ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5. Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

Non-linearly increasing resampling in racing algorithms

Verena Heidrich-Meisner[1] and Christian Igel[2]*

1- Ruhr-Universität Bochum, Institut für Neuroinformatik
Bochum, Germany

2- University of Copenhagen, Department of Computer Science
Copenhagen, Denmark

Abstract. Racing algorithms are iterative methods for identifying the best among several options with high probability. The quality of each option is a random variable. It is estimated by its empirical mean and concentration bounds obtained from repeated sampling. In each iteration of a standard racing algorithm each promising option is reevaluated once before being statistically compared with its competitors. We argue that Hoeffding and empirical Bernstein races benefit from generalizing the functional dependence of the racing iteration and the number of samples per option and illustrate this on an artificial benchmark problem.

1 Introduction

Racing algorithms try to find the best among several options with high probability. The quality of an option is a random quantity that can be observed by sampling and the sample size should be kept as small as possible (i.e., they tackle mutli-armed bandit problems) [1, 2]. For example, racing algorithms are applied to determine the best hyperparameters for supervised machine learning algorithms or for selection in evolutionary computing, [3, 4, 5, 6, 7]. They are iterative methods. In each iteration, options are reevaluated and then selected or discarded based on confidence intervals. When computing the confidences the algorithms have to accommodate this multiple testing. Based on ideas in [6, 7, 3, 8], we suggest to increase the number of new samples per option super-linearly to get tighter bounds and thus more efficient races.

2 Racing algorithms

Given a set Ω of λ options, the goal of racing algorithms is to determine the best option with a low error probability of δ. Instead of resampling all options equally often, they try not to waste samples on options that are not likely to be the best one. In each *race*, all options are resampled iteratively. If the performances estimates of two options have non-overlapping confidence intervals, these options are regarded as distinguishable. The race finishes as soon as the best option can

*VHM and CI acknowledge support from the German Federal Ministry of Education and Research within the National Network Computational Neuroscience, Bernstein Fokus: "Learning behavioral models: From human experiment to technical assistance", grant FKZ 01GQ0951.

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5. Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

be distinguished in this way. Options that can be identified as non-optimal in this fashion need not be resampled and drop out of the race.

In racing algorithms [1, 2, 4, 5], there is an upper bound on the number of racing steps. This ensures termination of the algorithm and allows to determine the confidence levels for the individual statistical tests. We will first review this type of racing algorithms before we focus on a variant that does not require such an upper bound.

2.1 Races with fixed maximum race length

In the following, we describe racing algorithms with an upper bound t_{limit} on the number of racing steps. In the standard scheme the number τ of racing steps is always the same as the number t of evaluations (samples) per option. Therefore, we follow the usual notation and use t for both quantities in this section. The quality of an option $o \in \Omega$ is a real-valued random variable Y_o. The tth sample of its quality (i.e., the tth time the option is evaluated to get a reliable estimate) is a realization of Y_o denoted by $y_{o,t}$, the confidence bounds are called $c_{o,t}$. The quality of an option is estimated by the empirical mean $\hat{y}_{o,t} = \frac{1}{t}\sum_{t'=1}^{t} y_{o,t'} \overset{\text{for } t>1}{=} \frac{1}{t}((t-1)\hat{y}_{o,t-1} + y_{o,t})$ of t realizations. We assume that the possible values of $y_{o,t'}$ are almost surely bounded with known bounds $y_{\min}$ and $y_{\max}$ (i.e., $\Pr(y_{o,t'} \in [y_{\min}, y_{\max}]) = 1$) and define the *range* $R = |y_{\min} - y_{\max}|$. No additional assumptions about the properties of the noise distribution are made.

We partition Ω into two sets. The set $\mathbb{D} \subset \Omega$ contains the options that have been *discarded* because they are not the best choice with a probability of at least $1-\delta$. All remaining options are in the set $\mathbb{U} \subseteq \Omega$ and have the label *undecided*. The number of options in $\mathbb{U}$ at iteration t is denoted by u_t. Initially, all options are labeled *undecided* and are evaluated once. In every following racing step, all options labeled *undecided* are reevaluated. The estimated mean performance and confidence interval are updated for each resampled option. If the lower bound of an option is better than the upper bounds of all other candidates, it is *selected* and the race terminates. If the upper bound of an option is worse than the lower bounds of at least one other candidate, it is discarded. The race stops if one option is selected or the number of racing iterations exceeds a predefined bound t_{limit}. We refer to this type of algorithm as *bounded racing*.

