Automatic Ranking of Classification Algorithms

A Comparison of Regression-Based Ranking and Preference Models

Bachelor Thesis Proposal & Work Plan

Helena Graf

matriculation number: 7011643

hgraf@mail.upb.de

November 20, 2017

Supervisors:

Prof. Dr. Eyke Hüllermeier

Prof. Dr. Axel-Cyrille Ngonga Ngomo

Contents

1 Motivation

Selecting a fitting classifier for a new problem is difficult, since algorithm performances can vary substantially among datasets, and it is not feasible to simply apply a large number of them to empirically to find a good match. For example, on a dataset about the electricity prices in the Australian state New South Wales, the predictive accuracy for the Multilayer Perceptron[1] is 0.7887 . The predictive accuracy of the Random Forests[2] algorithm on the same data set is 0.9236, a much higher value. On a different data set, with the topic of vehicle silhouetes, we get a predictive accuracy of 0.7979 for the Multilayer Perceptron, and 0.7518 for Random Forests, showing an advantage of the former on this dataset[3]. So in each case, one would have picked a different algorithm in order to achieve the best results.

Due to the fact that algorithm selection is a common problem in machine learning, one would generally like to automate that process. This approach is based on the idea, that one can predict a good classifier by generalizing from the algorithm's past performances. Hence, we attempt this by applying machine learning, regression models and preference learning, to be precise, to performance measures. Therefore, the same task would ideally not have to be solved repetitively by independent people, but instead, the insight gained about the learning process would be re-used.

Furthermore, the idea of selecting one single algorithm for a given problem can be relaxed in the context of this thesis. Since the prediction will most likely not be correct in all cases, and the performance of an algorithm is also dependent on the hyperparameter-tuning, returning a ranking of classifiers is sensible. This ranking could then be used in another framework to inspect top-rated algorithms closer, in order to return a better algorithm to the user.

[1] Hyperparameters: L:0.3,M:0.2,N:500,V:0,S:0,E:20,H:a.
[2] Hyperparameters: P:100,I:100,num-slots:1,K:0,M:1.0,V:0.001,S:1.
[3] Hyperparameters as above.

2 Related Work

2.1 Evolutionary Algorithms

2.2 Runtime Predictions

3 Goals

The overall goal of this thesis is to test the assumption that one can predict an accuracy-based ranking of classification algorithms for a new dataset given past performances of the classifiers. In this context, this will be done by implementing two different methods for generating a ranking, regression-based and preference based ranking, and comparing the results.

3.1 Required Goals

Implementation in Java to be done in two phases
Using jpl for rankers, ml-plan for regression, data from openML, based on weka,
Evaluation procedure via Kindall
Evaluate against??

3.2 Optional Goals

Adding another layer to search by including hyperparameters, possibly by means of iterative requests to tool.
Is succesful, add runtime and / or complexity and make combined predictions

4 Task Description

Section might be obsolete - approach described under 'goals'? Or need more details?

5 Preliminary Document Structure

1. Introduction
   1.1 Solution Approach
   2.2 Structure
2. Fundamentals
   2.1 ML-Plan
   2.2 JPL
3. Implementation
   3.1 Regression-based Ranking
      3.1.1 Preliminary Regression Algorithm
      3.2.2 Automatic Algorithm Selection
   3.2 Preference Models
      i. Preliminary Ranking Algorithm
      ii. Automatic Algorithm Selection
4. Evaluation
   4.1 Computing the Accuracy of the Results
   4.2 Assessment of the Accuracy
5. Related Work
   5.1 Evolutionary Algorithms
   5.2 Runtime Predictions
6. Conclusion
7. Literature
8. Appendix

