Machine Learning (2019) 108:297–329
https://doi.org/10.1007/s10994-018-5731-3

Extreme value correction: a method for correcting optimistic estimations in rule learning

Martin Možina[1] · **Janez Demšar**[1] · **Ivan Bratko**[1] · **Jure Žabkar**[1]

Received: 19 May 2011 / Accepted: 13 June 2018 / Published online: 26 June 2018

Abstract
Machine learning algorithms rely on their ability to evaluate the constructed hypotheses for choosing the optimal hypothesis during learning and assessing the quality of the model afterwards. Since these estimates, in particular the former ones, are based on the training data from which the hypotheses themselves were constructed, they are usually optimistic. The paper shows three different solutions; two for the artificial boundary cases with the smallest and the largest optimism and a general correction procedure called extreme value correction (EVC) based on extreme value distribution. We demonstrate the application of the technique to rule learning, specifically to estimating classification accuracy of a single rule, and evaluate it on an artificial data set and on a number of UCI data sets. We observed that the correction successfully improved the accuracy estimates. We also describe an approach for combining rules into a linear global classifier and show that using EVC estimates leads to more accurate classifiers.

Keywords Machine learning · Multiple comparisons · Extreme value distribution · Rule learning

1 Introduction

The task of classification models is to predict the classes of new, unseen examples or their probabilities of belonging to a certain class. Machine learning algorithms construct such

Editor: Johannes Fürnkranz.

✉ Martin Možina
martin.mozina@fri.uni-lj.si

Janez Demšar
janez.demsar@fri.uni-lj.si

Ivan Bratko
ivan.bratko@fri.uni-lj.si

Jure Žabkar
jure.zabkar@fri.uni-lj.si

[1] Faculty of Computer and Information Science, University of Ljubljana, Večna pot 113, 1000 Ljubljana, Slovenia

models by fitting hypotheses to training data. In presence of noise or in learning problems with a small number of training examples, algorithms that usually consider a huge number of hypotheses may find one which fits the training data well, but not the unseen rest of the population. Such overfitting causes misclassification of new examples, and avoiding it has been one of the core topics of machine learning research.

A less researched related problem is that of overconfident evaluation of hypotheses. Machine learning algorithms look for hypotheses that are as accurate as possible, which leads them to discovering the parts of the example space that are (possibly by chance) mostly populated by data instances from the same class. As a result, while the predicted classes may be correct, the probability that the example covered by the hypothesis belongs to the predicted class is often exaggerated.

To illustrate the phenomenon, we prepared a set of 300 artificial data sets with controlled class probabilities for each possible rule. We randomly defined a maximum class probability between 0.5 and 1.0 for each artificial data set. All data sets have ten binary attributes, five of which are unrelated with the class and for the other five we randomly defined class probability (lower than the maximum probability) for each combination of their values. We then generated 2^{10} examples for each data set, one for each combination of attribute values, and assigned the classes randomly according to the defined probabilities for the combination of informative attributes. This means that the actual class proportions in the data set do not necessarily match the defined probabilities for a particular combination of attribute values. We can calculate the true class probability for an arbitrary subset of attributes by computing the marginal class probability over the informative attributes in the subset.

For each data set, we learned an optimal rule with an algorithm that is similar to the algorithm for finding the best rule in CN2 (Clark and Niblett 1989). We set the beam width to 5 and use relative frequency as search heuristics. Figure 1 plots the probability of class membership predicted by the rule versus the known true probability. The algorithm was consistently able to discover a rule covering only examples of the same class and thus predicting the probability of 1.0, although the actual class probabilities for these rules were uniformly distributed between 0.5 and 1.0.

For a formal background, let $h_1, h_2, \ldots$ be all possible hypotheses considered by a certain learning algorithm, and let $q_1, q_2, \ldots$ be their corresponding qualities. The hypotheses can be, for instance, classification rules, and the qualities can be the probability of the predicted

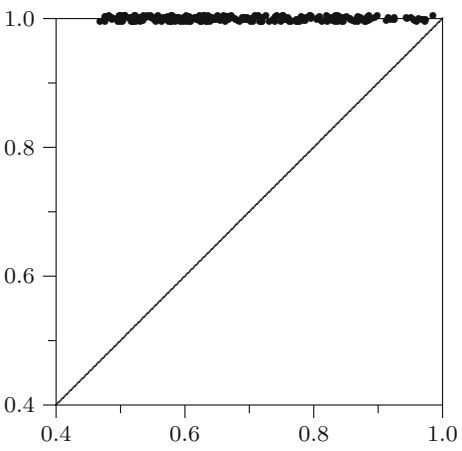

Fig. 1 Relation between the estimated (y-axis) and the true (x-axis) class probabilities of discovered rules. Minimal jitter is applied to improve visibility

classes or the χ^2 statistic computed on a 2×2 table of the hypothesis' predictions and the true classes. The task of the learning algorithm is to (a) compute the qualities of all hypotheses and (b) to select the best hypothesis h_{max}, such that $\forall i : q_{max} \geq q_i$.

In all practical cases, the qualities are estimated on a data sample. Let $\widehat{q_i}$ be the sample estimate of the true q_i, then $\widehat{q_i} = q_i + \epsilon_i$. It is usually assumed that $\widehat{q_i}$ is an unbiased estimate of q_i, so the expectation of the random error $E(\epsilon_i)$ over different possible samples equals zero. The problem occurs at point (b), where the algorithm picks up a single "optimal" hypothesis, selected not by its true q_i but by $q_i + \epsilon_i$. In common conditions in machine learning – the data sample is small relative to the huge number of competing hypotheses – the error terms can easily overwhelm the differences in qualities, so the algorithm chooses the "luckiest" hypothesis instead of the one with the highest true quality.

While it is possible to get unbiased estimates of the quality of individual hypotheses, the machine learning algorithm would most often choose one of those for which the estimate highly exceeds the true value. In this context, ϵ_i measures how much the estimated quality exceeds the true quality, namely the *optimism* of the assessment. The concept of optimism is formally defined in Sect. 3.

Optimistic selection has two consequences. First, among the competing hypotheses, the more complicated ones have a greater chance of overfitting and being selected. This may lead the algorithm to select a suboptimal hypothesis. Second, the selected hypothesis will yield exaggerated probabilities on new data instances, resulting in over-confident decisions.

The machine learning community has been aware of the effect of overfitting for a long time, and it developed a number of solutions, like fine-tuning the complexity of hypotheses by various regularizations. In this paper, we propose an alternative solution for controlling overfitting by estimating optimism of a method on randomized data.

This paper is a substantial generalization of our earlier conference paper on removing optimism in rule quality estimation in CN2 (Možina et al. 2006). We first define the general theory on estimation correction and then illustrate it using the CN2 algorithm. The algorithm related to CN2 is itself improved and illustrated in the coverage space (Fürnkranz and Flach 2005). In the last section, we provide a new approach to combining rules into a global classifier and present an extended evaluation on a larger collection of real-world data sets.

2 Related work

Quinlan and Cameron-Jones (1995) showed that extensive searching (oversearching) often produces less accurate results. They further showed that also complex models can achieve less accurate results, because complex models are usually obtained by more searching.

There are several approaches that deal with the problem of oversearching, which can be divided into two broad categories. One is to use a separate data set to validate the findings. The drawback of this procedure is that it reduces the training data set and is also unsuitable for comparing the competing hypotheses during the induction process. Besides, a single validation is again prone to random effects, while cross-validation estimates the performance of the learning algorithm and therefore also cannot be used during the induction process.

The other solution to overfitting is to penalize complex models. A common approach is to include a penalization term into the evaluation function, such as the ridge regression or the Lasso method (Hastie et al. 2009). These methods shrink the coefficients by imposing a

penalty on their size. They have been applied successfully in various learning methods, such as in linear and logistic regression, neural networks, SVMs, etc.

The penalization can be also viewed as an application of the Bayesian approach, in which we define a prior over possible hypotheses and combine it with the data to obtain the posterior. The most relevant Bayesian method for this paper is the m-estimate of probability (Cestnik 1990), which tries to estimate the quality of a rule by estimating its classification accuracy. The main idea of m-estimate is to balance between the prior probability and the probability assessed from the data. When the evidence is thin, the probability estimate is shifted from posterior to prior distribution. The m-estimate was frequently used in rule learning as a probability estimate. For example, Džeroski et al. (1993) applied it within the CN2 rule learning algorithm.

Janssen and Fürnkranz (2010) and Fürnkranz (2004) meticulously explored usefulness of the m-estimate and other rule learning heuristics. Fürnkranz (2004) looked for a function that would map from the rule's covered positive and negative training examples to the accuracy of the rule measured on the validation set. He obtained the best results with the m-estimate probability with $m = 1.6065$. His aim was similar to that of our method, which also tries to improve the accuracy measure of individual rules. Janssen and Fürnkranz (2010), however, focused on finding the optimal parameters for global classification. They experimented with several search heuristics, and m-estimate with value $m = 22.466$ turned out to be optimal. In the experimental comparison we use the proposed value for m in the m-estimate of probability.

Domingos (1999), on the other hand, suggested that penalization should in some way consider the number of hypotheses that were examined during the search process. Indeed, applying penalty does not directly deal with the problem of multiple hypotheses, however, by keeping the trained models small, it reduces the effect of the problem significantly. In his paper, Domingos proposed a formula to select the m parameter of the m-estimate method that reflects the number of explored hypotheses. This method is further discussed and also experimentally compared with our method in Sect. 4.3.2.

Scheffer (2005) proposed a measure for evaluating association rules that corrects confidence of the rule, given the support of the rule. This formula depends on the prior of all confidences over the given items for the given data set. This approach does not take the number of tested hypotheses into account, therefore its main idea is more related to the m-estimate, since it trades accuracy (confidence) for coverage (support). There are many other measures that trade coverage of rules for their accuracy; we will limit this study to the m-estimate as their most renowned representative.

The most extensive study of the problem, also called multiple comparison procedures in machine learning, was performed by Jensen and Cohen (2000). They list various related pathologies of induction algorithms, and find and theoretically investigate their causes. The basic supposition of Jensen and Cohen is similar to ours, that is, that learning algorithms measure $\widehat{q_i}$ and treat it as if it was an unbiased estimate of the true q_i, which becomes a problem when the hypothesis h_i is not a randomly chosen hypothesis but the one with the highest $\widehat{q_{max}}$, where the optimism term ϵ_i might have had a greater impact than the quality of the hypothesis itself. As solutions to the problem, they propose various techniques, such as the Bonferroni correction, using separate data for evaluation, cross validation and randomization.

Jensen and Cohen are, however, mostly analyzing two problems which are not in the focus of our paper: how to compare the winning hypotheses from two or more separate pools of hypotheses (for instance, selecting the best attribute among the binarized continuous attributes) and compute the probability of getting the hypothesis of the same quality at

random. Note that this is a different problem than assessing the actual quality of hypotheses, which is what our paper is concerned with.

The structural risk minimization (Vapnik 1995) theory is an alternative explanation of why learned hypotheses are optimistic and how penalization or ensemble methods deal with this problem. The idea of structural risk minimization is to estimate the hypothesis space complexity and then, from this complexity and the error obtained on learning sample, predict an upper bound on the test error of hypothesis. Complexity is usually considered as the VC-dimension (Vapnik 1995) in the case of infinite hypothesis space and the number of all possible hypotheses in finite hypothesis space. However, as both approaches consider only the hypotheses space and not the space of the data itself, several authors have proposed alternative data-dependent approaches to measure complexity of a hypothesis class, e.g. Rademacher complexity (Bartlett and Mendelson 2003). The empirical Rademacher complexity estimates the upper error bounds by fitting hypotheses to random data. This may be potentially useful for dealing with the problem discussed in this paper, yet we are unaware of any existing method using that approach in rule learning.

Another widely adopted solution of the multiple hypothesis problem is picking several hypotheses instead of only a single one. These ensemble methods, such as stacking, boosting, random forests, etc., average out the error term ϵ_i, which reduces the effect of the multiple hypotheses phenomena and makes the combined hypothesis less biased. This solution, however, does not achieve the main goal of this paper, namely removing the effect of testing multiple hypothesis from the estimated quality of the best hypothesis.

