Rules and redescriptions as features in binary classification tasks

Matej Mihelčić[1](✉), and Tomislav Šmuc[2]

[1] Department of Mathematics, Faculty of Science, University of Zagreb, 10000 Zagreb, Croatia
Bijenička cesta 30, 10000 Zagreb, Croatia
matmih@math.hr
[2] Ruđer Bošković Institute
Bijenička cesta 54, 10000 Zagreb, Croatia
tomislav.smuc@irb.hr

Abstract. We assess the suitability of using rules and redescriptions as features in binary classification tasks. Using rules as features is known to be able to increase the performance of classification algorithms, advantages of using redescriptions in this setting are still unknown. Redescriptions bring benefits over regular rules since discovered subsets of entities are supported by at least two rules (containing disjoint attributes) that share strong equivalence relations. Redescriptions allow discovering highly irregularly shaped clusters due to their ability to use conjunction, disjunction and negation logical operators. Rules and redescriptions are generated using the CLUS-RM redescription mining algorithm. Each selected rule and redescription forms one binary attribute that is added to the existing set of attributes. The performed results show increase in accuracy of several different classification algorithms applied to a set of different binary classification datasets when newly constructed set of features is used. Feature ranking performed in all experiments shows positive importance of many constructed redescriptions and rules for predicting binary target variable.

Keywords: feature construction, binary classification, redescription mining, rule mining, CLUS-RM

1 Introduction

Predictive tasks such as classification, regression and numerous variants thereof are omnipresent in majority of existing scientific disciplines. With the rise of popularity and awareness of different predictive machine learning algorithms that are able to provide huge number of, often highly accurate predictions for these tasks, there is also an increasing need to provide tools and techniques to aid in the proper preparation, construction, extraction and selection of predictive attributes (these for which relation with target variables are investigated). Many different feature selection techniques exist [28, 13] that aim to eliminate irrelevant features (these that provide no or very little information about the

target concept). Feature extraction techniques [28, 14, 51] map existing features (using some function) to a new (very often smaller) set of features that capture important information about the relation of original features and the target concept. Such features can be used independently from the original feature set, but can also be added and used in synergy with original features.

In this work, we concentrate on the third type of data preparation for predictive tasks, the feature construction [28, 38]. The main aim of feature construction is to find new features (utilizing algorithms or using background, domain specific knowledge) which capture non-trivial, possibly non-linear interactions between existing, original features. The utility of the feature construction step is assessed via increase in the predictive performance and/or high importance of newly constructed features for the predictive task, but also through better understanding of the underlying problem. Feature construction can increase model performance and allow predicting concepts that were to hard to predict using original features.

It is already known that rules, as a first order constructs, combining several existing features using conjunction or disjunction logical operator can increase performance of a classification algorithm [40, 42, 36, 56, 57]. In this work, in addition to using rules, we also utilize redescriptions to construct new features capturing different non-linear relations between original features. Redescriptions are obtained using a redescription mining [43] algorithm called CLUS-RM [35]. The aim of a redescription is to redescribe (describe in multiple ways) some subset of entities contained in a dataset. Redescription mining is unsupervised, descriptive task although work has been done to assess predictability of redescriptions [58, 34]. Constructed redescriptions can redescribe entities using conjunction, disjunction and negation logical operator, which allows the approach to detect potentially irregularly shaped clusters of entities. Furthermore, each cluster must be described by two different rules, which may also help detect important subsets in presence of noise. Thus, redescriptions are a second order construct (contain rule-pairs that are in equivalence relation), forming complex but fully interpretable features.

2 Notation and related work

In this section, we define the most important terms necessary to understand the approach and provide an overview of related work.

2.1 Notation and definition

In this work, we use one-view datasets $\mathcal{D}$, containing $|\mathcal{A}|$ attributes and $|E|$ entities. Since we deal with a binary classification task, each entity is assigned a target label $y \in \{0, 1\}$. We use $\mathcal{M}$ to denote an arbitrary machine learning classification model that is trained on some data $\mathcal{D}_{train}$, and it outputs a prediction $\hat{y}$ for each entity $e \in E_{test}$, where $E_{train} \cap E_{test} = \emptyset$. We use $AUC(\mathcal{M}, \mathcal{D}_{test})$ to denote the Area under the ROC curve [2] achieved by the model $\mathcal{M}$ on the

test data $\mathcal{D}_{test}$ and $AUPRC(\mathcal{M}, \mathcal{D}_{test})$ to denote the Area under precision-recall curve [45] achieved by the model $\mathcal{M}$ on the test data $\mathcal{D}_{test}$.

In general, redescription mining dataset consists of a set of entities E, a set of views $W_1, W_2, \ldots, W_n$ and a set of variables (attributes) $V_1, V_2, \ldots, V_n$. In this work, we use only two redescription mining views ($n = 2$). Since we deal with one-view datasets, the set of variables is the same for both redescription mining views ($V_1 = V_2 = \mathcal{A}$), also views $W_1 = W_2$. We denote the set of attributes contained within a query (logical formula) with $attr(q)$, and a set of entities described by this query by $supp(q)$. Queries can contain conjunction, negation and disjunction logical operators. Redescriptions are pairs of queries of a form $R = (q_1, q_2)$, where $attr(q_1), attr(q_2) \in \mathcal{A}$ and $attr(q_1) \cap attr(q_2) = \emptyset$. Analogously, $supp(R)$ denotes the set of entities re-described by a redescription R, $supp(R) = supp(q_1) \cap supp(q_2)$.

Jaccard index is used to measure redescription accuracy and is defined as:

$$J(R) = \frac{|supp(q_1) \cap supp(q_2)}{|supp(q_1) \cup supp(q_2)|}$$

The p-value (p_{val}) is defined as:

$$p_{val}(R) = \sum_{k=|supp(R)|}^{|E|} \binom{|E|}{k} (p_1 \cdot p_2)^k \cdot (1 - p_1 \cdot p_2)^{|E|-k}$$

$|E|$ equals the number of entities in the dataset and p_1, p_2 correspond to marginal probabilities of obtaining queries q_1 and q_2. It represents a probability of obtaining a set of size equal to or larger than $supp(R)$ by combining two randomly selected queries with marginal probabilities p_1 and p_2.

