Rectifying Predictions of Classifiers by Local Rules

Martin Možina and Ivan Bratko

Faculty of Computer and Information Science, University of Ljubljana, Slovenia

Abstract. The main advantage of unordered classification rules is in their power to spot and explain local regularities. However, using them in classification often poses problems due to conflicts between rules, when some resolution principle needs to be applied. On the other hand, most of the machine learning methods try to learn conflict-free hypotheses covering the whole domain space and are not concerned with single patterns only. In this paper we propose an algorithm named PILAR that combines the advantages of both approaches. Our algorithm aims at improving any machine learning algorithm by comparing its predictions with predictions of rules, and applying changes to the predictions of initial model when necessary. Moreover, if a dummy classifier (e.g. majority classifier) is used, then this procedure acts as a classifier from rules only and can be compared to other methods for classification from rules. We experimentally validated our method with two basic classification methods. In the first one dummy classifier was used and in the second logistic regression.

1 Introduction

One of the main motivations for inducing a set of unordered rules is their comprehensibility. Each rule concisely explains the correlation between class and a set of attributes on the subspace covered by the rule. Their locality, however, raises problems when used in classification, as, in some cases, clashes occur and sometimes there are no rules that would cover the classifying example. In the former case we need to apply any of the available conflict resolution strategies[8], while in the latter default class is the most obvious choice.

Most of the machine learning methods do not focus on particular subspaces, but rather induce theories that cover the whole domain and need not to be concerned with conflicts. Examples of such methods are ordered rules[2] (also known as decision lists), decision trees[11], logistic regression[4], etc. Their advantage, on the other hand, can also be their drawback; it is hard to spot all relevant patterns in data while optimizing for the overall performance.

In this paper, we try to combine the advantages of both approaches. We propose an algorithm named PILAR (Probabilistic Improvement of Learning Algorithms with Rules) that receives a base method as input, which can be any classifier that covers the whole example space, and a set of unordered rules. By comparing predictions of the base method on the learning set with given

rules, we observe whether in any of the patterns described with rules, there is a significant difference in probabilistic prediction of the base method and the corresponding rule. In a such situation, the prediction of the base method is accordingly corrected. Note that if a dummy classifier (e.g. one that always predicts 50% for both classes in two-class domains) is used as the base method, then the method classifies from rules only and can be regarded as a scheme for resolving conflicts between unordered rules.

We describe our algorithm in the following section. We first translate given rules as constraints and continue with a description of a procedure that makes the base method consistent with these constraints. In the third section we describe some necessary changes to the rule induction algorithm to produce rules that can be used in the algorithm. The algorithm is experimentally evaluated in section four with the selection of two base methods; in the first one the base method is majority classifier that always predicts the majority class, while in the second we used logistic regression.

2 Method

The problem described in the introduction can be formalized as follows.

Inputs:

- A set of learning examples $\mathbf{X} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where one example is as usual a pair of attribute-values and a class value. Let the domain of class variable be values $c_1, \ldots, c_m$.
- A global classification model M. The expression $M_j(x_i)$ returns the probability of class c_j for example x_i.
- A set of probabilistic unordered rules $\mathbf{R} = R_1, \ldots, R_m$. Each rule R_i has a condition part defining the subspace covered by the rule, a class predicted by rule, and the probability of predicted class given conditions.

Output: A corrected classification model M', based on M and consistent with probabilistic predictions of rules.

2.1 Rules As Constraints

The core idea of this paper is to use rules as constraints in the induction of the corrected model M'. We will begin by developing a general scheme for constraints, which are independent of the actual learning principle of model M'. This general scheme will be relatively abstract. Later, in the next section, we shall instantiate a specific learning algorithm and corresponding constraints to illustrate mentioned concepts and enable testing and experimentations.

A probabilistic IF-THEN rule R has the following structure:

$$\text{IF } Cond(\mathbf{X}) \text{ THEN } P(Y = c_j) = q(R) \tag{1}$$

The conditions term $Cond(\mathbf{X})$ determines examples that are covered by this rule, while the conclusion predicts class c_j (or $cl(R)$) and its conditional probability $P(Y = c_j|Cond(\mathbf{X}))$, called also as the quality $q(R)$ of rule R. We will assume that the probability $q(R)$ is *unbiased*, namely, it gives an estimation of the expected relative frequency of all possible subsets drawn from population where $Cond(\mathbf{X})$ are true.

