Rules, subgroups and redescriptions as features in classification tasks

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

[1] Department of Mathematics, Faculty of Science, University of Zagreb
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 evaluate the suitability of using supervised and unsupervised rules, subgroups and redescriptions as new features and meaningful, interpretable representations for classification tasks. Although using supervised rules as features is known to allow increase in performance of classification algorithms, advantages of using unsupervised rules, subgroups, redescriptions and in particular their synergy with rules are still largely unexplored for classification tasks. To research this topic, we developed a fully automated framework for feature construction, selection and testing called DAFNE – Descriptive Automated Feature Construction and Evaluation. As with other available tools for rule-based feature construction, DAFNE provides fully interpretable features with in-depth knowledge about the studied domain problem. The performed results show that DAFNE is capable of producing provably useful features that increase overall predictive performance of different classification algorithms on a set of different classification datasets.

Keywords: feature construction, classification, redescription mining, rule mining, subgroup discovery, CLUS-RM, JRip, M5Rules, CN2-SD

1 Introduction

With the rise of popularity and awareness of different predictive machine learning algorithms able to provide huge number of often highly accurate predictions for various tasks, there is also an increasing need to provide tools and techniques to aid in the construction, extraction and selection of predictive attributes.

The main aim of feature construction is to find new features which capture non-trivial, possibly non-linear interactions between existing, original features [21, 29]. Its utility is assessed via increase in the predictive performance, high importance of newly constructed features for the predictive task and through better understanding of the underlying problem. Various types of feature construction have been studied: creating rules [31] or using decision tree based algorithms (Random Forest [40], Deep Forest [45]). The main advantage of rules is

that they can simultaneously offer interpretative and performance improvement for various classifiers [31, 32, 28, 44]. Rules can also be used as local predictors [5, 11, 27] to form global classification models.

In this work, we extend the study of rule-based feature construction to include subgroups, descriptive (unsupervised) rules and redescriptions. The main goal is to asses if and when the latter can be more informative than supervised rules or be used in synergy with supervised rules to improve performance. Subgroups [42, 17] have the same form of a logical formula as regular rules but describe subsets of instances such that their distribution of target labels significantly deviates from the target label distribution on the entire dataset. Redescriptions [33, 9] are tuples of logical formulae that can contain conjunction, disjunction or negation operator, with the constraint that each formula in a tuple (also called a query) should describe very similar (or ideally the same) subsets of entities. Redescription mining is unsupervised, descriptive task, with redescriptions representing a second order constructs (tuples of rules that in a nearly equivalence relation), forming complex but fully interpretable features.

As previously mentioned, rule-based features necessarily increase the dimensionality of data. The detrimental effect of such increase can be alleviated using different feature selection techniques [21, 14]. These techniques aim to eliminate irrelevant features (these that provide no or very little information about the target concept). Alternatively, feature extraction techniques [21, 39] 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.

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 $\mathcal{D}$ (one data table), containing $|\mathcal{A}|$ attributes and $|E|$ entities. Since we deal with a classification task, each entity is assigned a target label $y \in \{c_1, \ldots, c_k\}$, where a special case of Binary classification has $y \in \{0, 1\}$. 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 $E_{train} \cap E_{test} = \emptyset$.

The input data is used to create rules, subgroups and redescriptions. Rules and subgroups are logical formulae containing conditions and conjunction logical operator, whereas redescriptions contain tuples of logical formulae containing conditions and conjunction, disjunction and negation logical operators. Each query in a redescription can contain only attributes that are disjoint from attribute of other queries in the redescription. In this work, we use redescriptions formed by pairs of queries.

2.2 Related work

Feature selection [21, 14] and feature construction [21, 29] 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 [16, 23]).

Feature selection approaches [21, 14] include correlation-based, forward selection using Gram-Schmidt orthogonalization, mutual information or model-based feature ranking, hybrid approaches, various feature subset selection methods, wrapper and filter methods [22]. Some ensemble algorithms (e.g. random forest) provide feature ranking which can be used for feature selection (see [15]). Feature selection methods using models can be divided in performance-based approaches and test-based approaches [15].

Performance-based approaches (e.g [35, 6]) combine feature selection with a classifier-based feedback on the quality of the selected set of features. Test-based approaches (e.g [1, 41]) combine permutation testing of attribute values with feature ranking obtained by random forest algorithm to assess the real significance of importance of original features.

