Why Is Rule Learning Optimistic and How to Correct It

Martin Možina, Janez Demšar, Jure Žabkar, and Ivan Bratko

Faculty of Computer and Information Science, University of Ljubljana, Tržaška cesta 25, SI-1001 Ljubljana

Abstract. In their search through a huge space of possible hypotheses, rule induction algorithms compare estimations of qualities of a large number of rules to find the one that appears to be best. This mechanism can easily find random patterns in the data which will – even though the estimating method itself may be unbiased (such as relative frequency) – have optimistically high quality estimates. It is generally believed that the problem, which eventually leads to overfitting, can be alleviated by using m-estimate of probability. We show that this can only partially mend the problem, and propose a novel solution to making the common rule evaluation functions account for multiple comparisons in the search. Experiments on artificial data sets and data sets from the UCI repository show a large improvement in accuracy of probability predictions and also a decent gain in AUC of the constructed models.

1 Introduction

Most rule learning algorithms [8] induce models by iteratively searching for the best rule and removing the examples covered by it. Rules are usually sought by a beam search, which gradually adds conditions to the rule with aim to decrease the number of covered (so-called) negative examples, while at the same time losing as few positive examples as possible. The search is guided by two measures, one which evaluates the partial rules and the other which selects between the final rule candidates; most often, as in the case of this paper, the same measure is used for both purposes.

A good rule should give accurate class predictions, or, in other words, have a high probability of the positive class among all examples (not only learning examples) covered by rule. Hence relative frequency, an unbiased estimator of probability, seems to be a reasonable choice for the measure of quality of rule:

$$Q(r) = \frac{s}{n} \tag{1}$$

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

However, the assumption that the relative frequency indeed estimates the probability of positive class is wrong. Fig. 1(a) shows how searching through a large space of rules, which tries to maximize relative frequencies, can always find rules with 100% positive subsets, though these are usually purely random patterns in the data and their true positive class probabilities are much lower.[1] Class proportions for the rules found by the search process are thus completely uncorrelated with the true class probabilities.

[1] Experimental details are provided in the next section.

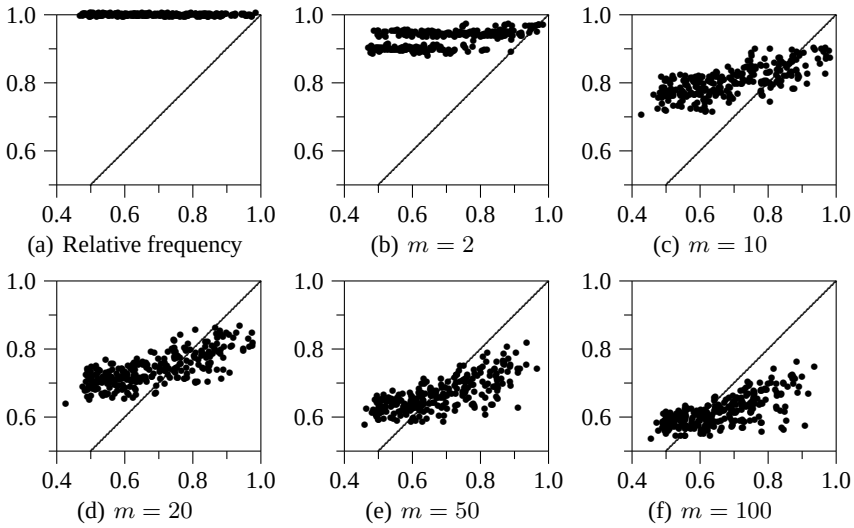


Fig. 1. Relation between the estimated (y-axis) and true (x-axis) class probabilities for rules from artificial data sets.

A more general version of this problem has been extensively explored by Jensen and Cohen [12] who blame multiple comparisons during the search to be responsible for plethora of pathologies in induction algorithms. Our paper proposes a method which can fix the relative frequency estimate and other rule evaluation measures by taking multiple comparisons into account through the use of extreme value distributions [7].

Since a review of all proposed improvements of evaluation measures would take the entire paper, the next section only studies the effect of the m-estimate of probability [2] as a good representative of such techniques. We then present our algorithm and, in the following section, validate it on several artificial and UCI data sets. The conclusion gives a list of several open questions and limitations of the methods.