In a global classification model, we need to estimate the class probability for a specific example $P(c_j|x_i)$. However, a rule R only gives an aggregated estimation of probability $q(R)$ for all covered examples, which is not enough to estimate $P(c_j|x_i)$ well, unless $q(R)$ is close to 0 or 1. Nevertheless, there is a relation between $q(R)$ and probabilities $P(c_j|x_i)$, which will be used as the first constraint on the final global classification model.

Let $C_j(x_i)$ be a random variable with value 1 if the example x_i has class value c_j, and 0 otherwise. The expected value of $C_j(x_i)$ is

$$E(C_j(x_i)) = 1 \times P(c_j|x_i) + 0 \times (1 - P(c_j|x_i)) = P(c_j|x_i) \tag{2}$$

We will denote examples covered by rule R with X_R. The expected number of covered positive examples is thus the expected sum of all $C_j(x_i)$ from X_R:

$$E\left(\sum_{x_i \in X_R} C_j(x_i)\right) = \sum_{x_i \in X_R} E(C_j(x_i)) = \sum_{x_i \in X_R} P(c_j|x_i) \tag{3}$$

Note, that the above equation identifies a way of computing the expected relative frequency. Let $M'_j(X_R)$ be the sum of estimated probabilities for class c_j predicted by corrected model M' in examples X_R:

$$M'_j(X_R) = \sum_{x_i \in X_R} M'_j(x_i) \tag{4}$$

Since $M'_j(x_i)$ estimates the probability $P(c_j|x_i)$, then we can presume that $M'_j(X_R)$ is an estimation of $\sum_{x_i \in X_R} P(c_j|x_i)$. Expected relative frequency is by definition computed as the ratio of expected number of positive examples and the number of all examples. According to the definition, $q(R)$ also represents expected relative frequency, therefore the most natural constraint would be:

$$\frac{M'_{cl(R)}(X_R)}{|X_R|} = q(R) \tag{5}$$

However, in real situations, the above constraint will rarely be satisfiable for all rules. For example, estimation of class probability in a rule is sometimes dependent on rule properties (e.g. rule length), and in such situations we could encounter two different rules covering the same set of examples X_R, while having different qualities $q(R)$. Hence, for the sake of applicability, we need to relax the "ideal" constraint.

Considering that qualities $q(R)$ are unbiased estimates of probability, the differences between $q(R)$ and $M'_{cl(R)}(X_R)/|X_R|$ should be close to zero (~ 0). The

differences could be computed with any of possible distance measures for probability, and we selected the log-likelihood (LL) measure for groups (equivalent to Kullback-Leibler divergence [6]):

$$LL(R|M') = |X_R| \times \left[q(R) \log \frac{q(R)}{\frac{M'_{cl(R)}(X_R)}{|X_R|}} + (1 - q(R)) \log \frac{1 - q(R)}{1 - \frac{M'_{cl(R)}(X_R)}{|X_R|}} \right] \quad (6)$$

The $LL(R|M')$ equals 0 when $q(R) = M'_{cl(R)}(X_R)/|X_R|$ and > 0 otherwise. The first constraint on M' is then:

Constraint 1.

$$\sum_{R_i \in \boldsymbol{R}} (\text{ sign } (R_i) \times LL(R_i|M')) \sim 0, \quad (7)$$

where

$$\text{sign } (R) = \begin{cases} 1, \frac{M'_{cl(R)}(X_R)}{|X_R|} - q(R) \geq 0; \\ -1, \frac{M'_{cl(R)}(X_R)}{|X_R|} - q(R) < 0. \end{cases} \quad (8)$$

The argument in favor of the above constraint is that the cumulative predictions of M' will be evenly spread around the qualities of rules. However, the constraint states nothing about the dispersion of predictions; there is nothing that would force the differences between $M'_{cl(R)}(X_R)$ and $q(R)$ to be as small as possible.