In the context of hypothesis testing, Bonferroni adjustment (Holm 1979) can sometimes be used to reduce the computed significance of hypotheses. This is however of little use in machine learning where millions and billions of hypotheses would make any finding insignificant. The Bonferroni correction also assumes mutual independence of hypotheses, which is highly violated in machine learning and makes the correction overly conservative. Since some hypotheses are considered only implicitly, for example pruned off by discarding a part of the search space, it is difficult to compute the effective number of tested hypotheses. Alternatively, correction of p value can also be determined empirically with randomization (e.g. Hanhijärvi 2011). In general, the corrective methods coming from the context of statistical hypothesis testing [including, for instance Holm (1979) and Hochberg (1988)] aim at correcting the probability of *finding a hypothesis with some estimated quality if the investigated effect is actually random*, while we are interested in *correcting the estimated quality of the hypothesis*.

The approach described in this paper is based on the theorems from extreme value theory (Fisher and Tippett 1928; Coles 2001). Similarly to the central limit theorem which states that the sample averages of random variables with finite variance are distributed approximately normally, the extremal types theorem states that all distributions of maximal values of data samples can be approximated by one of three distributions. For instance, for a normally distributed variable X, its maximal value

$$X_{max} = \max(X_1, X_2, X_3, \ldots, X_n)$$

is distributed according to Gumbel's distribution (Coles 2001). In machine learning, the distributions with such shapes have already been experimentally found (but not identified as such) by Jensen and Cohen (2000).

3 Extreme value correction

Many machine learning algorithms adopt in some way the following learning scheme (Jensen and Cohen 2000):

1. Generate n candidate hypotheses $h_1, \ldots, h_n$.
2. Evaluate them with evaluation function q on a given data sample S; $\widehat{q_i} = q(h_i, S)$.
3. Return the best hypothesis h_{max} according to the quality q; its quality is

$$\widehat{q_{max}} = \max(\widehat{q_1}, \ldots, \widehat{q_n}).$$

Examples of such methods are decision tree learning methods, rule induction algorithms, all stepwise procedures, and others.

Having an evaluation function q for measuring the quality of a hypothesis, we will henceforth use q_i to denote the true quality of hypothesis h_i, $\widehat{q_i}$ for its (potentially optimistic) estimate on training examples, and $\overline{q_i}$ for the quality as corrected by our proposed method. As noted above, h_{max} will be the hypothesis with the maximal estimated quality $\widehat{q_{max}}$, and our task is to find its corrected quality $\overline{q_{max}}$.

Let Q_i be a random variable representing the quality of h_i. We will consider $\widehat{q_i}$ to be an instantiation of Q_i whose value depends on the data sample, and q_i the quality of h_i on the population from which the sample was drawn. We assume that the $\widehat{q_i}$ is an unbiased estimate, namely, $\widehat{q_i} = q_i + \epsilon_i$ where the random error ϵ_i has $E(\epsilon_i) = 0$.

For h_{max}, which is not chosen randomly but based on $\widehat{q_{max}} = q_{max} + \epsilon_{max}$, its large estimated quality can be either due to a high true quality q_{max} or due to chance, ϵ_{max}. Different samples can yield different hypotheses h_{max}, and $\widehat{q_{max}}$ can be treated as instantiation of another random variable Q_{max} with distribution defined by maxima over all possible random samples, *i.e.* $P(Q_{max} > x_0)$ equals the proportion of samples for which the algorithm would find a hypothesis with $\widehat{q_{max}} > x_0$.

For illustration, let there be two hypotheses h_j and h_k with equal qualities $q_j = q_k$, while the qualities of other hypothesis are too low for the error term to have an affect, $\forall i, i \neq j \wedge i \neq k : q_j - q_i \gg \epsilon_j - \epsilon_i$. The learning algorithm would then choose either h_j or h_k, hence for a particular sample $\widehat{q_{max}} = \max(\widehat{q_j}, \widehat{q_k}) = q_j + \max(\epsilon_j, \epsilon_k)$. Jensen and Cohen (2000) showed that $E(\max(\epsilon_j, \epsilon_k)) \geq E(\epsilon_j)$ which in our case means that $E(\max(\epsilon_j, \epsilon_k)) \geq 0$. This positive bias, which appears as a consequence of trying several hypotheses, is in this paper called optimism of the learning algorithm. The expected optimism $E(\epsilon_{max})$ increases with the number of competing hypotheses, while increasing the number of inferior hypotheses does not affect the distribution of q_{max} and the related optimism. Note that the notion of optimism in this paper differs from what is usually regarded as optimism in statistics, where optimism is defined as the difference between the in-sample error and the training error (Hastie et al. 2009).

Generally, $\widehat{q_{max}}$ is a positively biased estimate of the true quality of h_{max}. In this section we will first analyze two unlikely border scenarios, where minimal and maximal optimism occur. Then, we will present a general method for correcting optimistic estimates that is in-between both border cases and it retains some favorable statistical properties.

The theory and corrections will be illustrated using artificial data sets with 1000 binary attributes and 100 examples in two classes C_1 and C_2, 50 examples in each. In each experiment we assign to each attribute x_i the true probability $P(x_i = 0|C_1) = P(x_i = 1|C_2)$; higher values of $P(x_i = 0|C_1)$ imply higher correlation between attribute x_i and class variable. The values of attributes are generated randomly according to the conditional probabilities.

Our goal is to discover, from a given sample of data, which of the attributes is the most related to the class. We use the chi square (χ^2) test for measuring the quality of the relation. Let o_1, o_2, o_3, o_4 be the observed numbers of examples from a 2x2 contingency table of attribute x_i and class:

	C_1	C_2
$x_i = 0$	o_1	o_2
$x_i = 1$	o_3	o_4

Since $P(x_i = 0|C_1) = P(x_i = 1|C_2)$ and the number of examples in both classes is exactly 50, expected values of all cells are 25. Therefore, the formula for χ^2 simplifies to:

$$\widehat{q_i} = \widehat{\chi_i^2} = \sum_j \frac{\left(o_j - e_j\right)^2}{e_j} = \sum_j \frac{\left(o_j - 25\right)^2}{25} \tag{1}$$

Furthermore, beside discovering the best hypothesis (attribute in our case), we also aimed at estimating, using the given sample only, the true quality of the best hypothesis. The estimation was compared with χ^2 computed directly from the pre-defined probabilities, which represented the true quality:

$$q_i = \chi_i^2 = 4\frac{(50 \cdot P(x_i = 0|C_1) - 25)^2}{25}. \tag{2}$$

3.1 The case of no optimism

There are a few circumstances in which the quality estimates are not optimistic. Since $\widehat{q_i}$ is assumed to be an unbiased estimator of q_i, no optimism occurs if hypotheses are chosen at random. There is also no optimism if we consider only a single hypothesis not constructed from the data, as common in statistical hypothesis testing.

For a realistic scenario, there is no optimism when there exists a hypothesis h_j with quality much higher than that of all other hypotheses (relative to errors ϵ). In this case, the ϵ_j does not influence the selection, so $E(\epsilon_j) = 0$ and $E(\widehat{q_j}) = q_j$ (and therefore $E(\widehat{q_{max}}) = q_{max}$). The corrected value should therefore equal the estimate $\overline{q_{max}} = \widehat{q_{max}}$.

For an experiment showing the circumstances in which the optimism disappears, we constructed 10 data sets with conditional probability $P(x_1 = 0|C_1)$ between 0.5 and 0.9 with a step 0.03, and $P(x_i = 0|C_1)$=0.5 for $i \neq 1$. Figure 2 shows a graph comparing averaged estimated $\widehat{\chi^2}$ of h_{max}'s and the theoretical χ^2. If the conditional probability of the first attribute is close to 0.5, the best evaluation suffers from high optimism, since the qualities of alternative hypotheses are similar. However, as the probability increases, the optimism diminishes and the estimated value $\widehat{\chi^2}$ becomes a good approximation of the theoretical value.

3.2 The case of maximal optimism

The largest optimism is manifested in an unrealistic scenario in which all hypotheses have equal true quality q. A correction assuming this scenario would impose the highest reduction

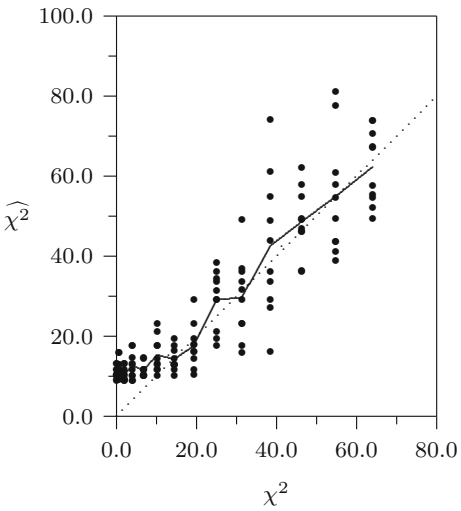

Fig. 2 No optimism: estimates of χ^2 versus the true values. One hypothesis has a known quality, others are random. Each dot corresponds to the selected hypothesis from one trial. The line shows the average for each $\widehat{\chi^2}$ at given χ^2 and the dotted line corresponds to ideal, unbiased estimates $\widehat{\chi^2} = \chi^2$

in best hypothesis' quality, and we call it the *pessimistic correction*. Let us assume that all hypotheses have the same true q. Then,

$$\widehat{q_{max}} = \max(\widehat{q_1}, \ldots, \widehat{q_n}) = q + \max(\epsilon_1, \ldots \epsilon_N). \tag{3}$$

Let Q be a random variable representing estimated qualities of all hypotheses. Then $P(Q > \widehat{q_{max}})$ is the probability of a random hypothesis exceeding the quality $\widehat{q_{max}}$. If q had been known, we could have computed the $P(Q > \widehat{q_{max}})$. But the opposite is also true: if the probability $P(Q > \widehat{q_{max}})$ was known, we could infer quality q.

We will now illustrate this idea on our χ^2 example. When all attributes have the same conditional probability $p_0 = P(x_i = 0|C_1)$ and p_0 was known, we could estimate the probability $P(Q > \widehat{q_{max}})$ by computing the χ^2 value with a formula similar to the one from Eq. (1), where estimated frequencies are adjusted for p_0: $e'_1 = e'_4 = 50p_0$ and $e'_2 = e'_3 = 50(1 - p_0)$. Since this χ^2 value comes from a chi-squared distribution with one degree of freedom, we could compute $P(Q > \widehat{q_{max}})$ using cumulative distribution function of $\chi^2(1)$. However, we could also reverse this reasoning, if $P(Q > \widehat{q_{max}})$ was known and p_0 unknown. We could find the conditional probability p_0 and the corresponding $\widehat{q_{max}}$ with a root-finding method, for example bisection.

To explain a procedure for computing $P(Q > \widehat{q_{max}})$, let us assume that the optimism $\max(\epsilon_1, \ldots \epsilon_N)$ in equation 3 is independent of the quality q when all hypotheses have the same true quality. Hence, if we were able to construct n hypotheses with the same known true quality (which can be different from q), where Q_p represents their estimated qualities and π_p is the expected quality of the best hypothesis among them, then, due to the same optimism in both sets of hypotheses, the probability $P(Q_p > \pi_p)$ would equal the probability $\mathrm{P}(Q > \widehat{q_{max}})$. There are cases where the assumption of the same optimism for all qualities q does not hold. For example, when the quality q approaches its upper bound, the optimism diminishes, since the estimated values $\widehat{q_i}$ are also bound by the same upper bound. In such cases, the pessimistic correction will decrease the quality estimate more than it should.

We can prepare n hypotheses with the same true quality by randomizing the data:

1. Permute classes of the training examples to remove correlations between hypotheses and the class.

Table 1 Contingency tables for the example of pessimistic correction

	C_1	C_2
(a) Expected contingency if $P(x_i = 0\|C_1)$ equals 0.7		
$x_i = 0$	35	15
$x_i = 1$	15	35
(b) Contingency table of the best scored attribute		
$x_i = 0$	44	6
$x_i = 1$	6	44
(c) Expected contingency of the best scored attribute		
$x_i = 0$	34.3	15.7
$x_i = 1$	15.7	34.3

2. Find and evaluate the best hypothesis. Store the evaluation.
3. Compute the average (π_p) of the stored evaluations.
4. Repeat steps 1–3 until π_p converges.

