PRETSL: Distributed Probabilistic Rule Evolution for Time-Series Classification

Babak Hodjat, Hormoz Shahrzad, Risto Miikkulainen, Lawrence Murray, Chris Holmes

Abstract The distributed evolutionary computation platform EC-Star is extended in this paper to probabilistic classifiers. This extension, called PRETSL, allows the distributed age-layered evolution of probabilistic rule sets, which in turn makes more fine-grained decisions possible. The method is tested on 20 UCI data problems, as well as a larger dataset of arterial blood pressure waveforms. The Results show consistent improvement in all cases compared to binary classification rule-sets. Probabilistic rule evolution is thus a promising approach to difficult classification tasks and particularly well suited for time-series classification.

Key words: Evolutionary Computation, Probabilistic Rule-sets, Distributed Processing, Time Series Classification

1 Introduction

Rule sets utilize the notion of predicate logic and form collections of statements of the form “IF antecedent condition A is met THEN consequence B occurs”. These

Babak Hodjat
Sentient Technologies, 1 California St. #2300, San Francisco, CA, USA

Hormoz Shahrzad
Sentient Technologies, 1 California St. #2300, San Francisco, CA, USA

Risto Miikkulainen
Sentient Technologies, 1 California St. #2300, San Francisco, CA, USA

Lawrence Murray
Oxford University, Oxford, England, UK

Chris Holmes
Oxford University, Oxford, England, UK

are ideal candidate models for use in medical diagnostic applications due to their explicit, interpretable, structure and their ability to uncover nonlinear relationships and interactions in large data domains. The interpretability of rules is a vital attribute for medical applications, where predictions need to be auditable so that experts can understand how and why a recommendation or forecast was made. The ability of rule sets to deal naturally with nonlinear relations and interactions is another key attraction. The recent emergence of large-scale genetic epidemiology case-control studies has taught us that simple genotype-phenotype models can only explain a small proportion of the known heritable (genetic) risk component of a disease. Probabilistic rule sets hold great potential for uncovering cryptic relationships and can maximize the use of available information contained in the data.

Up to now, rule-set models have been hampered by the computational challenges that are needed to implement them effectively. In addition, there has been no way to accommodate uncertainty into rule-set predictions, so that they cannot be statistically characterized. The computational challenge of rule sets arises from the enormous search space of potential rules that might apply for any particular system, due to all the possible combinations of antecedents and consequences. Conventional optimization methods are ill suited to scale to such spaces.

Age-varying fitness calculation is an approach suitable for data problems in which evolved solutions need to be applied to many fitness samples in order to measure a candidate's fitness confidently Hodjat and Shahrzad (2013). This approach is elitist: Best candidates of each generation are retained to be run on more fitness cases to improve confidence in the candidate's fitness. The number of fitness evaluations in this method depends on the relative fitness of a candidate solution compared to others at any given point.

EC-Star OReilly et al (2013) is a massively distributed evolutionary platform that uses age-varying fitness as the basis for distribution, thus allowing for easier distribution of big-data problems through sampling or hashing/feature reduction techniques, breaking the data stash into smaller chunks, each contributing to the overall evaluation of the candidates.

In this paper, the power of EC-star search is combined with a probabilistic extension of rule-based logic into a new method called PRETSL (Probabilistic Rule Evolution for Time-Series cLassification). In a probabilistic rule set the consequences of rules are used to update a conditional probability statement. For example, a probabilistic rule might be, "IF condition A is present THEN the probability of the disease occurring increases by Z", where A and Z are parameters to be learned by the system. The probabilities suggested by all rules of the set are combined and thresholded to produce the final classification.

The EC-Star platform and related work in probabilistic classifiers is first reviewed below. The PRETSL approach for using fuzzy logic and probabilistic rule-sets in an age-layered distributed evolutionary run is then outlined. Initial results are presented from experimentation on 20 data sets from the UCI collection, as well as on an application on a blood-pressure prediction task, comparing the probabilistic with a binary classifier rule-set representation. The results suggest that PRETSL is an effective approach, making it possible to combine knowledge at a more fine-grained level, and thus increasing classification accuracy.