An example redescription (R_{ex}), was obtained on a well known one-view dataset Iris [6], where $R_{ex} = (q_{1_{ex}}, q_{2_{ex}})$. The Iris dataset contains instances of three species of Iris flower for which four measurements were made: the width and the length of the sepal and the petals in centimetres.
R_{ex} is defined as:
$q_{1_{ex}}$: $\neg(4.3 \leq \text{sepalL} \leq 5.5 \ \wedge \ 2.9 \leq \text{sepalW} \leq 4.2)$
$q_{2_{ex}}$: $1.4 \leq \text{petalW} \leq 2.5 \ \vee \ 1.6 \leq \text{petalL} \leq 6.9$
Both queries of R_{ex} contain attributes from $\mathcal{A}$, but $attr(q_{1_{ex}}) \cap attr(q_{2_{ex}}) = \emptyset$. R_{ex} re-describes 60 instances of Iris flower (on the subset of the dataset selected for training) with Jaccard index 0.896.

A set of redescriptions $\mathcal{R}$ is constructed using score denoted $sc(R, \mathcal{R})$ [35] that takes into account redescription properties (such as accuracy, statistical significance, redescription query complexity) but also redescription set properties (such as redundancy with respect to entities and attributes). With $hom(supp(R), y)$ we denote the homogeneity of a support set of a redescription R with respect to a set of corresponding target variables y. $hom(supp(R), y) = max(\frac{|supp(R)^+|}{|supp(R)|}, \frac{|supp(R)^-|}{|supp(R)|})$, where $supp(R)^+ \subseteq supp(R) = \{e \in supp(R), \ y_e = 1\}$ and $supp(R)^- \subseteq supp(R) = \{e \in supp(R), \ y_e = 0\}$. Homogeneity of a rule is defined analogously. Final redescription score $sc_{fin}(R, \mathcal{R}) = 0.5 \cdot sc(R, \mathcal{R}) + (1 - hom(R, y))$.

We also define a score to select a first redescription $sc_{first}(R, \mathcal{R}) = 2 \cdot (1 - hom(supp(R), y))$.

2.2 Related work

Feature selection [28, 13] and feature construction [28, 38] are often used jointly in predictive tasks. As feature construction increases the number of variables, feature selection aims to choose the attributes containing the important information about the target variable allowing faster training/predicting with machine learning models and increasing their accuracy in practice (e.g [37, 17, 30]).

There exist various approaches for feature selection [28, 13], such as correlation based approaches (ranking using Pearson coefficient), forward selection using Gram-Schmidt orthogonalization, mutual information feature ranking, model-based feature ranking, hybrid approaches, various methods of feature subset selection, wrapper methods and filters [29]. Some ensemble algorithms such as random forest immediately provide feature ranking which can be used for feature selection (see [16]). Feature selection methods using models can be divided in performance-based approaches and test-based approaches [16].

Performance-based approaches (e.g [46, 19, 5, 9]) combine feature selection with a classifier-based feedback on the quality of the selected set of features.

Test-based approaches (e.g [44, 47, 1, 54]) combine permutation testing of attribute values with feature ranking obtained by random forest algorithm to assess the real significance of importance of original features.

There exist a large number of ways in which features can be constructed, including constructive induction [29], construction using fragmentary knowledge [29], greedy feature construction [38], genetic and co-evolutionary approaches [48, 24], hybrid approaches (e.g [49]).

Constructive induction approaches such as [40, 42, 36, 56, 57] construct new attributes as subsets of existing attributes. Attributes in the subset can be combined using conjunction, disjunction and negation logical operator [40, 42], or more complex operators such as M-of-N [36] (at least one conjunction of m out of N attributes is true), X-of-N [56] (for a given instance, it denotes the number of attribute-value pairs that are true) or using arithmetic combination of attributes [26]. Gomez and Morales [11] created a learning algorithm called RCA (restricted covering algorithm) which tries to build a single rule for each class with a predetermined number of terms. FRINGE, devised by Pagallo et al. [39] is one of the earliest decision-tree based feature construction algorithms (it adaptively enlarges the initial attribute set using negation and AND logical operators for learning DNF concepts). CITRE [32] and DC Fringe [55] combine existing attributes using conjunction and disjunction operators to construct new features. FICUS [31] generalizes previous approaches to allow combining existing features by some predefined user-defined function. Garcia et al. [8] create a fuzzy rule-based feature construction approach. It has also been shown [10] that embedding logical constraints (in a form of rules) to learning algorithms using quadratic programming can increase their performance.

Redescription mining [43] is a data mining task that aims to find subsets of entities that can be characterized in multiple ways (re-described). It discovers strong, equivalence-like relations between different subsets of attributes.

3 Feature construction procedure

In this section, we present the feature construction and machine learning algorithm testing framework.

Very important component of this framework is the CLUS-RM algorithm that uses the generalized redescription set construction procedure (GRSCP) [35]. We will denote the call of the CLUS-RM algorithm on dataset $\mathcal{D}$ containing E entities and a set of views $\mathbf{W}$, using user-defined constraints $\mathcal{C}$ and run settings *appset* with CLUSRM($\mathcal{D}, \mathbf{W}, \mathcal{C}, appset$). The algorithm has been modified for the purpose of feature construction to return two sets: a set of redescriptions $\mathcal{R}$ and a set of rules $\mathbf{r}$. The constraint set $\mathcal{C}$ consists of user-defined minimal and maximal redescription support size, minimally required redescription Jaccard index, maximally allowed redescription p-value and the minimally required Jaccard index required to perform conjunctive refinment [35]. The application settings contain information about the number of random restarts [34] and algorithm iteration, the use of logical operators (definition of a query language), definition of various parameters related to Predictive Clustering trees (PCTs) [22] (used to construct rules from which redescriptions are constructed). Besides tree-depth and target label type, parameters include the number of redescriptions to be returned and newly introduced parameter allowing redescription and rule selection based on target label homogeneity. Additionally introduced setting is a rule filtering based on redundancy with respect to rule's support set.

The feature construction procedure (Algorithm 1), takes as input a one-view binary classification dataset $\mathcal{D}$, a set of views $\mathbf{W}$, constraints $\mathcal{C}$ and application settings *set* for the CLUS-RM algorithm, a set of machine learning algorithms to be tested $S_{\mathcal{M}}$ and a set of algorithm parameters $params_{\mathcal{M}}$. The procedure first creates a stratified split of the data to train, validation and test set using standard 60%, 20%, 20% split (Algorithm 1, lines 2-3). The CLUS-RM algorithm is applied to the train data using predefined parameters and outputs a redescription and rule sets. After obtaining these sets, validation statistics and test supports are computed for obtained rules and redescriptions. Currently, the GRSCP relies solely on the predefined redescription measures on the train set (potentially in combination with the homogeneity of the target variable), but information about redescription predictivity on the validation set can also be incorporated into selection procedure. The obtained redescriptions and rules are added as features to the train, validation and test set (lines $9-11$ in Algorithm 1). Each redescription and rule forms one binary attribute, where value 1 denotes that entity is contained within support set of a redescription or a rule and 0 denotes the entity is not described with a given redescription or a rule. After obtaining dataset with new features, the framework computes model performance on this dataset using $\mathcal{D}_{train} \cup \mathcal{D}_{validation}$ as train and $\mathcal{D}_{test}$ as test set (lines $12-13$ in

Algorithm 1). Feature selection procedure similar to that of Svetnik et al. [46] is applied using Random Forest containing 600 PCTs (line 14 in Algorithm 1, also Algorithm 2).