Feature construction approaches include constructive induction [22], construction using fragmentary knowledge [22], greedy feature construction [29] and hybrid approaches (e.g [36]). New lines of research in this direction represent self-supervised learning frameworks [2, 37] for learning useful new representations for tabular data.

Constructive induction approaches such as [31, 32, 28, 44] construct new attributes from subsets of existing attributes. Attributes in the subset can be combined using conjunction, disjunction and negation logical operator [31, 32], or more complex operators such as M-of-N [28] (at least one conjunction of m out of N attributes is true), X-of-N [44] (for a given instance, it denotes the number of attribute-value pairs that are true) or using arithmetic combination of attributes [19]. Gomez and Morales [12] 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 by Pagallo et al. [30] is a decision-tree based feature construction algorithm (it adaptively enlarges the initial attribute set using NOT and AND logical operators for learning DNF concepts). CITRE [25] and DC Fringe [43] combine existing attributes using conjunction and disjunction operators to construct new features. FICUS [24] generalizes previous approaches to allow combining existing features by some user-predefined function. Garcia et al. [10] create a fuzzy rule-based feature construction approach. Another line of research uses rules as local patterns to form global prediction models [11, 5] or to rectify predictions of existing classification algorithms [27].

Subgroups have been used as local patterns to build a global regression model [13], as dummy variables to improve regression fit [7] and as local patterns to understand the behaviour of spammers in a classification use-case [3].

Redescription mining [33] aims to find subsets of entities that can be described in multiple ways (re-described), discovering in that manner strong, equivalence-like relations between different subsets of attributes.

3 The DAFNE framework

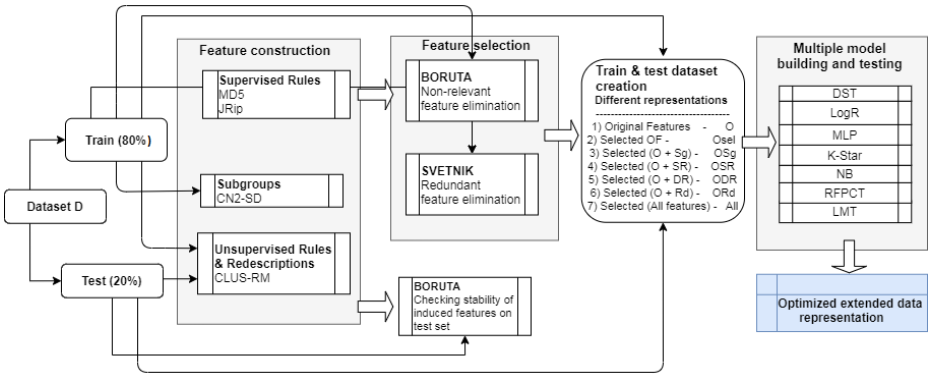

Fig. 1: DAFNE, framework for automated feature construction and evaluation.

The `DAFNE` framework (Figure 1) takes a standard, tabular dataset representing some kind of classification problem, creates a stratified split to train (80%) and test (20%) dataset. Train set is used to create supervised rules using state of the art algorithms `JRip` and `PART` implemented in Weka [8], subgroups are created using the well known `CN2-SD` algorithm [20]. Descriptive (unsupervised) rules and redescriptions are created using the state of the art redescription mining algorithm CLUS-RM [26] on the entire dataset. The CLUS-RM algorithm does not require knowledge about target labels and can thus utilize all entities to create rules and redescriptions. Descriptive rules are obtained as a by-product of redescription mining, these are actually query candidates for redescriptions. The fact that in many applications rule-pairs match accurately for groups with homogeneous target label (since these share many common properties) and the fact that CLUS-RM aims to find pairs of rules that describe common subsets of entities, led us to believe that this process might create useful unsupervised rules and redescriptions for classification tasks. After descriptive objects (rules, subgroups and redescriptions) have been obtained, the tool creates Binary features representing each obtained object and enriches the attribute set of both train and test data. The Boruta framework [18], which utilizes random forest of decision trees, is used to detect a subset of provably useful attributes to predict the given target label on the train set (these features will be used in further evaluation). In this evaluation setting, we also apply Boruta to obtain provably important set of features on the test set (users can observe changes in percentages of important attributes for different types of objects compared to train set). Selected set of provably useful features on a train set is further reduced using the feature selection approach proposed by Svetnik et al. [35] which returns