There is an appropriate solution for our problem. We can impose another constraint that prevents rules, where the final model is optimistic $M'_{cl(R)}(X_R)/|X_R| > q(R)$, to have any influence whatsoever on the model. Such a constraint will take care that $M'_{cl(R)}(X_R)/|X_R|$ are as close to the evaluation $q(R)$ as possible, while the first constraint assures a balance between positive and negative errors. The second constraint from rules on M' is thus:

Constraint 2. IF $\frac{M'_{cl(R)}(X_R)}{|X_R|} > q(R)$, then R should have no influence on probability prediction in M'.

2.2 PILAR

PILAR is an algorithm that exploits both constraints in practice. First, we will describe the model space that enables the use of predictions from M and predictions from rules. Afterwards, we will translate the learning problem to a nonlinear optimization problem, and describe the optimization algorithm.

Although our algorithm works for any number of classes, we will assume that we deal with a two-class problem with values c_0 and c_1, to simplify explanation of the algorithm. We selected the log-linear function as a model, because it enables simple and understandable explanations of classification[10]. The model is parameterized by a weight vector $\mathbf{W} \in \mathbb{R}$ (a weighted sum):

$$\begin{aligned} f_1(x) &= \ln \frac{M_1(x)}{1 - M_1(x)} + \mathbf{W} \cdot \mathbf{R}(x) = \\ &= \ln \frac{M_1(x)}{1 - M_1(x)} + w_0 + w_1 \times R_1(x) + \ldots, \end{aligned} \quad (9)$$

where $M_1(x)$ is the probability given by the base method for class c_1. Term $R_i(x)$ is defined as:

$$R_i(x) = \begin{cases} 0, \text{ if conditions of } R_i \text{ are false for } x; \\ 1, \text{ if } R_i \text{ predicts class } c_1; \\ -1, \text{ if } R_i \text{ predicts class } c_0. \end{cases} \tag{10}$$

The probability of class c_1 is computed from $f_1(x)$ through the logit link function:

$$M_1'(x) = \frac{1}{1 + e^{-f_1(x)}} \tag{11}$$

In order to fit weights $\mathbf{W}$ from data we will define the problem as a constrained optimization problem. Generally, we could use any criteria for optimization like AUC, classification accuracy, etc., however, as our goal is to improve probability prediction, the logical choice is log-likelihood:

$$LL(D|\mathbf{W}) = \sum_{i=1}^{N} \ln M_{y_i}(x_i) \tag{12}$$

Along with the fitting function, we define the following three constraints:

1. $w_i \leq 0$,
2. If $M'_{cl(R)}(X_{R_i}) > q(R_i)$, then $w_i = 0$,
3. $\sum_{R_i \in \mathbf{R}} (\text{ sign } (R_i) \times LL(R_i|M')) \leq 0$.

Regarding the first constraint (1), all weights should be nonnegative. Negative weights are implausible, since the rule's effect on M' would be inconsistent with rule's class in conclusion. For example, a rule predicting class c_1 should not be allowed to decrease the probability for class c_1 in M'. The second constraint (2) is a special version of the second constraint in the previous section. It prevents rules that are already sufficiently explained (average probability of $M'_{cl(R)}(X_R)$ is higher than its quality $q(R)$) to influence predictions of M'. In the last constraint (3) we only changed the sign from "closeness to zero" (~ 0) to "less or equal zero" (≤ 0), since equality constraints allow less manoeuver space for optimization. This change is possible due to contradicting effects of the third constraint and the goal function. In most of the cases, if $LL(D|\mathbf{W})$ increases, then the sum in (3) also increases, therefore, we expect in the best evaluated model to have the third constraint near zero.

Algorithms 1 and 2 show the pseudo code of our simple hill-climbing algorithm to find the best solution. In each step, (1) a rule is selected, (2) its weight is slightly changed, and (3) other weights are changed in a such way that consistency with constraints is fulfilled. These three sub-steps are then repeated until we are unable to find a weight that would still improve the quality of the current model.

Algorithm 1 A general hill-climbing algorithm for finding the best fit of weights **W**. In each step, procedure ChangeOneWeight is called that slight changes the weight of a single rule.

