

Multi-objective learning of accurate and comprehensible classifiers – a case study

Rok PILTAVER^{a,b}, Mitja LUŠTREK^a and Matjaž GAMS^{a,b}

^a*Jožef Stefan Institute - Department of Intelligent Systems, Ljubljana, Slovenia*

^b*Jožef Stefan International Postgraduate School, Ljubljana, Slovenia*

Abstract. Accuracy and comprehensibility are two important classifier properties, however they are typically conflicting. Research in the past years has shown that Pareto-based multi-objective approach for solving this problem is preferred to the traditional single-objective approach. Multi-objective learning can be represented as search that starts either from an accurate classifier and modifies it in order to produce more comprehensible classifiers (e.g. extracting rules from ANNs) or the other way around: starts from a comprehensible classifier and modifies it to produce more accurate classifiers. This paper presents a case study of applying a recent algorithm for multi-objective learning of hybrid trees MOLHC in human activity recognition domain. Advantages of MOLHC for the user and limitations of the algorithm are discussed on a number of datasets from the UCI repository.

Keywords. Multi-objective learning, hybrid classifier, hybrid tree, accuracy, comprehensibility.

Introduction

When evaluating a classifier, one is usually most interested in its predictive accuracy estimated by e.g. percent of correctly classified instances, confusion matrix, area under ROC, or other measures. However, there are also other classifier properties that are often important for the user: comprehensibility [1] – also referred to as understandability or interpretability – justifiability [2, 3], surprisingness [4], and others. This paper is limited to discussing accuracy and comprehensibility.

The comprehensibility is defined as: “the ability to understand the output of induction algorithm” [5] or “the ability to understand the logic behind a prediction of the model” [6]. According to Craven and Shavlik [7] it is important because it enables: classification explanation, classifier validation, knowledge discovery and supports classifier generalization improvement and refinement of approximately-correct domain theories. Furthermore, there are many application domains in which the importance of comprehensible classification models continues to be emphasized, such as: medicine, credit scoring, churn prediction, and bioinformatics [1].

The main problem in learning accurate and comprehensible classifiers is that the two objectives are conflicting [8]. There are two main approaches to solving this problem [8, 9]. The weighted-formula approach is conventional; it transforms the multi-objective problem into a single-objective one. The second approach is Pareto-based multi-objective approach. Its objective function is no longer a scalar value, but a vector so all the criteria are treated separately. This produces a number of Pareto-optimal solutions [10] (i.e. classifiers) instead of a single solution. Freitas [9] lists

arguments for and against each approach and concludes that the more complex Pareto-based approach is preferred because it avoids multiple runs of single-objective optimisation algorithm and the ad-hoc specification of its parameters (i.e. weights) as well as provides very informative set of non-dominated solutions [10]. Nevertheless, depending on the application domain, there are cases in which the weighted-formula approach is sufficient or the Pareto-based approach is too complex.

To learn accurate and comprehensible classifier an algorithm can start with an accurate classifier and transform it to produce more comprehensible ones. Examples of such approaches are extracting rules from artificial neural networks (ANN) [8] and pruning decision trees to find a trade-off between their size – which is related to its comprehensibility – and accuracy [11]. The search can also proceed inversely: start from a comprehensible classifier and transform it to produce more accurate classifiers. An example of such algorithm is the recently presented multi-objective learning of hybrid classifiers (MOLHC) algorithm [12], which is guaranteed to find the entire Pareto set of hybrid trees by replacing sub-trees in the initial classification tree with black-box classifiers (e.g. SVM, ANN, or random-forest).

This paper presents a case study of applying the MOLHC algorithm in human activity recognition domain: motivation for learning hybrid classifiers and the insights in the classification task provided by the visualization of the algorithm's output. Limitations and performance of MOLHC algorithm are discussed on a number of datasets from the UCI repository [13, 14].

The structure of the paper is as follows: Section 1 gives a quick overview of the MOLHC algorithm, Section 2.1 introduces the activity recognition domain used for the case study, Section 2.2 illustrates the use of the algorithm, its advantages and drawbacks, and finally Section 3 summarizes the paper and suggests directions for further research.

