high (98.90%–100.00%). The AM^+RML model produced output for all five datasets. However, for those with a relatively low number of attributes (h_h and h_swiss), the results are considerably poor and produced empty rules in several folds. As for the remaining three datasets (horse, hypo, and sick), results are much better. The large number of associated attributes (22–29) has helped in decreasing the percentage of instances with missing values in the remaining attributes post the attribute selection phase.

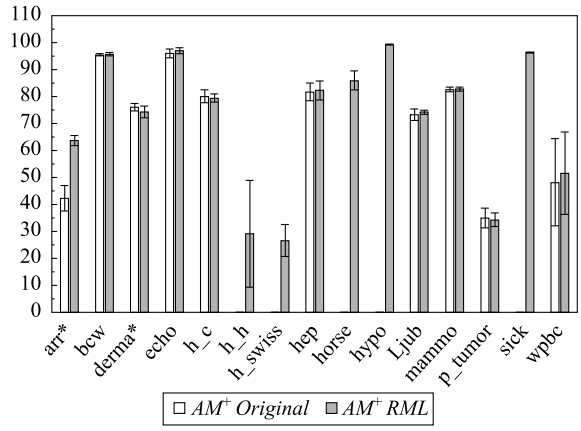

Fig. 1 Average predictive accuracy for the timing of removal of instances having missing values experiment

For the remaining 10 datasets, we note that the AM^+RML model achieved a significant win over $AM^+Original$ in the arr dataset. This is despite the fact that the class noise associated with this dataset is not low (11.28%). The $AM^+Original$ obtains a significant win in one dataset: derma. For this par-

ticular dataset, the percentage of class noise is not significant. It is expected that the attribute noise is high, which only adds more noise when adding more instances for this dataset, thus deteriorating classification performance. For the rest of the datasets, the difference in performance among the $AM^{+}Original$ and the $AM^{+}RML$ models is not considered statistically significant. Table 4 reports the average accuracy (Acc), number of rules (Rules), number of terms per rule (T/R), and rule-set induction time (Time) over all ten datasets for which the $AM^{+}Original$ has produced output. From this table it can be seen that overall, the $AM^{+}RML$ model has higher predictive accuracy than the $AM^{+}Original$.

Table 4 Averages for the timing of removal of instances with missing values

AntMiner^{+} version	Acc	Rules	T/R	Time/s
$AM^{+}Original$	70.99 ± 3.67	3.94 ± 0.88	2.91 ± 0.37	19.63 ± 4.26
$AM^{+}RML$	73.48 ± 3.02	4.10 ± 1.17	2.86 ± 0.34	18.85 ± 6.02

5.3 Discretization method

The AntMiner^{+} algorithm cannot directly handle numeric attributes. Among the benchmark datasets, 21 contain numeric attributes. Discretization is an essential step to transform these numeric attributes into a form that the AntMiner^{+} algorithm can handle. The numeric attributes can now be handled by AntMiner^{+} during rule induction as ordinal attributes. The Weka implementation of the four discretization methods is used. This results in ten models as will be shortly described. The best performing model for each dataset will be outlined. The default discretization method in the implementation adopted is fay. For binning discretization methods, the default number of bins is 10. Only datasets having continuous attributes (Table 5) are included in this experiment. The column AntMiner^{+} version describes the version used (Original or RML), as concluded in Section 5.2. The Friedman test is used to test whether the difference among the predictive accuracy of the selected models is considered statistically significant. The discretization method associated with the best rank is usually chosen. In few cases, a model other than the best ranked is selected as justified. Reasoning will be usually based on the rule set size and/or the computational time associated with these models.

• Results

The predictive accuracy with the associated standard deviation obtained by AntMiner^{+} in combination with each of the used discretization methods is shown in Fig. 2. The discretization method selected for each dataset is shown in Table 9.

Table 5 Datasets included in discretization method experiment

No.	Dataset	AntMiner^{+} version	No.	Dataset	AntMiner^{+} version
1	arr	$AM^{+}RML$	12	horse	$AM^{+}RML$
2	cmc	$AM^{+}Original$	13	hypo	$AM^{+}RML$
3	derma	$AM^{+}Original$	14	liver	$AM^{+}Original$
4	echo	$AM^{+}RML$	15	mammo	$AM^{+}RML$
5	ecoli	$AM^{+}Original$	16	new_thy	$AM^{+}Original$
6	haber	$AM^{+}Original$	17	park	$AM^{+}Original$
7	h_c	$AM^{+}RML$	18	pima	$AM^{+}Original$
8	h_h	$AM^{+}RML$	19	sick	$AM^{+}RML$
9	h_stat	$AM^{+}Original$	20	wdbc	$AM^{+}Original$
10	h_swiss	$AM^{+}RML$	21	wpbc	$AM^{+}RML$
11	hep	$AM^{+}RML$			

In addition, the predictive accuracy, number of rules, number of terms per rule, and computational time per rule set are averaged over all datasets for each discretization method and shown in Figs. 3 and 4. The grand average of all ten discretization methods over the 21 datasets is also displayed.

From Fig. 2, it can be seen that even for the same learner (AntMiner^{+}), the performance across different datasets differs according to the discretization method used. Among the 21 datasets employed in this experiment, the difference in AntMiner^{+} performance associated with the ten models for each dataset and resulting from the use of different discretization methods is found to be statistically significant in 12 datasets. In particular, the difference is quite large in three datasets. Namely, the following is noted.

1) In the h_h dataset, the relative improvement obtained by kon over fay exceeds 176% ($= \frac{80.17-28.96}{28.96} \times 100\%$).
2) In the liver_disorder dataset, efb10 improves over the default discretization method fay for more than 41% in predictive accuracy.
3) The improvement obtained when using eib5 over the default fay is over 54% for the wpbc dataset as well.

In the remaining nine datasets, the difference among the ten models for each dataset is not found to be statistically significant. This result is not surprising for the derma dataset. This dataset only has one numeric attribute against 32 nominal attributes. However, for two datasets, namely new_thy and wdbc, all the predictive attributes are numeric.

Among the ten models generated, the best rank is obtained by fay, followed by efb10 the highest number of times ($5:21$ and $4:21$, respectively). If the four eib models were aggregated (eib5, eib10, eib15, and eib20), then eib would score the best rank in ($9:21$) times followed by efb at ($7:21$). The highest number of times a model is selected belongs to

