Bayesian Estimation of Rule Accuracy in UCS

James A. R. Marshall*
Dept. of Computer Science
University of Bristol
Bristol BS8 1UB, UK
marshall@cs.bris.ac.uk

Gavin Brown
School of Computer Science
University of Manchester
Manchester M13 9PL, UK
gbrown@cs.man.ac.uk

Tim Kovacs
Dept. of Computer Science
University of Bristol
Bristol BS8 1UB, UK
kovacs@cs.bris.ac.uk

ABSTRACT

Learning Classifier Systems differ from many other classification techniques, in that new rules are constantly discovered and evaluated. This feature of LCS gives rise to an important problem, how to deal with estimates of rule accuracy that are unreliable due to the small number of performance samples available. In this paper we highlight the importance of this problem for LCS, summarise previous heuristic approaches to the problem, and propose instead the use of principles from Bayesian estimation. In particular we argue that discounting estimates of accuracy based on inexperience must be recognised as a crucially important part of the specification of LCS, and must be well motivated. We present experimental results on using the Bayesian approach to discounting, consider how to estimate the parameters for it, and identify benefits of its use for other areas of LCS.

Categories and Subject Descriptors

G.3 [**Mathematics of Computing**]: Probability and Statistics

General Terms

Algorithms

Keywords

Learning Classifier Systems, Bayesian estimation, inexperience, discounting, UCS

1. INTRODUCTION

Learning Classifier Systems are online learning algorithms that continuously discover and evaluate new classification rules during their execution. Thus, LCS frequently must evaluate the accuracy of classification rules that have previously matched only a few instances. An elementary understanding of probability theory tells us that estimates of such 'inexperienced' rules' accuracy will be unreliable, due to the small number of samples available. This feature of LCS contrasts sharply with many other machine-learning approaches where accuracy is only evaluated on sufficiently large sample populations as to give a high confidence in the predicted accuracy.

The 'inexperience problem' has prompted various LCS researchers to implement an inexperience discounting scheme. This scheme has typically been poorly documented (e.g. [4] §2.3.4.4, [2]), yet its impact on performance can be highly significant (e.g. [2]). Additionally, the motivation for the discounting scheme is primarily heuristic in nature. In this paper we seek to make explicit to the LCS community the importance of dealing with inexperienced rules, so that discounting schemes such as those described above firmly become part of the LCS specification. We further seek an alternative means of performing inexperience-based discounting, based on sound probabilistic principles, rather than purely heuristic reasoning. Fortunately such an approach is readily available from Bayesian estimation theory, and is widely used in other areas of Machine Learning.

2. A BRIEF HISTORY OF INEXPERIENCE DISCOUNTING IN LCS

Inexperience discounting in Learning Classifier Systems has a long history, beginning at the latest with XCS. As described in [4] (§2.3.4.4), in XCS Wilson made use of the following discounting mechanism when evaluating the fitness of a rule i, $g(i)$, given some set of observations of all classifications $\mathcal{D}_i$ and correct classifications $\mathcal{C}_i \subseteq \mathcal{D}_i$ made by the rule:

$$g(i) = \begin{cases} \gamma f(i) & \text{if } |\mathcal{D}_i| < \theta_{exp} \\ f(i) & \text{otherwise} \end{cases} , \quad (1)$$

with $\gamma = \frac{1}{16}$ and $\theta_{exp} = 20$; thus Wilson proposes that a rule's raw fitness $f(i)$ be scaled by a factor $\frac{1}{16}$ if it has been evaluated on fewer than 20 instances, otherwise it is unmodified. As this aspect of XCS is not part of the generally accepted specification [3], from this point on in the paper we consider another LCS, namely UCS. The same discounting rule is used in UCS with $\gamma = \frac{1}{100}$ and $\theta_{exp} = 10$ [6], despite unfortunately being left out of the published description of the algorithm [2].

The above scheme introduces two additional parameters to the LCS specification, the discount factor γ to be used, and the inexperience threshold θ_{exp} below which it is to be applied. It is unclear how these may be set in anything but

*To whom correspondence should be addressed

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IWLCS '07 London
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$5.00.

a heuristic, or empirical, manner. Additionally, eq. 1 introduces a potentially large discontinuity into the calculation of rule fitness.