1. The multi-objective learning of hybrid classifiers algorithm

The basic idea of the multi-objective learning of hybrid classifiers (MOLHC) algorithm [12] is to replace sets of leaves in a given comprehensible classification tree with black-box (BB) leaves that invoke a provided accurate BB classifier in order to increase the accuracy of the resulting hybrid trees compared to the initial tree. The algorithm is motivated by the fact that many machine learning domains as well as human expert knowledge can be partially explained with simple models (e.g. rules) but require much more complex and less comprehensible model in other parts. The algorithm guarantees to find the complete Pareto set of described hybrid trees efficiently: it outperforms a state of the art multi-objective optimisation algorithm NSGA-II [17] for the discussed task in terms of run-time, and in contrast with NSGA-II guarantees finding the complete Pareto set (i.e. is not stochastic) and does not require setting any search parameters [12]. It has been shown to produce sets of hybrid trees that considerably outperform the baseline algorithms (classification tree and BB classifier) in terms of hyper-volume under the attainment surface in many domains [12]. In simple words: the set of hybrid trees produced by MOLHC algorithm consists of classifiers that cannot be constructed with the baseline algorithms and offer useful trade-off between accuracy and comprehensibility.

In contrast with most related work MOLHC does not use the size of classification tree as a measure of comprehensibility because it operates with hybrid trees. Instead,

the comprehensibility c of a hybrid tree is defined by Eq. 1 as the ratio between the number of examples that are classified by the regular leaves i and the number of all examples N used to evaluate the comprehensibility of the hybrid tree.

$$c = (\sum_{i \text{ is non-replaced leaf}} N_i) / N \quad (1)$$

The comprehensibility of a hybrid tree is therefore equal to the probability of classifying an instance with a comprehensible model. By definition, the comprehensibility of the initial classification tree is 1. A classification tree is considered as perfectly comprehensible regardless of its size; however it is only sensible to use the measure and the algorithm on reasonably small classification trees, i.e. with less than ~ 50 leaves. Comprehensibility of the BB classifier is 0, meaning that it is not comprehensible at all. The comprehensibility of all other hybrid trees are between 0 and 1.

A naïve approach to finding the Pareto set of hybrid trees would be to generate, evaluate and compare all the possible hybrid trees. This yields a search space with 2^n hybrid trees where n is the number of leaves in the initial classification tree that are considered to be replaced with BB leaves. Only the leaves in which BB classifier achieves higher accuracy than the majority class classifier belonging to the leaf should be considered for replacing with BB leaves - replacing leaves for BB decreases comprehensibility and is therefore only worthwhile if it increases accuracy at the same time. The MOLHC examines the search space using an iterative search methods that avoids generating most hybrid trees not belonging to the Pareto set. The main loop of the algorithm considers a set of hybrid trees that have the same number of replaced leaves. It starts with hybrid tree that has zero replaced leaves (only the initial tree belongs to this set) and increases until it has considered replacing all the leaves. When considering a hybrid tree (from the set of hybrid trees processed in the current iteration), it produces a set of new hybrid trees that can be generated from the given hybrid tree by replacing exactly one non-dominated leaf. The set of non-dominated leaves L_n is defined by Eq. 2 and contains the leaves l that are better than all the other currently non-replaced leaves L considering the difference in accuracy a_l and comprehensibility c_l introduced by replacing the leaf for the BB – they increase accuracy more and at the same time decrease comprehensibility less than other leaves.

$$L_n = \{ l \in L; \forall i \in L: (a_l \geq a_i) \wedge (c_l \leq c_i) \} \quad (2)$$

Replacing only the non-dominated leaves limits the search but has been proven to find the complete Pareto set of hybrid trees [12]. The search optimisation enables exact multi-objective learning based on initial trees with less than ~ 50 leaves in under a second on a personal computer (e.g. 3 GHz Intel® Core™ 2 Duo) [12]. For comparison consider that a naïve algorithm takes over 3 minutes for an initial tree with 17 leaves and 18 minutes for 18 leaves; bigger trees cannot be used with the naïve algorithm as its time complexity increases exponentially with the number of leaves [12].

2. Case study of multi-objective learning of hybrid classifiers algorithm

In order to demonstrate the use of MOLHC algorithm in practice and show the advantages of multi-objective learning and hybrid trees with black-box (BB) leaves this section presents a case study of MOLHC application in activity recognition domain described in the following subsection. In addition we selected the datasets for testing from the set of 94 classification datasets from the UCI repository [13] available in ARFF format at the Weka webpage [14]. Among the 49 datasets with more than 300 instances we chose 23 datasets where the BB classifier achieved at least 10 % better accuracy than the tree with approximately 20 leaves. Finally 40 trees were used to calculate the results shown in Figure 2 and Figure 4: one small tree (~20 leaves) for each of the 23 datasets and another bigger tree (~40 leaves) for 17 dataset that allowed building larger trees. The choice of datasets and initial trees is the same as in [12].