2 Prior Work

In EC-Star, age is defined as the number of fitness samples upon which a candidate has been evaluated. This system uses a hub-and-spoke architecture for distribution, where the main evolutionary process is moved to the processing nodes (Figure 1). Each node, or Evolution Engine, has its own pool of candidate rule-sets, or individuals, and independently runs through the evolutionary cycle. At each new generation, an Evolution Engine submits its fittest candidates to the Evolution Coordinator (i.e., the server) for consideration. This step takes place typically after a set number of evaluations, referred to as the maturity age.

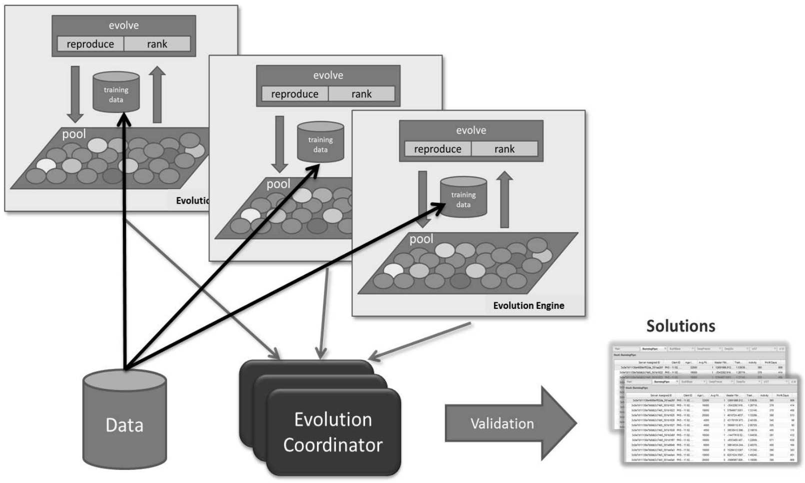


Fig. 1 The EC-Star hub-and-spoke distribution architecture. Each Evolution Engine runs an independent evolution on its own pool of candidates on a limited amount of data, and periodically reports the results to the Evolution Coordinator. The Evolution Coordinator maintains a list of the best candidates found so far and periodically sends the best of them back to the Evolution Engines for further evaluation. In this manner, EC-Star utilizes age-layering to speed up evolution, and takes advantage of heterogeneous and potentially unreliable computing resources across the internet.

The server side, or Evolution Coordinator, maintains a list of the best of the best candidates so far. EC-Star achieves scale through making copies of genes at the server, sending them to Evolution Engines for aging, and merging the aged results received back from them (Figure 2). This process also allows the spreading of the fitter genetic material. EC-Star is massively distributable by running each Evolution Engine on a processing node (e.g., CPU) with limited bandwidth and occasional availability Hodjat et al (2014). Typical runs utilize hundreds of thousands of processing units spanning across thousands of geographically dispersed sites. In the Evolution Coordinator, only candidates of the same age range are compared with one another (thus implementing age-layering). Each age range has a fixed quota.

EC-Star has previously been used e.g. in the blood-pressure prediction task, and found to be an effective implementation for rule evolution on time-series data sets - a class of problems that is not as well suited for traditional classification methods such as Random Forest Deng et al (2013). In this paper, it will be extended into probabilistic classification.

3 Design

EC-Star's default representation is a Pitts-style rule-based representation Smith (1980), where the genotype consists of a header and body. The header includes fields such as a unique ID, Age, and Master Fitness (which represents the aggregate fitness over samples evaluated so far). The gene body is a rule set with the following grammar:

<*rules*> ::= <*rule*> | <*rule*><*rules*>
<*rule*> ::= <*conditions*> → <*action*>
<*conditions*> ::= <*condition*> | <*condition*> & <*conditions*>
<*action*> ::= <*prediction label*> | <*action*>
<*condition*> ::= <*predicate*> |<*condition*> | <*condition*> [*lag*]
<*predicate*> ::= <*truth value on a feature*>

Predicates can be calculated as an inequality (e.g., less-than) against an evolved threshold on the data. For example, in the case of a normalized feature, a threshold between 0 and 1 is evolved into the predicate (say, 0.4), and it will return true, should the inequality (i.e., $feature < 0.4$) evaluate to true in the presence of that threshold.

EC-Star allows for applying fuzzy logic Klir and Yuan (1995) to the evaluation of predicates and rules. The fuzzy value for a predicate inequality is derived by applying a sigmoid function on the inequality: The closer the feature is to the threshold, the closer the resulting continuous value is to 1. Fuzzy logic is then used to calculate a fuzzy value for the rule as a whole.