3. BAYESIAN ESTIMATION OF RULE ACCURACY

In equation 1, the fitness before scaling $f(i)$ is a function of the accuracy of rule i. Both before and after scaling, accuracy is defined based on some set of observations of the rule's classifications $\mathcal{D}_i$, of which $\mathcal{C}_i \subseteq \mathcal{D}_i$ are correct classifications. In UCS $f(i) = \frac{|\mathcal{C}_i|}{|\mathcal{D}_i|}$, i.e. the proportion of the total number of matches that were correct classifications. This corresponds to a maximum likelihood estimation of the rule's probability of success, and while it is reliable for large $|\mathcal{D}_i|$, for small $|\mathcal{D}_i|$ it may deviate substantially from the rule's true success probability; for small $|\mathcal{D}_i|$ a large proportion of successes is quite possible even if the rule's true success probability is low. Consider the example of trying to estimate the bias of a coin by tossing it repeatedly and observing the outcomes (this is of course what we are doing in trying to estimate rule accuracy); obtaining 8 heads out of 10 trials tells us much less about the coin's bias than obtaining 80 heads out of 100 trials. It is this realisation that prompted discounting schemes such as eq. 1. However, as discussed in the previous section, an undesirable consequence of this discounting scheme is the addition of two parameters to the LCS specification, without principled guidance on how to set their values. As also discussed, the scaling function is discontinuous.

A more principled approach suggests itself, in the form of Bayesian estimation theory. In estimating the probability of success of a rule from a number of observations on its performance, we are solving a well known statistical problem; estimating the Bernoulli parameter of a binomial distribution. The binomial distribution is the distribution over the number of successes from tossing a biased coin a certain number of times; the Bernoulli parameter is the bias of this coin. As discussed above, in our situation the rule itself corresponds to the biased coin just described, and its bias is its probability of correctly classifying an arbitrary instance it is presented with; it is this quantity that we wish to estimate.

Instead of the maximum likelihood approach, which estimates rule accuracy as $\frac{|\mathcal{C}|}{|\mathcal{D}|}$, we can take a Bayesian approach. The Bayesian approach assumes a prior distribution over the possible rule accuracies, and combines this with evidence on the rule's accuracy to calculate the posterior distribution of its accuracy. Thus for some sequence of classifications $\mathcal{D}$, the distribution over the rule's accuracy a is given in the general case by

$$p(a|\mathcal{D}) = \frac{p(\mathcal{D}|a)p(a)}{\int p(\mathcal{D}|a)p(a)da}, \tag{2}$$

where $p(\mathcal{D}|a)$ is the probability density function for the observed performance given the rule's true accuracy, and $p(a)$ is the prior distribution over all possible rule accuracies. Given no information about the rule syntax and the problem at hand we might reasonably assume a uniform distribution for $p(a)$.

If the prior distribution $p(a)$ is a Beta distribution with parameters α and β, i.e.

$$p(a) = \frac{a^{\alpha-1}(1-a)^{\beta-1}}{B(\alpha,\beta)}, \tag{3}$$

where $B(x,y)$ is the Beta function with parameters α and β (a ratio of Gamma functions in the general case, a ratio of factorials in the case of integer α and β). Then we have what is called a *conjugate prior*. Then the posterior distribution will also be a Beta distribution

$$p(a|\mathcal{D}) = \frac{a^{n+\alpha-1}(1-a)^{|\mathcal{D}|-n+\beta-1}}{B(n+\alpha, |\mathcal{D}|-n+\beta)}. \tag{4}$$

Conveniently a uniform prior distribution can be written as a Beta distribution with parameters $\alpha = \beta = 1$. Given that the mean of a Beta distibution with parameters α and β is

$$\frac{\alpha}{\alpha+\beta}, \tag{5}$$

we can calculate the mean of the posterior distribution $p(a|\mathcal{D})$ for uniform prior distribution $p(a)$ as

$$\frac{|\mathcal{C}|+1}{|\mathcal{D}|+2}. \tag{6}$$

This is also known as the Laplace Correction.