2.1. Motivation for MOLHC application and the case study domain

The goal of activity recognition is to recognize the activity a person is performing using sensors and software for sensor data processing. The task can be to recognize: *basic activities* such as lying, sitting, standing, walking, running, cycling, transitions between activities, etc.; *events* such as falling, sitting down, standing up, stop moving, etc.; or *complex activities* such as performing house chores, preparing meal, eating, exercising, shopping, etc. Sensor data can be obtained from video cameras, real-time locating systems, inertial sensors, or 3D motion capture systems. Activity recognition is an important task in ambient intelligence as it is prerequisite for many applications such as sport applications, health monitoring, smart house automation, and others. This section considers learning a classifier that distinguishes between 10 basic activities (listed in the following paragraph) based on attributes extracted from data provided by a single 3-axis accelerometer mounted on person's chest. The recognized basic activities can then be used to recognize events and complex activities.

The training and testing dataset was recorded in a laboratory with 9 persons each performing a given sequence of activities lasting for approximately 1.5 hours: 22 % of the time lying, 17 % walking, 14 % cycling, 10 % standing, 7-8 % of sitting, kneeling, on all fours, and running (each), 4 % transitions between activities, and 3 % leaning. Two-second time windows of measured accelerations along each of the 3 accelerometer axes was used to compute 61 attributes suggested by literature, for example: mean value, area under the curve, amplitude, total energy, dominant frequency, mean crossing rate, entropy, variation coefficient, etc. The time windows were overlapping (1 second overlap with the previous window and 1 second overlap with the following window) therefore a total of around 48.000 instances were acquired.

We intended to enter the EvAAL live activity-recognition competition (<http://evaal.aaloa.org/>) therefore we required a classifier that we could trust to perform correctly in a situation substantially different from the one in the laboratory. The sequence and duration of activities that would be used for testing at the competition were not known. In addition the placement of the accelerometer could not be guaranteed to be exactly the same as in the laboratory and the motion of the person evaluating the activity recognition system at the competition could be different then the motion recorded in the laboratory, e.g. different posture or intensity of movement. A very accurate (90.6 % classification accuracy) black-box classifier was constructed using high-quality laboratory data, however it did not allow an expert to validate it. On

the other hand, completely comprehensible classifiers performed poorly (77.0 % classification accuracy) in comparison. Hence this was a problem that called for a hybrid approach using MOLHC to generate a set of hybrid classifiers ranging from the most comprehensible to the most accurate in order to get an insight in the classification problem at hand and choose a classifier (hybrid tree) that has high enough accuracy and is as comprehensible as possible.

2.2. MOLHC for activity recognition

The first step in using MOLHC is to **choose an initial classification tree** and a black-box (BB) classifier with high accuracy. First a classification tree was constructed using the original dataset but was difficult to validate. In order to support validation the domain expert choose a subset of 12 attributes that were known to be important for the classification and were easy to interpret. They were used to build a classification tree using C4.5 learning algorithm [15] implemented as J48 in Weka [16]. Its pruning parameters were set so that it produced a tree of appropriate size (12 leaves) and accuracy (76.1 %). The tree is shown in Figure 3 – attributes are renamed and numerical attribute values that split the data into sub-trees are replaced with words in order to improve comprehensibility for readers not familiar with the domain. The size of the initial tree should be small enough to prevent overfitting and enable the expert to analyse it in reasonable time. On the other hand pruning a tree too much will decrease its accuracy. The initial classification tree can also be built by a domain expert based on his knowledge: it should include the rules he knows are valid and possibly some additional rules he suspects are valid in most cases.

To **choose a BB classifier**, several classifiers should be trained using various learning algorithms and learning parameter settings, and compared according to their accuracy estimated on a test set. In our case, random forest classifier was chosen as the BB classifier (90.6 % classification accuracy) based on the experts' past experience with the domain. All the 61 available attributes were used for learning the random forest classifier – there is no point in holding back any data (except redundant and random attributes) from the algorithm that learns the BB classifier. A possible improvement of the algorithm could use multiple BB classifiers: one for each leaf or one for each sub-tree with enough examples to learn an accurate BB classifier.