2 Experimental Study of Rule Estimators

The m-estimate [2] computes the class probability (or, in our case, the rule quality) as

$$Q_m(r) = \frac{s + m \times p_a}{n + m} \tag{2}$$

where p_a is the prior probability and m is a parameter of the method. Fuernkranz and Flach [8] showed that the m-estimate presents a trade off between precision (relative frequency) and linear cost metrics (for instance, weighted relative accuracy [13, 16]). Different values of parameter m can be used to approximate many currently used evaluation functions. For instance, when $m = 0$, m-estimate equals the relative frequency. Instead of citing various proposals from the extensive related work, we shall thus concentrate on the more general m-estimate.

Table 1. 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 mean square error of the estimate.

m	avg. accuracy	Spearman	mean error
0	0.68	0.00	0.119
2	0.68	0.54	0.074
10	0.68	0.68	0.027
20	0.68	0.72	0.015
50	0.67	0.70	0.009
100	0.66	0.65	0.010

To observe the correlation between the true and the estimated class probabilities, we constructed a set of artificial data sets with controlled class probabilities for each possible rule. We have prepared 300 data sets with ten binary attributes. Five attributes in each data set were unrelated with the class. For the other five, we prescribed a (random) class 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 prescribed probabilities for the combination of informative attributes. Note that the actual class proportions in the data set do not necessarily match the defined probabilities for a particular combination of attribute values.[2]

For each of 300 data sets we learned a single rule using different values for m (0, 2, 10, 20, 50, 100). Fig. 1 shows the relation between the rule's estimated class probability $Q_m(r)$ and the known true probability, which we shall denote by $\widetilde{Q}(r)$. As we already mentioned in the introduction, at $m = 0$ (relative frequency), the method is extremely optimistic. With increasing values of m, the method is still optimistic for rules with lower true probability, but pessimistic for rules with higher true probability. It seems that m-estimate lowers the estimated quality by the same amount for all rules, which can not adjust the estimates to lie on (or at least close to) the ideal diagonal line representing the perfect correlation.

Table 1 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 $Q_m(r)$ and $\widetilde{Q}(r)$, 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;
- the mean square difference between $Q_m(r)$ and $\widetilde{Q}(r)$ which indicates the rule's accuracy when used as probabilistic predictor.

The first column of Table 1 suggests that lower values of m give (marginally) better rules than higher values. However, higher m's score better in terms of the Spearman correlation and give better probability estimates.

In conclusion, using the m-estimate with a suitably tuned m can considerably decrease the error of the estimated probabilities, yet, as seen from the graphs, the major

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

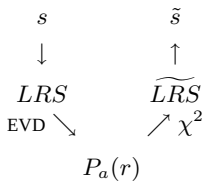


Fig. 2. An outline of the proposed procedure

effect comes from reducing the optimism, while the correlation between the true and the estimated probability remains rather poor. m-estimate and the many other similar techniques are thus not a satisfactory solution to the problem of overfitting, wrong rule quality estimates and optimistic probability predictions.

3 Algorithm for Improved Probability Estimate

Relative frequencies, m-estimates and other potential measures of rule quality are computed from the number of examples covered by the rule (n) and the number positive examples among them (s). We have seen that relative frequencies overestimate the true probability because the algorithm searches for the rule with the highest $s : n$ ratio. Since the training data presents only a limited sample from the population, the observed ratio for each rule is subject to random distribution, so the found rule is therefore not necessarily the optimal one, and it almost certainly has an optimistic $s : n$ ratio. One way of preventing these unwanted effects and improving the probability estimate s/n is to try to find the expected value of s, which we shall denote by $\tilde{s}$.

The outline of the proposed procedure is illustrated in Fig. 2. For reasons that will become clear later, we start by computing the log-likelihood ratio statistics (LRS) for 2×2 tables derived by Dunning [6]. It is usually assumed that LRS is distributed according to $\chi^2(1)$. This is, however, true only for randomly chosen rules, or, in our case, for LRS computed from the expected value of s, $\tilde{s}$.

The highest observed LRS (computed from s, not $\tilde{s}$) is distributed according to the Fisher-Tippet extreme value distribution (EVD) [7].[3] Since EVD depends only the number of rules considered in the search (it is more likely to get higher LRS if number of rules considered is higher), which is determined by the rule length, the chosen search algorithm and the properties of the data set (number of examples, number and type of attributes), we will be able to compute corresponding EVDs – for rules of different lengths for the selected algorithm and a particular data set – in advance.