In order to represent a probabilistic rule-set De Raedt and Thon (2010), an action is defined to be an evolvable probability between 0 and 1, representing the likelihood of a sample to belong to a class label defined over the data-set. In its simplest form, the probabilities of different rules that fire over a data sample are aggregated into a single probability for a binary classification system. For example, if three rules fire, returning 0.2, 0.4, and 0.6 respectively, the output verdict on the sample can be calculated as the average of the probabilities (0.4). Taking the fuzzy logic value of each rule into account gives us the opportunity to calculate the rule-set verdict as a weighted average using the fuzzy value of each rule as the weight. The last step, if needed by the domain, is to convert this value to the binary classification, with 0.5 as the threshold.

Note that it is possible that, for a given fitness sample, no rules fire, in which case, depending on the problem domain requirements, either a default action is selected, or the fitness sample is said to have resulted in a no-action state. The no-action state can thus be treated separately in the fitness function.

The fitness of a probabilistic rule-set is then calculated as the mean absolute error (MAE) of its predictions. Below, this method is referred to as PRETSL, for Probabilistic Rule Evolution for Time-Series cLassification.

4 Experiments

First, the PRETSL approach is demonstrated on 20 standard UCI data sets (Asuncion and Newman (2007). Each data set consists of a number of data points (e.g., patients), each with a number of predictors (e.g., biometrics and health history), some of which are missing for each data point. These data are partitioned randomly such that the training set has roughly 70% of data points, and the remaining 30% data points are witheld and used as the test set.

The EC-Star platform is used to train 50 binary and 50 probabilistic classifiers using the training set. For each entry in the dataset, the binary classifiers output $z = P(y=1|x) \in \{0,1\}$, while the probabilistic classifiers output $z = P(y=1|x) \in [0,1]$, in both cases giving the problem specific predicted probability (e.g., that the patient does not survive the study).

To compare methods, the mean squared error (MSE) of each classifier's predictions is calculated using the test set:

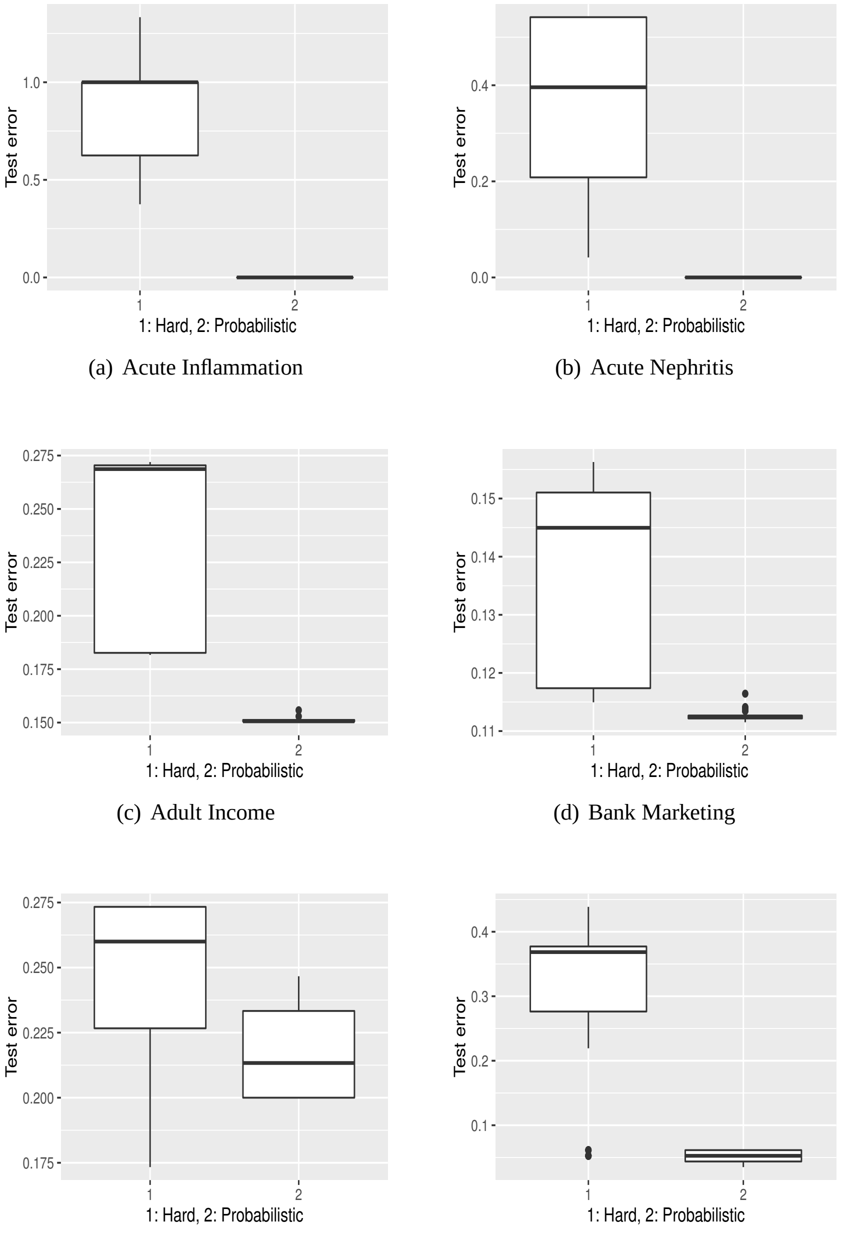
$$\mathrm{MSE}(d^v) = \frac{1}{N^v} \sum_{n=1}^{N^v} (P(y=1|x=x_n^v) - y_n^v)^2. \tag{1}$$