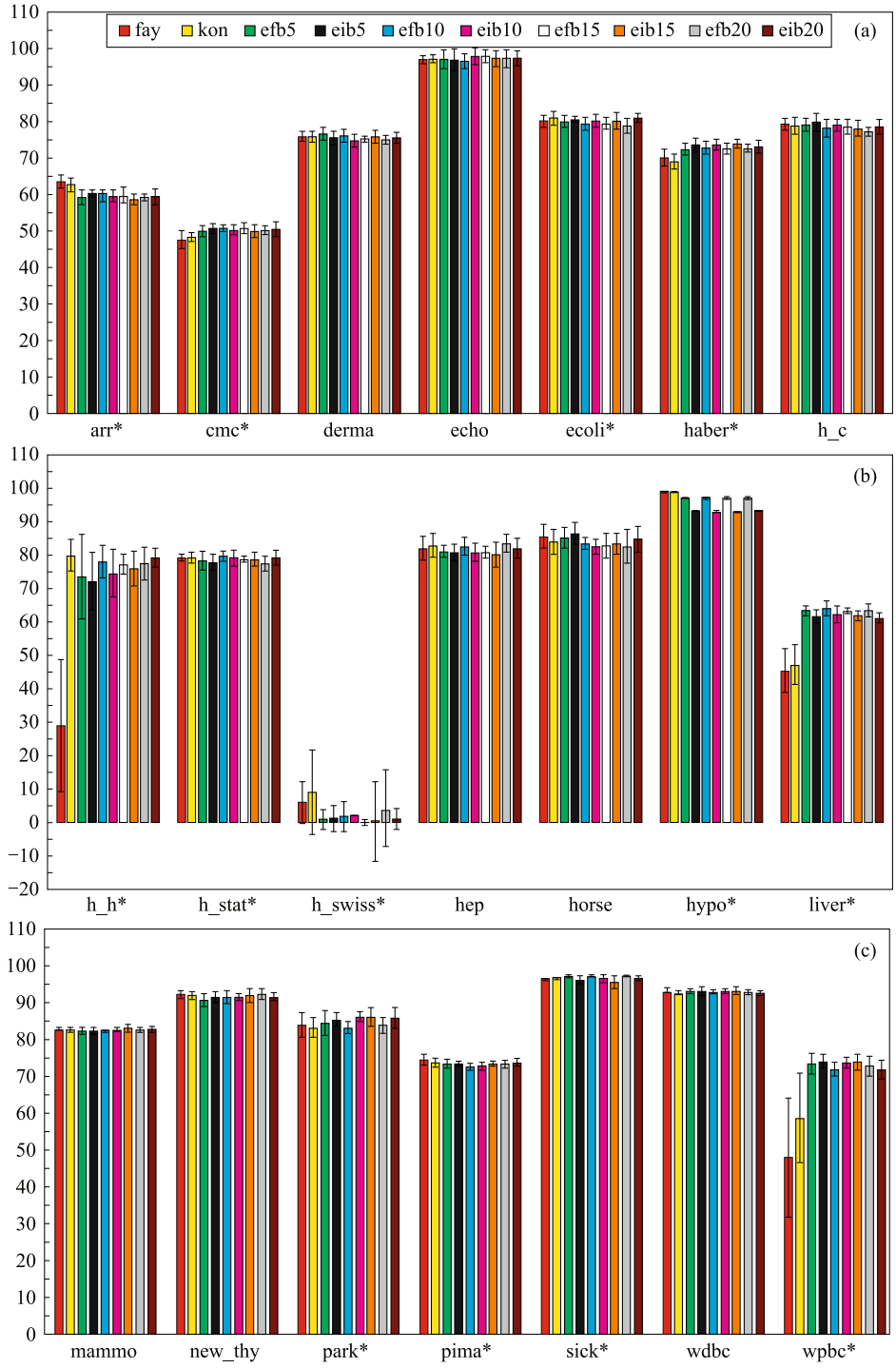

Fig. 2 Results summary of discretization tuning

fay and efb10 (4 : 21).

From Figs. 3 and 4, the following can be noticed.

1) Discretization methods that use binning obtain a higher overall predictive accuracy average over entropy-based methods. These models are also more robust as they have lower overall average standard deviation. The highest average accuracy is obtained by the efb10 method.
2) The difference among the model sizes obtained is not significant for the ten models. However, the use of entropy-based discretization methods (fay and kon) results in relatively smaller model sizes. The number of bins does not seem to significantly affect the size of the resulting model.
3) The shortest computational time belongs to models

using entropy-based discretization methods. Models based on binning discretization methods require almost double the time.

5.4 Feature subset selection method

A diverse combination of FSS methods is included in the comparison. For each dataset, we compare the AntMiner+ rule induction performance that uses each of the different feature subset selection methods listed in this section. We also include the AntMiner+ performance when no attribute selection is used in the preprocessing phase. This is done for all datasets in the benchmark except the (arr) dataset, where the number of attributes is relatively high. Weka Java implementation with default settings for attribute selection methods is used. A list of the included methods is presented along with the corresponding synonym used in tables.

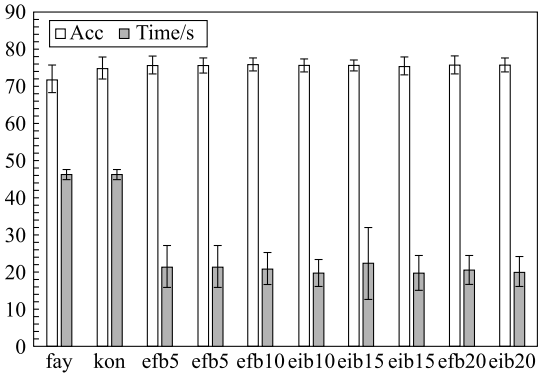

Fig. 3 Average predictive accuracy and time/s over all datasets for AntMiner+ per discretization method

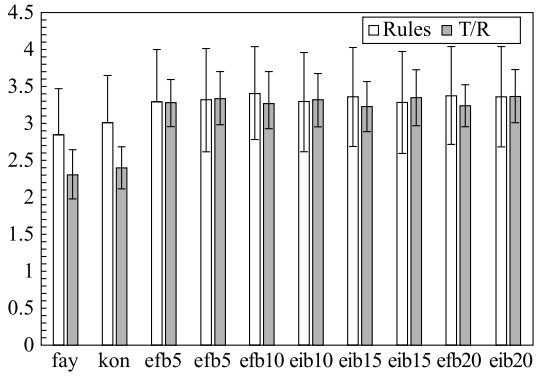

Fig. 4 Average model size over all datasets for AntMiner+ per discretization method

The default feature subset selection method in the implementation [58] adopted is rel with ten as the default number of attributes to retain. Therefore, only datasets having more than ten attributes are included in this experiment (Table 6). The column AntMiner+ version describes the version used (Original or RML) as concluded in Section 5.2. The column discretization method shows the discretization method as per Section 5.3. The non-parametric Friedman test is used to test whether the difference among the predictive accuracy of the selected models is considered statistically significant. The statistical comparison is only done among results associated with the eight FSS methods. The model where no attribute selection is employed (AS0) is not included in the statistical comparisons. The FSS method associated with the best rank is usually chosen. In few cases, a model other than the best ranked is selected as justified. Reasoning is usually based on the rule set size and/or the computational time associated with these models.

Table 6 Datasets included in FSS experiment

No.	Dataset	#Attr.	AntMiner+ version	Discretization method
1	arr	278	$AM^{+}RML$	fay
2	derma	33	$AM^{+}Original$	efb5
3	h_c	13	$AM^{+}RML$	eib5
4	h_h	13	$AM^{+}RML$	kon
5	h_stat	13	$AM^{+}Original$	efb10
6	h_swiss	13	$AM^{+}RML$	kon
7	hep	19	$AM^{+}RML$	efb20
8	horse	22	$AM^{+}RML$	eib5
9	hypo	29	$AM^{+}RML$	fay
10	lymph	18	$AM^{+}Original$	—
11	park	22	$AM^{+}Original$	eib10
12	p_tumor	17	$AM^{+}RML$	—
13	sick	29	$AM^{+}RML$	efb10
14	wdbc	32	$AM^{+}Original$	efb5
15	wpbc	33	$AM^{+}RML$	eib5

• Results

The predictive accuracy with the associated standard deviation obtained by AntMiner+, in combination with each of the used FSS methods, is shown in Fig. 5. The FSS method selected for each dataset is shown in Table 9.