The distribution of the random variable Q_p is known as it represents the quality of hypotheses on randomized data. Q represents the quality of hypotheses on the actual data and is therefore unknown. Since the optimism is the same in both cases, we have

$$P(Q_p > \pi_p) = P(Q > \widehat{q_{max}}). \tag{4}$$

The sought $\widehat{q_{max}}$ thus has a value which results in such a distribution for Q that $P(Q > \widehat{q_{max}})$ equals the computed $P(Q_p > \pi_p)$.

We will illustrate the method on a data set like the one in the previous section, except that all conditional probabilities will equal $P(x_i = 0|C_1) = P(x_i = 1|C_2) = 0.7$. The unknown "true" 2×2 contingency table of expected frequencies is shown in Table 1(a) and the true quality of the hypothesis is:

$$q_{max} = 4\frac{(50 \cdot 0.7 - 25)^2}{25} = 4\frac{(35 - 25)^2}{25} = 16. \tag{5}$$

The best scored attribute in our generated data set has the contingency table of observed frequencies as shown in Table 1(b). Its $\widehat{q_{max}}$ equals

$$\widehat{q_{max}} = \frac{(44 - 25)^2}{25} + \frac{(6 - 25)^2}{25} + \frac{(44 - 25)^2}{25} + \frac{(6 - 25)^2}{25} = 57.76. \tag{6}$$

Now we estimate the average quality π_p on randomized data. After 200 repetitions of randomization, the estimated average for our domain is 11.28. Since $Q_p \sim \chi^2(1)$, the probability $P(Q_p > \pi_p)$ equals 0.00078. Having $P(Q_p > \pi_p)$ and $\widehat{q_{max}}$, we need to find such probability $p_0 = P(x_i = 0|C_1)$ and corresponding expected frequencies ($e'_1 = e'_4 = 50p_0, e'_2 = e'_3 = 50(1 - p_0)$, where $P(Q > 56) = 0.00078$. Therefore, we have to solve the following equation:

$$\frac{\left(44 - e'_1\right)^2}{e'_1} + \frac{\left(6 - e'_2\right)^2}{e'_2} + \frac{\left(44 - e'_3\right)^2}{e'_3} + \frac{\left(6 - e'_4\right)^2}{e'_4} = 11.28 \tag{7}$$

As all expected frequency depend on p_0, we can find p_0 with bisection. Table 1(c) shows expected frequencies for which the χ^2 gives the correct expected value:

$$\chi^2 = \frac{(44 - 34.3)^2}{34.3} + \frac{(6 - 15.7)^2}{15.7} + \frac{(44 - 34.3)^2}{34.3} + \frac{(6 - 15.7)^2}{15.7} = 11.2 \tag{8}$$

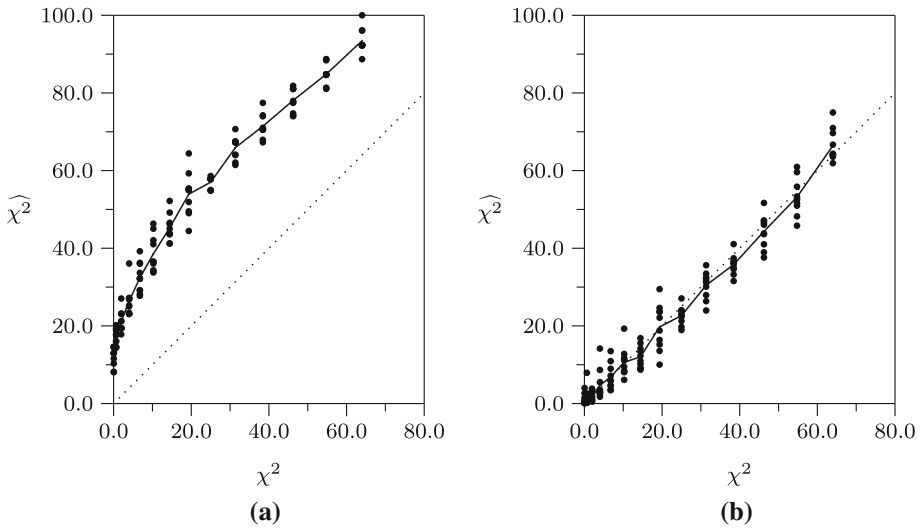

Fig. 3 Original and corrected estimates of χ^2. All hypotheses have the same true quality. **a** Uncorrected $\widehat{\chi^2}$ and **b** Pessimistic correction of $\widehat{\chi^2}$

In other words, if the best-scored hypothesis has distribution from Table 1(b) and we know that the true scores of all hypotheses are the same, then the expected class distribution of all hypotheses is the one from Table 1(c). With this in mind we can use these frequencies to compute the corrected value of $\widehat{q_{max}}$, $\overline{q_{max}}$:

$$\overline{q_{max}} = \frac{(34.3-25)^2}{25} + \frac{(15.7-25)^2}{25} + \frac{(34.3-25)^2}{25} + \frac{(15.7-25)^2}{25} = 13.8 \quad (9)$$

We repeated this experiment for different settings of $P(x_i = 0|C_1)$ with ten different random data sets for each setting. Each data set had again 1000 attributes and 100 examples. Figure 3 shows the original and corrected estimates. The corrected estimates fit the diagonal line almost perfectly.

3.3 General extreme value correction

This section describes the extreme value correction (EVC), which is the main contribution of this paper. Figure 4 depicts two steps of the EVC procedure:

1. Compute P_a from EVD_p and $\widehat{q_{max}}$. Value $\widehat{q_{max}}$ is the estimated quality of the best hypothesis h_{max}, P_a is the probability of finding a hypothesis with quality $\widehat{q_{max}}$ or larger assuming that all possible hypotheses h_i are random (their true quality equals the default quality). EVD_p is the corresponding extreme value probability distribution.
2. Find such $\overline{q_{max}}$, where $P(Q_r > \overline{q_{max}}) = P_a$. The random variable Q_r models qualities of random hypotheses and has probability distribution D_p.

We shall now explain both steps in detail. The learning algorithm finds the best hypothesis h_{max} with quality $\widehat{q_{max}}$ which is an optimistic estimate of the h_{max}'s true quality q_{max}. Since h_{max} was selected as the best among many hypotheses, the estimation $\widehat{q_{max}}$ is from a distribution of extreme values which we shall denote by EVD_p.

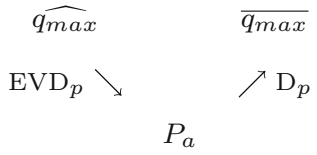

Fig. 4 Outline of the proposed procedure for general correction of estimates. Left branch shows the extreme value route: value $\widehat{q_{max}}$ is the estimated quality of selected hypothesis h_{max}, EVD_p is the corresponding extreme value distribution, and P_a is the probability of finding hypothesis with quality $\widehat{q_{max}}$ or larger when all hypotheses are actually random. Right branch shows the inverse of computing statistical significance of a single hypothesis; $\overline{q_{max}}$ is the corrected quality of the selected hypothesis and D_p is the distribution of quality measure

If we knew EVD_p, we could compute the significance of $\widehat{q_{max}}$ from the cumulative distribution of EVD_p. P_a is the probability that the learning algorithm would find a hypothesis with quality $\widehat{q_{max}}$ even if all possible hypotheses h_i were actually random.

We therefore need to model the extreme value distribution EVD_p. Fisher and Tippett (1928) have shown that extreme values, $\max\{X_1, X_2, \ldots X_n\}$, for X's coming from any distribution can be approximated by the general form

$$F(x; \mu, \sigma, \xi) = \exp\left\{-\left[1 + \xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}\right\}. \tag{10}$$

The three parameters describe the distribution's *location* (μ), *scale* (σ), and *shape* (ξ). When $\xi = 0$, the distribution reduces to Gumbel type:

$$G(x; \mu, \sigma) = \exp\left\{-\exp\left(-\frac{x-\mu}{\sigma}\right)\right\}. \tag{11}$$

We will use $G(x; \mu, \sigma)$ to model EVD_p throughout the paper, since we will be using only χ^2 distribution in our experiments, and the extreme values of χ^2 are covered by the Gumbel distribution. The derivation of EVD model assumes independent random variables, however Coles (2001) showed that in the case of dependent variables the shape of the limit distribution stays the same, but with different parameters.

We will fit the parameters of EVD_p with the following randomization procedure.

1. Permute classes of the training examples to remove correlations between hypotheses and the class.
2. Find and evaluate best hypothesis with the selected learning algorithm. Store best evaluation.
3. Fit parameters of EVD_p on all best evaluations computed until now.
4. Repeat steps 1–3 until convergence.

In the second step of the EVC procedure, we calculate the estimate $\overline{q_{max}}$ by following the steps of the standard statistical procedure for hypotheses testing, however in the reverse order. Unbiased estimates of random hypotheses are distributed by a known distribution D_p, say χ^2, depending on the used quality measure. Knowing the correct estimate $\overline{q_{max}}$ of the quality of h_{max}, we would be able to compute its significance (probability P_a) from D_p. On the other hand, since we already know the significance P_a, we can calculate $\overline{q_{max}}$ from P_a. The concrete method for calculation of the corrected estimate $\overline{q_{max}}$ which gives the same P_a then depends upon the chosen measure of quality and its distribution D_p.

The left branch of the algorithm in Fig. 4 is the approach used in machine learning to estimate the probability that the best hypothesis was found purely by chance. The right branch

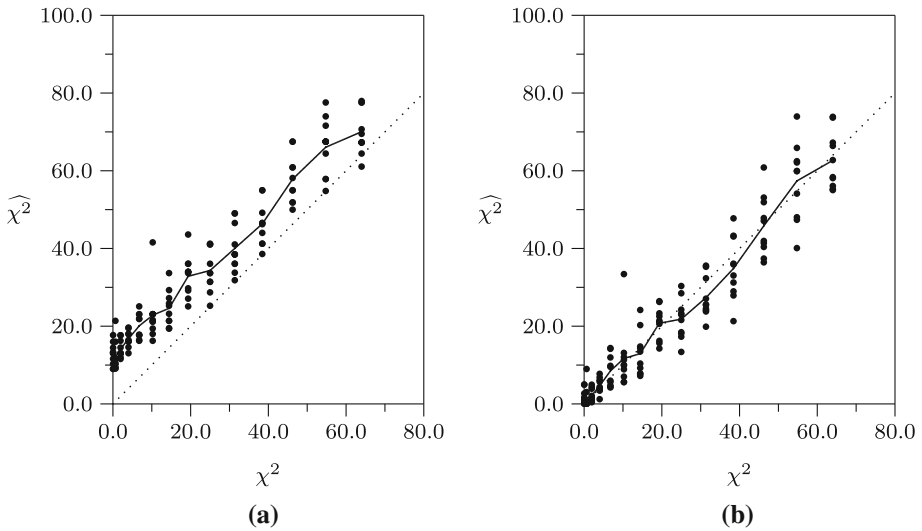

Fig. 5 Original and corrected estimates of χ^2. True qualities are distributed normally. **a** Uncorrected $\widehat{\chi^2}$ and **b** $\widehat{\chi^2}$ corrected by EVC

is the classical procedure for null-hypothesis significance testing: select a single hypothesis and evaluate it with an appropriate test. The idea of the EVC algorithm is to join these two procedures by assuming that we should obtain the same significance P_a by following either of the two paths. However, as we are not interested in P_a, but in the unknown $\overline{q_{max}}$, we compute P_a by taking the left side first and then go back up on the right side. Our procedure first estimates P_a given EVD_p, and then finds $\overline{q_{max}}$ which gives the same P_a. The computed $\overline{q_{max}}$ is the corrected estimate which we are looking for.

The basic requirement for applicability of extreme value distribution is that all measurements come from the same distribution. In our case, all $\widehat{q_i}$ for random hypotheses have to come from the same distribution. This requirement is met by some quality estimates, e.g. χ^2, but not by all, e.g. relative frequencies in rule evaluation have different β distributions. We will show a possible work-around for such cases in the following section.

