Measures for Combining Accuracy and Time for Meta-learning

Salisu Mamman Abdulrahman[1] and **Pavel Brazdil**[1,2]

Abstract. The vast majority of studies in meta-learning uses only few performance measures when characterizing different machine learning algorithms. The measure *Adjusted Ratios of Ratio (ARR)* addresses the problem of how to evaluate the quality of a model based on the *accuracy* and *training time*. Unfortunately, this measure suffers from a shortcoming that is described in this paper. A new solution is proposed and it is shown that the proposed function satisfies the criterion of monotonicity, unlike ARR.

1 INTRODUCTION

The major reason why data mining has attracted a great deal of attention in the information industry and in society as a whole in recent years is due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The information and knowledge gained can be used for applications ranging from market analysis, fraud detection, customer retention to production control and science exploration.

Data mining tools such as Weka, Knime, and RapidMiner contain hundreds of operators covering a wide range of data analysis tasks, but unfortunately provide only limited advice on how to select the right method according to the nature of the problem under analysis.

To alleviate these problems, different systems have been developed that "intelligently" help users to analyze their data. The goal of *Meta-learning* systems is to help the user by providing some guidance [1, 2, 3]. This is done by suggesting a particular algorithm or operation(s) (e.g. application of particular preprocessing operation or classification algorithm) to the user that would lead to good performance.

The vast majority of studies in meta-learning uses only few performance measures when characterizing different machine learning algorithms. Regards classification, for instance, one common measure is *predictive accuracy*. Other researchers have used also AUC, area under the ROC curve, or else precision, recall and F1. What is common to all these measures is the higher the value, the better. Costs of operations, and in particular training time, are different though, as the lower the value, the better.

An aggregate metric that combine both accuracy and time as metric was presented in [4], ARR, the *adjusted ratio of ratios*, which allows the user to add more emphasis either on the predictive accuracy or on the training time. This measure suffers however, from a shortcoming, which is described in the next section.

2 RANKING BASED ON ACCURACY AND TIME

The *Adjusted Ratio of Ratios* (ARR) measure aggregates information concerning accuracy and time. It can be seen as an extension of the *success rate ratios* (SRR) method. This method was presented in [4] together with two other basic measures, *average ranks* (AR) and *significant wins* (SW). This multicriteria evaluation measure combines the information about the accuracy and total training/execution time of learning algorithms and is defined as:

$$ARR^{d_i}_{a_p a_q} = \frac{\frac{SR^{d_i}_{a_p}}{SR^{d_i}_{a_q}}}{1+AccD*\log\left(\frac{T^{d_i}_{a_p}}{T^{d_i}_{a_q}}\right)} \quad (1)$$

where $SR^{d_i}_{a_p}$ and $T^{d_i}_{a_p}$ represent the success rate and time of algorithm a_p on dataset d_i, respectively. The term $SR^{d_i}_{a_p}/SR^{d_i}_{a_q}$ is the ratio of success rates which can be seen as a measure of the advantage of algorithm a_p over algorithm a_q (i.e., a benefit). The equivalent ratio for time, $T^{d_i}_{a_p}/T^{d_i}_{a_q}$, can be seen as a measure of the disadvantage of algorithm a_p over algorithm a_q (i.e., a cost). Thus, the authors have taken the ratio of the benefit and the cost, obtaining thus a measure of the overall quality of algorithm a_p.

However, we note that time ratios have, in general, a much wider range of possible values than success rate ratios. If a simple time ratio were used it would dominate the ratio of ratios. This effect can be controlled by re-scaling using $\log(T^{d_i}_{a_p}/T^{d_i}_{a_q})$ which provide a measure of the order of magnitude of the ratio. The relative importance between accuracy and time is taken into account by multiplying this expression by the *AccD* parameter. This parameter is provided by the user and represents the amount of accuracy he/she is willing to trade for a 10 times speedup or slowdown. For example, *AccD* = 10% means that the user is willing to trade 10% of accuracy for 10 times speedup/slowdown. Finally, the value of 1 is added to $\log(T^{d_i}_{a_p}/T^{d_i}_{a_q})$ to yield values that vary around 1, as happens with the success rate ratio.

