Predicting the Performance of SAT Solvers

SAT solving is one of the fundamental problems in computer science, with many real-world applications. Due to the NP-hardness of this problem, there is ongoing research on efficient search strategies, e.g., using conflict-driven clause learning (CDCL) approaches. International competitions like the SAT Competition 2020 compare solvers on a variety of SAT instances from different application areas. Solver runtime is the key performance measure, with a cut-off time and a penalty for long-running solvers. As in many areas of computer science, there is no single algorithm that performs best for all problem instances. However, getting insights into the reasons for good or bad solver performance are of vital interest for the community. Such knowledge can help to improve individual solvers or to improve the selection of benchmark instances.

Techniques from data science and machine learning can help to answer questions about solver performance. For example, one can predict the runtime of a particular solver on a particular SAT instance (a regression task). As another example, one can predict whether a solver terminates within the cut-off time or not (a classification task). Besides different prediction scenarios, the characterization of SAT instances is also an important research direction. Prediction models can only achieve good performance if they have sufficient features to learn from. There are many different ways to characterize SAT instances by features. Thus, resulting predictions models can be difficult to understand. A practitioner might be interested in finding the most important instance features and explaining particular predictions.

The goal of this thesis is to compare prediction approaches as well as features selection and model explanation techniques, all applied to the use case of SAT solver performance prediction. The following questions are particularly interesting:

- Which prediction problems can be formulated for the given data and how well can they be solved?
- Which features can describe SAT instances, and how do they affect solver performance?
- How much can feature selection techniques help to get simpler prediction models?
- How much can explainability techniques help to interpret predictions?

The following steps are part of your thesis:

- Review literature about performance prediction for SAT solving.
- Design and implement a pipeline to compare prediction scenarios, different instance feature sets, feature selection techniques and/or explanation techniques in a systematic manner. We will jointly define the exact scope in our first meetings with you.
- Evaluate your prediction pipeline experimentally. You can use our server infrastructure for that.

During your work on this thesis, you will gain insights into an active field of SAT solving research. You will acquire knowledge about state-of-the-art machine learning libraries. You will gain experience in running and evaluating large-scale scientific experiments. The thesis is part of an ongoing research project, and you will get supervision from the SAT perspective (Markus Iser, chair of Prof. Sanders) as well as the ML perspective (Jakob Bach, chair of Prof. Böhm).

You can write the thesis in English or German. Prior experience with programming in Python is beneficial, but not necessary.

Contact
Jakob Bach jakob.bach@kit.edu +49 721 608-43911 Room: 363
Am Fasanengarten 5 76131 Karlsruhe Building: 50.34