In addition, the predictive accuracy, number of rules, number of terms per rule, and computational time per rule set are averaged over all datasets for each FSS method and shown in Figs. 6 and 7. The grand average of all the eight FSS methods over the 15 datasets is also reported. Table 7 shows the (AS0) case, where no attribute selection is employed. Averages are limited over the ten datasets where AntMiner+ produced a non-zero output. The corresponding average for all the FSS methods over the same datasets is also shown.

Table 7 Averages over all datasets for AntMiner+ for FSS vs. all features

FSS?	Acc	Rules	T/R	Time/s
All FSS	75.97 ± 2.50	4.39 ± 1.01	3.22 ± 0.29	21.89 ± 5.48
0AS	73.79 ± 2.82	4.99 ± 1.04	3.89± 0.50	57.13 ± 9.61

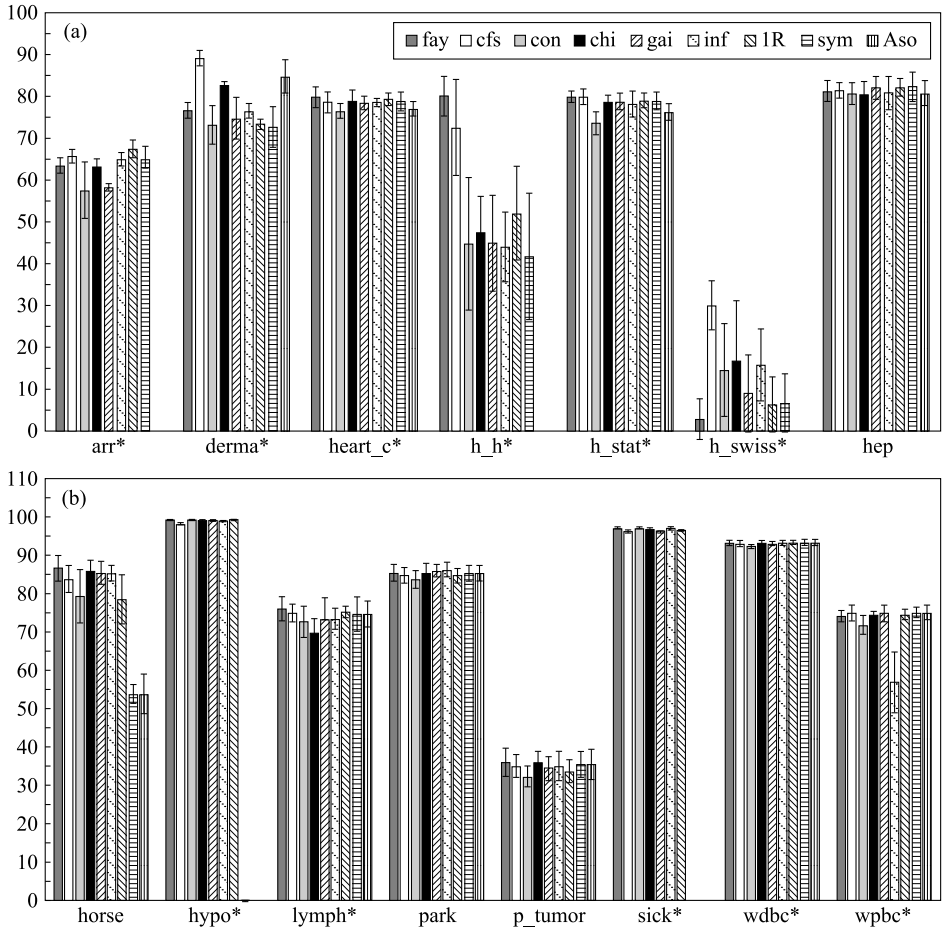

Fig. 5 FSS tuning experiment results summary

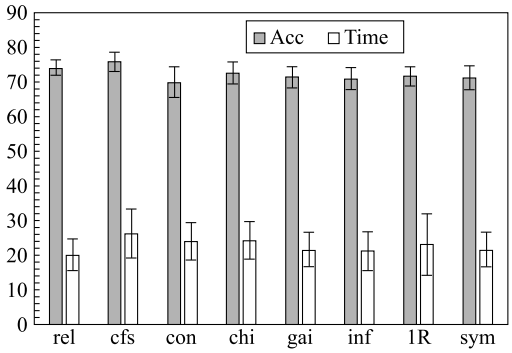

Fig. 6 Average predictive accuracy and time/s over all datasets for AntMiner$^+$ per FSS method

Figure 5 shows that three FSS methods equally score the best rank for the hep dataset: gai, 1R, and sym. The sym FSS method is chosen as it features the highest average and median among the three FSS methods.

From Fig. 5 and Table 7, several observations can be drawn. The first observation is noted when comparing rule induction combined with FSS with that of full attributes (AS0). The experiment confirms the benefit of FSS as a preprocessing step in this case. Without FSS, the rule induction process fails entirely in some datasets (h_h, h_swiss, hypo, and sick). For all these datasets, the percentage of instances having missing values is very large ($\geqslant 99.7\%$). Therefore, when the next step of preprocessing (removal of instances with missing values) is performed, no instances are left for induction. Effectively, FSS significantly reduces the percentage of instances having missing values and is thus fundamental in this case. The same reasoning is related to the inferior performance (predictive accuracy) obtained in similar datasets

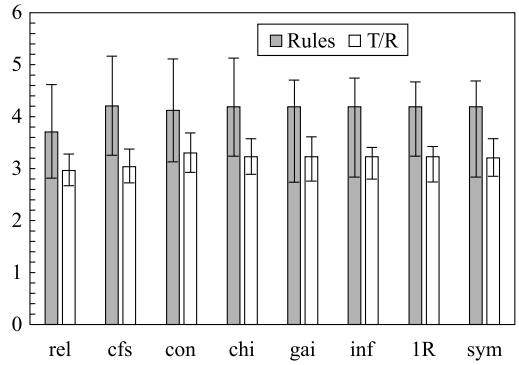

Fig. 7 Average model size over all datasets for AntMiner$^+$ per FSS method

(e.g., horse dataset).

The second confirmed advantage is the acceleration of search when using FSS methods in general. It is noticed that using FSS methods reduces the computational time. For example, this reduction is up to six times in the derma dataset. In general, the computational time of AntMiner$^+$ without FSS is more than twice as much as that obtained by averaging the computational time of AntMiner$^+$ combined with each of the eight FSS methods. Also, note that the solution size is larger and the accuracy is on average lower.

When comparing results using the different FSS methods, we note that out of the 15 datasets, the difference among AntMiner$^+$ results, when combined with each of the eight FSS methods, is considered statistically significant in 11 datasets. The high imbalance ratio in some (e.g., hypo and sick) seems not to affect the results. Among these, the difference is extremely significant in h_h, h_swiss, and wpbc datasets. For example, it can be seen that changing the FSS method used with AntMiner$^+$ for the h_h dataset can improve the predictive accuracy from (41.82%) when using sym to (80.17%) when using rel, thus effectively providing over 91% improvement in accuracy. These three datasets (h_h, h_swiss, and wpbc) exhibit the highest level of class noise combined with highest percentage of instances having missing values, and small number of instances (below 300). Most FSS methods showed to be the preferred for at least one dataset, however, the methods rel followed by cfs obtained the largest count of best ranks. When considering grand averages, Fig. 6 shows that the highest overall average is associated with the correlation-based FSS method (cfs). It also features the highest computational time. The follow-up is ReliefF method (rel). In all models, comparable rule set sizes are found, as depicted in Fig. 7.