Algorithm 1 A framework for feature construction and model testing

Input: Dataset $\mathcal{D}$, a set of views $\mathbf{W}$, Constraints $\mathcal{C}$, Settings *set*, A set of machine learning models $S_{\mathcal{M}}$, a set of model parameters $params_{\mathcal{M}}$

Output: A set of redescriptions $\mathcal{R}$, a set of rules $\mathbf{r}$, feature ranking $\mathcal{F}_{rank}$, a set of prediction results $S_{\mathcal{P}}$

1: **procedure** CRM-FC
2: $\mathbf{y} \leftarrow \mathcal{D}$.`targetVector()`
3: $\{\mathcal{D}_{train},\ \mathcal{D}_{validation},\ \mathcal{D}_{test}\} \leftarrow \mathcal{D}$.`splitDataStratified(`$0.6, 0.2, 0.2$`,y)`
4: $[\mathcal{R}, \mathbf{r}] \leftarrow$ `CLUSRM(`$\mathcal{D}_{train}$, $\mathbf{W}$, $\mathcal{C}$, *appset*$)$
5: $[\mathcal{R}', \mathbf{r}'] \leftarrow$ `computeValidStatsAndTestSupp(`$\mathcal{R}$, $\mathbf{r}$, $\mathcal{D}_{validation}$, $\mathcal{D}_{test}$$)$
6: $\mathbf{r}$.`filter(`*appset*`.maxEntRedundancy())`
7: **if** (*set*.`useClassHom()` **then**
8: $\mathbf{r}$.`filterHom(`*appset*`.minHom())`
9: $\mathcal{D}'_{train} \leftarrow$ `addFeatures(`$\mathcal{D}_{train}$, $\mathcal{R}'$, $\mathbf{r}'$$)$
10: $\mathcal{D}'_{validation} \leftarrow$ `addFeatures(`$\mathcal{D}_{validation}$, $\mathcal{R}'$, $\mathbf{r}'$$)$
11: $\mathcal{D}'_{test} \leftarrow$ `addFeatures(`$\mathcal{D}_{test}$, $\mathcal{R}'$, $\mathbf{r}'$$)$
12: $\mathcal{D}_{trainValidation} \leftarrow \mathcal{D}'_{train} \cup \mathcal{D}'_{validation}$
13: $\mathcal{P}_{all} \leftarrow$ `computeModelPerformance(`$\mathcal{D}_{trainValidation}$, $\mathcal{D}'_{test}$, $S_{\mathcal{M}}$, $params_{\mathcal{M}}$$)$
14: $\mathcal{F} \leftarrow$ `selectFeatures(`$\mathcal{D}_{trainValidation}$, *appset*`.fsIteration`, $\mathcal{M}_{RF600PCT}$$)$
15: $\mathcal{D}''_{train} \leftarrow$ `reduceFeatures(`$\mathcal{D}'_{train}$, $\mathcal{F}$$)$
16: $\mathcal{D}''_{validation} \leftarrow$ `reduceFeatures(`$\mathcal{D}'_{validation}$, $\mathcal{F}$$)$
17: $\mathcal{D}''_{test} \leftarrow$ `reduceFeatures(`$\mathcal{D}'_{test}$, $\mathcal{F}$$)$
18: $\mathcal{D}'_{trainValidation} \leftarrow \mathcal{D}''_{train} \cup \mathcal{D}''_{validation}$
19: $\mathcal{P}_{red} \leftarrow$ `computeModelPerformance(`$\mathcal{D}'_{trainValidation}$, $\mathcal{D}''_{test}$, $S_{\mathcal{M}}$, $params_{\mathcal{M}}$$)$
20: $\mathcal{P}_{orig} \leftarrow$ `computeModelPerformance(`$\mathcal{D}_{train} \cup \mathcal{D}_{validation}$, $\mathcal{D}_{test}$, $S_{\mathcal{M}}$, $params_{\mathcal{M}}$$)$
21: $\mathcal{F}' \leftarrow$ `selectFeatures(`$\mathcal{D}_{train} \cup \mathcal{D}_{validation}$, *appset*`.fsIteration`, $\mathcal{M}_{RF600PCT}$$)$
22: $\mathcal{D}_{trainS} \leftarrow$ `reduceFeatures(`$\mathcal{D}_{train}$, $\mathcal{F}'$$)$
23: $\mathcal{D}_{validationS} \leftarrow$ `reduceFeatures(`$\mathcal{D}_{validation}$, $\mathcal{F}'$$)$
24: $\mathcal{D}_{testS} \leftarrow$ `reduceFeatures(`$\mathcal{D}_{test}$, $\mathcal{F}'$$)$
25: $\mathcal{D}_{trainValidationS} \leftarrow \mathcal{D}_{trainS} \cup \mathcal{D}_{validationS}$
26: $\mathcal{P}_{origRed} \leftarrow$ `computeModelPerformance(`$\mathcal{D}_{trainValidationS}$, $\mathcal{D}_{test}$, $S_{\mathcal{M}}$,
27: $params_{\mathcal{M}}$$)$
28: $S_{\mathcal{P}} \leftarrow \{\mathcal{P}_{all},\ \mathcal{P}_{pred},\ \mathcal{P}_{orig},\ \mathcal{P}_{origRed}\}$
29: **return** $[\mathcal{R}, \mathbf{r}, \mathcal{F}, S_{\mathcal{P}}]$

Train, validation and test datasets are reduced using selected feature set and the performance measures for all models are computed on the reduced test set (lines $15-17$ in Algorithm 1). The same procedure is performed on the original data (not containing newly constructed features, lines $20-26$ in Algorithm 1). The framework returns a set of produced redescriptions $\mathcal{R}$, a set of rules $\mathbf{r}$, a set of features $\mathcal{F}$ and a set of performance results $S_{\mathcal{P}}$ containing the AUC and $AUPRC$ measures for each selected classifier for each of the four scenarios (all features, all after feature selection, original features, original after feature selection).