In this study, the confidence intervals are computed using either the Hoeffding or the empirical Bernstein bound [8]. If a confidence interval is recomputed, the highest lower and lowest upper bounds determined so far are stored. It holds (e.g., [5]): *Let the quality of each option be bounded almost surely between y_{min} and y_{max} and define $R = |y_{min} - y_{max}|$. Let t be the racing step, u_t be the number of options labeled as undecided, $n_{b,t} = \sum_{k=1}^{t-1} u_k + (t_{limit} - t + 1)u_t$ an upper bound for the number of tests in the race for a fixed maximum race length t_{limit}, $\delta_{n_{b,t}} = \frac{\delta}{n_{b,t}}$ the individual confidence levels, and $\hat{\sigma}_{o,t}$ the empirical variance of option o. If after t evaluations of option $o \in \Omega$ the confidence interval*

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5. Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

$[\hat{y}_{o,t} - c_{o,t}, \hat{y}_o + c_{o,t}]$ is computed with either

$$c_{o,t}^{Hoeffding} = R\sqrt{\frac{\log \frac{2}{\delta_{n_{b,t}}}}{2t}} \quad or$$

$$c_{o,t}^{Bernstein} = \hat{\sigma}_{o,t}\sqrt{2\frac{\log \frac{3}{\delta_{n_{b,t}}}}{t}} + 3R\frac{\log \frac{3}{\delta_{n_{b,t}}}}{t}$$

in the case of the empirical Bernstein bound, then with probability of at least $1 - \delta$ all confidence intervals computed during the complete race hold simultaneously. This implies that at step t we have $\max_{k=1,\dots t}(\hat{y}_{o,k} - c_{o,k}) \leq \mathbb{E}[Y_o] \leq \min_{k=1,\dots t}(\hat{y}_{o,k} + c_{o,k})$ with a probability of at least $1 - \delta$.

In bounded racing, races usually stop when an option is selected or a predefined maximum race length t_{limit} is reached.

2.2 Races without maximum race length

In the previous section a maximum race length t_{limit} has to be specified a priori to compute the confidence intervals. This is unsatisfactory and therefore we describe an alternative scheme referred to as *unbounded racing* that does not require an upper limit on the race length. This extension relies on an idea also applied in [6, 7, 3, 8]. Let us consider the nth application of the Hoeffding bound (for the empirical Bernstein bound similar results can be derived in the same way) in a race. We have

$$\Pr\left\{ |\hat{y}_{o,t} - \mathbb{E}[Y_o]| \leq R\sqrt{\frac{\log \frac{2}{\delta_n}}{2t}} \right\} \geq 1 - \delta_n \ , \tag{1}$$

where t is the number of all samples used for the considered random variable in the race so far and n is the number of tests spent on all options so far. In each iteration, all options in $\mathbb{U}$ have been evaluated equally often. In section 2, the number of samples t is the same as the number of racing steps and it is not necessary to distinguish between them. All bounds in a race should hold simultaneously with a probability of δ. To account for multiple testing, we consider the union bound and require $\delta \geq \sum_{n=1}^{n_{\max}} \delta_n$, where $n_{\max}$ is the maximum number of tests in the complete race. This can be achieved by choosing $\delta_n = \frac{\delta}{n_{\max}}$ for all $n \in \{1, \dots, n_{\max}\}$ which leads to the method presented in the previous section.[1]

Now, let $n_{\max} \to \infty$. Individual confidence levels δ_n strictly bounded from zero (e.g., a single constant confidence level) are no longer possible, but any proper convergent sequence can be chosen instead. As in [6, 7, 3, 8], we define

$$\delta_n = \frac{c\delta}{n^2} \ ,$$

[1] A conservative estimate for $n_{\max}$ is $\lambda \cdot t_{\text{limit}}$. We use the tighter dynamic estimate $n_{\text{b},t}$ suggested in [5]. Then the δ_n changes between racing steps, but is constant within one iteration and is still strictly lower bounded from zero.

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5. Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

where $c = 6/\pi^2$ is a normalization constant. Plugging this into the Hoeffding bound yields:

$$|\hat{y}_{o,t} - \mathbb{E}[Y_o]| \leq R\sqrt{\frac{\log \frac{2}{\delta_n}}{2t}} = R\sqrt{\frac{\log(\pi^2 n^2) - \log(3\delta)}{2t}} \tag{2}$$

This type of race will terminate successfully under the assumption that there is a single best option, because of the law of large numbers and because the upper/lower bounds decrease/increase with increasing t (general convergence can be ensured in different ways, the simplest stopping criterion is a maximum number of evaluations per option).

3 Decoupling racing step and samples per test

The algorithm outlined in the previous section has the desired advantage that it does not involve an explicit upper limit on the race length. However, the numerator explicitly depends on the number of tests n, which is strictly monotonically increasing in the race length. This can lead to loose confidence intervals and thus long races.

To control this effect, we "decouple" the number of racing steps and the number of evaluations per option still participating in the race. Let from now on the former be denoted by τ and the latter by θ. The idea is to let θ grow faster than τ and thereby faster than the number of tests. Until now, in every racing step every option was reevaluated only once and we had $t = \theta = \tau$. This is now generalized to allow the number of reevaluations to depend non-linearly on the current racing step. For example, in racing step $\tau \geq 1$ an option can be reevaluated $2^{\tau-1}$ times. We call $\theta(\tau)$ the evaluation function. Then the number of evaluations of an option in step τ is $\theta_{\exp}(\tau) = 2^\tau$. Substituting $\theta_{\exp}(\tau)$ into equation (1) yields

$$|\hat{y}_{o,\tau} - \mathbb{E}[Y_o]| \leq R\sqrt{\frac{\log(\pi^2 n^2) - \log(3\delta)}{2^{\tau+1}}}$$

and the bounds get tighter with increasing number of racing steps.

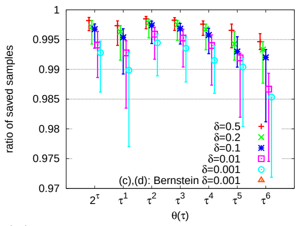
While $\theta_{\exp}(\tau)$ makes the bounds tighter on a logarithmic scale with every racing step, this exponential schedule can quickly demand too many samples per racing step. Thus, we also consider a polynomial schedule $\theta^p_{\text{poly}}(\tau) = \tau^p$. This is still sufficient to tighten the bounds, but allows for reasonable sampling rates. The Hoeffding bound then reads

$$|\hat{y}_{o,\tau} - \mathbb{E}[Y_o]| \leq R\sqrt{\frac{\log(\pi^2 n^2) - \log(3\delta)}{\tau^p}} \quad .$$

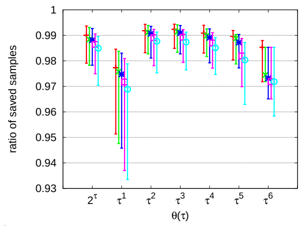
4 Experiments and results

Our experiments should serve as a proof of concept and aim at understanding the effect of different evaluation functions $\theta(\tau)$ on the performance of Hoeffding

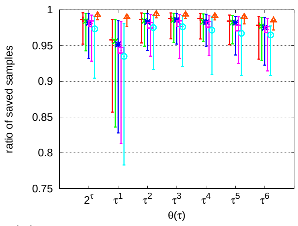
ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.



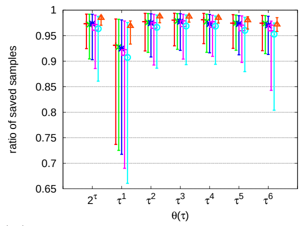
(a) Bounded Bernstein races.



(b) Unbounded Bernstein races stopped after 50000 evaluations per option.



(c) Bounded Hoeffing races.



(d) Unbounded Hoeffing races.

Fig. 1: Ratio of saved evaluations depending on the evaluation function $\theta(\tau)$ for bounded and unbounded Hoeffding and Bernstein races with confidence $\delta \in \{0.5, 0.2, 0.1, 0.01, 0.001\}$. The median of 100 trials is shown, the quartiles are given as error bars. In c) and d) the respective *worst* Bernstein race is included as reference.

and empirical Bernstein races. To this end, we tested both races on an artificial problem. Each option o in this task corresponds to a uniformly distributed random variable with interval $[a_o, b_o] \subset [0, 10]$, where the boundaries are sampled uniformly at random. The goal was to select the option promising the highest mean pay-off, that is, $o^* = \text{argmax}_{o \in \Omega}(a_o + b_o)$. One evaluation corresponded to drawing once from the associated interval. In each of the 100 trials ten new options were generated randomly. A fixed maximum number of evaluations per option $\theta_{\text{limit}} = 50000$ was available to Hoeffding and empirical Bernstein races. The ratio of samples saved compared to the complete budget $10\theta_{\text{limit}}$ served as performance measure. If the optimal option was not identified during a race the maximum number of evaluations was set to $10\theta_{\text{limit}}$. We tested both the bounded as well as the unbounded version of the algorithms. For a fair comparison, also the races not requiring an upper bound on the number of samples were stopped after consuming θ_{limit} evaluations for each still undecided option.

For all types of races, we compared $\theta^p_{\text{poly}}(\tau) = \tau^p, p = 1, \ldots, 6$, and $\theta_{\text{exp}}(\tau) = 2^\tau$. As can be seen in figures 1(a)-1(d) for all parameter settings Hoeffing and empirical Bernstein races were successful (i.e., they saved evaluations on average). Unsurprisingly, a low confidence $\delta = 0.5$ led to the shortest races. Empirical Bernstein races were on this task always more efficient than Hoeffding races, since sufficiently many samples were available for a good estimate of the empirical variance. In all experiments the coupled races $\theta^1_{\mathrm{poly}}(\tau) = \tau$ performed particularly bad, see figures 1(a)-1(d), and $\theta^2_{\mathrm{poly}}(\tau)$ performed best (for unbounded racing on par with $\theta^3_{\mathrm{poly}}(\tau)$). In particular, this evaluation function worked (at least slightly) better than θ_{exp}. Only bounded races with $p = 6$ performed worse than the races with $\theta = \tau$. Of course, this it to be expected because in this case only very few racing steps are possible before the evaluation budget is spend. The evaluation function θ_{exp} always led to good performance but produced on average longer races than θ^2_{poly}. Thus, our experiments verified that the common choice (see equation (2)) indeed leads to unnecessarily long races and that this effect can be successfully countered by allowing the number of samples to depend non-linearly on the racing step.

5 Conclusion

Racing algorithms rely on concentration bounds to estimate the reliability of the estimated quality of an option. The tightness of these bounds affects the number of evaluations saved by applying racing algorithms. We argue that the usual coupling of racing steps and number of samples per tests in racing algorithms is a particularly bad choice and should be generalized. Tighter bounds can be achieved by allowing the number of samples grow faster than the number of tests. This is confirmed by our experiments for Hoeffding and empirical Bernstein races.

References

[1] O. Maron and A. W. Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 59–66. Morgan Kaufmann Publishers, 1994.

[2] O. Maron and A. W. Moore. The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 11:193–225, 1997.

[3] K. Bringmann and T. Friedrich. Approximating the least hypervolume contributor: NP-hard in general, but fast in practice. *Theoretical Computer Science*. doi:10.1016/j.physletb.2003.10.071.

[4] M. Birattari, T. Stützle, L. Paquete, and K. Varrentrapp. A racing algorithm for configuring metaheuristics. In E. Cantu-Paz et al., editors, *Genetic and Evolutionary Computation Conference (GECCO 2002)*, pages 11–18. Morgan Kaufmann Publishers, 2002.

[5] V. Heidrich-Meisner and C. Igel. Hoeffding and Bernstein races for selecting policies in evolutionary direct policy search. In *International Conference on Machine Learning (ICML)*, pages 401–408. ACM, 2009.

[6] P. Rolet and O Teytaud. Bandit-based estimation of distribution algorithms for noisy optimization: Rigorous runtime analysis. In C. Blum and R. Battiti, editors, *Learning and Intelligent Optimization (LION 4)*, volume 6073 of *LNCS*, pages 97–110. Springer, 2010.

[7] J.-B. Hoock and O. Teytaud. Bandit-based genetic programming. In A. I. Esparcia-Alcazar et al., editors, *European Conference on Genetic Programming (EuroGP 2010)*, volume 6021 of *LNCS*, pages 268–277. Springer, 2010.

[8] V. Mnih, C. Szepesvári, and J.Y. Audibert. Empirical Bernstein stopping. In *International Conference on Machine Learning (ICML 2008)*, pages 672–679, 2008.