the non-redundant set of features useful to predict the target label. To analyse the usefulness of different types of objects, `DAFNE` creates a train/test dataset containing: a) all original features (O), b) all non-redundant provably useful original features (O_{sel}), c) all non-redundant provably useful features (All_{sel}), d) non-redundant provably useful original and features obtained from supervised rules (OSR_{del}), e) non-redundant provably useful original and features obtained from descriptive rules (ODR_{sel}), f) non-redundant provably useful original and features obtained from subgroups (OSg_{sel}), g) non-redundant provably useful original and features obtained from redescriptions (ORd_{sel}). DAFNE further trains each of the 8 different types of classification capable machine learning algorithms: multilayer perceptron, $J48$, Decision Stump, Naive Bayes, Logistic Model Trees, Logistic Regression and KStar available in Weka [8] and a Random Forest of 600 Predictive Clustering trees (PCTs) trained using the CLUS framework [4]. Trained models are evaluated on a test set and all constructed, selected features, model evaluation results and analyses are returned to the user. The optimized, extended feature set (`Optimized extended data representation` in Fig. 1) can be used to produce predictive models with improved performance and/or use these new features for better interpretation and understanding of the data and the problem domain.

The feature evaluation procedure performed by DAFNE is rigorous. From provably important Boruta computed features to non-redundant set of features and finally evaluation of selected features using different types of classifiers. It is well known that adding useless features reduces classification accuracy of many types of classification algorithms, thus newly constructed features on a train set must be predictive in order to increase classifier score on a separate test set. To further ensure that increase in classifier score is achieved only due to newly constructed features or their synergy with original features, default parameters are used to train all 8 classification algorithms. Using default parameters also greatly reduces the execution time of feature evaluation. Parameters of each classification algorithm would need to be tuned for every of 6 newly created datasets which is unfeasible for large scale experimentation.

If classification performance of one or more classification algorithms is increased on a test set compared to using only original features, DAFNE has achieved the goal of detecting predictive features. Since using supervised rules as features is known to improve classification accuracy and there exists use-cases where using subgroups is beneficial as well in this setting, we aim to broaden the evaluation of subgroups to more different datasets, investigate the use of descriptive rules and redescriptions and to evaluate the effects of synergy of this objects on classification accuracy.

3.1 Parameters used in DAFNE components

We fixed the parameters of DAFNE components as follows:

- `JRip` - default options, with minimal weight of entities per rule set to 1.0, 500 batch optimization runs and batch size of 200. Changes compared to defaults were made to obtain larger number of rules.

- `PART` - default options with a constraint of minimal 10 entities per rule.
- `CN2-SD` - default options (8 iterations, beam size of 5 and $\gamma = 0.7$).
- `CLUS-RM` - default options (redescription accuracy of 0.6) with 20 random runs, 10 iterations per run, tree depth of 8, using conjunctive refinement procedure [26], conjunction, disjunction and negation operator, support size in $[10, 0.8|E|]$, maximal redescription p-value of 0.01 and output non-redundant redescription set of maximal size 1000 [26]. The main aim is to increase the number of produced redescriptions. Maximal support enables pruning uninteresting redescriptions and tautologies and is often set to $0.8 \cdot |E|$.

Rules and subgroups are filtered to eliminate redundant objects, subgroups must have p-value ≤ 0.01. Fine-grained selection was performed by the Boruta [18] and the feature selection approach by Svetnik et al. [35]. We use default options for Boruta as suggested in [18] and increase the number of trees in a forest to 2000 as suggested in [6]. For non-redundant feature selection [35], we use default parameters as these are well justified in the `varSelRF` R package.

Classification algorithms were trained with default Weka options with maximal number of iterations of Logistic regression classifier set to 10000 to disallow lengthy executions. The random forest of PCTs contains 600 trees with standard $\sqrt{|\mathcal{A}|} + 1$ number of random subspaces. Maximal tree depth of 8 is used.