Now consider a specific rule. From n and the observed s, we can compute the LRS and then, knowing its distribution (EVD), find the probability that a rule this good (or better) is found under the null-hypothesis of no relation between the attribute and class values. We shall denote this probability, the "significance of the rule", by $P_a(r)$. Note that $P_a(r)$ takes the multiple comparisons into account, so this estimate is unbiased.

[3] For illustration, although the daily levels of a river are usually distributed normally, the distribution of maximal annual levels is not normal but Fisher-Tippet's.

On the other hand, imagine that we knew the expected $\tilde{s}$ of that rule. We could compute its true log-likelihood ratio $\widetilde{\text{LRS}}$ and, through $\chi^2(1)$ distribution, arrive at the same significance $P_a(r)$ as above.

The trick that we use in this paper is to reverse the second path. So, from the unbiased $P_a(r)$ which we get from the known (but optimistic) s through LRS and EVD, we shall compute the unbiased $\widetilde{\text{LRS}}$ and the corresponding $\tilde{s}$.

The reason for which we need to compute LRS instead of computing the extreme value distribution for $Q(r) = s/n$ directly, and estimate the unbiased $\widetilde{\text{Q}}$ through $P_a(r)$, is that the extreme value of a sampled random variable (such as s, $Q(r)$ or LRS) is distributed by the Fisher-Tippet (or some other) extreme value distribution only if the variable's values are taken from a fixed distribution (independent from s and n). LRS, as we just noted, fulfils this criterion, while s/n is distributed according to $\beta(s, n-s)$ and is thus not the same for all rules.

As a side note, our approach to correcting quality estimators can be generalized to other criteria beside $Q(r) = s/n$. If the density distribution of a criterion depends upon the rule (like is the case with s/n), we need to find a measure which is well-correlated with the criterion (additional explanation is given later), yet drawn from a fixed distribution (like LRS, which is drawn from $\chi^2(1)$ and still reasonably correlated with s/n).[4] If the observed criterion already comes from a fixed distribution (if, for example, LRS would be used as the main evaluation function), finding a correlated measure is not needed and we can immediately proceed to the computation of EVD.

This section will present the details of the algorithm, along with a running example for illustration.

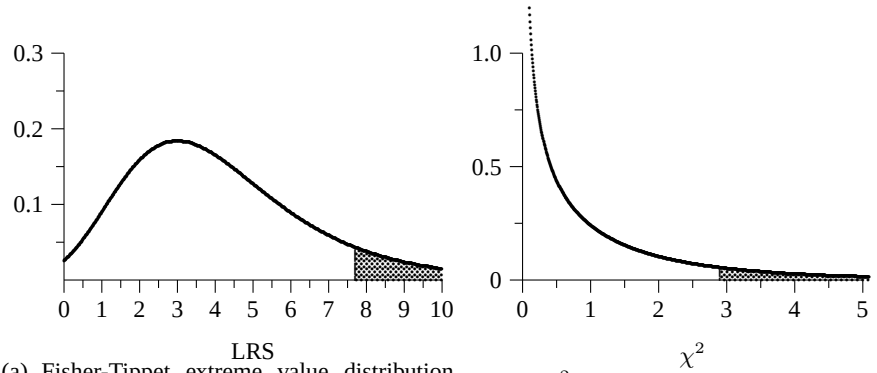
Step 1: From s to LRS. Let s again be the number of positive examples covered by rule and let s^c be the number of positive examples not covered by the rule. Similarly let n be the number of covered examples by the rule and n^c be the number of examples not covered by the rule. LRS is then defined as:

$$\text{LRS} = 2\left[s\log\frac{s}{e_s} + (n-s)\log\frac{n-s}{e_{n-s}} + s^c\log\frac{s^c}{e_{s^c}} + (n^c-s^c)\log\frac{n^c-s^c}{e_{n^c-s^c}}\right] \quad (3)$$

where e_x is the expected value of x. For instance, e_s is computed as $n\frac{s+s^c}{n+n^c}$. When computed on a randomly chosen rule, LRS is distributed according to $\chi^2(1)$ distribution, disregarding properties of the rule (length, s, $n\ldots$) and the data. Note that a similar formula for LRS, without the last two terms, was used in [4, 3] for computing significance of rules. However, as that formula is approximately correct only if n is small enough when compared to n^c, we prefer to use the formula 3 derived by Dunning [6].

