On the Evaluation of Algorithm Selection Problems

Jan N. van Rijn J.N.VAN.RIJN@LIACS.LEIDENUNIV.NL
Leiden Institute of Advanced Computer Science, Leiden University, P.O. Box 9512, 2300 RA, Leiden, Netherlands

Salisu Mamman Abdulrahman SMA@INESCPORTO.PT
Pavel Brazdil PBRAZDIL@INESCPORTO.PT
LIAAD-INESC TEC, University of Porto, Campus da FEUP, Rua Dr. Roberto Frias, 4200 - 465, Porto, Portugal

Joaquin Vanschoren J.VANSCHOREN@TUE.NL
Dept. of Mathematics and Computer Science, TU Eindhoven, P.O. Box 513, 5600 MB, Eindhoven, Netherlands

Keywords: Algorithm Selection, Meta-Learning, Subsampling

Abstract

Meta-learning aims to learn which learning techniques work well on what data. Rather than recommending a single classifier, a ranking should be created, ordering the classifiers on their estimated performance. It is unclear how to evaluate such rankings. In this paper we propose the use of Loss Time curves. We show that for meta-learning techniques to perform well on this measure, they should also take into consideration the amount of time spent to train an algorithm.

1. Introduction

Meta-learning aims to learn which learning techniques work well on what data (Vilalta & Drissi, 2002). A common task, known as the Algorithm Selection Problem, is to determine which classifier performs best on a given dataset. We can predict this by training a meta-model on meta-data comprised of dataset characterizations, i.e., *meta-features* (Brazdil et al., 1994), and the performances of different classifiers on these datasets. The same meta-features can be computed on each new dataset and fed to the meta-model to predict which classifiers will perform well.

In many meta-learning applications it is not enough to simply predict the single best classifier. When the recommended classifier does not perform well enough, an alternative should be at hand. Rather than recommending a single classifier, a ranking should be created, ordering the classifiers on their likelihood of performing well on the dataset. This way, the user can make an informed decision about which models to try based on the available time and resources. Typically, a cross-validation run will be performed on the top-ranked algorithm, and if the result was not good enough, the next one can be tried. One way of evaluating such rankings is using *Loss Curves* (Leite et al., 2012). Loss is defined to be the difference between current best classifier performance against global best classifier performance. A loss curve plots the amount of loss against the number of tests that were needed to obtain such loss (see Figure 1).

Loss Curves assume that every test will take the same amount of time, which is not realistic. For example, Multilayer Perceptrons take longer to train than Naive Bayes classifiers. Therefore, it is better to use *Loss Time Curves*, which plot the average loss against the time needed to obtain this loss (see Figure 2). It describes how much time is needed on average to converge to a certain loss (lower is better). The faster such curve goes to a loss of zero, the better the technique is. They have been used before in the Optimization literature (Hutter et al., 2010).

For a significant longer version of this paper, the reader is referred to van Rijn et al. (2015).

2. Combining Accuracy and Runtime

As Loss Time Curves also take into consideration the amount of time spent to train an algorithm, the meta-learning algorithm should be aware of this. Various measures of combining accuracy have been proposed,

Appearing in *Proceedings of Benelearn 2016*.

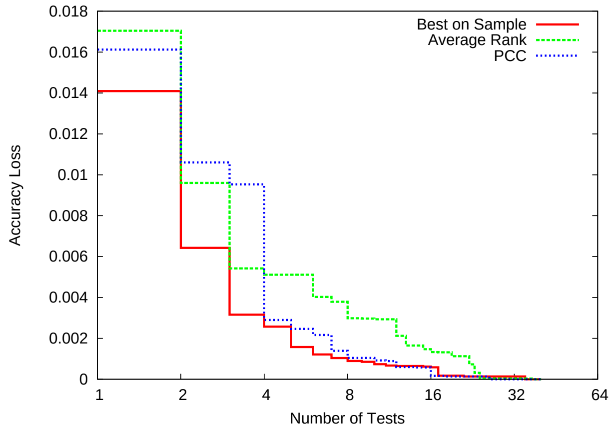

Figure 1. Loss Curves of three meta-learning methods.

