Meta-modelling Execution Time of Data Minning Algorithms

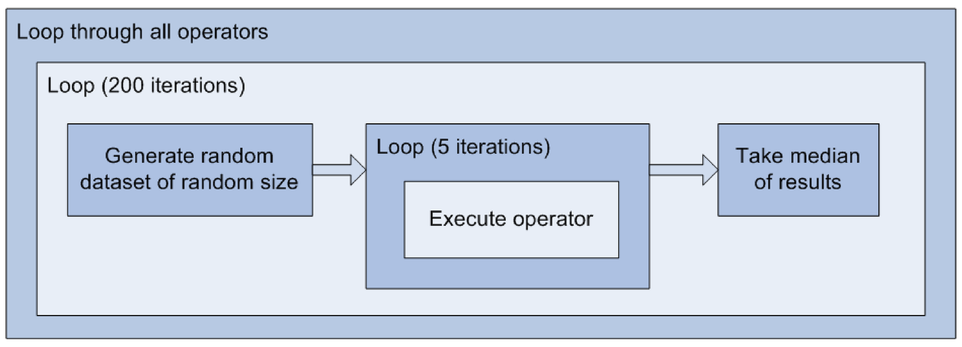
Matija Piškorec[1], Matko Bošnjak[2], Tomislav Šmuc[1]

[1]Ruđer Bošković Institute, Croatia
[2]Faculty of Engineering, University of Porto, Portugal

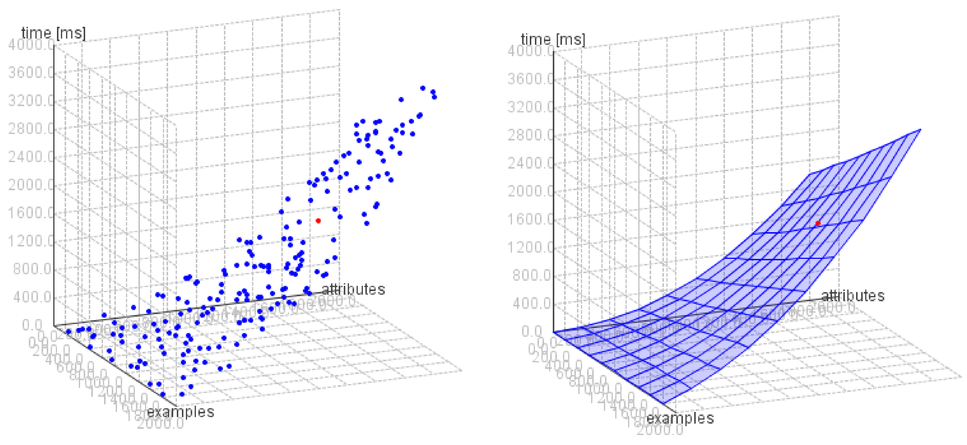
Abstract

Knowing the execution time of a computational model, especially when dealing with large data, is crucial in deciding whether the solution of the problem is attainable in acceptable time. In the case of data mining processes, typically both the modelling and the model application execution time are important. We developed a meta-mining framework for execution time estimation of data mining algorithms built in RapidMiner. Operator execution time estimation is treated as a machine learning problem for which we built prediction models using execution times obtained by running algorithms on a set of predetermined datasets. With the appropriate refitting, this experimental methodology is applicable to any data mining environment. We an present overall framework with modelling results for a subset of RapidMiner operators, and compare non-parametric distance measures based predictions with polynomial function fitting. Finally, we demonstrate the integration of these models in the form of a standalone RapidMiner extension and discuss issues related to reliability, scalability and applicability for the overall workflow execution time modelling.