For a simple test, we constructed 10 data sets for each $maxP$ between 0.5 and 0.9 with step 0.03. Each data set had 1000 attributes and 100 examples. The probability $P(x_i = 0|C_1)$ for each attribute was randomly drawn from normal distribution $N(0.5, (maxP - 0.5)/3.2)$. This way 99.9% of attributes have either $P(x_i = 0|C_1)$ or $P(x_i = 0|C_2)$ between 0.5 and $maxP$. We chose normal distribution as a good representative for true quality distributions on real data sets. The results of the extreme value correction are shown in Fig. 5. We observe that the original estimates are optimistic while the adjusted values match the true values very well.

Extreme value correction is an approach that consists of two parts: first, it uses randomization to assess the optimism produced by a learning method on random data, and, second, uses this distribution to remove the optimism from the estimated quality $\widehat{q_{max}}$ of the best hypothesis. The resulting qualities of hypotheses have the following three favorable properties:

1. Using EVD_p will result in a P_a corrected for multiple hypothesis testing. The corrected quality $\overline{q_{max}}$ therefore corresponds to a value that gives the same P_a, when tested with

a standard statistical test without the use of extreme value distribution. For example, if $\widehat{q_{max}}$ equals the median of the extreme value distribution, the corrected $\overline{q_{max}}$ will equal the median of the original distribution. Note that "uncorrected" value $\widehat{q_{max}}$ does not have this property.

2. If the same EVD_p is used, then the order of hypotheses is retained: $\widehat{q_i} > \widehat{q_j} \longrightarrow \overline{q_i} > \overline{q_j}$. Since the same EVD_p is used to calculate P_a, therefore P_a of $\widehat{q_i}$ will be lower than P_a of $\widehat{q_j}$, and hence $\overline{q_i} > \overline{q_j}$.
3. Comparing hypotheses with different complexities, where different number of hypotheses were tested, will result in different extreme value distributions, therefore the order of best-to-worst hypotheses can change in favor of simpler hypotheses. This property is demonstrated in the application of EVC to rule learning.

4 Extreme value correction in rule learning

Many rule learning algorithms induce models by iteratively searching for the best rule and removing the examples covered by it (Fürnkranz and Flach 2005). Rules are usually sought by a beam search, which gradually adds conditions to the rule with aim to decrease the number of covered "negative" examples, while at the same time keeping as many "positive" examples as possible. The search is guided by two measures, one which evaluates the partial rules and the other which chooses between the final rule candidates; here we will use the common approach where the same measure is used for both purposes.

A good rule should give accurate class predictions, that is, have a high probability of the positive class on the entire population (not only learning sample) covered by rule. A reasonable choice for the measure of rule's quality is the relative frequency of the predicted class:

$$q_i = \frac{s_i}{n_i}, \tag{12}$$

where n_i is the number of learning examples covered by the rule r_i, and s_i is the number of positive examples among them.

However, as q_i is estimated on a sample, $\widehat{q_i} = \widehat{s_i}/\widehat{n_i}$ is an optimistic estimate of the true relative frequency q_i, as we have already shown theoretically as well as experimentally (Figs. 1, 6a). We will assume that the estimate of the number of examples that the rule covers is unbiased, $E(\widehat{n_i}) = n_i$, and use a better estimate of s_i, $\overline{s_i}$. The alternative idea, where s_i would be fixed and n_i corrected, would still lead to the same result.

Machine learning algorithms often use the m-estimate (Cestnik 1990) to shift the probabilities toward the prior distributions,

$$Q_m(r_i) = \frac{s_i + m \cdot p_a}{n_i + m}, \tag{13}$$

where p_a is the prior probability and m is a parameter of the method. Fürnkranz and Flach (2005) showed that the m-estimate presents a trade off between precision (relative frequency) and linear cost metrics [for instance, weighted relative accuracy (Lavrač et al. 1999; Todorovski et al. 2000)]. Different values of parameter m can be used to approximate many common evaluation functions. For instance, when $m = 0$, m-estimate equals the relative frequency.

To put the m-estimate to a test, we again induced a single rule using the same algorithm and the same data sets as in the introduction, this time using the m-estimate with different values of m (0, 2, 10, 20, 50, 100). With increasing values of m, the method is still optimistic for rules

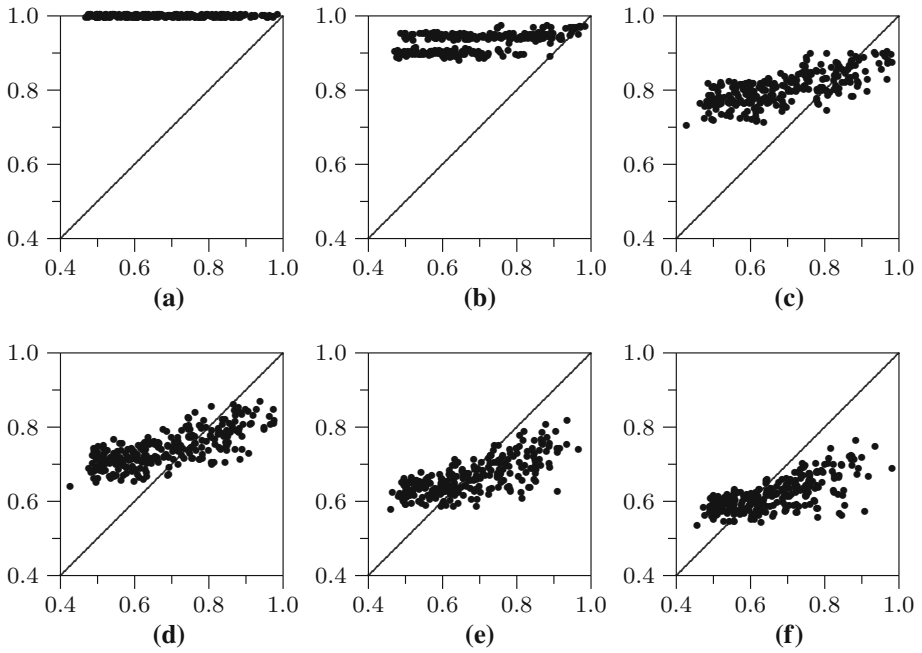

Fig. 6 Relation between the estimated class probability $\widehat{q_{max}}$ (y-axis) and true (x-axis) class probability q_{max} for the best rules constructed from artificial data sets. **a** Relative frequency, **b** $m = 2$, **c** $m = 10$, **d** $m = 20$, **e** $m = 50$ and **f** $m = 100$

Table 2 Comparison of rules obtained from artificial data sets with different values for m: the average true class probability, Spearman correlation between the true probability and the estimate, and the root mean squared error of the estimate

m	Avg. accuracy	Spearman	RMSE
0	0.68	0.0000	0.3509
2	0.68	0.6147	0.2780
10	0.68	0.7475	0.1674
20	0.68	0.7903	0.1214
50	0.67	0.7919	0.0995
100	0.66	0.7426	0.1053

with lower true probability, but pessimistic for rules with higher true probability (Fig. 6).[1] It seems that m-estimate lowers the estimated quality by roughly the same amount for all rules, which can not adjust the estimates to lie closer to the ideal diagonal line representing the perfect correlation.

Table 2 compares the measured evaluation functions by

- the average true prediction accuracy of the induced rules, which reveals the quality of the evaluation function as search heuristics;
- the Spearman correlation coefficient between the true rule accuracies $Q(r)$ and estimated values $\widehat{Q_m(r)}$, that shows the quality of the rule ordering, which is crucial when rules are used for classification, where we need to distinguish between "stronger" and "weaker" rules;

[1] We obtained similar results in experiments with other ways of constructing artificial data sets.

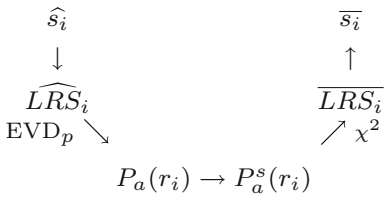

Fig. 7 An outline of the proposed procedure

– the root mean squared error (RMSE) between $Q(r)$ and $\widehat{Q_m(r)}$ which indicates the rule's accuracy when used as probabilistic predictor.

The first column of Table 2 suggests that lower values of m give (marginally) better rules than higher values. However, higher values of m score better in terms of the Spearman correlation and give more accurate probability estimates.

Although the m-estimate with a suitably tuned m can considerably decrease the error of the estimated probabilities, this effect seems to come from reducing the optimism by pushing the predicted probabilities towards the average, while the correlation between the true and the estimated probability remains rather poor. Thus, m-estimate and many other similar techniques are not a satisfactory solution to the problem of overfitting, wrong rule quality estimates and optimistic probability predictions.

4.1 EVC algorithm for relative frequencies

The outline of the proposed procedure is illustrated in Fig. 7. It differs slightly from the general algorithm described in Sect. 3.3, as it does not directly correct the evaluation s_i/n_i of a rule, but another well related measure of quality, likelihood ratio statistics (*LRS*) presented in Eq. (14). This is needed since the use of extreme value distributions requires that the values of a random variable come from a fixed distribution (Coles 2001). *LRS*, which is distributed according to $\chi^2(1)$, disregarding the number of covered positive and negative examples, fulfills this criterion, while s_i/n_i, which comes from $\beta(s_i, n_i - s_i)$ and is different for each rule, does not. The two measures are well correlated for rules that cover the same number of examples. The step $P_a(r_i) \rightarrow P_a^s(r_i)$ adjusts probability $P_a(r_i)$ considering that *LRS* is symmetrical with respect to positive and negative examples, e.g. a rule covering only negative examples would also have high *LRS*. As these high *LRS* values are disregarded in our method, we need to accordingly adjust the probability P_a. The specifics of the algorithm are given in the following step-by-step description.

Step 1: From $\widehat{s_i}$ to $\widehat{LRS_i}$.

Let s be the number of positive examples covered by some rule and let s^c be the number of positive examples not covered by the rule. Similarly, let n be the number of all covered examples and n^c be the number of examples that are not covered by the rule. *LRS* for 2×2 tables derived by Dunning (1993) is:

$$LRS = 2\left[s \log \frac{s}{e_s} + (n-s) \log \frac{n-s}{e_{n-s}} + s^c \log \frac{s^c}{e_{s^c}} + (n^c - s^c) \log \frac{n^c - s^c}{e_{n^c - s^c}}\right], \quad (14)$$

where e_x is the expected value of x. For instance, e_s is computed as $n\frac{s+s^c}{n+n^c}$. Note that a similar formula for LRS, without the last two terms, was used in Clark and Niblett (1989)

1. Let $L = 1$ (L is the maximum rule length).
2. Permute values of class in the data.
3. Learn a rule on this data (using LRS as evaluation measure), where the maximum length of rule is L.
4. Record the LRS of the rule learned.
5. Repeat steps 2-4 to collect a large enough (say 100) sample of LRSs
6. Set $\beta(L) = 2$ and $\mu(L) = median + 2 \ln \ln 2$, where $median$ is the median of stored LRSs (see Appendix A for details).
7. If $\mu(L) > \mu(L-1)$, then $L = L + 1$ and return to step 2.

Fig. 8 The algorithm for computing parameters of the Gumbel distributions

and Clark and Boswell (1991) for computing significance of rules. We prefer to use formula from Eq. (14), because it considers all examples in the estimation of likelihood and not just covered examples.

We compute the $\widehat{LRS_i}$ for all rules where the number of positive examples is higher than the expected number of positive examples, $s_i \geq e_s$, otherwise LRS is set to 0.

Example We have a data set with 20 examples where the prior probability of the positive class is 0.5. Learning from that data, the rule search algorithm found a rule r_i with two conditions which covers 10 examples with 8 of them belonging to the positive class. Its LRS is, according to (14), 7.7.

Step 2: From $\widehat{LRS_i}$ to $P_a(r_i)$.

Since LRS is distributed according to $\chi^2(1)$, its extreme value distribution is the Gumbel distribution (Generalized Extreme Value distribution Type-I). The cumulative distribution function of the Gumbel distribution is

$$F(x; \mu, \beta) = e^{-e^{-(x-\mu)/\beta}}. \tag{15}$$

Parameters μ and β depend upon the number of rules considered by the search, which in turn depends upon the rule length and the data set and, of course, the search algorithm. Due to their independence of the actual rule, we can compute values $\mu(L)$ and $\beta(L)$ for different rule lengths before we begin learning, using the algorithm shown in Fig. 8.