3.2 Use case scenario

DAFNE is constructed to tackle realistic problems in which one data table ($\mathcal{D}_{train}$) with target labels is available. Also, obtaining additional data table ($\mathcal{D}_{test}$) without target labels is possible (through data collection, domain-level experimentation or similar). The task is to predict the target label y for instances in $\mathcal{D}_{test}$. When both data tables are available, the DAFNE uses $\mathcal{D}_{train}$ to create supervised rules and subgroups and $\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{test}$ to create unsupervised rules and redescriptions. Notice that it is possible to iteratively extend the feature set with newly constructed unsupervised rules and redescriptions for any consecutive test set. DAFNE simulates this process by dividing the annotated data into artificial train and test set, performing the aforementioned feature construction procedure and evaluating newly constructed features using several machine learning algorithms and the target labels for $\mathcal{D}_{test}$, which were not used during any step of rule or model creation. The fact that DAFNE can utilize knowledge available in the test set is considered to be a significant advantage compared to majority of other state of the art feature construction approaches.

4 Data description

We used 5 datasets: Abalone, Arrhythmia, Breast cancer, Wine and Sports Articles, downloaded from the UCI Machine learning repository [38], to evaluate the proposed methodology. We removed rows containing missing values in all datasets since these are not supported by Boruta and tested DAFNE on the Arrhythmia dataset in the original multi-class and the derived binary classification setting. The binary setting simply predicts existence of arrhythmia (yes/no).

5 Experiments and results

DAFNE was run 40 times on each dataset to obtain statistics about the usefulness of supervised/descriptive rules, subgroups and redescriptions to predict a target concept on each of the aforementioned 5 datasets. Supervised rules and subgroups are created using the same seed every time, effectively returning the same set of rules and subgroups, redescriptions and descriptive rules change at each run. Thus, we assess what is the overall change of the system depending on the introduced descriptive rules and redescriptions.

Median percentage of selected rules, subgroups and redescriptions on a train and test set by Boruta is reported in Table 1. This table also contains the number of times (out of 40) at least one member of the object type was found in the non-redundant set of features. In Table 2, we report the median and maximal $AUPRC$ measure [34] for each classifier on each dataset. We underline the original or selected original features if they lead to the best performance of a given model or boldface every combination of features that allow this model to outperform identical model using original or selected original set of features. We also boldface the maximal $AUPRC$ score if the model achieves the same maximal score as using original features but there exist runs where using newly constructed features allowed outperforming a model trained only on original features or using newly constructed features in synergy with original features allows obtaining the same (maximal) result. If maximal result is achieved utilizing only original features, the result is not displayed in boldface.

Results presented in Table 1 suggest that Boruta found that high percentage of created subgroups are significant, followed by supervised rules, redescriptions and descriptive rules. This is the expected trend since subgroups and supervised rules utilize target label information during creation. It is important to notice that both redescriptions and descriptive rules are deemed important on majority of datasets and that there are representatives of these objects in the non-redundant sets of features used to train and evaluate classification models. Boruta also determined that large number of objects, found important on the train set, remains important for the prediction of target label on the test set.

Table 1: Median percentages of supervised rules (SR), descriptive rules (DR), subgroups (Sg) and redescriptions (Rd) deemed provably important for predicting the target class by the Boruta approach on the train (Tr) and test (Ts) set obtained from each dataset. *Num. nn.* reports numbers of runs in which at least one subgroup, supervised rule, descriptive rule and redesription occurred in the non-redundant set of features used for training/testing of different classifiers.

$\mathcal{D}$	Sg_{Tr}	SR_{Tr}	DR_{Tr}	Rd_{Tr}	$Num.nn$	Sg_{Ts}	SR_{Ts}	DR_{Ts}	Rd_{Ts}
Abalone	85	42	11	34	32/12/23/9	49	7	3	3
Arrhythmia	100	71	2	< 1	40/40/22/3	33	0	0	0
Arrhythmia$_b$	100	69	1	0	40/37/0/0	50	0	0	0
Breast cancer	100	83	5	13	40/37/2/1	100	33	3	11

Wine	100	100	19	27	40/38/30/20	100	100	10	27
Sports articles	100	61	1	1	40/40/2/1	33	11	1	1

Results presented in Table 2 show that newly created features improve performance (or allow obtaining the same maximal performance) of every of the 8 chosen classification models on at least 3 different datasets. Results confirm that subgroups seem to be the most important features, however other types of objects have a very important role as well. It is evident that using descriptive rules and redescriptions can significantly increase classifier performance. For example, the Decision Stump model has achieved the best performance using redescriptions as features on the Breast Cancer dataset or using descriptive rules on Arrhythmia dataset. If there existed supervised rules or subgroups more useful to predict the target label, these would surely be chosen instead by the feature selection procedure. Also, if there existed supervised, descriptive rules or subgroups with similar predictive power as redescriptions on Breast Cancer dataset, these would be used in at least some of the runs (where the predictive redescriptions were not present) to obtain similar predictive performance. Redescriptions and descriptive rules can also improve performance of complex classifiers such as Multilayer perceptron, Logistic model trees, Decision trees and Random Forest of PCTs. Thus, synergy or complementarity of different types of objects has played an important role, as noticeable from the results (there are instances where using selected subset from the set of all features yields the highest score, e.g Arrhythmia with binary class label).