The second step is the **execution of the MOLHC algorithm**, which uses the following inputs: the initial classification tree, BB classifier, and data that was not used to train the two input classifiers. The output of the algorithm is a Pareto set of hybrid trees, which is represented in the objective space as the Pareto front: a graph with accuracy on one axis, comprehensibility on the other, and points on the graph representing individual hybrid trees (see Figure 1). By analysing the Pareto front and data about the Pareto set, knowledge about the domain can be extracted as illustrated in the following paragraphs.

The first thing to observe is the **steepness of the Pareto front**. A steep Pareto front (e.g. Figure 1a) represents a case in which the difference in accuracy between the initial tree and BB classifier is small. In such cases the user should consider choosing the initial tree because it achieves accuracy similar to the BB classifier but is completely comprehensible as opposed to the BB. On the other hand, a Pareto front that decreases gradually (e.g. Figure 1c) represents a case with considerable difference in accuracy between the initial tree and BB classifier. In such cases the user should investigate the Pareto front further in order to select an appropriate hybrid tree with a

desired trade-off between accuracy and comprehensibility. The steepness of Pareto front corresponds to the difficulty of classification task for a decision tree (given a comprehensible decision tree and an accurate BB classifier). [Figure 1](#) shows that iris data set [13] is easy to classify using a comprehensible classifier, while a simple classifier does not suffice to classify activity or letter datasets [13] with high accuracy.

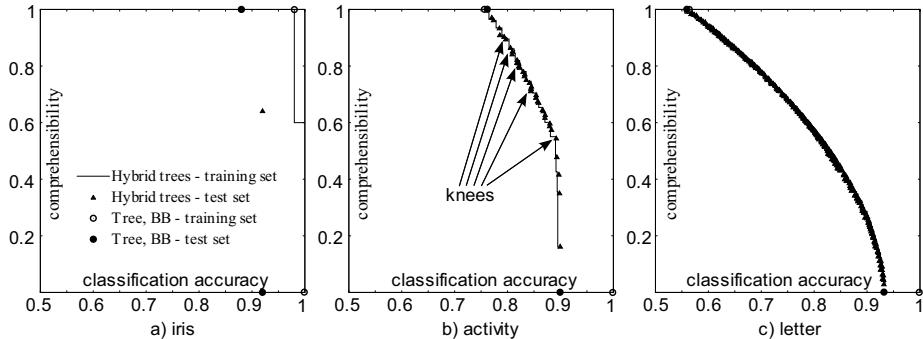


Figure 1. Pareto fronts of hybrid trees for a) iris, b) activity, and c) letter datasets

The second property of the Pareto front to be investigated is the **density and spread of hybrid trees along the Pareto front**. For instance the Pareto front for iris dataset ([Figure 1a](#)) is sparse (includes only two hybrid trees), while the Pareto front for letter dataset ([Figure 1c](#)) is dense (403 hybrid trees). Deb [10] lists several measures of spread, however a threshold between sparse and dense Pareto front depends on the application domain. If there are few hybrid trees on a Pareto front the user can inspect and compare all of them, otherwise he should concentrate on a subset of hybrid trees. Pareto front in [Figure 1c](#) is well spread along the entire range while the Pareto front in [Figure 1b](#) is considerably sparser in the low comprehensibility range. Hybrid trees in the sparse part of the Pareto front should be inspected thoroughly while only a subset of trees needs to be inspected in the dense part since it contains many similar trees – the similarities usually become obvious quickly, however a program with appropriate graphical interface could automate and simplify the task further. [Figure 2](#) shows that the number of hybrid trees and hence the density of Pareto front depends on the number of leaves in the initial tree for which the accuracy of BB classifier is higher than the accuracy of majority class classifier in the leaf. It also depends on the differences in accuracy and comprehensibility that is introduced by replacing each leaf for a BB leaf, which explains the outliers in [Figure 2](#).