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 with two conditions which covers 10 examples with 8 of them belonging to the positive class. Its LRS is, according to (3), 7.7.

[4] The correlation would be perfect if every rule covered the same number of examples.



(a) Fisher-Tippet extreme value distribution ($\mu = 3$, $\beta = 2$)

(b) χ^2 with 1 degree of freedom.

Fig. 3. Probability density functions

Step 2: From LRS to $P_a(r)$. $P_a(r)$ measures the probability that, given a random data with no relation between the attribute and class values, the rule found by the chosen search procedure will have the quality of at least LRS(r) (or another suitable measure of quality). This definition suggests a way of computing $P_a(r)$: like Jensen and Cohen [12], we permute the class values in the data set so that all rules are purely random and their true probability for positive class equals the prior probability. We then induce a rule on the randomized data set and compute its LRS. Repeating it for many times we get a distribution for LRSs.

Gumbel and Lieblein [9, 10] (cited in [14]) have shown that the limiting distribution of all χ^2 distributions is the Fisher-Tippet distribution (Fig. 3(a)). Fisher-Tippet distribution is characterized by two parameters, location (μ) and scale (β). For LRS, it can be shown that β always equals 2, and μ equals the median of the above sample to which we add $2 \ln \ln 2$ (see Appendix A for a proof). In general, values of μ and β depend upon the number of rules covered by the search (which does not necessarily equal the number of *explicitly* evaluated rules), 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. 4. The algorithm runs until $\mu(L)$ is smaller than $\mu(L-1)$, which signifies that rules of length $L-1$ can not be improved because they are perfect or they do not cover enough examples.

During learning we use the cumulative Fisher-Tippet distribution function (see the formula in Appendix A) with the pre-computed parameters to estimate $P_a(r)$.

Example (continued). Say that algorithm from Fig. 4 found $\mu(2) = 3$ and $\beta(2) = 2$ (remember that rule r has two conditions). The curve with such parameters is depicted

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.[5]
4. Record the LRS of the rule learned.
5. Repeat steps 2-4 to collect a large enough (say 100) sample of LRSs
6. Estimate parameters $\mu(L)$ and $\beta(L)$ of the Fisher-Tippet distribution (see Appendix A).
7. If $\mu(L) > \mu(L-1)$, then $L = L + 1$ and return to step 2.

Fig. 4. The algorithm for computing parameters of the Fisher-Tippet distributions

in Fig. 3(a), so the probability $P_a(r)$ 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 $P_a(r)$ to $\widetilde{\text{LRS}}(r)$. To compute $\widetilde{\text{LRS}}(r)$ we need to do the opposite from the last step. Looking at the $\chi^2(1)$ distribution (Fig. 3(b)), we need to find such a value $\widetilde{\text{LRS}}(r)$ that the area under the curve to the right of it will equal $P_a(r)$. In other words, the shaded areas under the curves in Fig. 3 should be the same.

Example (continued). The corresponding $\widetilde{\text{LRS}}$ for our examples as read from Fig. 3(b) is 2.9. 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.

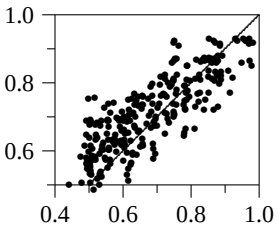
Step 4: From $\widetilde{\text{LRS}}$ to $\tilde{s}$. The remaining task is trivial: compute $\tilde{s}$ from the formula for $\widetilde{\text{LRS}}$ using an arbitrary root finding algorithm. Similar would be done for statistics other than $\widetilde{\text{LRS}}$ and $\tilde{s}$. In our task we are correcting probability estimates based on relative frequencies, so we shall compute them by dividing the corrected $\tilde{s}$ by n.

Example (conclusion). We used Brent's method [1] to find that $\widetilde{\text{LRS}} = 2.9$ corresponds to $\tilde{s} = 6.95$. The rule covers ten examples, so the corresponding class probability is $6.95/10 = 0.695$. Note that this estimate is quite smaller than the uncorrected 0.8.