Procedure FindBestFit()

Let W be the weights associated to rules. All weights are set to 0.
Let s be 2.
Let oldLL be $-\infty$.
while $s > 0.001$ **do**
  changed = True
  $s = s/2$
  while changed = True **do**
    changed = False
    for each $W_i \in$ **W do**
      W' = ChangeOneWeight(**W**,i,s)
      newLL = computeLikelihood(**W'**)
      if newLL > oldLL **then**
        oldLL = newLL
        W = **W'**
        changed = True
      end if
    end for
  end while
end while
Return **W**

3 "Unbiased" Induction of Rules

We explained in the previous section that we need an unbiased estimation of probability to effectively use rules as constraints in rule-based classification. In statistics, relative frequency is defined as the unbiased estimator of probability:

$$Q(r) = \frac{s}{n} \tag{13}$$

where n is the number of learning examples covered by the rule r and s is the number of positive examples among them.

However, the assumption that the relative frequency indeed estimates the probability of positive class in rule learning is wrong. The culprit is the extensive search for the best rule which was explored by Jensen and Cohen [5] who blame multiple comparisons during the search to be responsible for plethora of pathologies in induction algorithms. The same problem was found by Quinlan and Cameron-Jones[12] called oversearching. In [9] we proposed a method which can fix (make them less biased) many rule evaluation measures by taking multiple comparisons into account through the use of extreme value distributions. The method is called EVC (Extreme Value Correction)[1].

[1] The pdf of the paper as well as its short summary can be found at *http://www.ailab.si/martin/evc/*

Algorithm 2 A procedure that changes one weight and satisfies the constraints.

Procedure ChangeOneWeight(Weights ***W****, WeightIndex i, Step s)*

$W_i = W_i + s$
if $M'_{cl(R)}(X_{R_i}) > q(R_i)$ **then**
  $W_i = W_i - s$
  Return **W**
end if
for each W_k, where $k \neq i$ **do**
  while $M'_{cl(R)}(X_{R_k}) > q(R_k)$ AND $W_k \geq s$ **do**
    $W_k = W_k - s$
  end while
end for
Return **W**

3.1 Probabilistic Covering

The main problem of the mentioned correction method is that extreme distributions need to be prepared upfront for the whole domain. This makes removing learning examples after a single rule is learned unfeasible, as all distributions should be recomputed. Here we will propose an alternative strategy for removing examples named *probabilistic covering*[2].

Let $x.prob$ be the quality of the best rule covering x. If there are no rules covering x then $x.prob$ equals the prior probability of the example's class. When a new rule R is learned, the removing procedure updates all probabilities of covered examples as $x.prob = maximum(x.prob, q(R))$. We say that, when an example with $x.prob$ becomes covered by a new rule, where $r(Q)$ is higher than $x.prob$, then rule R *improves* the probability of this example; or shorter: rule R improves example. We call this probabilistic covering.

Futhermore, we have to inforce certain changes to the procedure of learning a single rule to prevent learning the same rule all over again, as probabilistic covering only records how well an example is explained and does not change the way in which a rule is learned. We propose the following changes:

Selection of best rule A new rule can be learned only if it improves at least one example. This must be added as a condition in the algorithm.

Selection of N most promising rules (star) In the original CN2 algorithm best N rules are selected according to $q(R)$. This heuristic fails in our case, as we seek for a rule that has high quality *and* will improve at least one example. We used:

$$EI(R) = \sum_{x_i \in X_R} [1.0 - (P(Z < x.prob) - 0.5) * 2] \tag{14}$$

[2] Since probabilistic covering is not the focus of this paper, we will only briefly explain its main idea.

to estimate expected number of improved examples for rule R. The random variable Z is distributed according to normal distribution with $\mu = r(Q) \times |X_R|$ and $\sigma^2 = |X_R| \times r(Q)(1 - r(Q))$.