The third important property of the Pareto front is **presence of knees**: parts of Pareto front with sudden jump in one of the objectives. Jin [18] argues that quantitative measure for knee should be defined according to the application domain. Example of a knees are shown in [Figure 1b](#); the most obvious one occurs where accuracy approaches 0.9. Knees are important because they limit the set of hybrid trees that needs to be examined by the user. For instance, if a hybrid tree with high accuracy is requested in the activity dataset ([Figure 1b](#)) then the one with accuracy 0.89 and comprehensibility 0.55 is a good candidate (an arrow pointing at it in [Figure 1b](#)). It has almost the same accuracy as the hybrid trees further down the Pareto front but it has considerably higher accuracy. Among the hybrid trees near a knee, the ones that have extreme values of an objective are the most interesting for the user.

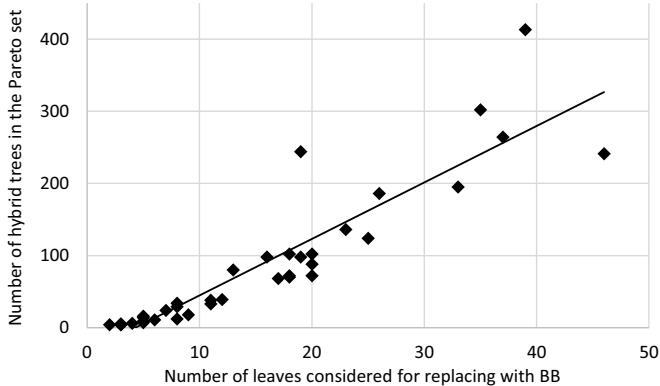


Figure 2. Number of hybrid trees in a Pareto set depends on the number of leaves in the initial tree that are considered for replacing with BB leaves (results obtained from 40 initial trees built on 23 UCI datasets).

Another insight offered by the MOLHC approach is **validation of classification tree leaves**. It is most useful if the initial tree is constructed by an expert (not by a machine learning algorithm) as it validates the experts knowledge and exposes expert's assumptions that are not in line with the provided data; it can be used to validate a learned tree as well. Among the 12 leaves in the initial tree (used for the activity recognition domain) BB achieved higher accuracy in all but the leaf number 8 (Figure 3). Because the BB classifier cannot improve classification accuracy for the instances belonging to that leaf, the user can accept the leaf as valid piece of extracted knowledge. For iris dataset, which can be accurately classified with a classification tree, there were three leaves in the initial tree and BB classifier outperformed them in only one leaf while the other two were confirmed as valid.

Besides checking in which leaves BB classifier achieves higher accuracy than the leaf, the Pareto set of hybrid classifiers is analysed in order to calculate the **relative quality of leaves**. The algorithm counts the number of Pareto optimal hybrid trees in which a leaf was replaced for a BB leaf. If the count for a leaf is low, it means that the leaf is good according to both objectives (accuracy and comprehensibility): it correctly classifies a large share of instances belonging to the leaf and provides classification explanation for a big number of instances. Discretized counts are depicted as stars under each leaf in Figure 3: a high number of black stars represents leaves with good accuracy and comprehensibility and vice versa. Figure 3 shows that leaves with high probability of the class assigned in the leaf (percent of instances belonging to the class are given in each leaf) receive good score, which is to be expected. However, these scores provide additional information: leaves 7 and 10 have similar classification accuracy (~56 %), but leaf 10 has lower score. This means that BB classifier is able to improve the classification accuracy in leaf 10, but not in leaf 7. A quick look at the number of stars in Figure 3 reveals that running, walking, cycling and lying activities are easy to recognize while sitting, kneeling and standing are often confused by the classification tree and are classified with low accuracy. Therefore, they should be replaced by BB classifier in order to improve accuracy, since comprehensible classification is not provided by the initial tree. The domain expert confirmed that additional accelerometer on thigh should be used in order to distinguish sitting from standing. He also confirmed that the suggested four activities are easy to classify: an

older version of activity recognition software that he developed used hand crafted rules, similar to the ones in the tree, to recognize the four activities. This illustrates how the MOLHC supports the knowledge discovery and classifier validation.

The domain expert finally **choose the hybrid tree** (Figure 3), which achieves 84.1 % accuracy and comprehensibility 0.72. By sacrificing some accuracy he obtained an accurate classifier enabling good classifier validation. The chosen hybrid tree replaced a sub-tree (containing leaves 1-3) with a single BB leaf, which increased the overall accuracy by 3.3 % and decreased comprehensibility by 0.13. It also replaced leaves 5 and 10, which increased accuracy by additional 4.7 % and decreased comprehensibility by additional 0.15. The expert could change the initial tree by adding sub-trees to those two leaves and run the MOLHC again if higher comprehensibility and similar accuracy was required. This illustrates how MOLHC supports classifier generalization improvement and refrainment of approximately-correct domain theories.