5.5 Rule evaluation function

This experiment involves all 25 datasets in the benchmark. AntMiner$^+$ is run on each dataset once for each of the six rule evaluation functions K, M, F, RCM, A+, and SS. The default parameter values recommended in the study by Minnaert et al. [58] for K, M, F, and RCM are used and are shown in Table 8. For each run, the settings concluded for each dataset in the previous sections regarding the timing of the removal of missing instances, discretization algorithm, and attribute selection method are used. The non-parametric Friedman test is used to test whether the difference among the predictive accuracy of the six models is considered statistically significant. The rule evaluation function associated with the best rank is usually chosen.

Table 8 Default parameters for rule evaluation functions

Rule evaluation function	Parameter	Default value
K	ω	0.44
M	m	0.28
F	β	7
RCM	c	0.028

• Results

The predictive accuracy with the associated standard deviation obtained by AntMiner$^+$ in combination with each of the used rule evaluation functions is shown in Fig. 8. The figure also displays the rule evaluation function selected for each dataset.

Additionally, the predictive accuracy, computational time

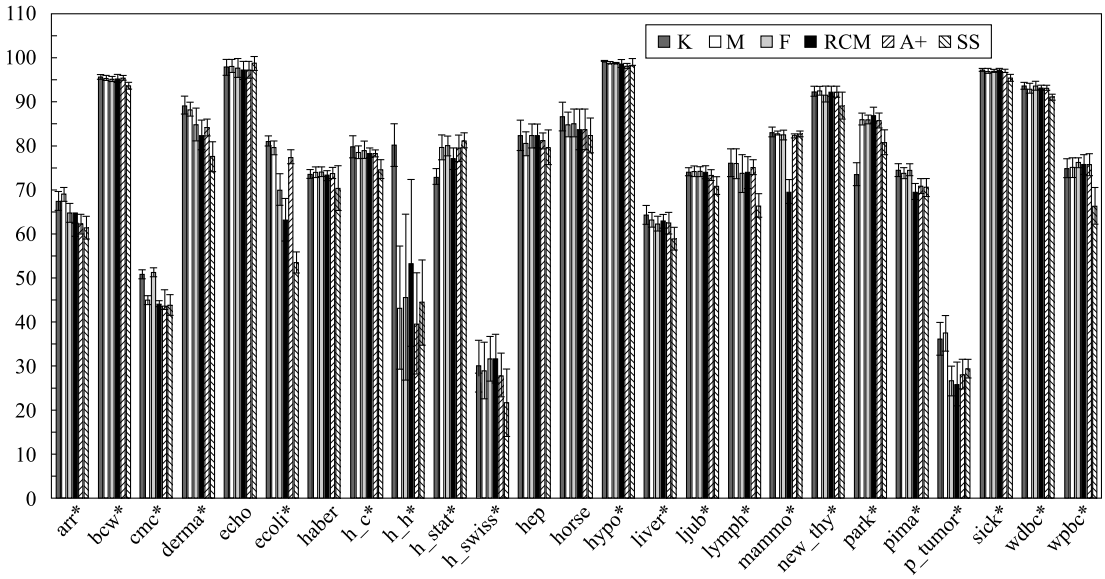

Fig. 8 Rule evaluation function tuning results summary

per rule set, number of rules, and number of terms per rule are averaged over all datasets for each discretization method and are shown in Figs. 9 and 10. The grand average of all ten discretization methods over the 25 datasets is also displayed. The policy on ties needs to be clarified. In the derma dataset, a tie is encountered between K and M. However, K is selected as it produces the highest accuracy average and smaller model size.

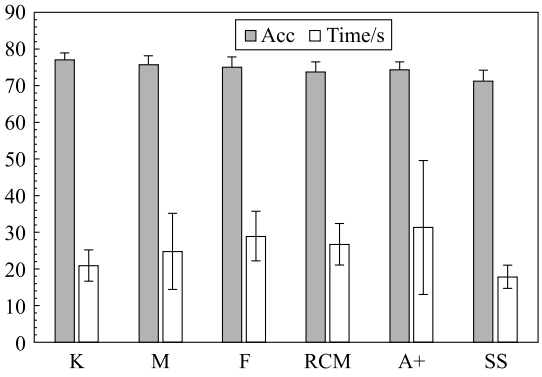

Fig. 9 Average predictive accuracy and time/s over all datasets for AntMiner+ per rule evaluation function

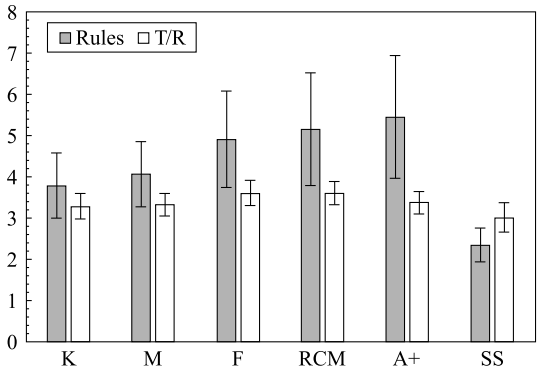

Fig. 10 Average model size over all datasets for AntMiner+ per rule evaluation function

The difference in predictive accuracy for each dataset among the six measures is found to be statistically significant in 21 out of 25 datasets. It is found that models associated with the K function obtained the best rank in Friedman test in 12 out of 25 datasets followed by M (5 : 25). From Figs. 9 and 10, it can be seen that when averaged over all datasets, except for the models associated with the SS rule evaluation measure, those associated with the K function obtain the highest predictive accuracy, shortest solution size, and require the least computational time.

5.6 Performance of the tuned AntMiner+

By the end of the tuning phase with its four steps, it is time to compare the results before and after tuning. Figure 11 shows the predictive accuracy along with the standard deviation for the original AntMiner+ with the default settings versus those for the tuned version. The final model for AntMiner+ after tuning is referred to as *AM+Tuned* hereafter. Table 9 summarizes those findings.