The ARR should ideally be monotonically increasing. Higher success rate ratios should lead to higher values of ARR. Higher time ratios should lead to lower values of ARR. The overall effect of combining the two should again be monotonic.

We have decided to verify whether this property can be verified on data. We have fixed the value of SRR to 1 and varied the time ratio from very small values (2^{-20}) to very high values (2^{20}) and calculated the ARR for three different values of AccD (0.2, 0.3 and 0.7). The result can be seen in the plot in Fig. 1. The horizontal axis shows the log of the time ratio (logRT). The vertical axis shows the ARR value.

As can be seen, the resulting ARR function is not monotonic and even approaching infinity at some point. Obviously, this can lead to incorrect rankings provided by the meta-learner. However, what is even more worrying is that this can affect the evaluation results. In the next section, we propose a solution to this problem.

3 OUR PROPOSED SOLUTION

When devising a new solution we did not wish to change the overall philosophy underlying ARR. We believe that it is indeed a good idea to work with ratios, as absolute numbers do not carry much meaning.

[1] LIAAD Inesc Tec, Porto, sma@inescporto.pt, pbrazdil@inescporto.pt

[2] FEP, University of Porto.

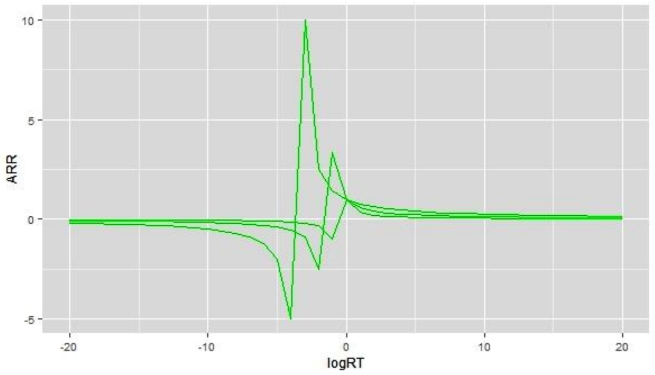

Figure 1. ARR with three different values for AccD (0.2, 0.3 and 0.7)

The accuracy of 90% can be considered good in one situation, but very bad in another. After some reflection, we have realized that the problem lies in the way how the time ratio has been re-scaled. So, we considered another way of re-scaling, which does not use *log*, but n-th root instead, where *n* is a parameter. The proposed function is referred to as A3R and is defined as follows:

$$A3R^{d_i}_{a_p a_q} = \frac{\dfrac{SR^{d_i}_{a_p}}{SR^{d_i}_{a_q}}}{\sqrt[n]{T^{d_i}_{a_p}/T^{d_i}_{a_q}}}, \qquad (2)$$

As Fig. 2 shows, this function is monotonic. The higher the A3R, the better.

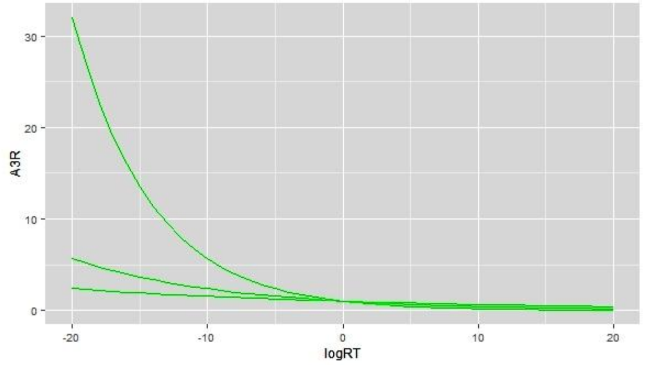

Figure 2. A3R for three different settings for the n-th root (4, 8, and 16)

Taking n-th root in the denominator of eq.(2) enables to rescale the ratio of times. The higher the value of n, the greater the rescaling. So, for instance, if one algorithm is 10 slower than another, the ratio is 10. Taking for 8-th (2nd) root of this will decrease it to 1.33 (3.16). If the ratio were 0.1 this would result in 0.74 (0.31). All numbers get closer to 1 after rescaling.