Table 2: Evaluation results of 8 selected classifiers.

$\mathcal{M}$	$\mathcal{D}$	O	O_{sel}	All_{sel}	OSg_{sel}	OSR_{sel}	ODR_{sel}	ORd_{sel}
MLP	AB	0.203	0.179	0.194	0.195	0.179	0.194	0.183
		0.203	0.202	**0.223**	**0.233**	**0.217**	**0.219**	**0.227**
	AR	0.668	0.608	0.653	**0.691**	0.569	0.594	0.611
		0.668	0.722	**0.733**	**0.729**	**0.738**	0.722	0.722
	AR_B	0.798	0.508	**0.825**	**0.822**	0.659	0.508	0.508
		0.798	0.526	**0.849**	**0.849**	0.694	0.526	0.526
	BC	0.984	0.966	0.911	0.939	0.956	0.966	0.966
		0.984	0.983	0.965	0.968	0.977	0.982	0.983
	W	1.0	0.905	0.988	0.927	0.923	0.968	0.905
		1.0	1.0	**1.0**	0.989	0.994	**1.0**	**1.0**
	SA	0.868	0.855	0.788	0.806	0.842	0.855	0.855
		0.868	0.892	0.848	0.855	0.877	0.892	0.892
LMT	AB	0.217	0.217	0.206	0.207	0.217	0.212	0.217
		0.217	0.217	**0.229**	**0.230**	**0.225**	**0.229**	**0.223**
	AR	0.788	0.717	0.718	0.715	0.682	0.714	0.717
		0.788	0.796	0.780	0.793	0.789	0.792	0.796
	AR_B	0.841	0.662	**0.859**	0.835	0.666	0.662	0.662
		0.841	0.662	**0.870**	**0.846**	0.726	0.662	0.662
	BC	0.992	0.967	0.948	0.939	0.969	0.967	0.967
		0.992	0.982	0.958	0.939	0.976	0.983	0.982
	W	1.0	0.928	0.948	0.890	0.887	0.985	0.952
		1.0	1.0	0.948	0.995	0.981	**1.0**	**1.0**
	SA	0.884	0.855	**0.888**	0.866	0.868	0.855	0.855
		0.884	0.881	**0.902**	**0.895**	**0.896**	0.881	0.881

Table 2: Evaluation results of 8 selected classifiers.