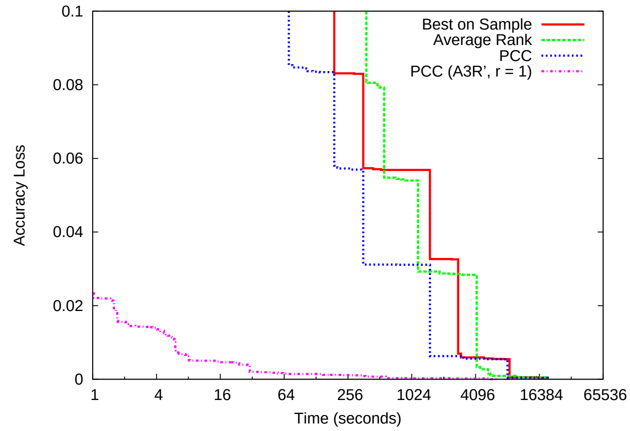

Figure 2. Loss Time Curves of meta-learning methods.

for example ARR and A3R (Abdulrahman & Brazdil, 2014). Most meta-learning methods can be adopted in such a way that they incorporate this measure. By using such measures, the methods are urged to first try relative fast methods that are expected to perform reasonably, before trying slow methods that are expected to perform best.

We compare three meta-learning techniques using Loss Time Curves. The *Average Rank* method ranks the classifiers in the order of their average rank on previously seen datasets and recommends the classifiers in that order. The *Best on Sample* method runs all classifiers using a given sample size (in this case 256 instances), and ranks the classifiers in the order of performance on that sample. The *Pairwise Curve Comparison* method (PCC) is an extension to a more sophisticated sample-based method, as presented by Leite & Brazdil (2010). We also include a version of this technique using A3R. We compare the methods on the task of ranking 53 classifiers on 39 datasets from OpenML [1] (Vanschoren et al., 2014).

The results are shown in Figure 1 and Figure 2. One observation is that performances in loss space do not necessarily agree with performances in loss time space. Furthermore, incorporating a measure that combines runtime and accuracy has a tremendous effect on the performances recorded in loss time space.

Acknowledgments

This work is supported by grant 612.001.206 from the Netherlands Organisation for Scientific Research (NWO).

[1]Full details: `http://www.openml.org/s/2`

References

Abdulrahman, S. M., & Brazdil, P. (2014). Measures for Combining Accuracy and Time for Meta-learning. *Meta-Learning and Algorithm Selection Workshop at ECAI 2014* (pp. 49–50).

Brazdil, P., Gama, J., & Henery, B. (1994). Characterizing the Applicability of Classification Algorithms Using Meta-Level Learning. In *Machine learning: Ecml-94*, vol. 784 of *Lecture Notes in Computer Science*, 83–102. Springer.

Hutter, F., Hoos, H. H., Leyton-Brown, K., & Murphy, K. (2010). Time-Bounded Sequential Parameter Optimization. In *Learning and intelligent optimization*, 281–298. Springer.

Leite, R., & Brazdil, P. (2010). Active Testing Strategy to Predict the Best Classification Algorithm via Sampling and Metalearning. *ECAI* (pp. 309–314).

Leite, R., Brazdil, P., & Vanschoren, J. (2012). Selecting Classification Algorithms with Active Testing. In *Machine learning and data mining in pattern recognition*, 117–131. Springer.

van Rijn, J. N., Abdulrahman, S. M., Brazdil, P., & Vanschoren, J. (2015). Fast algorithm selection using learning curves. In *Advances in intelligent data analysis xiv*, 298–309. Springer.

Vanschoren, J., van Rijn, J. N., Bischl, B., & Torgo, L. (2014). OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, *15*, 49–60.

Vilalta, R., & Drissi, Y. (2002). A Perspective View and Survey of Meta-Learning. *Artificial Intelligence Review*, *18*, 77–95.