This technique is in widespread use in other areas of Machine Learning, yet is of particular value in LCS. Its main benefit is that, while for large numbers of observations it asymptotically approaches the maximum likelihood estimate of the rule's accuracy, for small numbers of observations (which an LCS is frequently dealing with) a much more conservative estimate of rule accuracy is made. Unlike the discontinuous step-function of eq. 1, eq. 6 is continuous, and hence is referred to as a probability-smoothing technique. Assuming success in every observed trial, the Bayes estimate of rule accuracy is plotted in figure 1 for varying numbers of trials.

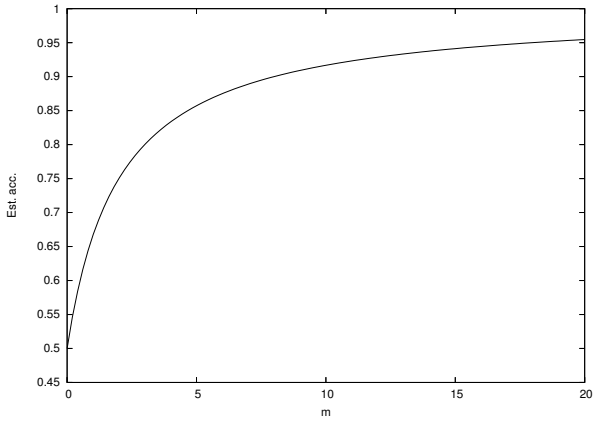

Figure 1: Bayesian estimate of rule accuracy for 100% successful rule given varying number of trials $\mathcal{D}$ and uniform prior

4. EXPERIMENTAL EVALUATION

We now compare the performance of our Bayesian estimation approach to the naïve undiscounted LCS approach,

and to the inexperience discounting scheme of eq. 1. The testbed for our evaluation is our own implementation[1] of UCS [1], and we evaluate its performance on two classic LCS testbeds, the 11-bit multiplexer (11-MUX), and the Monks UCI dataset with 5% noise (monks-3)[2]. UCS parameterisation was adapted from published parameters[3]. Tables 1 and 2 present results as Area Under Curve of classification accuracy after 5,000 learning iterations, with accuracy evaluated after every 50 iterations. Means and standard deviations are based on 20 replicates.

The results for the Bayesian approach we propose are promising. Intriguingly, for the multiplexer the heuristic discounting scheme of eq. 1, parameterised with $\gamma = \frac{1}{100}$ and $\theta_{exp} = 10$, is significantly outperformed by the naïve approach of no discounting ($t = 6.13$, $P < 0.0001$, $N = 20$; 2-sample t-test). Bayesian discounting is in turn significantly better than no discounting ($t = -3.31$, $P = 0.002$, $N = 20$; 2-sample t-test). For the monks-3 dataset, there is no significant difference between the performance of the traditional and Bayesian discounting schemes ($t = -0.39$, $P = 0.7$, $N = 20$; 2-sample t-test), although no discounting is significantly outperformed by both traditional discounting ($t = -15.95$, $P < 0.001$, $N = 20$; 2-sample t-test) and Bayesian discounting ($t = -15.94$, $P < 0.001$, $N = 20$; 2-sample t-test).

11-MUX	Naïve	Traditional	Bayes
Mean	98.54	97.81	**98.83**
Std. Dev.	0.25	0.47	**0.29**

Table 1: Mean and standard deviation for area under curve of accuracy on 11-multiplexer. Bold columns are significantly highest results

monks-3	Naïve	Traditional	Bayes
Mean	93.07	**96.79**	**96.85**
Std. Dev.	0.93	**0.48**	**0.52**

Table 2: Mean and standard deviation for area under curve of accuracy on monks-3. Bold columns are significantly highest results

5. REALISTIC PRIORS FOR RULE ACCURACY

The experimental results presented above make use of an estimation correction based on assuming a uniform prior distribution over rule accuracy. We might argue that in the absence of any information on the rules used and their fit to the problem at hand, a uniform distribution is appropriate. However such a distribution is actually highly unlikely, even for the simplest and most regular of problems. To illustrate this, in figure 2 we enumerate the accuracy of all possible ternary rules on all exemplars of 11-MUX. As can be clearly seen, the distribution is non-uniform, although it is still symmetric around mean 0.5. The experimental results of table 1 show that the uniform prior works well in this case, however.