$\mathcal{M}$	$\mathcal{D}$	O	O_{sel}	All_{sel}	OSg_{sel}	OSR_{sel}	ODR_{sel}	ORd_{sel}
	AB	0.160	0.160	0.160	**0.160**	0.160	0.160	0.160
		0.160	0.168	0.167	0.167	0.168	0.167	0.168
NB	AR	0.472	0.645	**0.713**	**0.714**	**0.656**	**0.647**	0.643
		0.472	0.715	**0.753**	**0.758**	**0.729**	0.708	0.715
	AR_B	0.719	0.617	**0.863**	**0.846**	0.719	0.617	0.617
		0.719	0.618	**0.872**	**0.858**	**0.720**	0.618	0.618
	BC	0.983	0.966	0.970	0.971	0.970	0.966	0.966
		0.983	0.979	0.974	0.976	0.978	0.979	0.979
	W	0.991	0.901	0.976	0.945	0.945	0.975	0.920
		0.991	1.0	**1.0**	0.998	0.998	**1.0**	1.0
	SA	0.845	0.819	0.838	0.828	0.830	0.819	0.819
		0.845	0.827	**0.855**	0.837	0.841	0.827	0.827
	AB	0.08	0.08	0.08	0.08	0.08	0.08	0.08
		0.08	0.08	0.08	0.08	0.08	0.08	0.08
	AR	0.163	0.163	0.162	0.162	0.138	0.163	0.163
		0.163	0.163	0.162	0.162	0.138	**0.20**	0.163
	AR_B	0.518	0.551	**0.682**	**0.682**	**0.620**	0.551	0.551
		0.518	0.551	**0.682**	**0.682**	**0.620**	0.551	0.551
DSt	BC	0.816	0.816	**0.839**	**0.839**	0.790	0.816	0.816
		0.816	0.864	0.839	0.839	0.790	**0.867**	**0.881**
	W	0.570	0.570	**0.604**	**0.604**	**0.630**	**0.605**	0.570
		0.570	0.610	**0.630**	**0.630**	**0.630**	**0.667**	**0.667**
	SA	0.724	0.724	0.722	0.722	0.724	0.724	0.724
		0.724	0.724	0.722	0.722	**0.724**	**0.724**	**0.724**
	AB	0.199	0.199	0.197	0.199	0.199	0.199	0.199
		0.199	0.210	0.210	0.210	0.210	0.210	0.210
	AR	0.40	0.707	0.647	0.640	0.660	0.707	0.707
		0.40	0.776	0.729	0.719	0.734	0.776	0.776
	AR_B	0.678	0.508	**0.857**	**0.835**	0.666	0.508	0.508
		0.678	0.520	**0.868**	**0.859**	**0.701**	0.520	0.520
$LogR$	BC	0.985	0.965	0.894	0.962	0.890	0.965	0.965
		0.985	0.986	0.952	0.974	0.962	**0.986**	0.986
	W	0.995	0.910	0.966	0.942	0.926	0.938	0.914
		0.995	1.0	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
	SA	0.883	0.867	**0.890**	**0.886**	0.865	0.865	0.867
		0.883	0.882	**0.90**	**0.891**	**0.903**	0.882	0.882
	AB	0.187	0.187	0.173	0.173	1.0	0.187	0.187
		0.187	0.194	0.191	0.192	0.194	0.193	0.194
	AR	0.468	0.50	**0.571**	**0.568**	**0.513**	**0.510**	0.50
		0.468	0.590	**0.644**	**0.640**	**0.607**	0.590	0.590
	AR_B	0.577	0.634	**0.814**	**0.792**	**0.737**	0.634	0.634
		0.577	0.634	**0.849**	**0.821**	**0.745**	0.634	0.634
KS	BC	0.975	0.966	0.969	0.970	0.968	0.966	0.966
		0.975	0.981	0.973	0.970	0.968	0.966	0.966
	W	0.980	0.878	**0.981**	0.946	0.955	**0.988**	0.926
		0.980	1.0	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
	SA	0.792	0.796	0.784	0.770	**0.814**	0.796	0.96
		0.792	0.837	0.831	0.813	**0.862**	0.837	0.837
	AB	0.10	0.10	**0.107**	**0.102**	0.10	**0.101**	0.10
		0.10	0.129	0.129	0.129	0.129	0.129	0.129
	AR	0.543	0.331	0.396	0.40	0.351	0.344	0.331
		0.543	0.542	0.527	0.530	0.433	**0.573**	0.542
	AR_B	0.697	0.575	**0.727**	**0.737**	0.658	0.575	0.575
		0.697	0.575	**0.730**	**0.754**	0.677	0.575	0.575
J_{48}	BC	0.875	0.816	0.844	0.853	0.837	0.816	0.816
		0.875	0.922	0.853	0.853	**0.934**	0.922	**0.960**
	W	0.95	0.738	0.880	0.886	0.880	0.852	0.781
		0.95	0.954	0.886	0.886	0.880	**1.0**	**0.963**
	SA	0.664	0.739	**0.793**	**0.777**	**0.785**	0.739	0.739
		0.664	0.773	**0.849**	**0.792**	**0.807**	0.773	0.773
	AB	0.180	0.176	0.174	0.173	0.176	0.176	0.176
		0.180	0.176	**0.194**	**0.185**	**0.199**	**0.196**	**0.189**
	AR	0.767	0.612	0.672	0.681	0.577	0.606	0.612
		0.767	0.728	0.747	0.744	0.707	0.734	0.728
	AR_B	0.826	0.633	0.812	0.791	0.752	0.633	0.633

Table 2: Evaluation results of 8 selected classifiers.