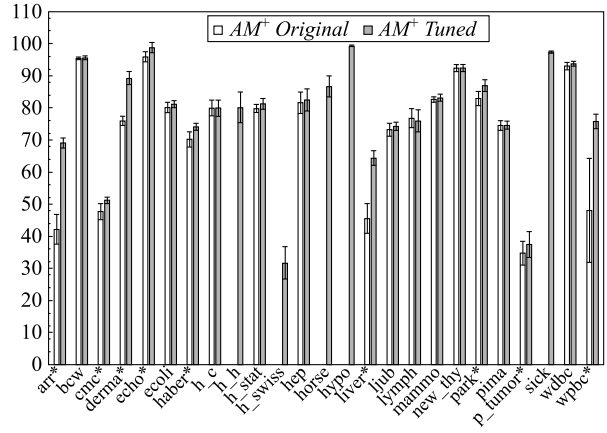

Fig. 11 Average predictive accuracy for AntMiner+ showing the difference before and after the tuning phase

Table 9 AntMiner+ tuning phase setting recommendations

No.	Dataset	AntMiner+ version	FSS method	Discretization method	Rule eval. measure
1	arr	*AM+RML*	1R	fay	M
2	bcw	*AM+RML*	—	—	K
3	cmc	*AM+Original*	—	efb10	F
4	derma	*AM+Original*	cfs	efb5	K
5	echo	*AM+RML*	—	eib10	SS
6	ecoli	*AM+Original*	—	eib20	K
7	haber	*AM+Original*	—	eib15	F
8	h_c	*AM+RML*	rel	eib5	K
9	h_h	*AM+RML*	rel	kon	K
10	h_stat	*AM+Original*	cfs	efb10	SS
11	h_swiss	*AM+RML*	cfs	kon	F
12	hep	*AM+RML*	sym	efb20	K
13	horse	*AM+RML*	eib5	rel	K
14	hypo	*AM+RML*	1R	fay	K
15	liver	*AM+Original*	—	efb10	K
16	ljub	*AM+RML*	—	—	M
17	lymph	*AM+Original*	rel	—	M
18	mammo	*AM+RML*	—	eib15	K
19	new_thy	*AM+Original*	—	fay	K
20	park	*AM+Original*	inf	eib10	RCM
21	pima	*AM+Original*	—	fay	F
22	p_tumor	*AM+RML*	rel	—	M
23	sick	*AM+RML*	rel	efb10	RCM
24	wdbc	*AM+Original*	sym	efb5	K
25	wpbc	*AM+RML*	gai	eib5	RCM

5.7 Global comparisons

As in any field where the model obtained will be used to sup-

port a decision used by a human, in the medical field, the acceptance of machine learning models presented for the purpose of medical diagnosis or prognosis is highly dependent on its ability to be interpreted and validated [79]. Otherwise, the user will not trust the model presented and even worst, it might lead to wrong decisions. Artificial neural networks and deep learning techniques are known for their competitive predictive accuracy. However, these techniques generate black-box models, in the form of complex mathematical functions that are difficult for humans to comprehend. Similarly, statistical learning algorithms including Bayes classifiers are also sub-symbolic classification methods that require some background knowledge about the prior probabilities, and thus are not appealing. Support vector machines [80,81] are among the newest and strongest classification techniques. However, the support vectors cannot easily communicate the obtained knowledge to medical experts. Case-based methods and nearest-neighbor techniques merely store training instances and do not create a classification model. Instances are matched to the closest one stored and classification is based on the closest match(es). Thus, this paradigm is also not suitable for the requirements in hand.

To compare the results of $AM^{+}Tuned$, a selection of competitive classification algorithms is included. PART [66] is a rule-based classification algorithm. PART extracts rules from decision trees created by the J48 algorithm [28]. In a large comparative study on thirty-three classification algorithms, Lim et al. [82] showed that the C4.5 algorithm had a good speed/accuracy performance. The implementation on the Weka benchmark is used for non-evolutionary learners with default settings. Since PART and J48 are deterministic algorithms, a single 10-fold cross-validation procedure is used for evaluation. Evolutionary learners include SLAVE [83] and UCS [84]. SLAVE is a genetic iterative rule learning system based on fuzzy logic theory. UCS is a Michigan-style learning classifier system derived from XCS [85] and specialized for supervised learning tasks. In a comparative study of genetic-based learning classifier systems [86], UCS stands out as a robust classification algorithm. A 10-time, 10-fold cross-validation procedure is used for evaluation. For UCS and SLAVE, the open-source software tool KEEL [7] is used with default settings.

Figure 12 depicts the predictive accuracy obtained with each of the algorithms when applied to the benchmark datasets. The highest overall average belongs to J48 (77.93%), closely followed by $AM^{+}Tuned$ (77.89%). The lowest is scored by Slave (68.05%). The Friedman non-parametric test shows that the difference in predictive accuracy among the five models ($AM^{+}Tuned$, Part, J48, UCS, and SLAVE) over the 25 datasets is considered statistically significant (p-value = 0.002). The best rank was scored by $AM^{+}Tuned$. However, the Holm post-hoc analysis did not detect a difference among the four models ($AM^{+}Tuned$, Part, J48, and UCS). Thus, the $AM^{+}Tuned$ algorithm is considered

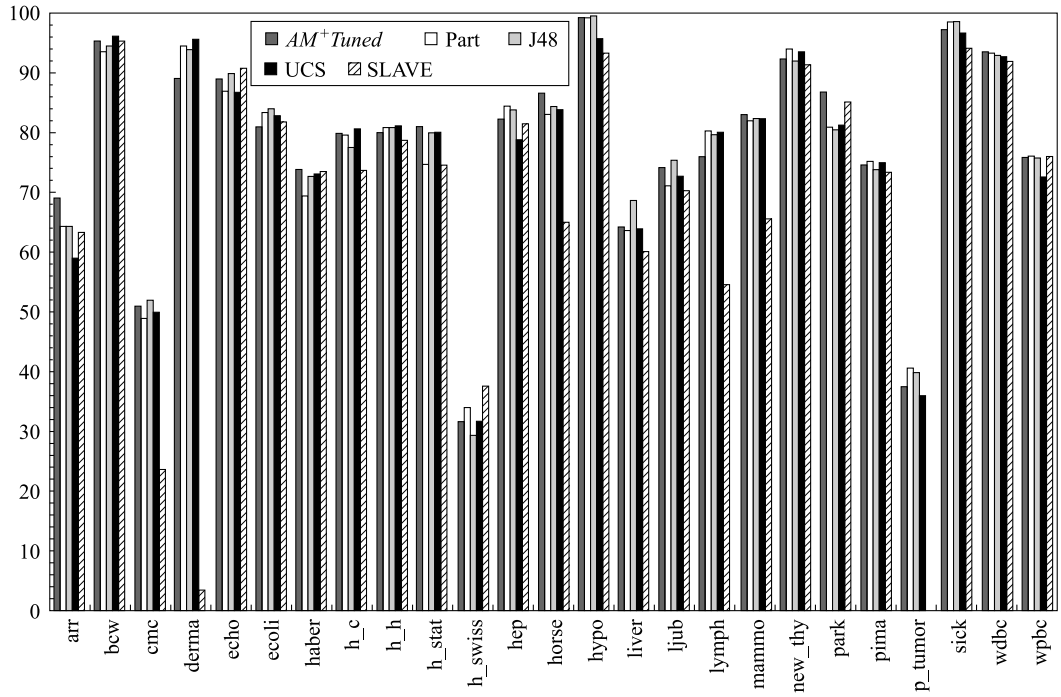

Fig. 12 Average predictive accuracy for $AM^{+}Tuned$, Part, J48, UCS, and SLAVE

comparable to state-of-the-art classification algorithms.