Algorithm 2 selectFeatures

Input: Dataset $\mathcal{D}$, Number of iterations $fsIteration$, Model capable of performing feature ranking $\mathcal{M}_{\mathcal{F}}$
Output: A reduced feature ranking $\mathcal{F}'$

1: **procedure** SELECTFEATURES
2: $\mathbf{y} \leftarrow \mathcal{D}$.`targetVector()`
3: $\mathcal{P}_{all} \leftarrow \emptyset$
4: **for** ($numTries = 0$; $numTries \leq fsIteration$; $numTries++$) **do**
5: $\mathcal{D}$.`randomize()`
6: $\{\mathcal{D}_{train}, \mathcal{D}_{validation}\} \leftarrow \mathcal{D}$.`splitDataStratified(0.75, 0.25,y)`
7: $numAttributes \leftarrow data$.`numAttributes()`
8: $\mathcal{D}'_{train} \leftarrow \mathcal{D}_{train}$
9: $\mathcal{D}'_{validation} \leftarrow \mathcal{D}_{validation}$
10: $\mathcal{F} \leftarrow$ `createFeatureRanking`$(\mathcal{D}'_{train}, \mathcal{M}_{\mathcal{F}})$
11: **while** ($numAttributes > 10$) **do**
12: $\mathcal{P}_t \leftarrow$ `computeModelPerformance`$(\mathcal{D}'_{train}, \mathcal{D}'_{validation}, \mathcal{M}_{\mathcal{F}})$
13: $\mathcal{P}_{all}[numAttributes] \leftarrow \mathcal{P}_{all}[numAttributes] + \frac{\mathcal{P}_t}{fsIteration}$
14: $numAttributes \leftarrow \frac{numAttributes}{2}$
15: $[\mathcal{D}'_{train}, \mathcal{D}'_{validation}] \leftarrow$ `reduceDataset`$(\mathcal{D}'_{train}, \mathcal{D}'_{validation},$
16: $numAttributes, \mathcal{F})$
17: $bestNumAttrs \leftarrow argmax_{\mathcal{P}_{all}.numAttributes}(\mathcal{P}_{all}[numAttributes])$
18: $\mathcal{F} \leftarrow$ `createFeatureRanking`$(\mathcal{D}, \mathcal{M}_{\mathcal{F}})$
19: $\mathcal{F}' \leftarrow$ `reduceFeatureRanking`$(\mathcal{F}, bestNumAttrs)$
20: **return** $\mathcal{F}'$

The feature selection procedure (Algorithm 2) follows guidelines set up by Svetnik et al. [46]. It takes as input a dataset $\mathcal{D}$ consisting of a train and validation part of our input dataset, the number of iterations to be performed, a model capable of performing feature ranking $\mathcal{M}_{\mathcal{F}}$ and it outputs reduced feature ranking $\mathcal{F}'$. The procedure performs $fsIteration$ iterations (line 4 in Algorithm 2). Instead of using five-fold cross validation procedure as in Svetnik et al., we validate the performance of our model $\mathcal{M}_{\mathcal{F}}$ and create feature rankings using train/validation split of the data. At each iteration, the data consisting of a train and a validation part is randomized and a stratified split to train/validation is created (lines $5-6$ in Algorithm 2). Feature ranking is created on the train dataset using model $\mathcal{M}_{\mathcal{F}}$ (line 10 in Algorithm 2). Following the procedure, the number of attributes is halved and model performance computed at each iteration of the algorithm (lines $11-16$ in Algorithm 2). The performance is computed as $0.5 \cdot AUC(\mathcal{M}_{\mathcal{F}}, \mathcal{D}_{validation}) + 0.5 \cdot AUPRC(\mathcal{M}_{\mathcal{F}}, \mathcal{D}_{validation})$. This choice is made to select the best features for both balanced and unbalanced classification tasks. The number of attributes with the highest average score after $fsIteration$ runs is selected (line 17 in Algorithm 2). A new feature ranking is performed on the dataset $\mathcal{D}$ and the feature ranking containing selected number of attributes is returned as output (lines $19-20$ in Algorithm 2).

The computational complexity of the approach depends on the time complexity of the CLUS-RM algorithm, which is $\mathcal{O}(\text{CLUS-RM}) = \mathcal{O}(z \cdot (|V_1| + |V_2|)) \cdot (log_2(|E|))^2 \cdot |E| + z^3 \cdot |E|$, the time complexity of a feature ranking perform-

ing model $\mathcal{M}_{\mathcal{F}}$, $\mathcal{O}(\mathcal{M}_{\mathcal{F}})$ and the time complexity of the models to be tested $\mathcal{O}(\mathcal{M} \in S_{\mathcal{M}})$. Thus, the overall complexity of the framework for feature construction is $max(\mathcal{O}(\text{CLUS-RM}), \mathcal{O}(\mathcal{M}_{\mathcal{F}}), max(\mathcal{O}(\mathcal{M} \in S_{\mathcal{M}})))$.

4 Data description

We used 8 datasets downloaded from the UCI Machine learning repository [50] and Kaggle [21] to evaluate the proposed feature construction methodology. Characteristics of the used datasets can be seen in Table 1.

Table 1: Data characteristics

Name	$\lvert\mathcal{A}\rvert$	$\lvert E\rvert$	Missing	Attr. type	Reference
Adult_{sub}	104	12211	Yes	Num./Bool.	[23]
Arrhytmia	279	452	Yes	Num./Bool.	[12]
Breast cancer	32	569	No	Num.	[53]
Epileptic seizure	178	11500	No	Num.	[7]
Iris	4	150	No	Num.	[6]
$\text{Secom}_{(1:3)}$	590	404	Yes	Num.	[33]
Sports articles	57	1000	No	Num.	[15]
Theorem proving	51	6118	No	Num.	[3]

The Secom dataset is highly unbalanced (1 : 14) in favour of the class 0, thus we create and tested the approach on the more balanced version (1 : 3). We took a stratified sub sample of the Adult data to reduce its size to 12211 instances.

5 Experiments and results

Datasets from Table 1 that originally do not have binary target label were transformed to binary classification problems. Arrhythmia - originally a multi-class classification problem with 16 values of class variable was transformed into binary: normal and not-normal, in Iris we predict each species of Iris flower separately (solve 3 binary classification problems). The Theorem proving dataset is a multi-label classification task (has 6 labels), we predict only the decline option (H_0) - did any of the five available solvers prove the theorem or not.