Note that in the case of hard classifiers, this measure reduces to the misclassification rate.

Figures 2, 3, and 4 give the results. The PRETSL approach improves classification performance in every single case and is comparable to results from random forest.

Second, PRETSL is demonstrated on a much larger real world problem of classifying time series of arterial blood pressure data. Our particular area of investigation is acute hypotensive episodes.

A large number of patient records are time series based. Some are at the granularity of high resolution physiological waveforms recorded in the ICU or via the remote monitoring systems. Given a time-series of training exemplars each of length T (in samples), to build a discriminative model capable of predicting an event, features are extracted by splitting the time series into non-overlapping (or overlapping), divisions of size k samples each, up to a certain point $h < T$ such that there are



(a) Acute Inflammation

(b) Acute Nephritis

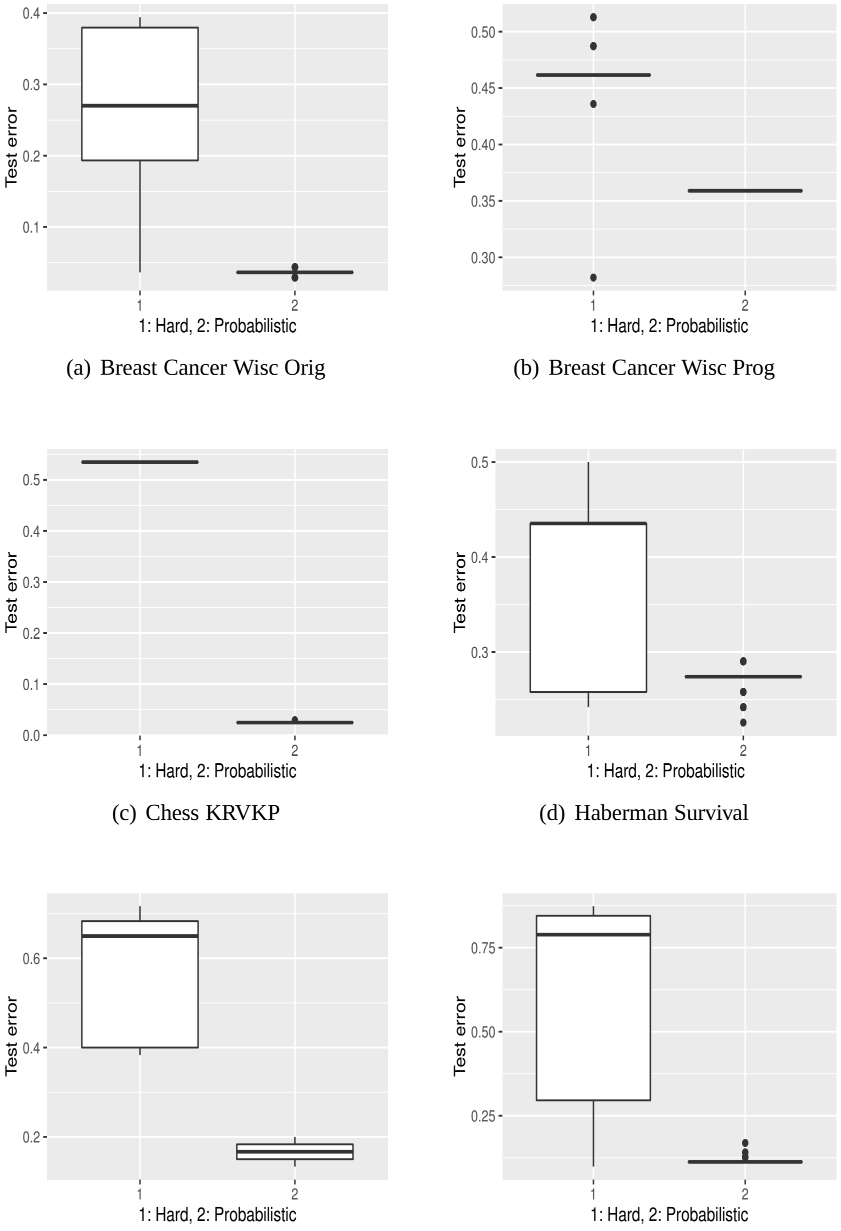
(c) Adult Income

(d) Bank Marketing

(e) Blood Transfusion

(f) Breast Cancer Wisc Diag

Fig. 2 Distribution of MSE on the test set for the 50 binary (i.e. hard) classifiers and the 50 probabilistic classifiers (i.e. PRETSL) for the first six of the 20 UCI datasets. The probabilistic classifiers outperformed the binary classifiers in each case in this figure as well as in figures 3 and 4, demonstrating the advantage of the PRETSL approach.