Measuring execution time



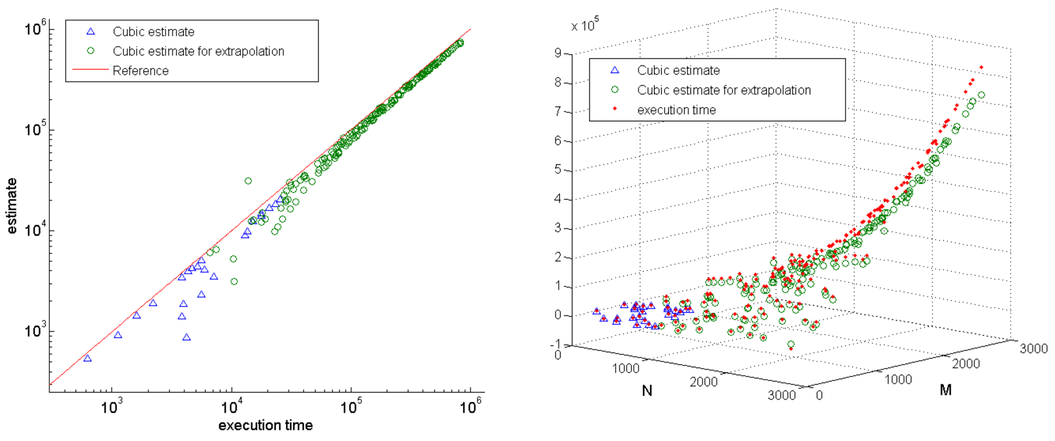
Estimation methodology



Pager: A kNN-based algorithm for regression [12], Pager is a parameterless algorithm, providing reliable and robust estimations without the need for providing the explicit regression equation. Our implementation is done in Java.

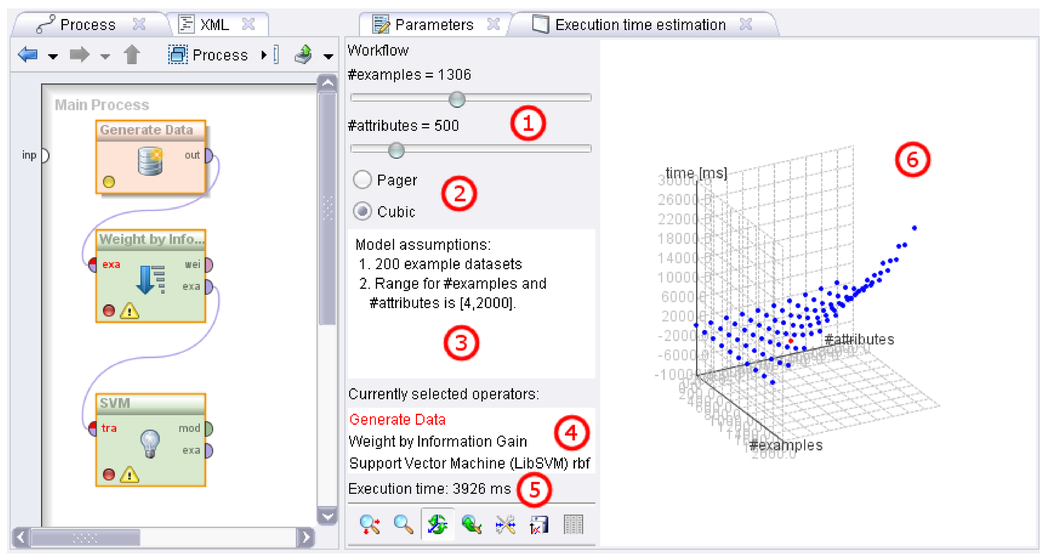
Polynomial: Cubic function fitting, suitable for a small number of meta-features. Our implementation is a simple gradient descent, optimised with the fminsearch function in Matlab.

Extrapolation of estimations



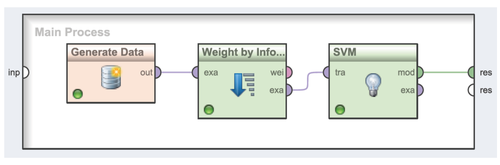
Extrapolation of estimated execution times to parameter ranges not used in experimentation for polynomial model. This example shows execution time estimates for a simple workflow consisting of "Weight by Information Gain" and "SVM" with rbf kernel operators where individual estimates for each operator are summed. Graphs show comparison between estimates and real values of execution times for test cases inside experimentation range (triangles) and outside experimentation range (circles). Red points on the right are real execution times while red line on the left is a reference representing ideal estimation.