CLUS-RM algorithm with GRSCP was used to generate redescriptions (see Table 2). Random Forest containing 600 PCTs [22] was used to produce feature ranking and feature selection (as a model $\mathcal{M}_{\mathcal{F}}$). The set of classifiers ($S_{\mathcal{M}}$) consists of $J48$ [41], Naive Bayes [20] (NB), logistic regression [27] (Log), Multilayer perceptron [52] (MLP), KStar [4] (KS), Decision Stump [18] (DSt), Logistic Model trees [25] (LMT) and a Random Forest containing 600 PCTs [22] (RF^{600}_{PCT}). Default parameters were used to train these models on all feature sets.

The performance results (AUC and $AUPRC$) of the selected 8 classifiers, for each configuration from Table 2 are provided in Table 3. We also perform feature selection, using procedure described in Algorithm 2 and write the corresponding results. Different feature configuration are denoted a-all, afs - all feature selection, o-original, ofs - original feature selection. Value in boldface represents the

Table 2: CLUS-RM parameters

Name	JS_m	$Supp_{min}$	$Supp_{max}$	nI	nr	$\|\mathbf{r}\|$	rJS_{min}	$cPure$	$rHom$	$\|\mathcal{R}\|$
Adult_{sub}	0.6	200	7200	20	10	187	0.5	true	0.8	88
Arrhytmia_{US}	0.6	40	250	30	200	2018	0.8	false	-	200
Arrhytmia_{S}	0.6	40	250	30	200	111	0.4	true	0.7	200
Breast cancer	0.6	40	340	10	5	18	0.8	true	0.6	4
Epileptic seizure	0.3	200	4050	30	10	49	0.4	false	-	200
Iris_1	0.6	10	80	10	5	21	0.8	false	-	6
Iris_2	0.6	10	80	10	5	9	0.2	true	1.0	1
Iris_3	0.6	10	80	10	5	9	0.2	false	-	1
$\text{Secom}_{(1:3)}$	0.4	10	300	10	5	376	0.8	false	-	200
Sports articles	0.6	160	500	10	20	9	0.2	false	-	20
Theorem proving	0.6	100	3500	30	10	63	0.8	true	0.6	20

best score for the given algorithm on the selected dataset and the underlined value denotes that the classifier achieved better performance when using all features than only original features. Since all classifiers achieved AUPRC and AUC score 1.0 on Iris_1 dataset using all feature sets, we omit it from Table 3.

Table 3: Evaluation results of 8 selected classifiers.

$\mathcal{D}$	$\mathcal{M}$	$auprc_a$	$auprc_{afs}$	$auprc_o$	$auprc_{ofs}$	auc_a	auc_{afs}	auc_o	auc_{ofs}
Adult_{sub}	MLP	0.820	**0.851**	0.817	0.817	0.872	**0.899**	0.878	0.878
	LMT	0.875	**0.876**	0.876	0.876	0.913	0.913	**0.914**	0.914
	NB	0.770	0.820	**0.835**	0.835	0.860	0.874	**0.885**	0.885
	DSt	**0.654**	0.654	0.653	0.653	**0.771**	0.771	0.766	0.766
	Log	0.873	**0.877**	0.855	0.855	0.908	**0.912**	0.903	0.903
	KS	0.759	**0.762**	0.745	0.745	**0.813**	0.813	0.785	0.785
	J_{48}	0.803	0.792	**0.823**	0.823	**0.861**	0.848	0.859	0.859
	RF^{600}_{PCT}	0.778	0.785	**0.795**	0.795	0.911	0.913	**0.918**	0.918
Arrhytmia_{US}	MLP	0.811	**0.870**	0.765	0.766	0.811	**0.866**	0.763	0.774
	LMT	0.804	**0.872**	0.787	0.812	0.814	**0.858**	0.785	0.797
	NB	0.733	0.857	0.805	**0.877**	0.787	0.871	0.841	**0.880**
	DSt	**0.617**	0.617	0.548	0.548	**0.665**	0.665	0.582	0.582
	Log	**0.773**	0.735	0.612	0.553	**0.787**	0.776	0.625	0.590
	KS	0.5	**0.546**	0.508	0.545	0.5	0.509	0.488	**0.543**
	J_{48}	0.624	0.720	0.750	**0.770**	0.670	0.761	0.802	**0.816**
	RF^{600}_{PCT}	0.880	**0.925**	0.921	0.924	0.883	**0.929**	0.922	0.927
Arrhytmia_{S}	MLP	0.760	**0.780**	0.765	0.766	0.744	0.765	0.763	**0.774**
	LMT	**0.847**	0.810	0.787	0.812	**0.868**	0.802	0.785	0.797
	NB	0.786	0.863	0.805	**0.877**	0.818	0.875	0.841	**0.880**
	DSt	0.548	0.548	0.548	0.548	0.582	0.582	0.582	0.582
	Log	**0.658**	0.649	0.612	0.553	0.677	**0.703**	0.625	0.590
	KS	0.509	**0.553**	0.508	0.545	0.521	0.528	0.488	**0.543**
	J_{48}	**0.776**	0.756	0.750	0.770	**0.823**	0.809	0.802	0.816
	RF^{600}_{PCT}	0.891	0.921	0.921	**0.924**	0.890	0.921	0.922	**0.927**
	MLP	0.986	**0.989**	0.984	0.988	0.985	**0.987**	0.982	0.986
	LMT	0.990	0.991	**0.992**	0.991	0.989	0.989	**0.991**	0.989

Table 3: Evaluation results of 8 selected classifiers.