(a) Breast Cancer Wisc Orig

(b) Breast Cancer Wisc Prog

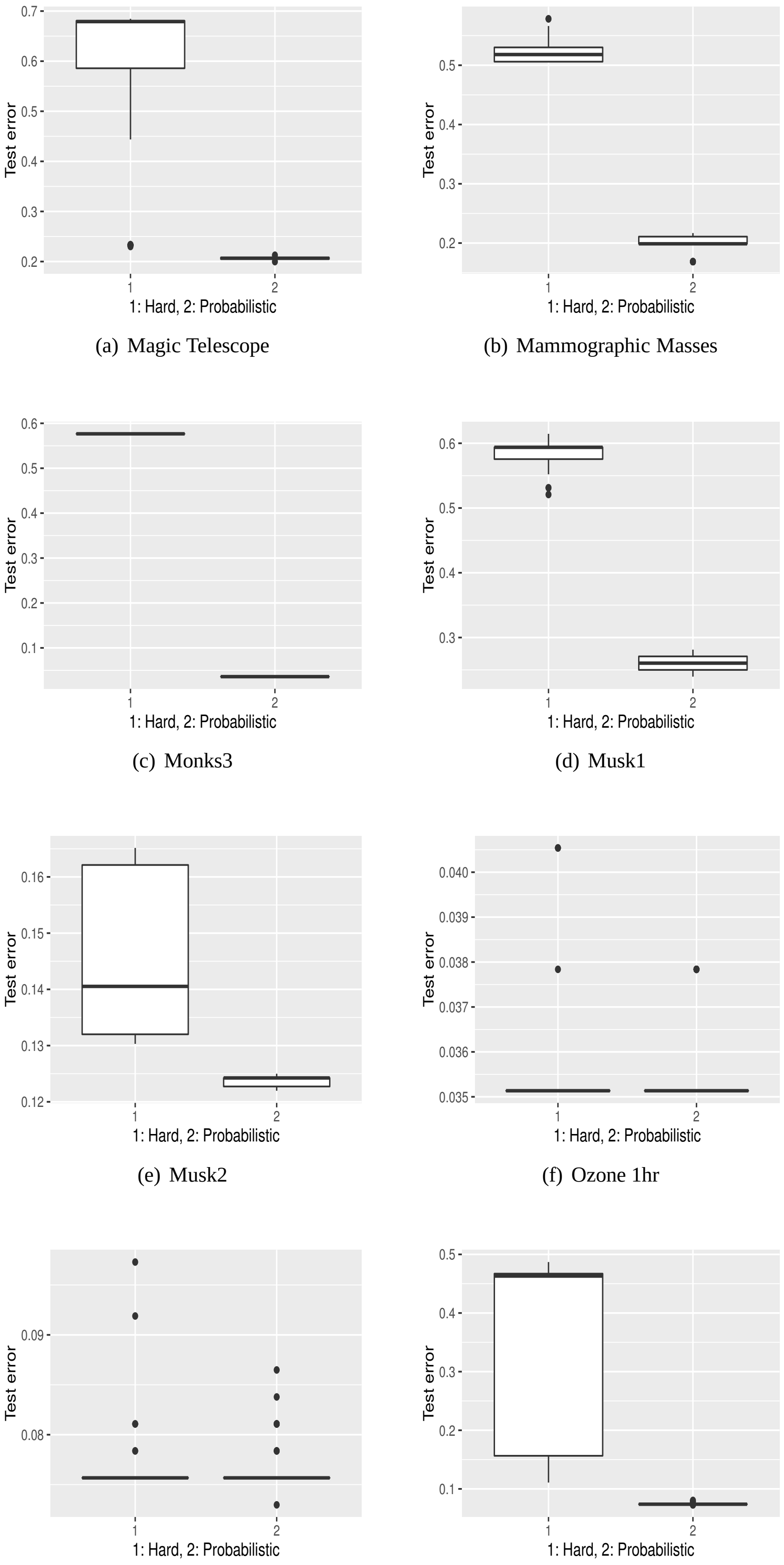
(c) Chess KRVKP

(d) Haberman Survival

(e) Heart Disease Cleveland

(f) Ionosphere

Fig. 3 Continuing from Figure 2, results for the second six of the 20 UCI datasets.



(a) Magic Telescope

(b) Mammographic Masses

(c) Monks3

(d) Musk1

(e) Musk2

(f) Ozone 1hr

(g) Ozone 8hr

(h) Spambase

Fig. 4 Continuing from Figure 3, results for the remaining eight of the 20 UCI datasets.

$m = h/k$ divisions. A number of aggregating functions are then applied to each of these divisions (a.k.a windows) to give features for the problem.

The blood-pressure dataset consists of roughly 4000 patient's ABP waveforms from MIMIC II v3, with a sampling rate of 125Hz Goldberger et al (2000), recorded invasively from one of the radial arteries. The raw data size was roughly one Terabyte. The labels in the data are imbalanced; the total number of Low events is just 1.9% of the total number of events. In total, there are 45,693 EC-Star data packages from 4,414 patient records. Of these, 32,898 packages with 100 data points each (i.e., events) were used as the training set and 12,795 samples as the test set.