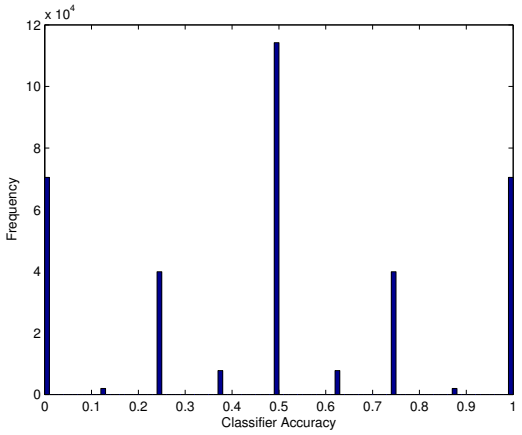

Figure 2: Distribution of rule accuracy for 11-MUX

Figure 3 shows the rule accuracies of all possible ternary rules that match at least one instance, each evaluated over the entire monks-3 training set. Here there is a clear majority of rules having accuracy 0 or accuracy 1. We can approximate this prior distribution of rule accuracies with a Beta distribution with parameters $\alpha = \beta = \frac{1}{2}$ (figure 4). Using this prior and applying eqs. 4 and 5 gives us a Bayesian estimate of rule accuracy based on $|\mathcal{C}|$ correct classifications from $|\mathcal{D}|$ trials as

$$\frac{|\mathcal{C}| + \frac{1}{2}}{|\mathcal{D}| + 1}. \tag{7}$$

This can be recognised as the m-estimate ([5], pp.179-180) of accuracy with $m = 1$. Applying this new correction makes no significant difference to UCS' performance on the monks-3 dataset however. Quite possibly the performance exhibited by UCS with the traditional and Laplace corrections is near the practical maximum that UCS can obtain on this problem.

Some problems and datasets such as 11-MUX and monks-3 are sufficiently small that the distribution of rule accuracies can be fully characterised. Of course such problems are toy problems and not of practical interest for 'real-world' applications of Machine Learning. However, we may estimate the distribution of rule accuracies for some larger problem by sampling uniformly from the spaces of all possible rules and instances, and parameterise our prior accordingly.

6. RULE ACCURACY AS A BERNOULLI PARAMETER IN LCS

It is interesting to consider under what circumstances we can treat rule accuracy as a Bernoulli parameter. In general we can write the expected accuracy of a rule on a binary classification problem such as those considered above as

$$E(a) = a_1 d_1 + a_2(1 - d_1), \tag{8}$$

where a_j is the rule's accuracy at classifying instances of class j, and d_1 is the proportion of all instances that belong

[1] `http://www.cs.man.ac.uk/~gbrown/ucs/`

[2] `http://www.ics.uci.edu/~mlearn/MLRepository.html`

[3] $N = 400$, $v = 10$, $P_\# = 0.33333333$, $acc_0 = 0.99$, $\chi = 0.8$, $\mu = 0.05$, $\delta = 0.1$, $\theta_{del} = 0.1$, $\theta_{GA} = 25$, $\theta_{sub} = 20$, $\theta_{exp} = 10$, $\gamma = 0.01$