During learning we use the cumulative Gumbel distribution function to estimate $P_a(r_i)$ for each candidate rule using the pre-computed parameters.[2]

Example (continued) Say that algorithm from Fig. 8 found $\mu(2) = 3$ and $\beta(2) = 2$ (remember that rule r has two conditions). The curve with such parameters is depicted in Fig. 9a, so the probability $P_a(r_i)$ for the rule from our example corresponds to the shaded area right of $LRS = 7.7$. $P_a(r)$ equals approximately 0.09.

Step 3: From $P_a(r_i)$ to $P_a^s(r_i)$

$P_a^s(r_i)$ is computed by multiplying $P_a(r_i)$ by 2. As approximately half of the hypotheses are automatically dismissed—their LRS is set to 0, only a half of the hypotheses is covered

[2] Note that using LRS at a given rule length will always order rules the same as would $\overline{LRS}$. However, as we will be using $\overline{s_i}/\widehat{n_i}$ in the actual learning phase, in order to correctly estimate parameters of Gumbel distribution, measures $\overline{s}/\widehat{n}$ and $\overline{LRS_i}$ should be well correlated.

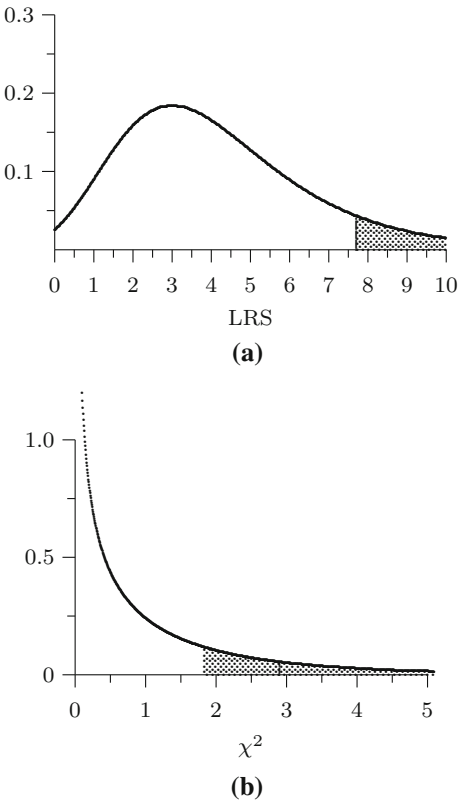

Fig. 9 Probability density functions. **a** Gumbel distribution ($\mu = 3$, $\beta = 2$) and **b** χ^2 with 1 degree of freedom

under the $\chi^2(1)$ curve, and hence the tail probabilities in this curve are twice as large as probabilities in the normal $\chi^2(1)$ curve.

Example (continued) In our example $P_a^s(r_i)$ equals $2 * 0.09 = 0.18$.

Step 4: From $P_a^s(r_i)$ to $\overline{LRS_i}$.

To compute $\overline{LRS_i}$ we need to do the opposite from the second step. Looking at the $\chi^2(1)$ distribution (Fig. 9b), we need to find such a value of $\overline{LRS_i}$ that the area under the curve to the right of it will equal the computed $P_a^s(r_i)$. In other words, the shaded area under the curve in Fig. 9b should equal the shaded are under the curve in Fig. 9a multiplied by 2.

Example (continued) The corresponding $\overline{\text{LRS}_i}$ for our examples as read from Fig. 9b is 1.82. Note that this is much less than $LRS = 7.7$, which we computed directly from the data and which would essentially be used by an unmodified rule induction algorithm.

Step 5: From $\overline{LRS_i}$ to $\overline{s_i}$.

The remaining task is trivial: compute $\overline{s_i}$ from the formula for $\overline{LRS_i}$ using an arbitrary root finding algorithm. In our task we are correcting probability estimates based on relative frequencies, so we shall compute them by dividing the corrected $\overline{s_i}$ by $\widehat{n_i}$.

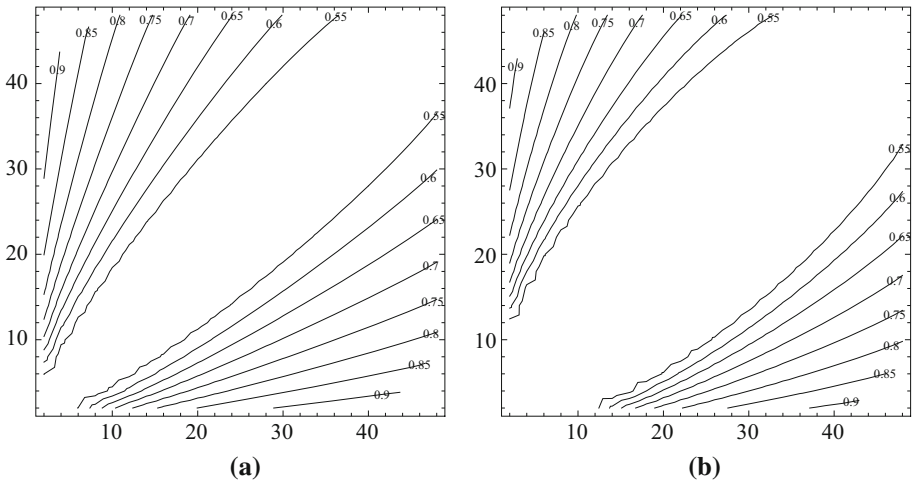

Fig. 10 Coverage space for EVC with different values of parameters in the Gumbel distribution. Labels on isometrics correspond to corrected relative frequencies. The horizontal axis is the number of covered negative examples, the vertical axis is the number of covered positive examples. Upper left and lower right parts are symmetric since they correspond the cases in which one or another class contains the majority of the examples covered by the rule. **a** $\mu = 3$, $\beta = 2$ and **b** $\mu = 10$, $\beta = 2$

Example (conclusion) We used Brent's method (Atkinson 1989) to find that $\overline{LRS_i} = 1.82$ corresponds to $\overline{s_i} = 6.5$. The rule covers ten examples, so the corresponding class probability is $6.5/10 = 0.65$. Note that this estimate is quite smaller than the uncorrected 0.8.

4.2 Extreme value corrected relative frequency in coverage space

Coverage space, introduced by Fürnkranz and Flach (2005), is a visualization of rule evaluation metrics and their behavior at different coverages and ratios between positive and negative examples. The isometrics in such diagram connect different combinations of covered positive and negative examples that are given the same quality by the selected measure.

Figure 10 shows isometrics for EVC using two different extreme value distributions. In both cases we have 50 positive and 50 negative examples. In the left diagram we used Gumbel with location parameter $\mu = 3$ and in the right diagram μ was set to 10. Higher location parameter is usually used when the algorithm compared a larger number of candidate hypotheses, therefore we can look at the latter metric also as one for rules with more conditions (where search was deeper), while the former (on the left) as one for rules with less conditions.

Both diagrams contain a large central space where the qualities of rules are less than 0.55. These rules have high probability of being found by chance, hence their qualities are penalized the most. Due to the higher location parameter of EVD in the right diagram, its central space is larger, since the probability to find by chance an equal ratio of positives and negatives increases by extending the search. The diagrams also nicely show that rules of different lengths and with the same covered class distribution get a different correction. Therefore, longer rules are penalized more, as their expected optimism is higher due to a larger search space.

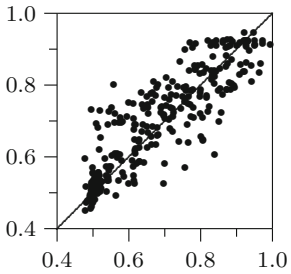

Fig. 11 Relation between the corrected (y-axis) and the true (x-axis) class probability

4.3 Experiments

We have tested the algorithm on artificial data described in introduction and on a selection of data sets from the UCI repository (Lichman 2013). In all experiments we used a modified version of CN2 (Clark and Boswell 1991) for learning unordered rules implemented as a component for the rule-based learner in the machine learning system Orange (Demšar et al. 2013).[3]

4.3.1 Artificial data set

The results of using the corrected measure on the same artificial data as in Fig. 6 are shown in Fig. 11. For each data set, we learned a single rule with the same beam-search algorithm and EV-corrected relative frequency. The estimated class probabilities are nicely scattered close to the diagonal axis, which is a clear improvement in comparison with the results from Fig. 6. This is also confirmed by the quantitative measure of fit: the average true probability is the same as the highest values in Table 2, the root mean squared error is better than that of m-estimates, and the Spearman coefficient is clearly superior.

4.3.2 UCI data sets

The previous experiment demonstrated favorable properties of the EVC method when compared with the m-estimate on artificial data. In this section, we will observe whether we can use EVC to improve accuracy estimations of individual rules on a set of 39 UCI data sets (Lichman 2013).[4]

We compared EVC with six other methods for evaluating rules. Rules were learned with a modified version of CN2, where the evaluation function was replaced accordingly. Each rule was evaluated on the full data set to enable comparison between estimated accuracies and true accuracies. To prevent learning the same rule after removing covered examples, a rule had to cover at least one uncovered example. We did not use any pre-pruning or post-pruning techniques.

In the first method, m was fixed to 2, an often used value, since it resembles the Laplace correction of probability in two-class problems. Fürnkranz (2004) experimentally showed that a value close to 2 (1.6065) as m performs best for evaluating individual rules. In the second

[3] More details about implementation and a link to git server with source code can be found at www.ailab.si/wiki in the *Papers* section.

[4] See Table 6 in "Appendix Appendix B:" for the list of datasets.

method, m was set to 22, the value determined to perform best on the UCI data sets (Janssen and Fürnkranz 2010). The third method M(Pro) implements the idea of Domingos (1999), who suggested a formula for selecting m value from the number of attributes in the domain and the number of conditions in the rule: $m = 1 + \log(l \cdot b \cdot a \cdot v)$, where l is the length of the rule being evaluated, b is the beam width, a is the number of attributes in the domain and v is the maximum number of values of any attribute. Such process-oriented selection of m-value is in a way analogous to our approach, since the correction of the probability depends on the number of different rules that were tested during the learning phase. In the fourth method M(IC), we selected the "optimal" m parameter with internal-cross validation (IC) maximizing classification accuracy of the best rule. The next method (Split) uses $m = 2$, however learns only from 70% of data. The remaining 30% were used to obtain "unbiased" class probabilities. The last method uses uncorrected likelihood ratio statistics (LRS) to search for the best rules and then calculates classification accuracy of the learned rules with relative frequency.

The main difference between artificial and real data is, however, that the true qualities of induced rules are unknown, therefore we need to estimate them. We separated each data set to two equally sized sets: learning and testing set. The first one was used to learn a set of rules that cover all examples from the learning set for each class. Then, we used the examples from the other fold to count the number of positive and the number of all examples covered by the induced rules. The ratio between positive examples and all examples was taken to be the true positive class probability; although this is still only an estimate, it is unbiased, since it is computed from the test data.

We compared methods using the same measures of fit as before: average accuracy of individual rules, Spearman correlation between estimated and true class probability, and root mean squared error (RMSE) of estimated probabilities given true probabilities. These measures are complementary. RMSE (and the related Spearman correlation) may favor the algorithms that induce rules with higher number of covered examples where the probabilities are easier to estimate. Applying the EVC may introduce such bias and stir the algorithm towards finding general rules of mediocre accuracy instead of interesting specific rules of high accuracy. Testing this directly by checking whether an algorithm finds a prescribed number of rules above some prescribed accuracy threshold is infeasible for a general collection of data sets without necessary domain knowledge. As a proxy, we compute the average true accuracies of five rules with highest estimated accuracy. Averaging accuracy over all rules would not measure the capability of the algorithm to spot the rare rules with high accuracy, while observing only the single most accurate rule would decrease the robustness of experiments. The threshold of five rules seems a reasonable (although arbitrary) choice in our experimental design.