Figure 5 gives the results, again showing that the probabilistic classifiers outperform the binary classifiers. Indeed, the worst performing soft classifier outperforms the best of the hard classifiers. An example probabilistic rule-set evolved by the system is given below, where V_n represents features from the wavelets in the data set, and prob is the probability for the patient to have developed critically low blood pressure after a 30 minute blackout window:

$(!V_4 < 35.13 \wedge V_{78} < 176.75 \wedge V_{52} < 6) \implies prob = 0.04$
$(V_{78} < 79.3 \wedge V_{36} < 3.09 \wedge V_{69} < 0.08 \wedge !V_{38} < -0.25) \implies prob = 0.95$
$(V_{78} < 79.3 \wedge V_{36} < 3.09 \wedge !V_{38} < -2.61) \implies prob = 0.95$
$(V_{78} < 128.03 \wedge V_{63} < 0) \implies prob = 0.14$
$(V_1 < 143.24) \implies prob = 0.84$

5 Discussion and Future Work

One key advantage of probabilistic predictions is that they can be combined with a formal loss function for misclassification in order to make optimal risk-based decisions, such as whether a patient should be given a new drug, or whether the patient requires further tests to make an accurate diagnosis or prognosis. Such an extension will allow for the integration of rule set models directly into the clinic.

Note that the rule sets are readily interpretable and may provide scientific insight; their probabilistic combination reduces the risk of overfitting that accompanies the use of a single classifier, and may facilitate model selection and hypothesis testing.