$\mathcal{M}$	$\mathcal{D}$	O	O_{sel}	All_{sel}	OSg_{sel}	OSR_{sel}	ODR_{sel}	ORd_{sel}
RF^{600}_{PCT}	BC	0.826	0.633	**0.830**	0.825	0.765	0.633	0.633
		0.988	0.956	0.964	0.960	0.968	0.956	0.956
		0.988	0.977	0.973	0.966	0.976	0.976	0.977
	W	1.0	0.850	0.991	0.928	0.923	0.986	0.895
		1.0	1.0	**1.0**	0.998	0.998	**1.0**	**1.0**
	SA	0.938	0.827	0.868	0.827	0.860	0.827	0.827
		0.938	0.867	0.875	0.852	0.880	0.867	0.867

6 Conclusion and future work

In this work we created several types of interpretable data models to improve representations of tabular data. A new feature construction and evaluation framework DAFNE includes a set of feature generating algorithms, producing supervised and unsupervised rules, subgroups and redescriptions, and advanced feature selection methodology to construct relevant and non-redundant feature sets. These are used to extend original problem representation with new interpretable and informative features for downstream supervised tasks. Evaluation results across 5 different datasets confirmed benefits of using supervised rules as features in classification tasks and showed that subgroups represent highly relevant features across tested datasets. Our study also shows that rules and redescriptions, constructed in a specific unsupervised manner, can form informative features increasing performance of various classification algorithms. Finally, the synergy of different types of features often allowed increasing classification performance compared to the original representation. Future work includes evaluating DAFNE on more challenging datasets or tasks and comparing against the state-of-the-art self-supervised learning frameworks for learning useful new representations for tabular data.

Acknowledgement

The authors acknowledge support by the Research Cooperability Program of the Croatian Science Foundation, funded by the European Union from the European Social Fund under the Operational Programme Efficient Human Resources 2014-2020, through the Grant 8525: Augmented Intelligence Workflows for Prediction, Discovery, and Understanding in Genomics and Pharmacogenomics.

References

1. Altmann, A., Toloşi, L., Sander, O., Lengauer, T.: Permutation importance: a corrected feature importance measure. Bioinformatics 26(10), 1340–1347 (2010)
2. Arik, S.O., Pfister, T.: Tabnet: Attentive interpretable tabular learning. AAAI 35(8), 6679–6687 (2021)
3. Atzmueller, M., Lemmerich, F., Krause, B., Hotho, A.: Towards understanding spammers–discovering local patterns for concept description. In: LeGo ECML/PKDD Workshop (2009)

4. Blockeel, H., Raedt, L.D., Ramon, J.: Top-down induction of clustering trees. In: ICML. pp. 55–63. Morgan Kaufmann (1998)
5. Dembczyński, K., Kotłowski, W., Słowiński, R.: A general framework for learning an ensemble of decision rules. In: LeGo ECML/PKDD Workshop (2008)
6. Díaz-Uriarte, R., Alvarez de Andrés, S.: Gene selection and classification of microarray data using random forest. BMC Bioinformatics 7(1), 3 (2006)
7. Duivesteijn, W., Feelders, A.J., Knobbe, A.: Exceptional model mining. Data Mining and Knowledge Discovery 30(1), 47–98 (2016)
8. Eibe, F., Hall, M.A., Witten, I.H.: The weka workbench. online appendix for data mining: practical machine learning tools and techniques. In: Morgan Kaufmann. Morgan Kaufmann Publishers (2016)
9. Galbrun, E., Miettinen, P.: Redescription Mining. Springer Briefs in Computer Science, Springer (2017)
10. García, D., Stavrakoudis, D., González, A., Pérez, R., Theocharis, J.B.: A fuzzy rule-based feature construction approach applied to remotely sensed imagery. In: IFSA-EUSFLAT. Atlantis Press (2015)
11. Giacometti, A., Miyaneh, E.K., Marcel, P., Soulet, A.: A generic framework for rule-based classification. LeGo ECML/PKDD Workshop pp. 37–54 (2008)
12. Gomez, G., Morales, E.F.: Automatic feature construction and a simple rule induction algorithm for skin detection. In: ICML Workshop on Machine Learning in Computer Vision. pp. 31–38 (2002)
13. Grosskreutz, H.: Cascaded subgroups discovery with an application to regression. In: ECML/PKDD. vol. 5211, p. 33 (2008)
14. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of machine learning research 3, 1157–1182 (2003)
15. Hapfelmeier, A., Ulm, K.: A new variable selection approach using random forests. Computational Statistics & Data Analysis 60, 50 – 69 (2013)
16. 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)
17. Herrera, F., Carmona, C.J., González, P., del Jesus, M.J.: An overview on subgroup discovery: foundations and applications. Knowl. Inf. Syst. 29(3), 495–525 (2011)
18. Kursa, M.B., Jankowski, A., Rudnicki, W.R.: Boruta–a system for feature selection. Fundamenta Informaticae 101(4), 271–285 (2010)
19. Langley, P., Bradshaw, G.L., Simon, H.A.: Rediscovering Chemistry with the Bacon System, pp. 307–329. Springer Berlin Heidelberg, Berlin, Heidelberg (1983)
20. Lavrac, N., Kavsek, B., Flach, P., Todorovski, L.: Subgroup discovery with cn2-sd. J. Mach. Learn. Res. 5(2), 153–188 (2004)
21. Liu, H., Motoda, H., Yu, L., Ye, N.: Feature extraction, selection, and construction. The handbook of data mining pp. 409–424 (2003)
22. Liu, H., Motoda, H.: Feature extraction, construction and selection: A data mining perspective, vol. 453. Springer Science & Business Media (1998)
23. 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)
24. Markovitch, S., Rosenstein, D.: Feature generation using general constructor functions. Machine Learning 49(1), 59–98 (2002)
25. Matheus, C.J., Rendell, L.A.: Constructive induction on decision trees. In: IJCAI - Volume 1. pp. 645–650. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1989)