4 Evaluation and Discussion

PILAR tries to improve a learning method on attribute subspaces indicated by classification rules. In the extreme case, where the base method is majority classifier, PILAR acts as an approach for classification from rules. This will also be the subject of our initial evaluation, where PILAR will be compared to some other strategies for resolving conflicting rules. Later, in the other part of this section, we will use PILAR to improve logistic regression and compare its results with the basic logistic regression. PILAR is implemented within the data-mining software Orange[3].

4.1 Classification from Unordered Rules

Currently there are several approaches available for classification from unordered rules. They could be classified in two classes; some of them are simple [1] and enable classification "by hand", but they usually perform worse with respect to accuracy than some more sophisticated methods [8]. Understandability is usually stated as the most important advantage attributed to rules, and we believe that it is also critical for experts to understand how classification works. The models produced by PILAR are easily understandable, since they are a simple weighted sum of rules. Hence, we shall compare our method to those with the same property, and will not concern other, more complicated methods (an overview of several methods was written by Lindgren [8]).

As we previously explained, PILAR requires evaluations of rules to be unbiased, which was achieved by using extreme value correction. Hence, we need to conduct a controlled experiment that will give answers to the following two questions:

1. How impaired is PILAR, when evaluations are biased (non-EVC)?
2. Is PILAR better than competing methods, when evaluations are EV-corrected?

We used three different classification methods in our experiments:

CN2 classical CN2 [2] classification that sums class distributions of all covering rules,
BAY a classifier that combines rules with naive-Bayesian formula,
PIL PILAR,

and two different evaluation functions:

LAP Laplacian formula, as used in standard CN2, and
EVC EV-correction of Laplace formula.

Together they call for six possible combinations, each described by a pair (classification method, evaluation method). For example, (CN2,LAP) is the classical CN2 rule learning algorithm, while (PIL,EVC) is the method suggested by this paper.

The results of experiments on several UCI domains are shown in Tables 1 and 2. The method (PIL,EVC) significantly outperforms all other methods. As its results are better than (PIL,LAP), we conclude that unbiased evaluation is important for PILAR. Similarly, since it works better then (CN2,EVC) and (BAY,EVC), we believe that PILAR is better than the other two methods.

Table 1. Brier scores of CN2 and EVC-CN2 with different classifiers on several UCI data sets. Bold values mark the best method(s) for given data set. The significances in the last row obtained by Wilcoxon test give a reason to believe that (PIL,EVC) performs significantly better than other.

Data set	(CN2,LAP)	(BAY,LAP)	(PIL,LAP)	(CN2,EVC)	(BAY,EVC)	(PIL,EVC)
abalone	0.41	0.38	0.40	0.40	0.49	**0.31**
adult	0.30	0.27	0.26	0.29	0.35	**0.22**
auto-mpg	0.15	0.15	0.16	**0.13**	0.16	**0.13**
breast-cancer	0.52	0.46	0.44	0.38	0.44	**0.37**
breast-cancer-wis	0.07	**0.06**	0.09	0.07	0.07	**0.06**
bupa	0.50	0.45	**0.42**	0.46	0.57	0.44
crx	0.26	0.24	0.26	0.26	0.26	**0.21**
galaxy	0.10	**0.09**	0.10	0.13	0.11	0.11
german	0.43	0.40	0.39	0.39	0.46	**0.35**
heart_disease	0.34	0.31	0.32	**0.25**	0.32	0.27
housing	0.22	0.21	0.24	0.22	0.27	**0.20**
imports-85	0.14	0.14	0.18	**0.13**	0.16	0.14
ionosphere	0.15	0.13	0.17	**0.12**	**0.12**	**0.12**
monks-1	**0.00**	**0.00**	0.01	0.17	0.05	0.03
monks-2	0.48	0.53	0.55	0.46	0.44	**0.39**
monks-3	0.06	0.07	0.05	**0.03**	**0.03**	**0.03**
promoters	0.31	0.26	0.27	0.23	**0.21**	0.22
prostate	0.44	**0.39**	0.41	**0.39**	0.42	**0.39**
SAheart	0.55	0.50	0.47	0.39	0.53	**0.38**
servo	0.10	**0.08**	0.10	0.11	0.11	0.12
shuttle-landing	**0.02**	**0.02**	0.06	0.09	0.08	0.04
tic_tac_toe	**0.01**	**0.01**	**0.01**	0.19	0.09	0.04
titanic	**0.31**	**0.31**	**0.31**	0.38	0.35	**0.31**
voting	0.08	0.08	0.12	0.10	0.10	**0.07**
Sig. vs (PIL,EVC)	< 0.001	0.007	< 0.001	< 0.001	< 0.001	-