The first three rows in Table 3 show average ranks of the three measures aggregated over 39 data sets. The second three rows contain ranks of properties of learned rules: average rule length (shorter is better), average rule coverage (higher coverage results in lower rank), and number of induced rules (less rules is better). Table 6 in "Appendix Appendix B:" contains RMSE values over all heuristics and all data sets. We see that the EVC has best rank in the case of RMSE, therefore the probability estimates by our method are more accurate than those by any m in the m-estimate measure. The average rank of EVC is 1.644, while ranks of other estimates are 5.638 ($m = 2$), 3.333 ($m = 22$), 4.103 (Pro), 4.546 (IC), 4.575 (Split), and 4.161 (LRS). The differences are highly significant (the Friedman test gives $p < 0.001$), and the Bonferroni–Dunn test at $\alpha = 0.05$ shows that EVC is significantly better than any method we compared it with.

Table 3 Average ranks of accuracies, RMSEs, Spearman correlations, rule lengths, rule coverages, and the number of induced rules on 39 UCI datasets

Measure	EVC	M(2)	M(22)	M(pro)	M(IC)	M(Split)	LRS
Accuracy	3.109	3.460	4.494	3.305	3.730	5.862	4.040
RMSE	1.644	5.638	3.333	4.103	4.546	4.575	4.161
Spearman	3.109	5.385	3.552	4.207	4.690	4.874	2.184
Rule length	2.075	4.960	4.569	4.098	5.264	3.253	3.782
Rule cov.	1.879	5.874	2.397	4.466	4.626	5.256	3.483
No. of rules	2.757	6.287	3.960	5.443	5.489	2.293	1.770

The first column contains results of learning with EVC, the next 5 are different variants of m-estimate, in the last column, LRS was used to evaluate rules. The differences between methods are highly significant (the Friedman test gives $p < 0.001$ in all three cases). The Bonferroni–Dunn post-hoc test shows that EVC has significantly smaller RMSE when compared with other methods. In the case of Spearman, EVC does better than other methods, but the difference between EVC and M(22) is not significant. Methods EVC and M(pro) produce most accurate rules, closely followed by M(2)

The general differences according to Friedman are also significant in the cases of accuracy and Spearman correlation ($p < 0.001$). Methods EVC, M(Pro), M(2), and M(IC) learn most accurate rules, however the differences among these methods according to Bonferroni–Dunn are not significant. From these results, we conclude that using EVC leads to finding rules that are at least as accurate as rules produced by other methods, and at the same time have significantly better estimations of classification accuracy (as implied by RMSEs).

The LRS method achieved best rank in Spearman correlation of all, followed by EVC, and M(22). These methods produce the most reliable estimations of accuracy for ranking, which seems to be related to the number of covered examples by rules, as the same three methods learn rules with the most covered examples. Furthermore, since EVC enforces larger corrections to quality estimations of longer rules, it consequently learns the shortest rules. The number of rules produced by EVC is on average lower than the number of rules of other m-estimate based methods. The only exceptions are M(Split), which has less examples to learn from, and LRS.

These experiments show the main advantage of the EVC heuristics: it guides learning to induction of rules with high accuracy, and, at the same time, the EVC's estimates of class probabilities, when compared with other methods, are significantly closer to their true values.

4.3.3 Rules in a global classifier

Rules are used to spot and explain local patterns. Sometimes we would like to construct a classifier from rules, which often turns out to be problematic due to conflicts between rules and some resolution principle needs to be applied. In the original CN2, the sum of class distributions of rules was computed to select the most probable class for the example being classified (Clark and Boswell 1991). However, Lindgren (2004) showed that CN2's approach does not handle conflicts well and suggested some alternatives.

We chose to linearly combine rules as suggested by Friedman and Popescu (2008) due to two reasons. First, as each rule is easy to understand, so should be a linear ensemble of these rules. Second, even though a linear model can model additive patterns in data, it fails to model certain types of interactions between attributes and class values. A combination of rules and linear models should thus result in more accurate models.

We used logistic regression, one of the most common linear models for classification problems. Each learned rule was encoded as a binary attribute with value 1 if an instance is covered by the rule and 0 otherwise. We extended the set of rule-based attributes with the original set of attributes, as implemented by Friedman and Popescu (2008) in their RuleFit algorithm. They claim that using both types of attributes together might increase the accuracy and interpretability, because it is hard to build a rule-based model where relation between attributes and class is linear.

A major problem of using rules as attributes is that it inherently contains a typical example of overfitting: we learn rules from learning data and then use the same data to infer the parameters of rule-based attributes. Using attributes in learning that have been induced from the same data is very problematic (Domingos 2000).

A possible solution is to learn rules and logistic regression on separate data sets, which is viable when data is abundant. Our proposed solution works also with small data sets, hence we need to use all data for training. To avoid overfitting, we propose to additionally penalize coefficients for rule-based attributes, as described later on in this section.

In the case of binary response (two classes), the logistic regression model uses a logit function to link the linear formula and probability. Let $y \in \{-1, 1\}$ be a binary class value, $\boldsymbol{x}$ a vector of original attributes, x_i the value of ith attribute, $r_i(\boldsymbol{x}) \in \{0, 1\}$ a binary rule-based attribute, a_i and b_i parameters of the model, and $\phi(x)$ the logit function:

$$\phi(x) = \frac{1}{1 + e^{-x}}. \tag{16}$$

Then, if our domain contains M original attributes and K rule-based attributes, the posterior class probability estimated by logistic regression is:

$$\hat{p}(y|\boldsymbol{x}) = \phi\left(a_0 + \sum_{i=1}^{M} a_i x_i + \sum_{i=1}^{K} b_i r_i(\boldsymbol{x})\right). \tag{17}$$

Let $\boldsymbol{e}$ be a vector representing an example containing a constant value and all attribute values: the original attribute values and attribute values derived from rules. If $\boldsymbol{w}$ is a vector containing all parameters (a_i and b_i), the formula simplifies to:

$$\hat{p}(y|\boldsymbol{e}) = \phi\left(\boldsymbol{w}^T \boldsymbol{e}\right). \tag{18}$$

A standard approach to induce the parameters $\boldsymbol{w}$ from learning data is to minimize the log-likelihood function. In the case of the L2-regularized logistic regression, the loss function is the penalized negative log-likelihood function:

$$L(\boldsymbol{w}) = -\sum_{i=1}^{n} log\left(\phi\left(y_i \boldsymbol{w}^T \boldsymbol{e_i}\right)\right) + \frac{\lambda}{2}\boldsymbol{w}^T \boldsymbol{w}, \tag{19}$$

where n is the number of all learning examples and λ is the penalty factor. The gradient of the loss function is

$$\nabla_{\boldsymbol{w}} L(\boldsymbol{w}) = \sum_{i=1}^{n} y_i \boldsymbol{e_i}\left(\phi\left(y_i \boldsymbol{w}^T \boldsymbol{e_i}\right) - 1\right) + \lambda \boldsymbol{w}. \tag{20}$$

To find the minimum of (20), we used the trust region Newton optimization method (Lin et al. 2008), which is the same as the algorithm used in the LIBLINEAR package (Fan et al. 2008).

As previously mentioned, we suggest to impose an additional penalty term for rule-based attributes to prevent ovefitting that stems from using the data twice, first to learn attributes and then to learn weights for these attributes:

$$L(\boldsymbol{w}) = -\sum_{i=1}^{n} log\left(\phi\left(y_i \boldsymbol{w}^T \boldsymbol{e_i}\right)\right) + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w} + \boldsymbol{\gamma}^T|\boldsymbol{b}|. \tag{21}$$

A single value γ_k in the $\boldsymbol{\gamma}$ vector should be assigned the value which prevents logistic regression optimizer from overfitting to instances covered by the kth rule. We can use EVC-corrected accuracy q_k of the kth rule to compute γ_k. A corrected classification accuracy of a rule specifies the class probability among examples covered by the rule, therefore the average predicted class probability on these instances should not exceed the probability estimated by EVC. If the gradient $\partial L(\boldsymbol{w})/\partial b_k$ was 0 when predicted probabilities $\hat{p}(y|e)$ of instances covered by rule r_k equal the rule classification accuracy q_k, this would prevent the logistic regression optimizer to increase the parameter b_k after reaching average predicted probability q_k for these examples.

Let q_k be the accuracy of kth rule, n_k the number of covered instances by the rule and s_k the number of covered positive instances. To implement the idea from the above paragraph, γ_k needs to be set as $\gamma_k = -\partial L(\boldsymbol{w})/\partial b_k$ when predicted probabilities on covered examples equal q_k. This, after some basic algebra, simplifies to:

$$\gamma_k = s_k - q_k n_k \tag{22}$$

If the measure for rule classification accuracy contains no penalty, say relative frequency $q_k = s_k/n_k$, the additional penalty in logistic regression will be 0. When some penalty is introduced and the difference between g_k and s_k/n_k increases, γ_k will also increase.

Tables 4 and 7 (the latter in "Appendix Appendix B:") contain results of 6 methods tested on 31 UCI domains with a binary class that were evaluated with four score functions: classification accuracy (CA), area under curve (AUC), Brier score, and the logarithmic loss (LogLoss). Rules were learned with the same learner as in the previous section. The methods are:

LR is the logistic regression learned from the original set of attributes.

LRR-EVC is the logistic regression learned from rule-based attributes and original set of attributes. EVC was used to learn rules. This method does not additionally penalize rule-based attributes, which is the same as in the RuleFit algorithm.

LRRP-EVC is the same as LRR-EVC with additional penalty in the log-likelihood loss function, as described in Eq. (21). Comparing LRRP-EVC to LRR-EVC will show whether additional penalties are useful or not.

LRRP-M2 is the same as LRRP-EVC, however with m-estimate ($m = 2$) instead of EVC in rule learning. Both, $m = 2$ and EVC lead to accurate rules (see previous section). Comparing this method and LRRP-EVC will show whether $m = 2$ estimations can also be used to infer penalty.

LRRS-EVC splits the learning data into two equal data sets. The first half is used to learn rules, the second to learn parameters of logistic regression. Comparing LRRP-EVC to LRRS-EVC and to the following method will answer whether is it better to additionally penalize logistic regression or simply learn logistic regression on a separate data set.

LRRS-M22 is the same as LRRS-EVC, but m-estimate with $m = 22$ was used in rule learning. We decided to use $m = 22$, because this value seemed to perform well on UCI domains in previous experiments (Janssen and Fürnkranz 2010).

Table 4 Average ranks of several variants of logistic regression on 31 UCI data sets with binary class. Rows correspond to different score functions

Measure	LR	LRR-EVC	LRRP-EVC	LRRS-EVC	LRRP-M2	LRRS-M22
CA	3.081	3.613	2.468	4.049	4.032	3.758
AUC	3.323	3.419	2.468	4.290	3.581	3.919
Brier	3.097	4.065	2.229	4.016	3.742	3.855
LogLoss	3.048	4.161	2.145	4.000	3.774	3.871

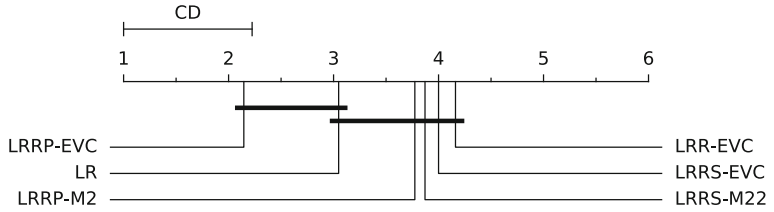

Fig. 12 Critical differences for the LogLoss measure from Table 4. We used Bonferroni–Dunn test at $\alpha = 0.05$

The λ of the L2-regularization term in the LR method was selected for each data set with an internal cross-validation procedure. The remaining 5 methods used the same λ as LR.

The average ranks of results are given in Table 4, the corresponding critical differences are visualized in Fig. 12. According to all scores, the LRRP-EVC (logistic regression with rules and EVC and additional penalty) performs best, followed by the simple logistic regression without rules (LR). According to Friedman test, the differences between ranks are significant for all measures ($p < 0.01$), and Bonferroni–Dunn test at $\alpha = 0.05$ shows that LRRP-EVC is significantly better in the case of Brier and LogLoss than any other combination of logistic regression and rules. The difference between LR and LRRP-EVC is, however, not significant. Table 7 in "Appendix" contains per data set results of AUC and provides additional explanation why and when LRRP-EVC is better than LR.