By framing the rule sets within a probabilistic system, formal methods from Bayesian statistics can be utilized to combine predictions across the population of rule sets in a coherent fashion Polson et al (2013); this approach should improve the performance further in future work.

More work is also in order to determine the source of consistently improved performance of PRETSL versus binary classification, as demonstrated in the experiments above.

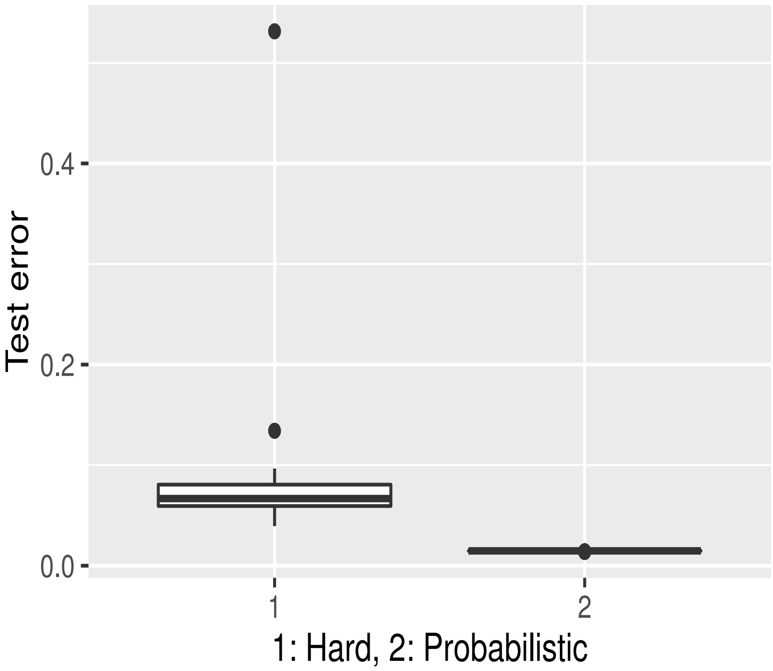


Fig. 5 Distribution of MSE on the test set for the 50 binary (i.e. hard) classifiers and the 50 probabilistic classifiers (i.e. PRETSL) for the MIMIC arterial blood pressure dataset. All PRETSL classifiers outperformed all binary classifiers in this scale-up experiment, demonstrating the power of the PRETSL approach in challenging problems in general, and time-series classification in particular.

6 Conclusion

In this paper, evolution of rule sets for classification tasks is extended into probabilistic rule sets. This method, PRETSL, is implemented in the EC-Star distributed computing platform and evaluated in 20 UCI datasets as well as in a scale-up application of blood-pressure prediction. Probabilistic formulation allows making more refined decisions, which leads to improved performance in all cases. PRETSL is therefore a promising approach to difficult classification tasks.

References

Asuncion A, Newman D (2007) Uci machine learning repository

De Raedt L, Thon I (2010) Probabilistic rule learning. In: Inductive Logic Programming, Springer, pp 47–58

Deng H, Runger G, Tuv E, Vladimir M (2013) A time series forest for classification and feature extraction. Information Sciences 239:142–153

Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, Mietus JE, Moody GB, Peng CK, Stanley HE (2000) Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals. Circulation 101(23):e215–e220

Hodjat B, Shahrzad H (2013) Introducing an age-varying fitness estimation function. In: Genetic Programming Theory and Practice X, Springer, pp 59–71

Hodjat B, Hemberg E, Shahrzad H, OReilly UM (2014) Maintenance of a long running distributed genetic programming system for solving problems requiring big data. In: Genetic Programming Theory and Practice XI, Springer, pp 65–83

Klir G, Yuan B (1995) Fuzzy sets and fuzzy logic, vol 4. Prentice hall New Jersey

OReilly UM, Wagy M, Hodjat B (2013) Ec-star: A massive-scale, hub and spoke, distributed genetic programming system. In: Genetic Programming Theory and Practice X, Springer, pp 73–85

Polson NG, Scott JG, Windle J (2013) Bayesian inference for logistic models using pólya–gamma latent variables. Journal of the American statistical Association 108(504):1339–1349

Smith SF (1980) A learning system based on genetic adaptive algorithms