Since the user chooses a hybrid tree based on the Pareto front, it is very important that the comprehensibility and accuracy values used for drawing the Pareto front are accurate. They are estimated on the training set and therefore depend on the number of training instances as is shown in Figure 4. Insufficient number of training instances may lead to **errors in the estimated comprehensibility and accuracy** and therefore mislead the user while choosing a hybrid tree. Figure 1a shows one such example, which occurred because only 50 instances were used for the iris dataset. The problem is only amplified by the fact that MOLHC approach requires three datasets: one for learning the initial tree and BB classifier, another for learning the Pareto set of hybrid trees and usually also the third for evaluating the hybrid trees. An improvement of the algorithm, which would enable it to perform reliably with small datasets and would limit the errors of the predicted comprehensibility and accuracy of the hybrid trees, would be welcome. It could probably be achieved using internal n-fold cross-validation.

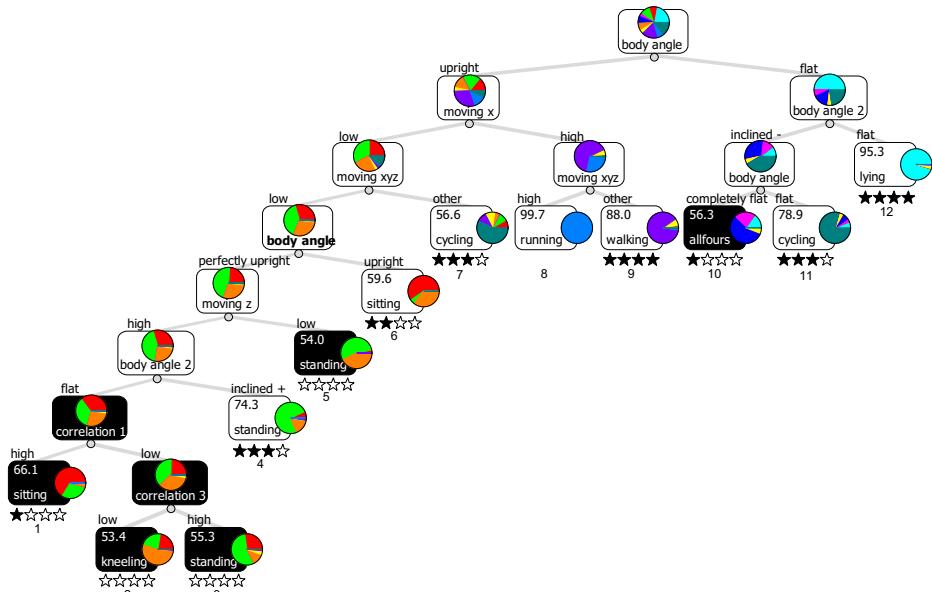


Figure 3. Output of the MOLHC algorithm for the activity recognition domain: quality of the leaves (stars), black-box leaves of the chosen hybrid tree (black leaves), and pie charts representing class distributions in each node.

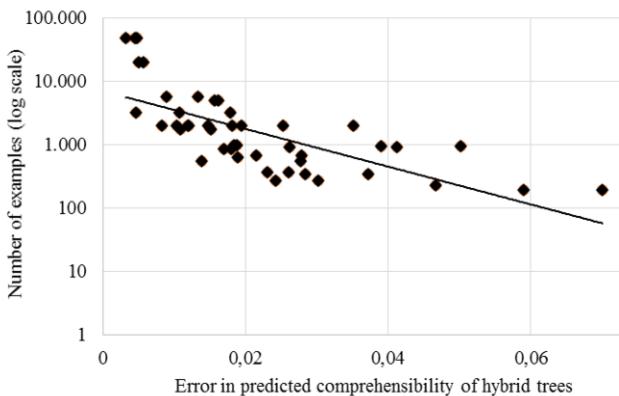


Figure 4. Error in predicted comprehensibility of hybrid trees depends on the number of learning examples (results obtained from 40 initial trees built on 23 UCI datasets).