Execution time extension for RapidMiner

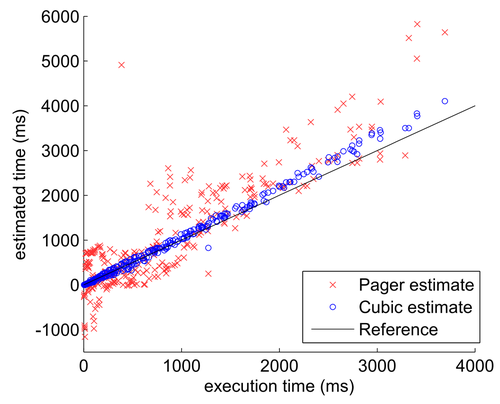


We implemented a RapidMiner extension that interactively displays execution time information for operators selected in the main process. Basic functionalities are: (1) sliders for choosing number of attributes and examples for the input dataset, (2) prediction model (Pager or polynomial), (3) info box about for the estimation model, (4) list of currently selected operators and (5) their execution time estimate, (6) interactive display showing execution time surface in relation to the number of examples and number of attributes.

Use case: a simple classification workflow



Simple use case consisting of feature weighting operator ("Weight by Information Gain") and classification operator ("SVM" with rbf kernel). On the right is the comparison of estimated and real execution times for estimation with Pager and polynomial model.



Literature

[1] S. D. Abdelmessih, F. Shafait, M. Reif, and M. Goldstein. Landmarking for meta-learning using rapidminer. In Proceedings of RapidMiner Community Meeting and Conference, September 2010.
[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and Stein C. Introduction to Algorithms. The MIT Press, 3rd edition, 2009.
[3] J. Engblom, A. Ermedahl, M. Sjödin, J. Gustavsson, and H. Hansson. Towards industry strength worst-case execution time analysis. Technical Report ASTEC 99/02, and DoCS 99/109, Uppsala University, April 1999.
[4] T. K. Ho, M. Basu, and M. Law. Measures of geometrical complexity in classication problems. Data Complexity in Pattern Recognition, pages 123, 2006.
[5] T. K. Ho and Basu. M. Complexity measures of supervised classication problems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(3):289300, 2002.
[6] M. A. Iverson, F. Özgüner, and L. C. Potter. Statistical prediction of task execution times through analytic benchmarking for scheduling in a heterogeneous environment. IEEE Transactions on Computers, 48:1374 1379, 1999.
[7] C. Koepf, C. Taylor, and J. Keller. Meta-analysis: Data characterisation for classication and regression on a meta-level. In Proceedings of International Symphosium on Data Mining and Statistics, 2000.
[8] Y. Peng, P.A. Flach, C. Soares, and P. Brazdil. Improved dataset characterisation for meta-learning. Discovery Science, pages 141152, 2002.
[9] B. Pfahringer, H. Bensusan, and C. Giraud-Carrier. Meta-learning by landmarking various learning algorithms. In Proceedings of International Conference on Machine Learning, 2000.
[10] P. Puschner and Ch. Koza. Calculating the maximum execution time ofreal-time programs. Real-Time Systems, 1(2):159176, 1989.
[11] M. Reif, F. Shafait, and A. Dengel. Prediction of classier training timeincluding parameter optimization. In Proceedings of the 34th Annual Ger-man Conference on Articial Intelligence, volume 7006, pages 260271,2011.
[12] H. Singh, A. Desai, and V. Pudi. Pager: Parameterless, accurate, generic,ecient knn-based regression. Database and expert systems applications,6262:168176, 2012.

Acknowledgments

This work is supported by the European community 7th framework ICT-2007.4 (No 231519) "e-LICO: An e-Laboratory for Interdisciplinary Collaborative Research in Data Mining and Data-Intensive Science".

Disclosure: This poster is based on paper "Meta-Modeling Execution Times of RapidMiner Operators" by same authors approved for publication on RCOMM 2012 conference in Budapest held from 27th until 31st of August, 2012.