$\mathcal{D}$	$\mathcal{M}$	$auprc_a$	$auprc_{afs}$	$auprc_o$	$auprc_{ofs}$	auc_a	auc_{afs}	auc_o	auc_{ofs}
B. cancer	NB	0.972	0.981	**0.983**	0.981	0.974	0.981	**0.984**	0.981
	DSt	0.816	0.816	0.816	0.816	0.865	0.865	0.865	0.865
	Log	0.942	0.993	0.974	**0.994**	0.951	0.993	0.984	**0.994**
	KS	0.974	0.976	0.975	**0.978**	0.976	0.978	0.976	**0.979**
	J_{48}	**0.921**	0.899	0.875	0.890	0.918	**0.925**	0.871	0.893
	RF^{600}_{PCT}	0.988	**0.990**	0.988	0.989	0.990	**0.990**	0.990	0.990
Ep. seizure	MLP	0.915	0.969	**0.978**	0.978	0.942	0.978	**0.986**	0.986
	LMT	**0.932**	0.932	0.930	0.930	0.946	0.943	**0.949**	0.949
	NB	0.904	0.922	**0.924**	0.924	0.960	**0.967**	0.965	0.965
	DSt	**0.641**	0.619	0.619	0.619	**0.777**	0.648	0.648	0.648
	Log	**0.924**	0.875	0.584	0.58	**0.949**	0.921	0.521	0.521
	KS	0.502	0.503	0.503	0.503	0.507	0.508	0.508	0.508
	J_{48}	0.858	**0.875**	0.859	0.859	0.876	**0.898**	0.872	0.872
	RF^{600}_{PCT}	0.980	**0.982**	0.982	0.982	0.995	**0.995**	0.995	0.995
Iris_2	MLP	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	LMT	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	NB	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	DSt	0.67	0.67	0.67	0.67	0.75	0.75	0.75	0.75
	Log	**1.0**	1.0	0.821	0.821	**1.0**	1.0	0.815	0.815
	KS	0.989	0.989	0.989	0.989	0.99	0.99	0.99	0.99
	J_{48}	0.943	0.943	0.943	0.943	0.95	0.95	0.95	0.95
	RF^{600}_{PCT}	**1.0**	1.0	0.99	0.99	**1.0**	1.0	0.995	0.995
Iris_3	MLP	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	LMT	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	NB	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	DSt	0.946	0.946	0.946	0.946	0.975	0.975	0.975	0.975
	Log	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	KS	**0.994**	0.994	0.988	0.988	**0.995**	0.995	0.99	0.99
	J_{48}	0.946	0.946	0.946	0.946	0.975	0.975	0.975	0.975
	RF^{600}_{PCT}	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
$\text{Secom}_{(1:3)}$	MLP	0.583	**0.654**	0.578	0.595	0.573	**0.644**	0.598	0.598
	LMT	0.619	0.629	**0.650**	0.645	0.648	0.660	0.687	**0.699**
	NB	**0.572**	0.568	0.563	0.546	**0.548**	0.543	0.539	0.520
	DSt	0.581	0.581	0.581	0.581	0.639	0.639	0.639	0.639
	Log	0.509	0.558	0.593	**0.619**	0.511	0.590	0.652	**0.675**
	KS	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	J_{48}	**0.591**	0.564	0.587	0.531	**0.663**	0.609	0.623	0.562
	RF^{600}_{PCT}	**0.551**	0.495	0.531	0.474	**0.777**	0.744	0.758	0.770
Sports articles	MLP	0.861	0.861	**0.874**	0.874	0.852	0.852	**0.880**	0.880
	LMT	**0.885**	0.885	0.884	0.884	0.892	0.892	**0.893**	0.893
	NB	0.833	0.833	**0.845**	0.845	0.857	0.857	**0.866**	0.866
	DSt	0.724	0.724	0.724	0.724	0.793	0.793	0.793	0.793
	Log	**0.886**	0.886	0.883	0.883	0.888	0.888	**0.892**	0.892
	KS	**0.806**	0.806	0.792	0.792	**0.813**	0.813	0.804	0.804
	J_{48}	**0.740**	0.740	0.664	0.664	**0.784**	0.784	0.709	0.709
	RF^{600}_{PCT}	**0.939**	0.939	0.938	0.938	**0.903**	0.903	0.902	0.902

Table 3: Evaluation results of 8 selected classifiers.

$\mathcal{D}$	$\mathcal{M}$	$auprc_a$	$auprc_{afs}$	$auprc_o$	$auprc_{ofs}$	auc_a	auc_{afs}	auc_o	auc_{ofs}
Th. proving	MLP	0.791	**0.843**	0.835	0.835	0.799	**0.852**	0.842	0.842
	LMT	0.830	0.833	**0.845**	0.845	0.859	0.862	**0.866**	0.866
	NB	**0.678**	0.654	0.646	0.646	**0.708**	0.683	0.676	0.676
	DSt	0.558	0.558	0.558	0.558	0.586	0.586	0.586	0.586
	Log	**0.778**	0.750	0.733	0.733	**0.797**	0.772	0.756	0.756
	KS	**0.874**	0.872	0.872	0.872	**0.892**	0.890	0.890	0.890
	J_{48}	**0.780**	0.766	0.773	0.773	**0.826**	0.817	0.823	0.822
	RF_{PCT}^{600}	0.905	0.905	**0.906**	0.906	0.930	**0.930**	0.930	0.930

Results from Table 3 demonstrate the potential of this approach to increase performance of multiple classification algorithms (using all attributes or after applying feature selection). Arrhythmia dataset example demonstrates that different runs and modes of rule and redescription selection can increase performance of different classification algorithms. Iris example shows this feature construction procedure can boost classifier performance even on datasets with very small number of attributes. Given the AUPRC measure, the approach increases performance of logistic regression on 6 out of 8 datasets (75%), J_{48}, KStar on 5 out of 8 datasets (62, 5%), the Multilayer perceptron and the Random Forest of Predictive Clustering trees on 4 out of 8 datasets (50%), the Logistic Model trees, and Decision Stump on 3 out of 8 datasets (37, 5%) and the least improvement is seen in Naive Bayes approach 2 out of 8 datasets (25%) - which is expected given its assumption on conditional independence of attributes.

Table 4: Number of rules (first) and redescriptions (second in the pair) with positive feature ranking score contained in the listed feature ranking intervals for all datasets, given $|\mathcal{A}_{or}|$ original features and $|\mathcal{A}_{all}| - |\mathcal{A}_{or}|$ constructed features.