Table 2. Classification accuracies of CN2 and EVC-CN2 with different classifiers on several UCI data sets. Bold values mark the best method(s) for given data set. The significances in the last row obtained by Wilcoxon test suggest, to some extent, that (PIL,EVC) achieves on average higher classification accuracies.

Data set	(CN2,LAP)	(BAY,LAP)	(PIL,LAP)	(CN2,EVC)	(BAY,EVC)	(PIL,EVC)
abalone	0.74	0.73	0.71	0.74	0.74	**0.76**
adult	0.82	0.82	0.81	0.77	0.80	**0.85**
auto-mpg	0.89	0.89	0.88	**0.91**	**0.91**	**0.91**
breast-cancer	0.70	0.69	0.70	0.73	0.71	**0.74**
breast-cancer-w	**0.96**	**0.96**	0.94	0.95	**0.96**	**0.96**
bupa	0.68	0.69	**0.70**	0.63	0.62	0.64
crx	0.83	0.83	0.81	0.84	**0.86**	0.84
galaxy	**0.94**	**0.94**	0.92	0.91	**0.94**	0.93
german	**0.74**	0.73	0.72	0.70	0.73	0.73
heart_disease	0.79	0.80	0.76	**0.84**	0.82	0.80
housing	0.87	**0.88**	0.83	0.86	0.86	**0.88**
imports-85	0.91	0.91	0.88	**0.92**	0.91	0.91
ionosphere	0.90	0.92	0.88	0.92	**0.93**	**0.93**
monks-1	**1.00**	**1.00**	**1.00**	0.99	0.99	0.99
monks-2	**0.73**	0.66	0.65	0.66	0.66	0.66
monks-3	**0.99**	**0.99**	**0.99**	0.98	**0.99**	**0.99**
promoters	0.78	0.78	0.81	0.80	0.80	**0.87**
prostate	**0.75**	**0.75**	0.74	0.72	0.74	0.74
SAheart	0.67	0.68	0.67	0.68	**0.70**	**0.70**
servo	**0.94**	**0.94**	0.93	0.93	0.93	0.93
shuttle-landing	0.98	0.98	0.97	0.93	0.93	**0.99**
tic_tac_toe	0.99	0.99	**1.00**	0.88	0.94	0.99
titanic	**0.79**	**0.79**	**0.79**	0.68	0.78	0.78
voting	0.95	0.95	0.92	0.93	0.94	**0.96**
Sig. vs (PIL,EVC)	0.139	0.099	0.002	< 0.001	0.045	-

4.2 Improving Logistic Regression

In this experiment, we test the ability of PILAR to improve another general method. We used stepwise logistic regression [4] as the base method. We point out two interesting questions:

1. Is it possible to improve probabilistic predictions of logistic regression with rules?
2. Can PILAR decrease the quality of logistic regression?

One would expect that the answer to the first question is yes. Logistic regression is basically a weighted sum of attributes, and can not by itself successfully exploit interactions between attributes. This can be a critical weakness in certain domains. On the other hand, we could also assume that increased complexity (with the use of rules) will sometimes decrease the quality of the base method, as complexity of learning method is related to overfitting.

The results of experiment (Brier scores) of logistic regression and corrected logistic regression are shown in Table 3. These results suggest that the answer to the first question is yes, and to the second is no. The corrected logistic regression is statistically better than normal logistic regression, moreover, the corrected logistic regression actually did not perform worse on a single data set, for the brier score was either better or stayed the same. Naturally, it is vital to regard these results as defeasible, since the results were obtained only on some domains, yet still the results are very promising.