6 Discussion

Several medical datasets are associated with a considerable percentage of missing values. However, when the intension is to mine knowledge and extract conclusions, one must avoid drawing unwarranted conclusions as much as possible; for example, flagging imputed values when suitable and evaluating their significance [87]. The safe handling of missing values is not a trivial task. In fact, the selection of the most appropriate missing data handling method is a hard and complex task [39]. This study adopts the removal of missing values in medical datasets and investigates the best timing to apply it. The removal of instances having missing values results in information loss. However, if this step is delayed after FSS, then the percentage of information loss is considerably decreased, thus allowing more instances for the training and testing process. This conclusion particularly holds for datasets having a larger number of predictor features. The study also finds that despite the noise normally associated with medical datasets, providing more instances to the learning algorithm improves the classification results. In some cases, the handling of instances with missing values may explain the difference between an algorithm that completely fails to run and another that produces good results.

Discretization enhances model comprehensibility and excels the search. The selection of a discretization method depends on the problem tackled and the learner used. It is important to find a balance between the number of intervals generated and the performance obtained, as the search space grows exponentially with the number of intervals. The choice of the discretization method has a profound effect on the model performance. It is not possible to identify a single discretization method as the best over all medical datasets. Among the discretization methods included in this experiment, the use of entropy-based discretization methods has a computational cost advantage in terms of model complexity and computational time. Discretization methods based on binning obtain overall higher averages in predictive accuracy and lower variance than those based on entropy. The best way is to experiment with different discretization methods and select the one producing the best balance of performance versus cost defined as computational time and model complexity.

Like removing instances having missing values, FSS results in a loss of information as entire attributes are eliminated. However, the aim is to remove irrelevant, redundant, or noisy features. FSS not only reduces storage and computational complexity but also enhances model comprehensibility and classification accuracy, particularly in small sample size datasets [29]. Finding the optimal m feature subset from all possible $\binom{n}{m}$ feature subsets has been shown to be an NP-hard combinatorial optimization problem [88]. In this study, it is found that for datasets having more than 10 attributes, using FSS methods always proves fruitful in comparison to induction without FSS. No FSS method performs the best over all datasets, and this is an expected result. However, when averaged over all datasets, induction using the cfs FSS method scores the highest predictive accuracy with no payoff in rule set size.

When considering different features of the datasets, including number of instances, number and type of attributes, imbalance ratio, noise, and missing values, the impact of class noise, number of instances, and missing values on FSS are found to be high. Missing values especially have a strong effect on induction using FSS. From this experiment, and among the FSS methods considered, the preferred FSS method with each medical dataset has been highlighted.

As for the rule evaluation function, the results obtained in this experiment confirm those found by Minnaert et al. [58] that, overall, the K function is the best among the previous six functions for the AntMiner$^+$ algorithm.

The tuning process described here is not meant to be exhaustive due to time limitations, and thus, we do not claim that the declared results are the best settings for each dataset. Even within a single dataset, there may be different requirements related to different attributes. For example, each numeric attribute, within the same dataset, may have a different optimal discretization method. In this regard, Bacardit et al. [89] include the selection of the best discretization method for each attribute within the genetic algorithm (GA) cycle. However, these steps are intended to improve the initial configuration and tailor it, to some extent, for each dataset in the described benchmark.

When evaluating the performance of AM^+Tuned, a closer look at Fig. 11 shows that no significant difference is encountered in 11 out of the 25 datasets. One interesting result is that there is no improvement obtained at all during the tuning process for the new_thy dataset. By investigating the dataset characteristics, we note that it has no missing values, and no attribute selection is needed as it contains only five attributes. The tuning step concludes that the fay discretization method and K rule evaluation function are found to be the best suited. These are the same settings in the original AntMiner$^+$ implementation, and that explains the situation.

In five datasets out of the remaining fourteen datasets (h_h,

h_swiss, horse, hypo, and sick), the difference is of success/failure in obtaining an output. The difference is statistically significant in the nine remaining datasets (arr, cmc, derma, echo, haber, liver, park, p_tumor, and wpbc). Thus, in the majority of datasets, there is a significant improvement achieved via the tuning process. The grand average over the 20 datasets in the benchmark, where the output is obtained by AM^{+} *Original*, shows an overall significant improvement obtained through the tuning process, as confirmed by the Wilcoxon test that is applied to compare the two models (AM^{+} *Original* [72.65 ± 2.89] and $AM^{+}Tuned$ [78.08 ± 1.74]) for the same 20 datasets. The resulting model is robust and comparable to state-of-the-art classification algorithms.

Although this study specifically addresses medical datasets, the recommended preprocessing procedure can be applied to arbitrary datasets with similar features. First, the existence of missing values is addressed. If the dataset contains missing values, then the timing of removing instances with missing values should be examined, whether it is done before or after the FSS step. Next, the discretization process is examined. The discretization procedure applies to datasets with numeric attributes, especially if the selected classification algorithm cannot deal directly with numeric attributes. If this is the case, then a number of discretization methods should be investigated, and the associated classification results for the resulting models compared. Once the best model is chosen according to measures of concern such as predictive accuracy or model complexity, FSS step is considered. This step is particularly recommended for datasets with a small number of instances but a large number of features. Similar to the discretization step, a number of feature subset selection methods are to be explored and the resulting classification models compared.

7 Conclusion

Data preprocessing has a profound effect on the performance of the learner. Each dataset is different, and there is no preprocessing method that is best across all datasets. Deciding the best combination of preprocessing methods for a specific dataset is not possible without trial and comparisons. Technology is advancing rapidly. The advent of various open-source libraries, like Weka and KEEL, hosting an extensive set of off-the-shelf preprocessing methods, combined with the leisure of standard formats like the ARFF, and advances in computer hardware technology, persuades the integration of automatic tuning for preprocessing operations into the data mining task and for each dataset on an individual basis. The idea is suitable for off-line applications.

This work shows the results of the tuning for the preprocessing stage, which is applied to AntMiner^{+} as an illustrative example. For each dataset, the timing of removing instances with missing values was examined. Experimentations are done with different feature subset selection and discretization methods for each dataset. The library used includes a variety of common discretization and attribute selection methods, from open-source, on-line libraries. Also, the use of several rule evaluation functions inside the AntMiner^{+} algorithm is investigated. As with any tuning process, the proposed method is time consuming. However, the disposal of open-source libraries associated with rudimentary standard formats facilitates a convenient custom preprocessing stage tuning. Experiments show that there is a significant improvement in classification performance measured by predictive accuracy and obtained in the majority of datasets in the benchmark through the individualized tuning of the preprocessing operations. Moreover, given a certain classification algorithm, the design of the preprocessing stage can make the difference between complete failure and the achievement of results that are competitive to rival classification algorithms in the same datasets.

Several datasets in the benchmark are associated with a large percentage of missing values. Removing instances with missing values is definitely not the best choice for handling such instances, particularly if performed before attribute selection. Despite the noise associated with medical data, the experiments show that providing more instances for training the rule induction algorithm improves the rule induction experience. Imputation techniques that are based on machine learning methods like ANN and SVM require significant computational time. The mechanism by which data are missing must be first identified and the method for handling the missing values accordingly decided. The use of the replacement method for missing values prevents the loss of a significant amount of information.

Among the different discretization methods examined in this research, experiments find that entropy-based methods are faster and result in smaller models than those based on binning. However, binning methods achieved a higher predictive accuracy with less variance. As for feature subset selection methods, the impact of class noise, the number of instances, and missing values on feature subset selection are found to be high. Missing values especially have a strong effect. However, these findings are specific to the AntMiner+ algorithm. The real bounty of this step is that improving

the classification potential of a dataset is now a convenient problem-centered approach to computation.