$\mathcal{D}$	$\lvert\mathcal{A}_{or}\rvert$	$\lvert\mathcal{A}_{all}\rvert$	$[1, 30]$	$[31, 80]$	$[81, 150]$	$[151, 250]$	$[251, 450]$	451^+	Top rank
Ad._{sub}	104	379	$(7, 6)$	$(27, 8)$	$(43, 19)$	$(51, 34)$	$(59, 21)$	$(-, -)$	$(6., 3.)$
Arr._{US}	279	2497	$(0, 0)$	$(0, 0)$	$(5, 2)$	$(79, 6)$	$(185, 12)$	$(1713, 165)$	$(103., 138.)$
Arr._{S}	279	590	$(0, 0)$	$(0, 0)$	$(2, 0)$	$(43, 44)$	$(58, 129)$	$(8, 27)$	$(136., 153.)$
BC	32	52	$(0, 0)$	$(18, 4)$	$(-, -)$	$(-, -)$	$(-, -)$	$(-, -)$	$(31., 32.)$
ES	178	427	$(0, 0)$	$(0, 0)$	$(0, 0)$	$(23, 49)$	$(26, 151)$	$(-, -)$	$(181., 184.)$
Iris_1	4	31	$(20, 6)$	$(1, 0)$	$(-, -)$	$(-, -)$	$(-, -)$	$(-, -)$	$(8., 3.)$
Iris_2	4	9	$(3, 2)$	$(-, -)$	$(-, -)$	$(-, -)$	$(-, -)$	$(-, -)$	$(5., 6.)$
Iris_3	4	9	$(3, 2)$	$(-, -)$	$(-, -)$	$(-, -)$	$(-, -)$	$(-, -)$	$(5., 6.)$
$\text{Sec}_{(1;3)}$	590	1166	$(0, 0)$	$(0, 0)$	$(0, 0)$	$(1, 0)$	$(16, 10)$	$(138, 90)$	$(175., 322.)$
SA	57	86	$(0, 0)$	$(9, 16)$	$(0, 4)$	$(-, -)$	$(-, -)$	$(-, -)$	$(55., 53.)$
TP	51	134	$(0, 0)$	$(24, 5)$	$(39, 15)$	$(-, -)$	$(-, -)$	$(-, -)$	$(54., 52.)$

Many redescriptions and rules have positive feature importance on all datasets (some have very high importance, see Table 4). The prime example is the Adult dataset, where the best ranking rule is 6. and the best ranking redescription is 3. out of 379 features (from which 104 are original).

6 Conclusion and future work

We have presented a feature construction approach that uses redescriptions and rules, created by the CLUS-RM algorithm, to create interpretable attributes that increase classifier performance in binary classification tasks. Experimental results, performed on 8 different datasets using 8 different classification algorithms have confirmed the ability of this approach to increase performance of multiple classifiers. This can also be considered its main drawback (instability) necessitating experimentation to obtain suitable features. Homogeneity filter was a first attempt at guided feature construction using information about target variable, however as it shows, this approach does not always yield the best performance. Despite that, the presented approach may reduce the need for manual feature construction since it is capable of producing useful features expressing linear or non-linear relations between existing attributes.

Future work will include testing this methodology on more challenging datasets (low average predictive preformance), on different tasks such as multi-class and multi-label classification, single variable or multi-target regression, hierarchical multi-label classification etc. It will also assess the suitability of this framework to increase performance of multi-view classification algorithms or to incorporate information from additional views via redescriptions in order to increase the performance of single-view classification algorithms.

References

1. Altmann, A., Toloşi, L., Sander, O., Lengauer, T.: Permutation importance: a corrected feature importance measure. Bioinformatics 26(10), 1340–1347 (2010)
2. Bradley, A.P.: The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recogn. 30(7), 1145–1159 (1997)
3. Bridge, J.P., Holden, S.B., Paulson, L.C.: Machine learning for first-order theorem proving. Journal of Automated Reasoning 53(2), 141–172 (2014)
4. Cleary, J.G., Trigg, L.E.: K*: An instance-based learner using an entropic distance measure. In: Prieditis, A., Russell, S. (eds.) Machine Learning Proceedings 1995, pp. 108 – 114. Morgan Kaufmann, San Francisco (CA) (1995)
5. Díaz-Uriarte, R., Alvarez de Andrés, S.: Gene selection and classification of microarray data using random forest. BMC Bioinformatics 7(1), 3 (2006)
6. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of eugenics 7(2), 179–188 (1936)
7. G. Andrzejak, R., Lehnertz, K., Mormann, F., Rieke, C., David, P., Elger, C.: Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. Phys. Rev. E, Statistical, nonlinear, and soft matter physics 64, 061907 (2002)

8. García, D., Stavrakoudis, D., González, A., Pérez, R., Theocharis, J.B.: A fuzzy rule-based feature construction approach applied to remotely sensed imagery. In: 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT-15). Atlantis Press (2015)
9. Genuer, R., Poggi, J.M., Tuleau-Malot, C.: Variable selection using random forests. Pattern Recogn. Lett. 31(14), 2225–2236 (2010)
10. Giannini, F., Diligenti, M., Gori, M., Maggini, M.: Learning łukasiewicz logic fragments by quadratic programming. In: Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part I. pp. 410–426 (2017)
11. Gomez, G., Morales, E.F.: Automatic feature construction and a simple rule induction algorithm for skin detection. In: In Proc. of the ICML Workshop on Machine Learning in Computer Vision. pp. 31–38 (2002)
12. Guvenir, H.A., Acar, B., Demiroz, G., Cekin, A.: A supervised machine learning algorithm for arrhythmia analysis. In: Computers in Cardiology. pp. 433–436 (1997)
13. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of machine learning research 3, 1157–1182 (2003)
14. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A.: Feature extraction: foundations and applications, vol. 207. Springer (2008)
15. Hajj, N., Rizk, Y., Awad, M.: A subjectivity classification framework for sports articles using improved cortical algorithms. Neural Computing and Applications pp. 1–17 (2018)
16. Hapfelmeier, A., Ulm, K.: A new variable selection approach using random forests. Computational Statistics & Data Analysis 60, 50 – 69 (2013)
17. Haury, A.C., Gestraud, P., Vert, J.P.: The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. PLOS ONE 6(12), 1–12 (2011)
18. Iba, W., Langley, P.: Induction of one-level decision trees. In: Sleeman, D., Edwards, P. (eds.) Machine Learning Proceedings 1992, pp. 233 – 240. Morgan Kaufmann, San Francisco (CA) (1992)
19. Jiang, H., Deng, Y., Chen, H.S., Tao, L., Sha, Q., Chen, J., Tsai, C.J., Zhang, S.: Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. BMC Bioinformatics 5(1), 81 (2004)
20. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. pp. 338–345. UAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1995)
21. Kaggle: Kaggle datasets (Last access: 23/06/2019), `https://www.kaggle.com/`
22. Kocev, D., Vens, C., Struyf, J., Džeroski, S.: Tree ensembles for predicting structured outputs. Pattern Recognition 46(3), 817 – 833 (2013)
23. Kohavi, R.: Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In: PROCEEDINGS OF THE SECOND INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING. pp. 202–207. AAAI Press (1996)
24. Krawiec, K., Włodarski, L.: Coevolutionary feature construction for transformation of representation of machine learners. In: Kłopotek, M.A., Wierzchoń, S.T., Trojanowski, K. (eds.) Intelligent Information Processing and Web Mining. pp. 139–150. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
25. Landwehr, N., Hall, M., Frank, E.: Logistic model trees. Machine Learning 59(1), 161–205 (2005)