5 Conclusion

In this paper, we presented PILAR, a method for improving probabilistic prediction of methods based on classification rules. Rules are used as constraints on the final model, requiring that its probabilistic predictions are on average similar to the probabilistic estimates given by rules. We studied the usefulness of our method as a rule classification technique and as a correction method for logistic regression. In both cases, PILAR proved to be a promising method and statistically outperformed competing methods. Given our experimental findings, we believe that it would improve (or at least not worsen) any general classification method.

The core of our method is a nonlinear optimization problem with nonlinear constraints. Currently, we use a simple hill-climbing strategy that may not be optimal. As future work, we plan to apply an advanced optimization technique, e.g. an evaluation strategy, that would find global optimum more often, which should further increase the quality of PILAR's models.

There already exist some approaches that combine linear classifiers (e.g. logistic regression) with non-linear (decision trees) [7]. While these approaches are conceptually different from ours, it would be still interesting to explore, if it is, on average, better to correct a linear classifier with rules or induce one model that contains both ideas.

Table 3. Brier scores of logistic regression, and PILAR with logistic regression on several UCI data sets. Bold values mark the best method(s) for given data set. The significances in the last row show that PILAR significantly improves predictions of logistic regression.

Data set	LR	(PIL,EVC)+LR
abalone	**0.29**	**0.29**
adult	0.25	**0.23**
auto-mpg	**0.14**	**0.14**
breast-cancer	**0.39**	**0.39**
breast-cancer-w	**0.05**	**0.05**
bupa	0.42	**0.40**
crx	**0.20**	**0.20**
galaxy	0.20	**0.10**
german	**0.33**	**0.33**
heart_disease	**0.23**	**0.23**
housing	0.19	**0.18**
imports-85	**0.17**	**0.17**
ionosphere	0.26	**0.23**
monks-1	0.51	**0.03**
monks-2	0.46	**0.43**
monks-3	0.36	**0.03**
promoters	**0.30**	**0.30**
prostate	**0.26**	**0.26**
SAheart	**0.36**	**0.36**
servo	0.37	**0.12**
shuttle-landing	0.49	**0.04**
tic_tac_toe	0.32	**0.12**
titanic	0.33	**0.32**
voting	**0.07**	**0.07**
Sig. vs (PIL,EVC)	< 0.001	-

References

1. Peter Clark and Robin Boswell. Rule induction with CN2: Some recent improvements. In *Machine Learning - Proceeding of the Fifth Europen Conference (EWSL-91)*, pages 151–163, Berlin, 1991.
2. Peter Clark and Tim Niblett. The CN2 induction algorithm. *Machine Learning Journal*, 4(3):261–283, 1989.
3. J. Demšar and B. Zupan. Orange: From experimental machine learning to interactive data mining. White Paper `[http://www.ailab.si/orange]`, Faculty of Computer and Information Science, University of Ljubljana, 2004.
4. D. W. Hosmer and S. Lemeshow. *Applied Logistic Regression, 2nd Edition.* Wiley-Interscience, 2000.
5. David D. Jensen and Paul R. Cohen. Multiple comparisons in induction algorithms. *Machine Learning*, 38(3):309–338, March 2000.
6. S. Kullback and R. Leibler. On information and sufficiency. *Ann. Math. Stat.*, 22:7686, 1951.
7. N. Landwehr, M. Hall, and E. Frank. Logistic model trees. In *Proceedings of the 16th European Conference on Machine Learning*, 2003.
8. Tony Lindgren. Methods for rule conflict resolution. In *In Proceedings of the 15th European Conference on Machine Learning (ECML-04)*, pages 262–273, Pisa, 2004. Springer.
9. Martin Možina, Janez Demšar, Jure Žabkar, and Ivan Bratko. Why is rule learning optimistic and how to correct it. In Johannes Fuernkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Proceedings of 17th European Conference on Machine Learning (ECML 2006)*, pages 330–340, Berlin, 2006. Springer-Verlag.
10. Martin Mozina, Janez Demsar, Michael W. Kattan, and Blaz Zupan. Nomograms for visualization of naive bayesian classifier. In *PKDD*, pages 337–348, 2004.
11. J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann,San Diego, 1993.
12. J. R. Quinlan and R. Cameron-Jones. Oversearching and layered search in empirical learning. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1019–1024, Montreal, Canada, August 1995.