Acknowledgements This work was supported by the Research Center of College of Computer and Information Sciences, King Saud University, Saudi Arabia. The authors are grateful for this support.

References

1. Pham H N A, Triantaphyllou E. An application of a new metaheuristic for optimizing the classification accuracy when analyzing some medical datasets. Expert Systems with Applications, 2009, 36: 9240–9249
2. Almuhaideb S, El-Bachir Menai M. Hybrid metaheuristics for medical data classification. In: El-Ghazali T, ed. Hybrid Metaheuristics. Springer, 2013, 187–217
3. Penã-Reyes C A, Sipper M. Evolutionary computation in medicine: an overview. Artificial Intelligence in Medicine, 2000, 19(1): 1–23
4. Tanwani A K, Afridi J, Shafiq M Z, Farooq M. Guidelines to select machine learning scheme for classification of biomedical datasets. In: Pizzuti C, Ritchie M D, Giacobini M, eds. Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics. Springer, 2009, 28–139
5. Almuhaideb S, El-Bachir Menai M. A new hybrid metaheuristic for medical data classification. International Journal of Metaheuristics, 2014, 3(1): 59–80
6. Milne D, Witten I H. An open-source toolkit for mining Wikipedia. Artificial Intelligence, 2013, 194: 222–239
7. Alcalá-fdez J, L. Sánchez L, García S, del Jesus M J, Ventura S, Garrell J M, Otero J, Bacardit J, Rivas V M, Fernández J C, Herrera F. KEEL: a software tool to assess evolutionary algorithms to data mining problems. Soft Computing, 2009, 13(3): 307–318
8. Martens D, de Backer M, Haesen R, Vanthienen J, Snoeck M, Baesens B. Classification with ant colony optimization. IEEE Transactions on Evolutionary Computation, 2007, 11(5): 651–665
9. Tanwani A K, Farooq M. Performance evaluation of evolutionary algorithms in classification of biomedical datasets. In: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation: Late Breaking Papers. 2009, 2617–2624
10. Tanwani A K, Farooq M. The role of biomedical dataset inclassification. In: Proceedings of Conference on Artificial Intelligence in Medicine in Europe. 2009
11. Tanwani A K, Farooq M. Classification potential vs. classification accuracy: a comprehensive study of evolutionary algorithms with biomedical datasets. Learning Classifier System, 2010: 127–144
12. Kotsiantis S B. Feature selection for machine learning classification problems: a recent overview. Artificial Intelligence Review, 2011: 249–268
13. Whitney A W. A direct method of nonparametric measurement selection. IEEE Transactions on Computers, 1971, 20(9): 1100–1103
14. Marill T, Green D. On the effectiveness of receptors in recognition systems. IEEE Transactions on Information Theory, 1963, 9(1): 11–17
15. Pudil P, Novovičová J, Kittler J. Floating search methods in features election. Pattern Recognition Letters, 1994, 15(10): 1119–1125
16. Yusta S C. Different metaheuristic strategies to solve the feature selection problem. Pattern Recognition Letters, 2009, 30(5): 525–534
17. Jourdan L, Dhaenens C, Talbi E G. A genetic algorithm for features election in datamining for genetics. In: Proceedings of the 4th Metaheuristics International Conference Porto. 2010: 29–34
18. Huang J J, Cai Y Z, Xu X M. A hybrid genetic algorithm for features election wrapper based on mutual information. Pattern Recognition Letters, 2007, 28(13): 1825–1844
19. AI-Ani A. Feature subset selection using ant colony optimization. International Journal of Computational Intelligence, 2005, 2(1): 53–58
20. Unler A, Murat A. A discrete particle swarm optimization method for feature selection in binary classification problems. European Journal of Operational Research, 2010, 206(3): 528–539
21. Bekkerman R, El-Yaniv R, Tishby N, Winter Y. Distributional word clusters vs. words for text categorization. Journal of Machine Learning Research, 2003, 3: 1183–1208
22. Liu H, Yu L. Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge Discovery and Data Engineering, 2005, 17(4): 491–502
23. Shin K, Fernandes D, Miyazaki S. Consistency measures for features election: a formal definition, relative sensitivity comparison, and a fast algorithm. In: Proceedings of International Conference on Artificial Intelligence (IJCAI). 2011, 1491–1497
24. Kerber R. ChiMerge: discretization of numeric attributes. In: Proceedings of the 10th National Conference on Artificial Intelligence. 1992, 123–128
25. Liu H, Setiono R. Feature selection via discretization. IEEE Transactions on Knowledge and Data Engineering, 1997, 9(4): 642–645
26. Fayyad U M, Irani K B. Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of International Conference on Artificial Intelligence. 1993, 1022–1029
27. Jin R M, Breitbart Y, Muoh C. Data discretization unification. Knowledge and Information Systems, 2009, 19(1): 1–29
28. Quinlan R. C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann Publishers, 1993
29. Guyon I, Elisseeff A. An introduction to variable and feature selection. The Journal of Machine Learning Research, 2003, 3: 1157–1182
30. Kohavi R, John G H. Wrappers for feature subsets election. Artificial Intelligence, 1997, 97(1–2): 273–324
31. Caruana R, Freitag D. Greedy attribute selection. In: Proceedings of International Conference on Machine Learning. 1994, 28–36
32. Koza J R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA: MIT Press, 1992
33. Breiman L, Friedman J H, Olshen R A, Stone C J. Classification and Regression Trees. New York: Chapman & Hall, 1984
34. Das S. Filters, wrappers and a boosting-based hybrid for feature selection. In: Proceedings of International Conference on Machine Learning. 2001, 74–81
35. Han J W, Kamber M. Data Mining: Concepts and Techniques. 2nd edition. London: Morgan Kaufmann Publishers, 2006
36. Chlebus B S, Nguyen S H. On finding optimal discretizations for two attributes. In: Polkowski L, Skowron A, eds. Rough Sets and Current Trends in Computing. Springer, 1998, 537–544
37. García S, Luengo J, Sáez J A, López V, Herrera F. A survey of discretization techniques: taxonomy and empirical analysis in supervised