6 Schedule

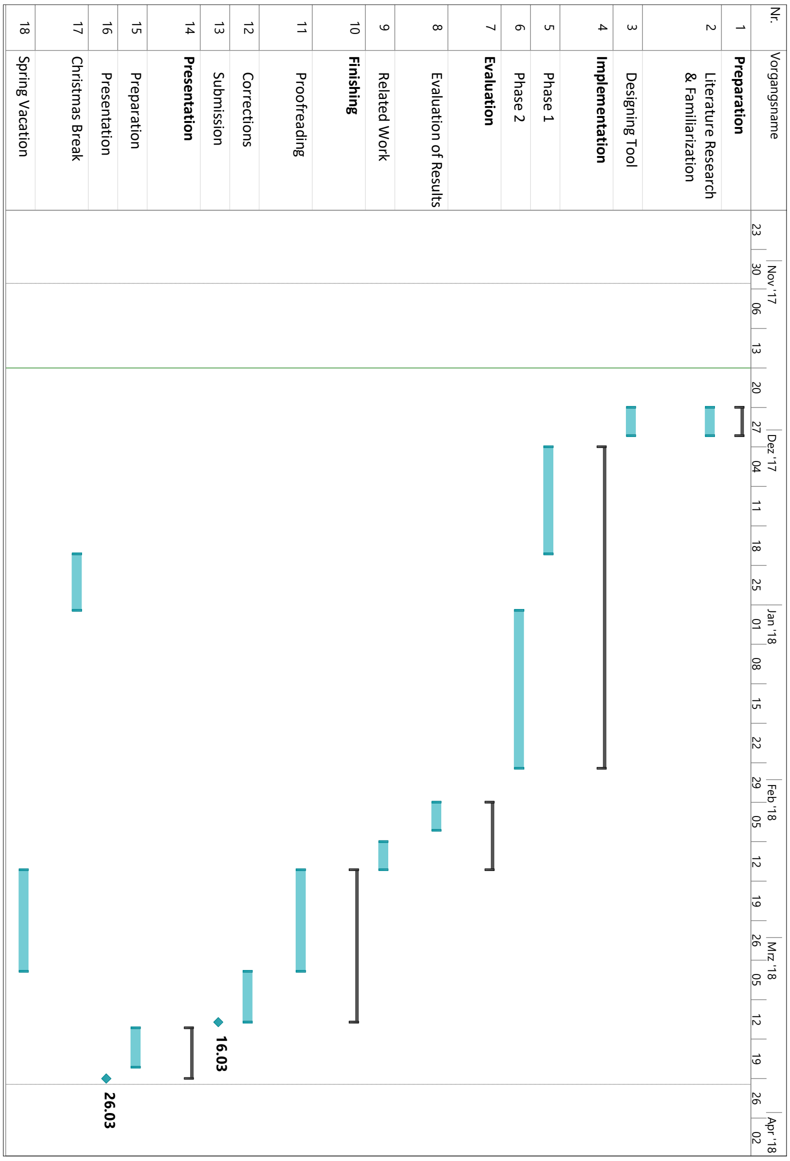

Figure 1: Sketch of the time schedule for the work on the thesis

References

[BL05] Bongard, Josh; Lipson, Hod: Active Coevolutionary Learning of Deterministic Finite Automata. In: *J. Mach. Learn. Res.* 6 (2005), Dezember, S. 1651–1678. – ISSN 1532–4435

[CHT03] Channabasavaiah, Kishore; Holley, Kerrie; Tuggle, Edward: Migrating to a service-oriented architecture. In: *IBM DeveloperWorks* 16 (2003)

[DPAM00] Deb, Kalyanmoy; Pratap, Amrit; Agarwal, Sameer; Meyarivan, T.: A Fast Elitist Multi-Objective Genetic Algorithm: NSGA-II. In: *IEEE Transactions on Evolutionary Computation* 6 (2000), S. 182–197

[Hor51] Horn, Alfred: On Sentences Which are True of Direct Unions of Algebras. In: *The Journal of Symbolic Logic* 16 (1951), Nr. 1, S. 14–21. – ISSN 00224812

[Lan94] Lankhorst, Marc M.: Breeding Grammars: Grammatical Inference with a Genetic Algorithm. In: *Proceedings of the 1994 EUROSIM conference on massively parallel processing applications and development*, Elsevier, 1994, S. 423–430

[LPP98] Lang, Kevin J.; Pearlmutter, Barak A.; Price, Rodney A.: Results of the Abbadingo One DFA Learning Competition and a New Evidence-Driven State Merging Algorithm. In: *Proceedings of the 4th International Colloquium on Grammatical Inference*. London, UK, UK : Springer-Verlag, 1998 (ICGI '98). – ISBN 3–540–64776–7, S. 1–12

[LR05] Lucas, Simon M.; Reynolds, T. J.: Learning Deterministic Finite Automata with a Smart State Labeling Evolutionary Algorithm. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (2005), Juli, Nr. 7, S. 1063–1074. – ISSN 0162–8828

[Obj15] Object Management Group (OMG): *OMG Unified Modelling Language*. v 2.5. March 2015

[RH16] van Rooijen, Lorijn; Hamann, Heiko: Requirements Specification-by-Example Using a Multi-Objective Evolutionary Algorithm. (2016). – AIRE16

Supervisor	Student
(Prof. Dr. Eyke Hüllermeier)	(Helena Graf)