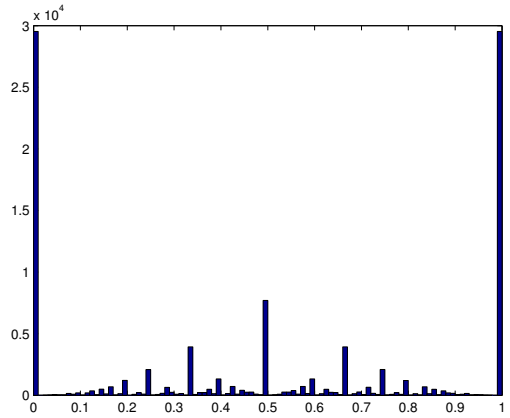

Figure 3: Distribution of rule accuracy for monks-3 training set

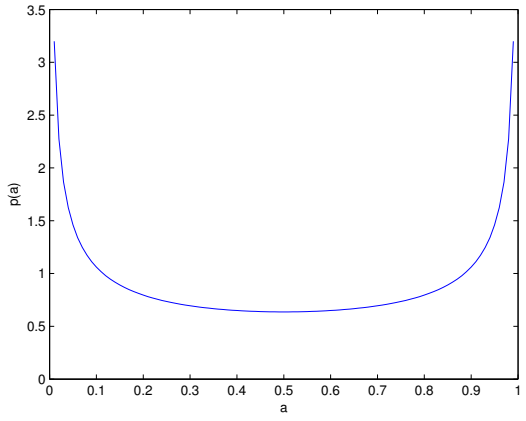

Figure 4: Beta distribution with parameters $\alpha = \beta = 0.5$

to class 1. In other words the rule's expected accuracy is a combination of its true positive and true negative rates, and as such is a function of both the class distribution and the rule's accuracy at classifying instances of each class. If $a_1 = a_2$ then then eq. 8 simplifies nicely and the class distribution becomes irrelevant. This seems like a reasonable simplifying assumption. For most LCS, however, the situation is even simpler; under most LCS representations a rule predicts a single class for the instances it matches, hence there is only one a_j.

So far we have been considering binary classification problems. However multi-class problems are common. Yet in this case we may still consider the rule's accuracy as a Bernoulli parameter by generalising eq. 8 to be

$$E(a) = \sum_{j=1}^{c} a_j d_j, \tag{9}$$

where c is the number of classes in the problem and $\sum_{i=1}^{c} d_i = 1$, and then assuming $a_1 = \ldots = a_c$. Again, as before, for most LCS a single class is predicted by a rule, hence there is only one a_j to consider per rule.

We conclude, based on the above, that for traditional LCS where each rule predicts a single class, unlike other pattern-matching techniques such as Artificial Neural Networks, it is valid to treat rule performance as a Bernoulli parameter and estimate rule accuracy accordingly.

7. DISCUSSION

In this paper we have highlighted the importance for LCS of discounting estimates of rule accuracy based on the number of samples available. In LCS, unlike many other Machine Learning algorithms, this problem is acute because of the continuous generation and evaluation of new rules during learning. Hence in LCS, rules must be compared against each other using estimates of accuracy that are based on variable, and often small, numbers of samples. The previous approach to such 'inexperience' discounting described in this paper is discontinous, heuristically motivated, and introduces two additional parameters to the LCS specification without principled guidelines for how to set them. We propose the use of a Bayesian estimation approach, commonly used in other areas of Machine Learning. This approach is theoretically well founded, and introduces no additional parameters to the LCS specification in the reasonable case of assuming all rule accuracies are equally likely. Alternatively, the approach turns parameterisation of the discounting scheme into a problem of fitting a known type of distribution to a population of samples of rule accuracy. Such distribution fitting can be tackled in a principled way with existing techniques.

One further benefit of the discounting scheme presented here is that estimated rule accuracy for completely accurate rules converges asymptotically to 1 with number of samples; in other words accuracies for such rules are never estimated to be 1. This can be of great benefit to weighting schemes for rule combination that break down with rules of estimated accuracy 1 (e.g. [2]).

We hope this paper will prove of value to LCS practitioners, by incorporating a formally justified inexperience discounting scheme into the specification of any LCS applied to supervised learning.

8. ACKNOWLEDGMENTS

We thank P. Flach and L. Bull for helpful comments.

9. REFERENCES

[1] E. Bernadó-Mansilla and J. M. Garrell-Guiu. Accuracy-Based Learning Classifier Systems: Models, Analysis and Applications to Classification Tasks. *Evolutionary Computation*, 11(3):209–238, 2003.

[2] G. Brown, T. Kovacs, and J. A. R. Marshall. UCSpv: Principled Voting in UCS Rule Populations. In *Proceedings of the Genetic and Evolutionary Computing Conference (GECCO)*, 2007.

[3] M. V. Butz and S. W. Wilson. An algorithmic description of xcs. *Soft Computing*, 6:144–153, 2001.

[4] T. Kovacs. *Strength or Accuracy: Credit Assignment in Learning Classifier Systems*. Springer, 2004.

[5] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[6] A. Orriols. Personal comunication, January 2007.