The difference between LRRP-EVC and LRR-EVC supports our initial conjecture that learning rules and logistic regression parameters on the same data leads to overfitting and warrants the use of penalty in Eq. (21). Furthermore, as LRRP-EVC is significantly better than LRRP-M2, the EVC heuristic seems to be more appropriate than m-estimate for learning rules that are to be used as attributes. LRRP-EVC is also significantly better than LRRS-EVC and LRRS-M22, even when learning on separate data sets is theoretically correct, because the data sets were not large enough to enable splitting. Finally, the average ranks of LRRP-EVC are better than the ranks of LR, however according to the Bonferroni–Dunn test, the differences are not significant. From the Table 7, that contains AUC's case by case, we can observe that in most cases (24 out of 31) the difference between methods is negligible (within 1%). In the other 7 domains, adding rules to the model notably helps, since average difference in AUC between methods is approximately 7%. It is therefore likely that in the 24 domains the original set of attributes is sufficient and new derived attributes are not needed, while they are needed in the remaining 7.

The above experiment showed that imposing penalty based on corrected accuracies improves the accuracy of a global classifier, when everything else remains the same. In Table 5, we present the results of a comparison of our algorithm LRRP-EVC and LRRP-EVC(R), the latter using only rule-based attributes, to a selected set of state-of-the art rule

Table 5 A comparison of LRRP-EVC to some other state-of-the art rule-learning algorithms. LRRP-EVC uses rules and linear factors, while LRRP-EVC(R) uses only rule-based attributes

Measure	LRRP-EVC	LRRP-EVC (R)	RuleFit	MLRules	CN2	JRip
CA	2.017	3.983	2.833	2.700	4.733	4.733
AUC	1.433	3.033	2.367	4.867	3.567	5.733
LogLoss	1.267	2.467	3.233	5.200	3.033	5.800

learning algorithms. The first selected algorithm is the RuleFit (Friedman and Popescu 2008) algorithm, since our method is based on this algorithm. We used the RuleFit3 version with the default values of parameters, where rules and linear factors are combined, rules are generated from a decision tree ensemble and Lasso penalty is used to prevent overfitting. A similar approach that also builds a linear model of rules by minimizing the negative likelihood is MLRules (Dembczynski et al. 2008), a specific version of the ENDER algorithm (Dembczynski et al. 2010). MLRules incrementally generates one rule at a time, where each rule minimizes the error of the global model. The parameters of MLRules were set as suggested in their original paper. Finally, we compare our algorithm with the Orange implementation of the unordered CN2 algorithm (Clark and Boswell 1991) and JRip, a Weka implementation of the Ripper algorithm (Cohen 1995).

In terms of classification accuracy, LRRP-EVC achieved the best rank among all methods, MLRules was second, RuleFit third, LRRP-EVC(R) fourth, followed by CN2 and JRip. According to Bonferroni–Dunn test at $\alpha = 0.05$, the differences between LRRP-EVC, MLRules, and RuleFit are not significant. The difference between LRRP-EVC and RuleFit is also not significant in AUC and LogLoss cases, although LRRP-EVC did achieve better average ranks. As LRRP-EVC and RuleFit are similar algorithms and use the same attributes (rules and linear factors), the additional penalty in LRRP-EVC is probably the reason for the difference. Furthermore, LRRP-EVC(R) with only rule-based attributes performed worse than LRRP-EVC, RuleFit and MLRules. This result shows that adding linear factors is indeed useful. MLRules, however, can achieve good results even without linear factors. We suppose that the strategy of MLRules, where rules are generated to decrease the error of the global model, will induce such rules that can substitute linear factors.

5 Conclusion

We presented a general procedure for correcting the overly optimistic estimates of qualities of hypotheses induced by machine learning. While the general principle is simple enough and directly applicable to toy learning algorithms like classification by the value of a single attribute, its practical application to the real-world machine learning algorithms may require a few technical tricks. For a case study, we demonstrated a corrected version of the standard rule learning algorithm CN2. The results on artificial data constructed specifically for proving the correctness of the approach are excellent. The experiments on the real-world and artificial examples from the UCI confirmed the enhanced performance of the algorithm with respect to the root mean squared error of the predicted probability and accuracy of the induced rules. For the final test, we implemented a global classifier based on logistic regression using rules as features and demonstrated the importance of EVC estimates in the process of fitting the parameters of the logistic model. Using m-estimate instead or even neglecting

correction of accuracies altogether lead to significantly worse classification models in terms of classification accuracy, area under curve, Brier score, and logarithmic loss.

The general correction procedure assumes that optimism of the method can be estimated with randomization. The open question is whether techniques such as bootstrapping could be used to estimate the optimism on the learn data itself, as such method would avoid the difference between optimism on random data and on the actual data itself. This question points to one possible direction for further work on correction methods.

Acknowledgements This work was partly supported by the Slovene Agency for Research and Development (ARRS).

Appendix A: Computing parameters of extreme-value distribution

Section 4.1 describes an algorithm for computing extreme distributions of rules learned from random data which involves calculating the parameters of extreme value distribution for a vector of maxima of evaluations of rules distributed by χ^2 with 1 degree of freedom. The limiting distribution of all χ^2 distributions is Gumbel distribution (Fisher and Tippett 1928; Gumbel 1954; Gumbel and Lieblein 1954). The cumulative distribution function of this distribution is

$$P(x < x_0) = e^{-e^{\frac{\mu - x_0}{\beta}}}, \tag{23}$$

where μ and β are parameters of the distribution. Distribution's mean, median, and variance are

$$\text{mean} = \mu + \beta\gamma, \quad \text{median} = \mu - \beta \ln \ln 2, \quad \text{var} = \pi^2\beta^2/6, \tag{24}$$

where γ is Euler-Mascheroni constant 0.57721. The natural way to compute the parameters μ and β from the sample would be to first estimate the variance from the data and use it to compute β, followed by the estimation of μ from the sample's mean or median. However, error of estimation of variance and mean propagates to estimations of parameters μ and β, where variance is a bigger problem than mean, as it is used for estimation of both parameters.

Gupta (1960) showed that for p independent and identically distributed values taken from χ^2 with one degree of freedom, where p is large, the following properties holds for their maxima M:

$$E(M) = 2 \ln p - \ln \ln p - \ln \pi + 2\gamma \tag{25}$$

$$m(M) = 2 \ln p - \ln \ln p - \ln \pi - 2 \ln \ln 2 \tag{26}$$

$$\sigma(M) = \sqrt{2/3\pi^2} \tag{27}$$

Since $\sigma(M)$ is independent of the number of values (or the number of considered rules, in our case), combining Eqs. (24) and (27) gives $\beta = 2$. We thus only need to estimate the remaining parameter μ. In our algorithm we computed the median from the vector of maximum values, so μ equals the median plus $2 \ln \ln 2$.

Appendix B: Experimental results on UCI domains

See Tables 6 and 7.

Table 6 RMSE values

Data set	EVC	M(2)	M(22)	M(Pro)	M(IC)	M(Split)	LRS
Abalone = 0	0.018	0.089	0.024	0.056	0.039	0.025	0.009
Abalone = 1	0.029	0.130	0.038	0.094	0.036	0.060	0.016
Adult =<= 50K	0.043	0.090	0.045	0.074	0.090	0.049	0.043
Adult => 50K	0.072	0.265	0.089	0.166	0.264	0.114	0.063
Ailerons =< avg	0.038	0.082	0.038	0.056	0.050	0.047	0.025
Ailerons => avg	0.015	0.046	0.020	0.032	0.029	0.027	0.033
Auto-mpg = 0	0.046	0.084	0.066	0.073	0.084	0.070	0.165
Auto-mpg = 1	0.044	0.107	0.080	0.082	0.108	0.063	0.141
Balance-scale = L	0.119	0.112	0.142	0.121	0.112	0.132	0.115
Balance-scale = R	0.100	0.133	0.058	0.136	0.133	0.323	0.043
Breast-cancer-wis = benign	0.019	0.028	0.029	0.022	0.022	0.037	0.075
Breast-cancer-wis = malign	0.066	0.050	0.107	0.060	0.050	0.075	0.060
Breast-cancer = no-recurrence-events	0.075	0.182	0.079	0.163	0.192	0.200	0.271
Breast-cancer = recurrence-events	0.199	0.315	0.218	0.235	0.316	0.272	0.259
Bupa = 1	0.093	0.266	0.107	0.179	0.109	0.158	0.178
Bupa = 2	0.075	0.233	0.112	0.182	0.112	0.220	0.217
Car = acc	0.153	0.213	0.111	0.173	0.213	0.168	0.162
Car = good	0.115	0.144	0.066	0.115	0.086	0.135	0.172
Car = unacc	0.057	0.159	0.108	0.157	0.159	0.181	0.169
Car = v-good	0.136	0.156	0.064	0.113	0.156	0.161	0.180
Chronic-kidney-disease = ckd	0.063	0.111	0.066	0.098	0.110	0.088	0.069
Chronic-kidney-disease = notckd	0.047	0.063	0.108	0.052	0.301	0.054	0.031
Cmc = 1	0.121	0.219	0.115	0.152	0.134	0.200	0.147
Cmc = 2	0.051	0.217	0.079	0.156	0.219	0.083	0.128

Table 6 continued

Data set	EVC	M(2)	M(22)	M(Pro)	M(IC)	M(Split)	LRS
Cmc = 3	0.047	0.258	0.105	0.182	0.153	0.154	0.143
Coil = 0	0.007	0.041	0.033	0.039	0.017	0.024	0.021
Coil = 1	0.069	0.359	0.136	0.219	0.135	0.106	0.018
Crx = +	0.069	0.162	0.074	0.110	0.091	0.121	0.100
Crx = −	0.040	0.169	0.062	0.101	0.086	0.099	0.081
Forest = d	0.100	0.105	0.207	0.105	0.200	0.223	0.130
Forest = h	0.085	0.171	0.173	0.087	0.098	0.099	0.213
Forest = o	0.091	0.293	0.178	0.199	0.289	0.153	0.158
Forest = s	0.072	0.139	0.157	0.087	0.142	0.165	0.096
Galaxy = 0	0.047	0.133	0.071	0.093	0.133	0.250	0.092
Galaxy = 1	0.045	0.131	0.135	0.112	0.131	0.070	0.114
German = bad-credit	0.157	0.356	0.183	0.276	0.360	0.172	0.127
German = good-credit	0.051	0.165	0.085	0.137	0.164	0.097	0.154
Heart-disease = 0	0.067	0.126	0.064	0.090	0.124	0.092	0.227
Heart-disease = 1	0.087	0.186	0.085	0.146	0.086	0.158	0.160
Hepatitis = 1	0.160	0.347	0.108	0.228	0.164	0.210	0.242
Hepatitis = 2	0.063	0.099	0.052	0.080	0.055	0.089	0.298
Horse-colic = died	0.153	0.318	0.169	0.196	0.318	0.257	0.208
Horse-colic = euthanized	0.156	0.365	0.151	0.252	0.367	0.255	0.250
Horse-colic = lived	0.081	0.195	0.080	0.133	0.206	0.097	0.279
Housing = 0	0.038	0.100	0.029	0.077	0.029	0.053	0.052
Housing = 1	0.062	0.123	0.065	0.086	0.068	0.118	0.050
Imports-85 = 0	0.050	0.064	0.073	0.043	0.128	0.080	0.133