4 Experiments

We have tested the algorithm on artificial data described in Section 2 and on a selection of data sets from the UCI repository [15]. In all experiments we used beam search [3, 4] with a beam width set to 5. The algorithm was implemented as a component for the rule based learner in machine learning system Orange [5].

[5] Note that using LRS at a given rule length will always order rules the same as would $\widetilde{\text{LRS}}$. However, as we will be using $\tilde{(s)}/n$ in the actual learning phase, in order to correctly estimate parameters of Fisher-Tippet distribution, measures $\tilde{s}/n$ and $\widetilde{\text{LRS}}$ should be well correlated.



Average quality: 0.68
Spearman correlation: 0.83
Mean squared error: 0.007

Fig. 5. Relation between the corrected (y-axis) and the true (x-axis) class probability.

The results of using the corrected measure on the artificial data are shown in Fig. 5. The estimated class probabilities are nicely strewn close to the diagonal axis, which is a clear improvement in comparison with the results from Fig. 1. This is also confirmed by the quantitative measure of fit: the average true probability is the same as the highest values in Table 1, the mean quadratic error is a little better than that of m-estimates, while the Spearman coefficient is clearly superior.

We mentioned that LRS is perfectly correlated with class probabilities only if every rule covers the same number of examples. Our data is constructed in that way, while real data sets certainly do not possess that property. To test the practical impact of our correction, we observed its behaviour on a set of UCI data sets. Each data set was split evenly onto learn and test sets. For learning we then generated ten bootstrap samples from the learn set.

We ran the algorithm on the bootstrap samples and then used the examples from the test set to count the number of positive and the number of all examples covered by each induced rule. We took this ratio to be the true positive class probability for the rule (although it is, as a matter of fact, still only an estimate, it is at least an unbiased one, since it is computed from the test data). Results in Table 1(a) show that we succeeded in improving the probability estimates: the probability estimates by our method are far more accurate than those by any m in m-estimate measure.

This would, however, be easily achieved and surpassed by a method returning a single rule covering all examples and which would estimate the probability with the prior class probability. To test that our gains are not due to oversimplification we also computed the average AUC over the ten bootstrap samples. To make predictions from lists of rules, we used a simple classifier that takes the first rule that triggers for each class (we get one rule for each class), and normalize the class probabilities of these rules to sum up to 1. Although there exist better classifiers from a set of rules, we believe that using them would not considerably change the ranking of examples and the related AUC. Table 1(b) shows that the performance of our method in terms of AUC is comparable to that of the other methods.

5 Conclusion

We have described a correction for removing the optimism in rule evaluation measures which arises since the rule was selected among many other rules considered during

(a) Mean squared error over all rules

Data set	rel	m=2	m=10	m=20	m=50	m=100	EVD
adult	0.43	0.13	0.08	0.06	0.06	0.06	**0.04**
australian	0.41	0.12	**0.05**	**0.05**	**0.05**	0.08	**0.05**
balance	0.25	0.12	0.11	0.11	0.08	0.07	**0.05**
breast (lju)	0.39	0.14	0.09	0.09	0.08	**0.06**	**0.06**
breast (wsc)	0.15	0.08	0.13	0.18	0.26	0.30	**0.05**
car	0.07	0.06	0.05	0.04	0.03	**0.02**	0.04
credit	0.41	0.11	0.07	0.07	**0.06**	0.07	**0.06**
german	0.42	0.13	0.06	0.06	0.05	0.05	**0.04**
hayes-roth	0.26	0.10	0.16	0.21	0.26	0.29	**0.08**
hepatitis	0.35	0.12	**0.05**	0.06	0.09	0.09	0.07
ionosphere	0.27	0.05	0.06	0.09	0.13	0.13	**0.03**
iris	0.20	0.07	0.09	0.12	0.17	0.22	**0.04**
lymphography	0.28	0.10	0.17	0.21	0.22	0.24	**0.05**
monks-1	0.07	0.07	0.16	0.13	0.15	0.20	**0.06**
monks-2	0.40	0.13	0.10	0.11	0.07	0.08	**0.05**
monks-3	0.32	0.09	0.08	0.11	0.14	0.13	**0.03**
mushroom	**0.00**	0.01	0.08	0.13	0.18	0.25	0.01
pima	0.48	0.15	0.05	**0.04**	**0.04**	**0.04**	0.05
SAHeart	0.46	0.19	0.08	0.07	**0.05**	0.07	0.07
shuttle	0.26	0.13	0.18	0.14	0.17	0.19	**0.11**
tic-tac-toe	0.19	0.03	0.07	0.14	0.24	0.30	**0.01**
titanic	**0.01**	0.02	0.04	0.04	0.04	0.03	0.02
voting	0.28	0.08	0.10	0.12	0.11	0.10	**0.04**
wine	0.09	0.07	0.14	0.20	0.24	0.31	**0.05**
zoo	0.16	0.09	0.22	0.31	0.42	0.47	**0.04**

