ICON Challenge on Algorithm Selection

http://challenge.icon-fet.eu

Lars Kotthoff

January 27, 2015

1 Setting

The algorithm selection problem is to select the best algorithm for solving a given problem [6]. It is relevant where algorithm portfolios [3] are employed and instead of tackling a set of problem instances with just a single algorithm, a set of them is used with a subset, which may be of size 1, being selected for each instance.

A common approach to algorithm selection in practice is to characterise the problem instance to be solved through sets of features that can be extracted in a computationally efficient manner. These features, along with ground truth data of algorithm performance on some problem instances, are then used to induce performance models of the portfolio and its constituent algorithms. Usually, machine learning is used to induce such models. To solve a given new problem instance, the learned model makes a prediction as to the most suitable algorithm(s).

Many applications of the algorithm selection problem are in the area of NP-complete (or harder) problems, where there can be vast differences in performance for different algorithms on the same problem instance. Indeed, one of the most successful algorithm selection systems, SATzilla [7], applies algorithm selection techniques to SAT. The interested reader is referred to a recent survey on the area [5].

Almost all of the approaches presented and evaluated use different data, which often is not available. This makes direct comparisons almost impossible. The algorithm selection benchmark library ASlib[1] aims to mitigate this by collecting algorithm selection problems from different application areas from the literature. It enables easy comparison of different systems on the same data and the quantitative evaluation of different approaches.

Aslib release 1.0[2] comprises 13 algorithm selection scenarios with different numbers of algorithms, problem instances, and features. We leverage this diverse data set for the ICON challenge on algorithm selection.

[1] http://aslib.net
[2] https://github.com/coseal/aslib_data/releases

2 Challenge

Your mission, should you choose to accept it, is to build algorithm selection systems based on the ASlib data sets. Your systems must accept input in the ASlib data format to build the performance model(s) and to make predictions for new problem instances (determined by their features).

You will not be required to run any algorithms or compute any features yourself – all of this information is contained in the ASlib data. Your task is to make the best use of this information.

As the algorithm selection scenarios in ASlib use different features and have different algorithms, you will need to build different systems for different scenarios. You are free to submit systems for as few or as many scenarios as you like. Submissions of the same system for different scenarios will be bundled into a group. The scenarios considered here are the ones contained in ASlib release 1.0, any scenarios added later are not part of this challenge.

You are free to determine a single algorithm to run for the cutoff time specified in the description of the respective data set or provide a schedule for several algorithms that are to be run for a specified amount of time sequentially. The cutoff time is CPU time, not elapsed time, and there is no communication between algorithms. Therefore, running schedules in parallel provides no benefit.

Your systems must be runnable on Linux and may only use components that are either open source or free for academic use. You must submit the systems to us, along with the source and a brief description, and we will evaluate them. The description should specify how to compile (if appropriate) and run the systems (for both training and testing). The description should also, for each scenario that you submit to, list the feature groups that should be given to the system, and what presolver should be run for how long, if any. That is, you are allowed to specify a single solver from the portfolio that is to be run on each instance for an amount of time that you specify before feature computation starts. If the instance is solved during this step, no further action will be taken on it. You are allowed exactly one solver, presolving schedules of several solvers are not permitted.

2.1 Data formats

The data format for the training and testing data is the ASlib data format. The only difference is that for testing, your system will not be provided the information on algorithm performance.

The output of the training phase should be a runnable "system" (which may be written to a separate file, or provided as a set of parameters for executing an existing file) that will make predictions, given new data. The predictions made given test data should have the following format,

```
instanceID,runID,solver,timeLimit
```

where the instance ID designates the instance to solve, the run ID determines the order in which solvers are run on the instance (lowest first), the solver determines the solver to run on the instance, and the time limit is the maximum CPU time to run the solver for before switching to the next.

For example, an approach that chooses only a single solver to run would produce output like this for a time limit of one hour:

```
myProblem1.cnf,1,mySolver1,3600
myProblem2.cnf,1,mySolver1,3600
myProblem3.cnf,1,mySolver2,3600
```

Approaches that make schedules for solvers would produce output like this:

```
myProblem1.cnf,1,mySolver1,900
myProblem1.cnf,2,mySolver2,1800
myProblem1.cnf,3,mySolver3,900
myProblem2.cnf,1,mySolver1,10
myProblem2.cnf,2,mySolver2,3590
```

Systems that do not read and write these data formats properly may be disqualified.

3 Evaluation

Each submitted system will be trained on subsets of the data for the respective ASlib scenario. The limit for training is 12 CPU hours in total for everything. Any system that takes longer will be disqualified.