Table 6 continued

Data set	EVC	M(2)	M(22)	M(Pro)	M(IC)	M(Split)	LRS
Imports-85 = 1	0.098	0.157	0.152	0.120	0.154	0.074	0.116
Indian-liver = 1	0.049	0.155	0.072	0.126	0.155	0.102	0.099
Indian-liver = 2	0.097	0.378	0.136	0.291	0.378	0.152	0.117
Ionosphere = b	0.116	0.148	0.147	0.109	0.151	0.166	0.088
Ionosphere = g	0.042	0.082	0.033	0.068	0.048	0.056	0.098
Iris = Iris-setosa	0.068	0.062	0.233	0.086	0.062	0.090	0.110
Iris = Iris-versicolor	0.074	0.081	0.184	0.089	0.117	0.178	0.081
Iris = Iris-virginica	0.050	0.073	0.142	0.072	0.142	0.061	0.085
Monks-1 = 0	0.200	0.230	0.186	0.189	0.230	0.173	0.176
Monks-1 = 1	0.040	0.272	0.200	0.245	0.272	0.157	0.190
Monks-2 = 0	0.110	0.175	0.155	0.138	0.175	0.205	0.164
Monks-3 = 0	0.051	0.167	0.116	0.155	0.167	0.174	0.231
Monks-3 = 1	0.117	0.181	0.154	0.162	0.181	0.178	0.144
Parkinsons = 0	0.104	0.263	0.147	0.152	0.136	0.232	0.147
Parkinsons = 1	0.036	0.086	0.038	0.064	0.086	0.068	0.102
Pima-indians-diabetes = 0	0.043	0.128	0.064	0.087	0.056	0.054	0.123
Pima-indians-diabetes = 1	0.077	0.266	0.099	0.180	0.146	0.106	0.099
Pop-failures = 0	0.178	0.510	0.180	0.328	0.501	0.174	0.145
Pop-failures = 1	0.016	0.052	0.045	0.050	0.052	0.026	0.153
Promoters = mm	0.156	0.340	0.249	0.276	0.328	0.485	0.344
Promoters = pp	0.141	0.140	0.231	0.131	0.192	0.306	0.371
Prostate = 0	0.108	0.210	0.129	0.140	0.112	0.167	0.211
Prostate = 1	0.079	0.192	0.073	0.109	0.073	0.106	0.279
Servo = 1	0.113	0.172	0.157	0.191	0.172	0.232	0.021

Table 6 continued

Data set	EVC	M(2)	M(22)	M(Pro)	M(IC)	M(Split)	LRS
Shuttle-landing-control = 1	0.064	0.322	0.256	0.296	0.322	0.415	0.195
Tic-tac-toe = n	0.107	0.238	0.144	0.177	0.236	0.241	0.142
Tic-tac-toe = p	0.049	0.143	0.091	0.103	0.143	0.148	0.161
Titanic = no	0.039	0.041	0.032	0.039	0.041	0.038	0.064
Titanic = yes	0.056	0.098	0.222	0.121	0.175	0.111	0.057
Vehicle = bus	0.081	0.157	0.105	0.111	0.159	0.093	0.121
Vehicle = opel	0.090	0.378	0.116	0.251	0.373	0.120	0.062
Vehicle = saab	0.084	0.287	0.086	0.175	0.086	0.131	0.122
Vehicle = van	0.084	0.181	0.149	0.132	0.127	0.107	0.150
Voting = democrat	0.042	0.048	0.055	0.042	0.048	0.061	0.065
Voting = republican	0.058	0.073	0.104	0.069	0.079	0.088	0.128
Wdbc = B	0.030	0.047	0.034	0.034	0.031	0.043	0.070
Wdbc = M	0.041	0.098	0.066	0.058	0.046	0.087	0.048
Wine = 1	0.074	0.078	0.194	0.082	0.194	0.105	0.127
Wine = 2	0.086	0.140	0.128	0.102	0.090	0.155	0.083
Wine = 3	0.091	0.132	0.185	0.092	0.128	0.127	0.082

Rows corresponds to a single class in a domain with at least 20 examples. Each dataset was split into two equally sized subsets. The first part was used to learn rules, the second to obtain unbiased estimates of accuracies in rules. RMSEs measure differences between accuracies estimated on learning data and unbiased estimates. The table's header enlists heuristics used to estimate accuracies on learn data: EVC and m-estimate (m values 2, 22, (pro)cess-oriented selection, internal cross-validation). The method Split learns with $m = 2$ on 70% of learning data and estimates classification accuracy on the remaining 30%. The method LRS uses likelihood ratio statistic to learn rules.

Table 7 Area under curve (AUC) values for 6 methods on 31 UCI domains

Measure	LR	LRR-EVC	LRRP-EVC	LRRS-EVC	LRRP-M2	LRRS-M22
Abalone	0.874	0.846	0.878	0.870	0.844	0.871
Adult	0.877	0.856	0.878	0.872	0.848	0.865
Auto-mpg	0.971	0.978	0.975	0.975	0.977	0.976
Breast-cancer	0.694	0.688	0.694	0.680	0.674	0.677
Breast-wis	0.995	0.990	0.995	0.992	0.992	0.992
Bupa	0.709	0.734	0.742	0.712	0.767	0.742
Chronic-kidney	0.998	1.000	0.998	0.999	0.999	0.999
Coil	0.698	0.659	0.703	0.689	0.636	0.668
Crx	0.925	0.916	0.936	0.927	0.916	0.923
Galaxy	0.994	0.984	0.995	0.987	0.990	0.987
Heart-disease	0.903	0.891	0.904	0.891	0.883	0.896
Hepatitis	0.850	0.825	0.847	0.820	0.818	0.821
Housing	0.929	0.929	0.934	0.927	0.933	0.931
Imports-85	0.973	0.970	0.973	0.972	0.972	0.970
Indian	0.752	0.710	0.740	0.731	0.710	0.736
Ionosphere	0.873	0.961	0.955	0.954	0.963	0.954
Monks-1	0.720	1.000	1.000	1.000	1.000	1.000
Monks-2	0.547	0.679	0.633	0.658	0.742	0.817
Monks-3	0.986	0.990	0.987	0.986	0.987	0.982
Parkinsons	0.896	0.919	0.917	0.896	0.935	0.907
Pima-indians	0.826	0.801	0.822	0.817	0.778	0.817
Pop-failures	0.942	0.912	0.943	0.921	0.880	0.906
Promoters	0.970	0.981	0.970	0.956	0.970	0.961
Prostate	0.850	0.821	0.842	0.834	0.822	0.838
Servo	0.985	0.985	0.985	0.970	0.985	0.970
Shuttle-landing	0.997	0.998	0.997	0.994	0.997	0.994
Thoraric	0.645	0.619	0.641	0.589	0.627	0.603
Tic-tac-toe	0.996	1.000	0.996	1.000	0.997	0.999
Titanic	0.752	0.768	0.765	0.765	0.768	0.766
Voting	0.989	0.990	0.991	0.989	0.992	0.988
Wdbc	0.990	0.992	0.991	0.989	0.993	0.991

The LRRP-EVC outperforms other methods, however it is not significantly better than LR (see Table 4). Comparing these two methods case by case reveals that in most cases (24 out of 31) the difference between LRRP-EVC and LR is negligible (within 1%). Otherwise rules help significantly, as the average difference on the remaining 7 data sets is approximately 7%. In those cases, the original set of attributes was insufficient and needed to be extended

References

Atkinson, K. E. (1989). *An introduction to numerical analysis*. New York: Wiley.

Bartlett, P., & Mendelson, S. (2003). Rademacher and Gaussian complexities: Risk bounds and structural results. *The Journal of Machine Learning Research*, *3*(5), 463–482.

Cestnik, B. (1990). Estimating probabilities: A crucial task in machine learning. In *Proceedings of the ninth European conference on artificial intelligence (ECAI'90)* (pp. 147–149).

Clark, P., & Boswell, R. (1991). Rule induction with CN2: Some recent improvements. In *Proceeding of the fifth European working session on learning (EWSL'91)*, Berlin (pp. 151–163).
Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning Journal*, *4*(3), 261–283.
Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the Twelfth international conference on machine learning (ICML'95)* (pp. 115–123).
Coles, S. (2001). *An introduction to statistical modeling of extreme values* (1st ed.). London: Springer.
Dembczynski, K., Kotlowski, W., & Slowinski, R. (2008). Maximum likelihood rule ensembles. In *Proceedings of the twenty-fifth international conference on machine learning (ICML'08)* (pp. 224–231).
Dembczynski, K., Kotlowski, W., & Slowinski, R. (2010). ENDER: A statistical framework for boosting decision rules. *Data Mining and Knowledge Discovery*, *21*, 52–90.
Demšar, J., Curk, T., Erjavec, A., Gorup, Črt, Hočevar, T., Milutinovič, M., et al. (2013). Orange: Data mining toolbox in Python. *Journal of Machine Learning Research*, *14*, 2349–2353.
Domingos, P. (1999). Process-oriented estimation of generalization error. In *Proceedings of the 16th international joint conference on artificial intelligence (IJCAI'99)* (pp. 714–719).
Domingos, P. (2000). Bayesian averaging of classifiers and the overfitting problem. In *Proceedings of the 17th international conference on machine learning (ICML'00)* (pp. 223–230).
Dunning, T. E. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, *19*(1), 61–74.
Džeroski, S., Cestnik, B., & Petrovski, I. (1993). Using the m-estimate in rule induction. *Journal of Computing and Information Technology*, *1*(1), 37–46.
Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., & Lin, C. J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, *9*, 1871–1874.
Fisher, R., & Tippett, L. (1928). Limiting forms of the frequency distribution of the largest or smallest member of a sample. *Mathematical Proceedings of the Cambridge Philosophical Society*, *24*, 180–190.
Friedman, J. H., & Popescu, B. E. (2008). Predictive learning via rule ensembles. *The Annals of Applied Statistics*, *2*(3), 916–954.
Fürnkranz, J. (2004). From local to global patterns: Evaluation issues in rule learning algorithms. In *Local pattern detection*, International Seminar (pp. 20–38). Dagstuhl Castle, Germany.
Fürnkranz, J., & Flach, P. A. (2005). ROC 'n' Rule learning—Towards a better understanding of covering algorithms. *Machine Learning*, *58*(1), 39–77.
Gumbel, E. J. (1954). *Statistical theory of extreme values and some practical applications*. National Bureau of Standards Applied Mathematics Series (US Government Printing Office) (p. 33).
Gumbel, E. J., & Lieblein, J. (1954). Some applications of extreme-value models. *American Statistician*, *8*(5), 14–17.
Gupta, S. S. (1960). Order statistics from the Gamma distribution. *Technometrics*, *2*, 243–262.
Hanhijärvi, S. (2011). Multiple hypothesis testing in pattern discovery. In *Proceedings of the 14th international conference on discovery science (DS'11)* (pp. 122–134).
Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*. New York: Springer.
Hochberg, Y. (1988). A sharper Bonferroni procedure for multiple tests of significance. *Biometrika*, *75*, 800–803.
Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, *6*, 65–70.
Janssen, F., & Fürnkranz, J. (2010). On the quest for optimal rule learning heuristics. *Machine Learning*, *78*(3), 343–379.
Jensen, D. D., & Cohen, P. R. (2000). Multiple comparisons in induction algorithms. *Machine Learning*, *38*(3), 309–338.
Lavrač, N., Flach, P., & Zupan, B. (1999). Rule evaluation measures: A unifying view. In *Proceedings of the 9th international workshop on inductive logic programming (ILP'99)*, Bled, Slovenia (pp. 174–185).
Lichman, M. (2013). *UCI machine learning repository*. http://archive.ics.uci.edu/ml. Accessed 2 June 2016.
Lin, C. J., Weng, R. C., & Keerthi, S. S. (2008). Trust region Newton method for logistic regression. *Journal of Machine Learning Reasearch*, *9*, 627–650.
Lindgren, T. (2004). Methods for rule conflict resolution. In *Proceedings of the 15th European conference on machine learning (ECML'04)* (pp. 262–273), Pisa: Springer.
Možina, M., Demšar, J., Žabkar, J., & Bratko, I. (2006). Why is rule learning optimistic and how to correct it. In *Proceedings of 17th European conference on machine learning (ECML'06)* (pp. 330–340), Berlin: Springer.
Quinlan, J. R., & Cameron-Jones, R. M. (1995). Oversearching and layered search in empirical learning. In *Proceedings of the 14th international joint conference on artificial intelligence (IJCAI'95)*, Montreal, Canada (pp. 1019–1024).

Scheffer, T. (2005). Finding association rules that trade support optimally against confidence. *Intelligent Data Analysis*, *9*(4), 381–395.

Todorovski, L., Flach, P., & Lavrač, N. (2000). Predictive performance of weighted relative accuracy. In *Proceedings of the 4th European Conference of Principles of Data Mining and Knowledge Discovery (PKDD'00)*, Lyon, France (pp. 255–264).

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer.