Bayesian Inference for Algorithm Ranking Analysis

Borja Calvo
University of the Basque Country
UPV/EHU
Donostia-San Sebastian, Spain
borja.calvo@ehu.eus

Josu Ceberio
University of the Basque Country
UPV/EHU
Donostia-San Sebastian, Spain
josu.ceberio@ehu.eus

Jose A. Lozano
University of the Basque Country
UPV/EHU
Basque Center for Applied
Mathematics (BCAM)
Spain
ja.lozano@ehu.eus

ABSTRACT

The statistical assessment of the empirical comparison of algorithms is an essential step in heuristic optimization. Classically, researchers have relied on the use of statistical tests. However, recently, concerns about their use have arisen and, in many fields, other (Bayesian) alternatives are being considered. For a proper analysis, different aspects should be considered. In this work we focus on the question: what is the probability of a given algorithm being the best? To tackle this question, we propose a Bayesian analysis based on the Plackett-Luce model over rankings that allows several algorithms to be considered at the same time.

CCS CONCEPTS

• **Mathematics of computing → Bayesian computation**; *Hypothesis testing and confidence interval computation*; *Distribution functions*; *Statistical software*;

KEYWORDS

Bayesian analysis, statistical analysis, algorithm comparison, ranking models, Plackett-Luce model.

ACM Reference Format:
Borja Calvo, Josu Ceberio, and Jose A. Lozano. 2018. Bayesian Inference for Algorithm Ranking Analysis. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3205651.3205658

1 MOTIVATION

One of the most important research tasks in heuristic optimization is comparing the performance of different algorithms in solving the problem of interest. In academic papers where new algorithms are proposed, a proper comparison with state-of-the-art methods is absolutely mandatory. In real-life problems, the need to determine the most suitable algorithm (and set of parameters) renders the comparison equally critical.

Testing the algorithms in all the possible instances of a problem is certainly unfeasible and, thus, experimental comparisons typically involve running the algorithms for a subset of those instances. As a result, we have some empirical data –a *sample* of the algorithms' performance in that problem– to draw conclusions about their behavior.

The process usually includes two steps. First, the average results are displayed, paying attention to the variability of the results. This can be easily achieved using, for instance, simple boxplots. Although such representations can be enough to draw conclusions, usually some kind of statistical inference is used in a second stage.

Quite often, null hypothesis statistical tests (NHST) are used for the inference. When applied to the comparison of algorithms, NHST focus on testing the hypothesis of both algorithms being 'equal', i.e., having identical average performance. To that end, the observed performances are used to compute a statistic and, then, the so-called p-value is estimated. Then (after the due p-value corrections for multiple testing), the algorithms are declared different if and only if the (corrected) p-value is below the generally accepted threshold of 0.05 (0.01 if one wants to be more restrictive).

This practice is surrounded by controversy and its use (in certain situations) has been long criticized [4]. The main point in this long-lasting controversy has to do with the lack of interpretability of the produced results (the so-called p-values). Indeed, NHST are quite easy to apply, but far too often the produced results are misinterpreted in scientific studies [6].

The concern about the use of NHST has been growing in the last few years. A recent example of this concern is the statement about p-values published by the American Statistical Association (ASA) [8]. To briefly illustrate the main drawbacks of using NHST for the comparison of algorithms, consider these[1]:

- From a probabilistic point of view, the p-value is computed assuming that the null hypothesis (i.e., that the average behavior of both algorithms is identical) is true. However, no matter how small is the difference, this hypothesis is hardly true and, thus, the p-value is a probability computed under a false assumption.
- The p-value is usually (sometimes unconsciously) assumed to be a proxy of the magnitude of the true difference (i.e., the difference of the average performance of both algorithms when applied to all the instances). Therefore, if the p-value is very small, we assume that the differences are big. However, the p-value is not only affected by the magnitude of the difference, it is also affected by the sample size and, thus, with big enough samples we can get a p-value as small as needed, no matter how small the average difference is.

[1] These are some examples of usual problems with NHST, a more comprehensive list can be found in [6]

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07…$15.00
https://doi.org/10.1145/3205651.3205658

- In many papers the p-value is taken as a kind of oraculus: If the p-value is below 0.05 one can say (prove) that the algorithms are (significantly) different. Moreover, the opposite, that is, that a p-value above 0.05 means that the algorithms perform equally, is quite often assumed true when it is not.

Now the question is, what is the contribution of NHST to the analysis? As a way of making inference about the magnitude of the difference, they are useless as they only analyze what would happen if there where no differences[2]. In terms of interpreting the p-value, they are (in this context) quite limited, as the p-value represents the frequency of incorrectly affirming that two algorithms are different when they are actually not (something that, most of the times, is false).

As their use for proper inference in this type of context is limited, in different fields researchers are moving from the classical NHST-based analyses to other alternatives. In the particular case of algorithm comparison, a recent publication in JMLR [1] proposes shifting the analysis from the frequentist to the Bayesian approach. It is worth noting that one of the authors of the paper is Janez Demsâr, one of the main references in the use of NHST for the comparison of multiple classifiers.