26. Mihelčić, M., Džeroski, S., Lavrač, N., Šmuc, T.: A framework for redescription set construction. Expert Systems with Applications 68, 196 – 215 (2017)
27. Mozina, M., Bratko, I.: Rectifying predictions of classifiers by local rules. In: LeGo ECML/PKDD Workshop (2008)
28. 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)
29. Oglic, D., Gärtner, T.: Greedy feature construction. In: NIPS, pp. 3945–3953. Curran Associates, Inc. (2016)
30. Pagallo, G.: Learning dnf by decision trees. In: IJCAI - Volume 1. pp. 639–644. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1989)
31. Pagallo, G.M.: Adaptative decision tree algorithms for learning from examples (ph.d. thesis). Tech. rep., Santa Cruz, CA, USA (1990)
32. Ragavan, H., Rendell, L.A.: Lookahead feature construction for learning hard concepts. In: ICML. pp. 252–259. Morgan Kaufmann Publishers Inc. (1993)
33. Ramakrishnan, N., Kumar, D., Mishra, B., Potts, M., Helm, R.F.: Turning cartwheels: An alternating algorithm for mining redescriptions. In: KDD. pp. 266–275. ACM, New York, NY, USA (2004)
34. 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)
35. Svetnik, V., Liaw, A., Tong, C., Wang, T.: Application of breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules. In: Multiple Classifier Systems. pp. 334–343. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
36. 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)
37. Ucar, T., Hajiramezanali, E., Edwards, L.: Subtab: Subsetting features of tabular data for self-supervised representation learning. In: NeurIPS. pp. 18853–18865 (2021)
38. UCI: Uci machine learning repository (Last access: 05/07/2022), `https://archive.ics.uci.edu/ml/index.php`
39. Van Der Maaten, L., Postma, E., Van den Herik, J.: Dimensionality reduction: a comparative review. Journal of machine learning research 10, 66–71 (2009)
40. Vens, C., Costa, F.: Random forest based feature induction. In: Cook, D.J., Pei, J., Wang, W., Zaïane, O.R., Wu, X. (eds.) ICDM. pp. 744–753. IEEE Computer Society (2011)
41. Wang, M., Chen, X., Zhang, H.: Maximal conditional chi-square importance in random forests. Bioinformatics 26 6, 831–7 (2010)
42. Wrobel, S.: An algorithm for multi-relational discovery of subgroups. In: Komorowski, H.J., Zytkow, J.M. (eds.) PKDD. Lecture Notes in Computer Science, vol. 1263, pp. 78–87. Springer (1997)
43. Yang, D.S., Rendell, L., Blix, G.: A scheme for feature construction and a comparison of empirical methods. In: IJCAI - Volume 2. pp. 699–704. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1991)
44. Zheng, Z.: Constructing nominal x-of-n attributes. In: IJCAI - Volume 2. pp. 1064–1070. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1995)
45. Zhou, Z., Feng, J.: Deep forest: Towards an alternative to deep neural networks. In: Sierra, C. (ed.) IJCAI. pp. 3553–3559. ijcai.org (2017)