(b) AUC

Data set	rel	m=2	m=10	m=20	m=50	m=100	EVD
adult	0.74	0.76	0.76	0.77	0.77	0.78	**0.84**
australian	0.85	0.87	0.88	0.88	0.88	0.88	**0.91**
balance	**0.82**	0.81	0.81	**0.82**	**0.82**	0.81	**0.82**
breast (lju)	0.60	**0.62**	0.60	0.58	0.60	0.60	**0.62**
breast (wsc)	0.97	0.97	0.97	0.97	0.96	0.96	**0.98**
car	0.84	0.84	0.85	0.86	0.89	**0.90**	**0.90**
credit	0.82	0.88	0.88	0.88	0.87	0.88	**0.91**
german	0.69	0.68	0.69	0.68	0.69	0.69	**0.73**
hayes-roth	0.88	0.89	0.87	0.86	0.87	0.86	**0.90**
hepatitis	**0.77**	0.76	0.76	0.73	0.73	0.71	**0.77**
ionosphere	0.90	0.91	0.89	0.89	0.89	0.91	**0.92**
iris	**0.97**	0.95	0.95	0.95	0.95	0.95	0.95
lymphography	0.78	0.81	0.83	**0.85**	0.84	0.83	0.81
monks-1	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
monks-2	0.66	**0.67**	0.66	0.66	0.64	0.65	0.64
monks-3	0.97	0.98	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
mushroom	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
pima	0.68	0.72	0.72	0.72	0.72	0.73	**0.76**
SAHeart	0.59	0.62	0.63	0.63	**0.65**	**0.65**	**0.65**
shuttle	**0.99**	0.98	0.98	**0.99**	**0.99**	**0.99**	0.98
tic-tac-toe	0.96	**1.00**	**1.00**	**1.00**	0.99	0.94	**1.00**
titanic	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**
voting	0.95	0.95	0.96	0.96	**0.97**	**0.97**	0.96
wine	0.97	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	0.94
zoo	**1.00**	0.99	0.99	0.99	0.99	0.99	0.99

Table 2. Mean squared errors and AUCs for estimation by relative frequencies (rel.)m, m-estimate and our method (EVD)

the search based on this same measure. The correction is based on the idea that the optimistic statistics, which is distributed according to extreme-value distribution, and the sought for statistics, which is (in case of LRS, which we used) distributed by $\chi^2(1)$ should predict the same probability that the rule was found by chance.

Tests on artificial data sets show that the correction works well. Experiments on real-world UCI data sets also confirm the gain in terms of probability predictions without decreasing the accuracy of predictive models.

There remain several limitations and unsolved problems. First, extreme value distributions are computed in advance, using entire data set. Common rule learning algorithms use a separate-and-conquer approach in which the covered examples are removed at each step, therefore changing the statistical properties of the data set. As a most obvious consequence, removing examples reduces the effective search space, which makes our correction too strict. We should therefore recompute the parameters of EVD distributions after each step, which is not practically feasible. The alternative would be to develop a rule learning algorithm that does not remove learning examples.

Extreme value distributions, as computed in the paper, account for multiple comparisons between the rules of the same length, but not between the rules of different lengths. We have developed and tested a remedy for this, but we omitted it in the paper since the impact of this correction is minimal – the number of comparisons between rules with different length is usually small.

We believe that the proposed method has a lot of potential. Although we here applied it only for correcting the class probability estimates, the same trick could, in principle, be applied to correcting other measures of rule quality that are being optimized by the search process. It may be even adoptable to other machine learning methods that extensively search through the space of possible hypotheses, such as learning decision trees, and which could significantly benefit from such corrections.