learning. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(4): 734–750

38. Wong A K C, Chiu D K Y. Synthesizing statistical knowledge from incomplete mixed-mode data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1987, 9(6): 796–805
39. Garcá-Laencina P J, Sancho-Gómez J L, Figueiras-Vidal A R. Pattern classification with missing data: a review. Neural Computing and Applications, 2010, 19(2): 263–282
40. Grzymala-Busse J W, Goodwin L K, Grzymala-Busse W J, Zheng X Q. Handling missing attribute values in preterm birth data sets. In: Slezak D, Yao J T, Peters J F, Ziarko W, Hu X H, eds. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. Springer, 2005, 342–351
41. Batista G E A P A, Monard M C. An analysis of four missing data treatment methods for supervised learning. Applied Artificial Intelligence, 2003, 17(5–6): 519–533
42. Feng H H, Chen G S, Yin C, Yang B R, Chen Y M. A SVM regression based approach to filling in missing values. In: Khosla R, Howlett R J, Jain L C, eds. Knowledge-Based Intelligent Information and Engineering Systems. Springer, 2005, 581–587
43. Gupta A, Lam M S. Estimating missing values using neural networks. Journal of the Operational Research Society, 1996, 47(2): 229–238
44. Dempster A P, Laird N M, Rubin D B. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B (Methodological), 1977, 39(1): 1–38
45. Schneider T. Analysis of incomplete climate data: estimation of mean values and covariance matrices and imputation of missing values. Journal of Climate, 2001, 14: 853–871
46. Gourraud P A, Génin E, Cambon-Thomsen A. Handling missing values in population data: consequences for maximum likelihood estimation of haplotype frequencies. European Journal of Human Genetics, 2004, 12: 805–812
47. Mcculloch W, Pitts W. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 1943, 5: 115–133
48. Holland J H. Adaptation in Natural and Artificial Systems. Ann Arbor: The University of Michigan Press, 1975
49. Dorigo M. Optimization, learning and natural algorithms. Dissertation for the Doctoral Degree. Politecnico di Milano, Italy, 1992
50. Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks. 1995, 1942–1948
51. Sato T, Hagiwara M. Bee system: finding solution by a concentrated search. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics. 1997
52. Karaboga D. An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, 2005
53. Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53–66
54. Parpinelli R S, Lopes H S, Freitas A A. Data mining with an ant colony optimization algorithm. IEEE Transactions on Evolutionary Computation, 2002, 6(4): 321–332
55. Stützle T, Hoos H H. MAX-MIN ant system. Future Generation Computer Systems, 2000, 16(8): 889–914
56. Pellegrini P, Ellero A. The small world of pheromone trails. In: Dorigo M, Birattari M, Blum C, Clerc M, Stützle T, Winfield A F T, eds. Ant Colony Optimzation and Swarm Intelligence. Springer, 2008, 387–394
57. Cohen W W. Fast effective rule induction. In: Prieditis A, Russell S J, eds. International Conference on Machine Learning. Morgan Kaufmann, 1995, 115–123
58. Minnaert B, Martens D, de Baker M, Baesens B. To tune or not to tune: rule evaluation for metaheuristic-based sequential covering algorithms. Data Mining and Knowledge Discovery, 2015, 29(1): 237–272
59. Almuhaideb S, ElBachir Menai M. A new hybrid metaheuristic for medical data classification. International Journal of Metaheuristics, 2014: 1–17
60. Rissanen J. Modeling by shortest data description. Automatica, 1978, 14(5): 465–471
61. Kononenko I. On biases in estimating multi-valued attributes. In: Proceedings of International Conference on Artificial Intelligence. 1995, 1034–1040
62. Kira K, Rendell L A. A practical approach to feature selection. In: Proceedings of the 9th International Workshop on Machine Learning. 1992
63. Kononenko I. Estimating attributes: analysis and extensions of RELIEF. In: Proceedings of European Conference on Machine Learning. 1994, 171–182
64. Hall M A. Correlation-based feature selection for machine learning. Dissertation for the Dotoral Degree. Hamilton, New Zealand: University of Waikato, 1999
65. Liu H, Setiono R. A probabilistic approach to feature selection—a filter solution. In: Proceedings of International Conference on Machine Learning. 1996, 319–327
66. Frank E, Witten I H. Generating accurate rule sets without global optimization. In: Proceedings of the 15th International Conference on Machine Learning. 1998, 144–151
67. Holte R C. Very simple classification rules perform well on most commonly used datasets. Machine Learning, 1993, 11(1): 63–91
68. Klösgan W. Problems for knowledge discovery in databases and their treatment in the statistics interpreter explora. International Journal of Intelligent Systems, 1992, 7(7): 649–673
69. Janssen F, Fürnkranz J. On the quest for optimal rule learning heuristics. Machine Learning, 2010, 78(3): 343–379
70. Martens D, Baesens B, Fawcett T. Editorial survey: swarm intelligence for data mining. Machine Learning, 2010, 82(1): 1–42
71. Hanczara B, Dougherty E R. The reliability of estimated confidence intervals for classification error rates when only a single sample is available. Pattern Recognition, 2013, 64(3): 1067–1077
72. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of International Conference on Artificial Intelligence. 1995, 1137–1145
73. García S, Fernández A, Luengo J, Herrera F. A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Computing, 2009, 13(10): 959–977
74. Wilcoxon F. Individual comparisons by ranking methods. Biometrics Bulletin, 1945, 1(6): 80–83
75. Friedman M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. American Statistical Association, 1937, 32(200): 675–701
76. Frank A, Asuncion A. UCI machine learning repository. Irvine, CA:

University of California, 2010

77. Napierala K, Stefanowski J. BRACID: a comprehensive approach to learning rules from imbalanced data. Journal of Intelligent Information Systems, 2012, 39(2): 335–373
78. Orriols-Puig A, Bernadó-Mansilla E. The class imbalance problem in UCS classifier system: a preliminary study. In: Proceedings of the 2003–2005 International Conference on Learning Classifier Systems. 2007, 161–180
79. Pazzani M J, Mani S, Shankle W R. Acceptance of rules generated by machine learning among medical experts. Methods of Information in Medicine, 2001, 40(5): 380–385
80. Vapnik V N. Estimation of Dependences Based on Empirical Data. Springer-Verlag, 1982
81. Vapnik V N. The Nature of Statistical Learning Theory. New York: Springer, 1995
82. Lim T S, Loh W Y, Shih Y S. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Machine Learning, 2000, 40(3): 203–228
83. Gonzalez A, Perez R. Slave: a genetic learning system based on an iterative approach. IEEE Transactions on Fuzzy Systems, 1999, 7(2): 176–191
84. Bernadó-Mansilla E, Garrell-Guiu J M. Accuracy based learning classifier systems: models, analysis and applications to classification tasks. Evolutionary Computation, 2003, 11(3): 209–238
85. Wilson S W. Classifier fitness based on accuracy. Evolutionary Computation, 1995, 3(2): 149–175
86. Orriols-Puig A, Casillas J, Bernadó-Mansilla E. A comparative study of several geneticbased supervised learning systems. In: Bull L, Bernadó-Mansilla E, Holmes J H, eds. Learning Classifier Systems in Data Mining. Springer, 2008, 205–230
87. Troyanskaya O G, Cantor M, Sherlock G, Brown P O, Hastie T, Tibshirani R, Botstein D, Altman R B. Missing value estimation methods for DNA microarrays. Bioinformatics, 2001, 17(6): 520–525
88. Amaldi E, Kann V. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. Theoretical Computer Science, 1998, 209(1–2): 237–260
89. Bacardit J, Butz M. Data mining in learning classifier systems: comparing XCS with gassist. In: Proceedings of International Conference on Learning Classifier Systems (IWLCS 2003–2005). 2004, 282–290

Sarab Almuhaideb is a PhD student in the Department of Computer Science, King Saud University, Saudi Arabia. She is a lecturer in the Department of Computer Science, Prince Sultan University, Saudi Arabia. Her research interests include issues related to machine learning, evolutionary computation, and hybrid metaheuristics.

Mohamed El Bachir Menai received his PhD degree in computer science from Mentouri University of Constantine, Algeria, and University of Paris VIII, France in 2005. He also received a “Habilitation universitaire” in computer science from Mentouri University of Constantine, in 2007 (it is the highest academic qualification in Algeria, France and Germany). He is currently a professor in the Department of Computer Science at King Saud University, Saudi Arabia. His main interests include evolutionary computing, data mining, machine learning, natural language processing, and satisfiability problems.