3. Conclusion

This paper presents the motivation for and advantages of using multi-objective learning algorithm MOLHC in a real world use case. The algorithm graphically presents the difficulty of classification task for a comprehensible classifier, identifies parts of the domain that can be classified accurately with understandable classifier, and parts of the domain that are more challenging and should be classified with a black-box (BB) classifier instead. It offers an insight into the analysed classification problem and supports classifier validation, knowledge discovery, and refinement and improvement of classifiers, which are important features according to [7]. The output of the algorithm is the Pareto set of hybrid trees, which range from most comprehensible to the most accurate. The paper shows that the size of the Pareto set depends mostly on the number of leaves in initial tree in which BB classifier achieves higher accuracy than the majority class classifier of the leaf. The Pareto front supports the user in taking well informed decision when choosing a hybrid tree that should be both accurate and comprehensible. Furthermore, the paper shows that the error of predicted comprehensibility depends on the number of learning instances, which could be a limiting factor for MOLHC application on domains with few instances. Another drawback of using MOLHC is a possible large number of hybrid trees presented on the Pareto front; however, the case study shows that the user needs to focus only on a subset of those hybrid trees that satisfies the requirements about accuracy and comprehensibility. Furthermore, presence of knees on the Pareto front and scoring of leaf quality further decreases the number of hybrid trees that must be compared by the user.

Future work should be devoted to decreasing the error of predicted comprehensibility and accuracy of hybrid trees and enabling reliable algorithm performance on small datasets. Using multiple BB classifiers should be investigated as it could provide improvements in accuracy of the hybrid trees. A program with appropriate graphical user interface could improve the user experience with the MOLHC algorithm and provide additional insights into the classification problem.

Systematic investigation of exploiting the Pareto set of hybrid trees to calculate the qualities of individual leaves also seems promising.

References

- [1] A. A. Freitas, Comprehensible classification models - a position paper. *ACM SIGKDD Explorations*, vol 15-1 (2013), 1-10.
- [2] H. Allahyari, N. Lavesson, User-oriented Assessment of Classification Model Understandability, in *Proceedings of the Eleventh Scandinavian Conference on Artificial Intelligence*, (2011), 11-19, IOS Press, ISBN 978-1-60750-753-6
- [3] D. Martens, B. Baesens, Building Acceptable Classification Models, *Data Mining - Annals of Information Systems*, vol 8 (2010), 53-74
- [4] D. R. Carvalho, A. A. Freitas, N. F. F. Ebecken, A Critical Review of Rule Surprisingness Measures, in *Proceedings of Data Mining IV - International Conference on Data Mining*, (2003) 545-556.
- [5] R. Kohavi, Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid, *Second International Conference on Knowledge Discovery and Data Mining*, (1996), 202-207, AAAI Press.
- [6] D. Martens , J. Vanthienen, W. Verbeke, B. Baesens, Performance of classification models from a user perspective, *Decision Support Systems*, vol 51-4 (2011), 782–793.
- [7] M. W. Craven, J. W. Shavlik, Extracting Comprehensible Concept Representations from Trained Neural Networks, In Working Notes on the IJCAI'95 Workshop on Comprehensibility in Machine Learning (1995) 61-75
- [8] Y. Jin, Pareto-Based Multiobjective Machine Learning: An Overview and Case Studies, *IEEE transactions on systems, man, and cybernetics - part C: applications and reviews*, vol. 38-3 (2008), 397-415.
- [9] A. A. Freitas, A critical review of multi-objective optimization in data mining: a position paper, *ACM SIGKDD Explorations Newsletter*, vol 6-2 (2004), 77-86.
- [10] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Hoboken (2009).
- [11] M. Bohanec, I. Bratko, Trading accuracy for simplicity in decision trees, *Machine Learning* vol. 15-3 (1994), 223-250.
- [12] R. Piltaver, M. Luštrek, J. Zupančič, S. Džeroski, and M. Gams, Multi-objective learning of hybrid classifiers, *21st European Conference on Artificial Intelligence* (2014).
- [13] A. Frank, A. Asuncion, UCI Machine Learning Repository, <http://archive.ics.uci.edu/ml>
- [14] Weka: Collections of Datasets, <http://www.cs.waikato.ac.nz/ml/weka/datasets.html>
- [15] R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [16] I. H. Witten, E. Frank, M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Third Edition. Morgan Kaufmann, San Francisco (2011)
- [17] K. Deb, A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, vol. 6- 2 (2002).
- [18] J. Yin, *Multi-objective Machine Learning*, Springer, Berlin (2006).