26. Langley, P., Bradshaw, G.L., Simon, H.A.: Rediscovering Chemistry with the Bacon System, pp. 307–329. Springer Berlin Heidelberg, Berlin, Heidelberg (1983)
27. Lee, A.H., Silvapulle, M.J.: Ridge estimation in logistic regression. Communications in Statistics - Simulation and Computation 17(4), 1231–1257 (1988)
28. Liu, H., Motoda, H., Yu, L., Ye, N.: Feature extraction, selection, and construction. The handbook of data mining pp. 409–424 (2003)
29. Liu, H., Motoda, H.: Feature extraction, construction and selection: A data mining perspective, vol. 453. Springer Science & Business Media (1998)
30. Mansbridge, N., Mitsch, J., Bollard, N., Ellis, K., Miguel-Pacheco, G., Dottorini, T., Kaler, J.: Feature selection and comparison of machine learning algorithms in classification of grazing and rumination behaviour in sheep. Sensors 18(10), 3532 (2018)
31. Markovitch, S., Rosenstein, D.: Feature generation using general constructor functions. Machine Learning 49(1), 59–98 (2002)
32. Matheus, C.J., Rendell, L.A.: Constructive induction on decision trees. In: Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 1. pp. 645–650. IJCAI'89, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1989)
33. McCann, M., Li, Y., Maguire, L., Johnston, A.: Causality challenge: Benchmarking relevant signal components for effective monitoring and process control. In: Guyon, I., Janzing, D., Schölkopf, B. (eds.) Proceedings of Workshop on Causality: Objectives and Assessment at NIPS 2008. Proceedings of Machine Learning Research, vol. 6, pp. 277–288. PMLR, Whistler, Canada (2010)
34. Mihelčić, M.: Construction and Exploration of Redescription Sets. Ph.D. thesis, International Postgraduate School Jožef Stefan, Slovenia (2018)
35. Mihelčić, M., Džeroski, S., Lavrač, N., Šmuc, T.: A framework for redescription set construction. Expert Systems with Applications 68, 196 – 215 (2017)
36. Murphy, P.M., Pazzani, M.J.: Id2-of-3: Constructive induction of m-of-n concepts for discriminators in decision trees. In: Machine Learning Proceedings 1991, pp. 183–187. Elsevier (1991)
37. Muthukumaran, K., Rallapalli, A., Murthy, N.L.B.: Impact of feature selection techniques on bug prediction models. In: Proceedings of the 8th India Software Engineering Conference. pp. 120–129. ISEC '15, ACM, New York, NY, USA (2015)
38. Oglic, D., Gärtner, T.: Greedy feature construction. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 3945–3953. Curran Associates, Inc. (2016)
39. Pagallo, G.: Learning dnf by decision trees. In: Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 1. pp. 639–644. IJCAI'89, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1989)
40. Pagallo, G.M.: Adaptative decision tree algorithms for learning from examples (ph.d. thesis). Tech. rep., Santa Cruz, CA, USA (1990)
41. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
42. Ragavan, H., Rendell, L.A.: Lookahead feature construction for learning hard concepts. In: Proceedings of the Tenth International Conference on Machine Learning. pp. 252–259. Morgan Kaufmann Publishers Inc. (1993)
43. Ramakrishnan, N., Kumar, D., Mishra, B., Potts, M., Helm, R.F.: Turning cartwheels: An alternating algorithm for mining redescriptions. In: Proceedings of the 10Th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 266–275. KDD 2004, ACM, New York, NY, USA (2004)

44. Rodenburg, W., Heidema, A.G., Boer, J.M.A., Bovee-Oudenhoven, I.M.J., Feskens, E.J.M., Mariman, E.C.M., Keijer, J.: A framework to identify physiological responses in microarray-based gene expression studies: selection and interpretation of biologically relevant genes. Physiological Genomics 33(1), 78–90 (2008)
45. Saito, T., Rehmsmeier, M.: The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE 10(3), 1–21 (2015)
46. Svetnik, V., Liaw, A., Tong, C., Wang, T.: Application of breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules. In: Roli, F., Kittler, J., Windeatt, T. (eds.) Multiple Classifier Systems. pp. 334–343. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
47. Tang, R., Sinnwell, J., Li, J., N Rider, D., Andrade, M., Biernacka, J.: Identification of genes and haplotypes that predict rheumatoid arthritis using random forests. BMC proceedings 3 Suppl 7, S68 (2009)
48. Tran, B., Xue, B., Zhang, M.: Genetic programming for feature construction and selection in classification on high-dimensional data. Memetic Computing 8(1), 3–15 (2016)
49. Tran, B., Xue, B., Zhang, M.: Using feature clustering for gp-based feature construction on high-dimensional data. In: McDermott, J., Castelli, M., Sekanina, L., Haasdijk, E., García-Sánchez, P. (eds.) Genetic Programming. pp. 210–226. Springer International Publishing, Cham (2017)
50. UCI: Uci machine learning repository (Last access: 23/06/2019), https://archive.ics.uci.edu/ml/index.php
51. Van Der Maaten, L., Postma, E., Van den Herik, J.: Dimensionality reduction: a comparative review. Journal of machine learning research 10, 66–71 (2009)
52. Van Der Malsburg, C.: Frank rosenblatt: Principles of neurodynamics: Perceptrons and the theory of brain mechanisms. In: Palm, G., Aertsen, A. (eds.) Brain Theory. pp. 245–248. Springer Berlin Heidelberg, Berlin, Heidelberg (1986)
53. W. Nick Street, W. H. Wolberg, O.L.M.: Nuclear feature extraction for breast tumor diagnosis (1993), https://doi.org/10.1117/12.148698
54. Wang, M., Chen, X., Zhang, H.: Maximal conditional chi-square importance in random forests. Bioinformatics 26 6, 831–7 (2010)
55. Yang, D.S., Rendell, L., Blix, G.: A scheme for feature construction and a comparison of empirical methods. In: Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 2. pp. 699–704. IJCAI'91, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1991)
56. Zheng, Z.: Constructing nominal x-of-n attributes. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2. pp. 1064–1070. IJCAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1995)
57. Zheng, Z.: A comparison of constructing different types of new feature for decision tree learning. In: Feature Extraction, Construction and Selection, pp. 239–255. Springer (1998)
58. Zinchenko, T., Galbrun, E., Miettinen, P.: Mining predictive redescriptions with trees. In: 2015 IEEE International Conference on Data Mining Workshop (ICDMW). pp. 1672–1675 (2015)