We will run the trained systems on the test data to get their predictions. We will simulate running the selected algorithm(s) according to the schedule you provide, using ground truth performance data. The simulated execution stops as soon as the solution to the problem instance is found or the cutoff time is reached, whichever comes sooner. The runtime up to this point is counted as the cost of solving the instance.

In addition of the runtime to solve the problem, you will be charged the cost of computing the features. Each system must specify the feature groups of the respective ASlib scenario it wishes to use as part of the submission process. It will be given the values of the features in those groups and the runtime to compute all of those features will be added to the runtime of the returned schedule.

If a problem instance is solved by the predicted schedule just within the cutoff time, but the cost of computing the necessary features is greater than the difference to the cutoff time, the instance will be counted as not solved and the cutoff time assumed as the runtime.

If a problem instance is solved during the computation of the feature values of a specific group and you requested this group to be computed, the runtime for the respective instance will be the cost of computing this group of features.

If the system returns no or a malformed prediction, the single best algorithm will be selected for the respective instance. If a system is disqualified on a scenario, the performance of the single best algorithm will be assumed.

Each system will be evaluated on the test data for the ASlib scenario it was submitted for and will be scored according to the following three criteria, based on the runtime computed as described above:

solved instances The fraction of instances solved within the cutoff time.

PAR10 The penalized average runtime (factor 10) across the instances of a scenario. That is, if an instance is solved within the specified cutoff, the actual runtime is taken. Otherwise, the cutoff value times 10 is assumed.

misclassification penalty The additional time spent solving the instances compared to the virtual best solver that picks always the best algorithm with no feature computation.

In addition to the other submissions, each system will be compared to the virtual best and the single best algorithms. The virtual and single best performances and indicative performances of basic algorithm selection systems for the *entire* scenario are listed at the respective benchmark result scenario pages on ASlib[3].

The final score of a submission group (i.e. a system with incarnations for different ASlib scenarios) is computed as the average score over all ASlib scenarios. For scenarios for which no system belonging to the group was submitted, the performance of the single best algorithm is assumed – we encourage you to submit for as many ASlib scenarios as you can.

4 Support

Contestants are encouraged to implement the ASlib data format, but are not required to do so in order to participate. The following options are available for participants wishing to leverage existing implementations.

claspfolio Claspfolio [2] is an algorithm selection system implemented in Python that can read the ASlib data format. It is available at `http://www.cs.uni-potsdam.de/claspfolio/`.

LLAMA LLAMA [4] is an algorithm selection toolkit implemented in R on top of mlr [1] that can read the ASlib format through the `algselbench` package[4]. It is available at `https://bitbucket.org/lkotthoff/llama`.

A discussion group for this challenge is available at `http://groups.google.co.uk/d/forum/icon-aslib` and `icon-aslib@googlegroups.com`. If you intend to participate in the challenge, you are strongly encouraged to subscribe, as clarifications and announcements will be posted there.

Please use the discussion group for all questions and do not email the organisers privately.

5 Dates

end of January 2015 (AAAI) Challenge announced

10 February 2015 Submission opens

10 July 2015 Submission closes

end of July 2015 (IJCAI) Results announced

[3]see e.g. `http://berndbischl.github.io/coseal-algsel-benchmark-repo/task-pages/QBF-2011/llama.html`

[4]`https://github.com/berndbischl/coseal-algsel-benchmark-repo`

6 Submission

Submission will be available on the challenge website at `http://challenge.icon-fet.eu`.

References

[1] Bernd Bischl, Michel Lang, Jakob Richter, Jakob Bossek, Leonard Judt, Tobias Kuehn, Erich Studerus, and Lars Kotthoff. *mlr: Machine Learning in R.* R package version 2.3.

[2] Holger Hoos, Marius Lindauer, and Torsten Schaub. claspfolio 2: Advances in algorithm selection for answer set programming. *TPLP*, 14(4-5):569–585, 2014.

[3] Bernardo A. Huberman, Rajan M. Lukose, and Tad Hogg. An economics approach to hard computational problems. *Science*, 275(5296):51–54, 1997.

[4] Lars Kotthoff. LLAMA: Leveraging learning to automatically manage algorithms. Technical Report arXiv:1306.1031, June 2013.

[5] Lars Kotthoff. Algorithm selection for combinatorial search problems: A survey. *AI Magazine*, 35(3):48–60, 2014.

[6] John R. Rice. The algorithm selection problem. *Advances in Computers*, 15:65–118, 1976.

[7] Lin Xu, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. SATzilla: Portfolio-based algorithm selection for SAT. *J. Artif. Intell. Res. (JAIR)*, 32:565–606, 2008.