A Appendix: Computing parameters of extreme-value distribution

Section 3 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 Fisher-Tippet [7, 9, 10]. The cumulative distribution function of this distribution is

$$P(x < x_0) = e^{-e^{\frac{\mu - x_0}{\beta}}} \tag{4}$$

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

$$\text{mean} = \mu + \beta\gamma, \quad \text{median} = \mu - \beta * \ln\ln 2, \quad \text{var} = \pi^2\beta^2/6 \tag{5}$$

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 [11] 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:

$$E(M) = 2\ln p - \ln\ln p - \ln\pi + 2\gamma \tag{6}$$

$$m(M) = 2\ln p - \ln\ln p - \ln\pi - 2\ln\ln 2 \tag{7}$$

$$\sigma(M) = \sqrt{2/3\pi^2} \tag{8}$$

Since $\sigma(M)$ is independent of the number of values (or the number of considered rules, in our case), combining 5 and 8 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 $2\ln\ln 2$.

Acknowledgements

This work was partly supported by the European Commission's Information Society Technologies (IST) programme, through Project ASPIC (IST-FP6-002307), and Slovene Agency for Research and Development (ARRS).

References

1. Kendall E. Atkinson. *An Introduction to Numerical Analysis*. John Wiley and Sons, New York, 1989.
2. B. Cestnik. Estimating probabilities: A crucial task in machine learning. In *Proceedings of the Ninth European Conference on Artificial Intelligence*, pages 147–149, 1990.
3. Peter Clark and Robin Boswell. Rule induction with CN2: Some recent improvements. In *Machine Learning - Proceeding of the Fifth Europen Conference (EWSL-91)*, pages 151–163, Berlin, 1991.
4. Peter Clark and Tim Niblett. The CN2 induction algorithm. *Machine Learning Journal*, 4(3):261–283, 1989.
5. J. Demšar and B. Zupan. Orange: From experimental machine learning to interactive data mining. White Paper `[http://www.ailab.si/orange]`, Faculty of Computer and Information Science, University of Ljubljana, 2004.
6. Ted E. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.
7. R.A. Fisher and L.H.C. Tippett. Limiting forms of the frequency distribution of the largest and smallest member of a sample. *Proc. Camb. Phil. Soc.*, 24:180–190, 1928.
8. Johannes Fuernkranz and Peter A. Flach. Roc 'n' rule learning – towards a better understanding of covering algorithms. *Machine Learning*, 58(1):39–77, January 2005.
9. Emil J. Gumbel. Statistical theory of extreme values and some practical applications. *National Bureau of Standards Applied Mathematics Series (US Government Printing Office)*, 33, 1954.
10. Emil J. Gumbel and J. Lieblein. Some applications of extreme-value models. *American Statistician*, 8(5):14–17, 1954.

11. Shanti S. Gupta. Order statistics from the gamma distribution. *Technometrics*, 2:243–262, 1960.
12. David D. Jensen and Paul R. Cohen. Multiple comparisons in induction algorithms. *Machine Learning*, 38(3):309–338, March 2000.
13. Nada Lavrač, Peter Flach, and Blaž Zupan. Rule evaluation measures: A unifying view. In Saša Džeroski and Peter Flach, editors, *Proceedings of the 9th International Workshop on Inductive Logic Programming (ILP-99)*, pages 174–185, Bled, Slovenia, 1999.
14. Wentian Li, Fengzhu Sun, and Ivo Grosse. Extreme value distribution based gene selection criteria for discriminant microarray data analysis using logistic regression. *Journal of Computational Biology*, 11(2/3):215–226, 2004.
15. P. M. Murphy and D. W. Aha. UCI Repository of machine learning databases `[http://www.ics.uci.edu/~mlearn/mlrepository.html]`. Irvine, CA: University of California, Department of Information and Computer Science, 1994.
16. Ljupčo Todorovski, Peter Flach, and Nada Lavrač. Predictive performance of weighted relative accuracy. In D. Zighed, J. Komorowski, and J. Zytkow, editors, *Proceedings of the 4th European Conference of Principles of Data Mining and Knowledge Discovery (PKDD-00)*, pages 255–264, Lyon, France, 2000.