Bayesian analysis is not the only way to do inference and, certainly, even in Bayesian statistics there are different approaches (hypothesis testing included). In fact, instead of relying on a single methodology, more robust analysis would be obtained if different tools are used to assess the different aspects of the comparison.

With that aim, in this work we propose a Bayesian approach based the the Plackett-Luce (PL) model [7] to answer a simple question: How likely is an algorithm the best to solve a problem? In particular, we will focus on the question from a ranking point of view, i.e., without considering the magnitude of the differences. Nonetheless, the magnitude is also very relevant and, to assess that aspect, there are other proposals (see, for instance, [1, 3]) and tools such as visualization methods (e.g., boxplots).

There are a number of Bayesian methods proposed to analyze data coming from the comparison of algorithms. However, as far as we know, all of them focus on the pairwise comparison of algorithms. Despite the great relevance of these methods, they are not enough to have a simple answer to the big question we usually want to answer: How likely is my proposal to be the best algorithm to solve a problem?

2 PLACKETT-LUCE BAYESIAN MODEL

There are many probabilistic models defined for the space of rankings (permutations). Among them there is one, the PL model [7], that has one interesting property: the (normalized) parameters of the model directly represent the marginal probability of an algorithm[3].

In this work we propose using the PL model with a Dirichlet prior, that is:

$$P(\mathbf{w}|R) \propto Dir(\mathbf{w};\alpha)\prod_{\sigma\in R} P_{PL}(\sigma;\mathbf{w}), \tag{1}$$

where α are the hyper-parameters of the prior distribution.

The above equation cannot be analytically solved, but the posterior distribution of weights can be easily sampled using Markov chain Monte Carlo (MCMC) methods [5]. When applied to the ranking data derived from the comparison we obtain an approximation of the posterior distribution of weights that can be used to answer different questions.

The weight associated to a given algorithm can be interpreted as its probability of being the best. Therefore, for the most simple analysis we can pay attention to the sample of each weight individually, which represents the distribution of the probability of the algorithm being the best.

As a way of illustrating the use of the method we have applied it to an existing comparison of algorithms [2] where the original analysis was conducted using the classical NHST approach[4].

3 CONCLUSIONS

Both the originally used NHST approach and our proposed method are based on rankings and, as such, the conclusions drawn by both analysis are similar. However, there is big difference in the interpretation of the results. In the case of NHST we have the average ranking and, for each pair of algorithms, we say whether there are significant differences or not. Conversely, in the case of the Bayesian analysis we can estimate the expected probability of a given algorithm (or subset of them) being the best. Moreover, having the posterior distribution of weights allows us not only to get the expected probabilities but also the uncertainty about their estimation (e.g., getting the interquartile rage). To sum up, the Bayesian inference approach allows us to make more precise statements.

ACKNOWLEDGMENTS

This work has been partially supported by the Research Groups 2013-2018 (IT-609-13) and Elkartek programs (Basque Government), and TIN2016-78365R and Severo Ochoa Program SEV-2013-0323 (Spanish Ministry of Economy, Industry and Competitiveness). Jose A. Lozano is also supported by BERC 2014-2017.

REFERENCES

[1] Alessio Benavoli, Giorgio Corani, Janez Demšar, and Marco Zaffalon. 2017. Time for a Change: a Tutorial for Comparing Multiple Classifiers Through Bayesian Analysis. *Journal of Machine Learning Research* 18, 77 (2017), 1–36.

[2] Josu Ceberio, Ekhine Irurozki, Alexander Mendiburu, and Jose A Lozano. 2012. A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems. *Progress in Artificial Intelligence* 1, 1 (2012), 103–117.

[3] C.P. de Campos and A. Benavoli. 2016. Joint Analysis of Multiple Algorithms and Performance Measures. *New Generation Computing* 35, 1 (2016), 69–86.

[4] Gerd Gigerenzer and Julian N. Marewski. 2015. Surrogate Science: The Idol of a Universal Method for Scientific Inference. *Journal of Management* 41, 2 (2015), 421–440.

[5] Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. 1995. *Markov chain Monte Carlo in practice.* CRC press.

[6] Greenland, Sander and Senn, Stephen J and Rothman, Kenneth J and Carlin, John B and Poole, Charles and Goodman, Steven N and Altman, Douglas G. 2016. Statistical tests, P values, confidence intervals, and power: a guide to misinterpretations. *European journal of epidemiology* 31, 4 (2016), 337–350.

[7] Robin L. Plackett. 1975. The Analysis of Permutations. *Journal of the Royal Statistical Society* 24, 10 (1975), 193–202.

[8] Wasserstein, Ronald L and Lazar, Nicole A. 2016. The ASA's statement on p-values: context, process, and purpose. *The American Statistician* 70, 2 (2016), 129–133.

[2] Unless we conduct a power analysis, something that is by no means the most common situation.

[3] The elements or items of the permutations will represent, in our context, algorithms and, thus, from now on we will refer to them as algorithms.

[4] More details on the method and the conducted experimentation can be consulted at http://www.sc.ehu.es/ccwbayes/members/borxa/bPL