The change from ARR to A3R is important, as we wish to recalculate many meta-learning experiments and consider both accuracy ratios (and possibly AUC ratios) together with time ratios, suitably rescaled.

To understand the relationship between the success rate ratios (SRR) and time ratios (RT), we have constructed iso-A3R curves (Fig.3). The horizontal axis plots *logRT* in an increasing order of time rate ratios. Thus negative values on the left characterize fast algorithms, while the positive values on the right characterize slow ones. The vertical axis shows the success rate ratios (SRR). Each curve shows the values of A3R where the values are constant. The blue (red, green) curve represents situations where *A3R* is 0.9 (1.0, 1.1). As the ratio of times decreases (i.e. the algorithm is faster), it is sufficient to have lower values of the success rate ratio (SRR) to obtain the same value of A3R.

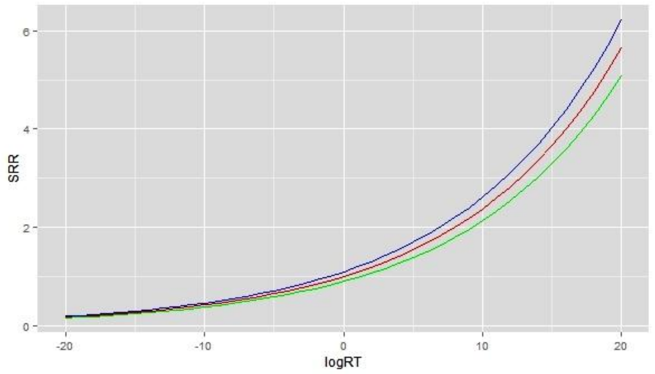

Figure 3. Iso-curves with three different values of A3R (0.9, 1.0 and 1.1). Here n=8 was used to calculate the root.

4 FUTURE PLANS

We intend to improve the methods presented in [5] which rely on relatively pairwise comparison involving two algorithms. We plan to upgrade this work by considering the information concerning both accuracy (or AUC) ratios and time ratios. Hence, the new function proposed will be very useful.

Besides, another challenge is that the new set-up would use many more algorithms (in the order of 100's) than in previous studies. We will exploit the OpenML [6] database in this process and collaboration is underway with U.Leiden on running some of the experiments and re-using the results. Considering that the number of algorithms is high, we need to re-think the method based on pairwise comparisons.

Furthermore, we plan to use the method based on sampling landmarks, as in [5]. To simplify the whole procedure, we will probably use a fixed set of samples, rather than using some dynamic sampling strategy, as proposed in [5]. Still, we need to evaluate what the best number of samples is from the benefit-cost perspective.

5 CONCLUSION

We have presented a new measure A3R for evaluating the performance of algorithms that considers both accuracy and time ratios suitably re-scaled. We have shown that this measure satisfies the criterion of monotonicity, unlike the previous version ARR. We have discussed the usage of A3R in further experiments on meta-learning.

This work is funded (or part-funded) by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project «FCOMP - 01-0124-FEDER-022701»

REFERENCES

[1] P. Brazdil, C. Giraud-Carrier, C.Soares, and R. Vilalta, *Metalearning:* Applications *to Data Mining*, Springer, 2009.

[2] Kalousis, A. (2002). Algorithm selection via meta-learning. *PhD* Thesis. *University of Geneva*.

[3] Gama, J. and P. Brazdil (1995). Characterization of classification algorithms. *Lecture Notes in Computer Science 990*, 189–200.

[4] Brazdil, P. B., C. Soares, and Joaquin Pinto Da Costa. "Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results." *Machine Learning* 50.3 (2003): 251-277.

[5] Leite, R., and P. Brazdil. "Active Testing Strategy to Predict the Best Classification Algorithm via Sampling and Metalearning." *ECAI*. 2010.

[6] Vanschoren, J.. "The experiment database for machine learning." *5th Planning to Learn Workshop, WS28 at ECAI-2012*. 2012.