SUNNY with Algorithm Configuration

Tong Liu T.LIU@UNIBO.IT
Roberto Amadini ROBERTO.AMADINI@UNIMELB.EDU.AU
Jacopo Mauro MAURO.JACOPO@GMAIL.COM

1. Description

SUNNY-OASC is an per instance algorithm scheduling strategy based on K-NN technique. SUNNY-OASC is an extension of the SUNNY-AS tool (see Amadini et al. (2015); Amadini and Mauro (2015)) with some ideas borrowed from the work of Lindauer et al. (2016).

1.1. Execution modalities

SUNNY-OASC has two execution modalities: `autok` and `fkvar`.

- The `autok` is a variant of T-SUNNY as defined by Lindauer et al. (2016) where SUNNY-AS has been improved training also on the size of the value of the neighborhood K.[1]
- The `fkvar` instead trains for the neighborhood value K and the subset of features to consider by using a wrapper method Kohavi and John (1997). SUNNY is used as the evaluator and a greedy forward selection is adopted to select the subset of features for computing the neighborhood. The selection cycle is defined as follow: the unselected feature set is considered and we pick one feature at the time adding it to the selected features set (initially empty) to form a test feature set. By tuning also the value k, SUNNY calculates the best par10 score that it can achieve with the test feature set. Based on the outcome, a new feature is added until the performance decrease or we have performed a given number of evaluations. In the end, `fkvar` produces a combination of features and a value K for which SUNNY performs the best on training data.

1.2. Representative instances

For performance reasons, SUNNY-OASC is not used on all the instances available but only on some selected ones. The representative instances use for the training are selected as follow: i) SUNNY-OASC first associates each instance to a solver that solves it in the least time, ii) for each solver instances are ordered from hard to easy in terms of runtime, ii) for each solvers one instance at the time is picked until a global limit on the number of representative instances is reached.

1. `autok` is slightly different than T-SUNNY since the reimplementation of T-SUNNY used a different algorithm to select the solvers to use. To chose the solvers we used the original SUNNY-AS algorithm.

1.3. Parameters for the Challenge

The experiments performed by Amadini et al. (2015) suggest that a handful subset of features (e.g., 5 or less) is often enough for SUNNY to obtain a competitive performance. For this reason, in `fkvar` we fixed 5 as the amount of feature to select. In order to guarantee an acceptable execution runtime, for the `fkvar` approach, we have chosen to consider only 1500 and not more instances to be included in the representative instance set. We also fixed the interval of K as [3,30].

When `fkvar` is executed, we also run `autok` with $K \in [3, 80]$ as a backup. If SUNNY runs better with the entire feature set, we then use the solution produced by `autok`.

For the `autok` version submitted, differently to the one used as a backup when running SUNNY-OASC in the `fkvar` modality, we consider the full training set as effective training data (i.e., more than 1500 instances are used to train if available).

2. Setup Instruction

The source code of SUNNY-OACS is available at Liu (2017) and requires Python v2. There are five folders: 'data' and 'results' contain oasc-challenge data and solution results respectively, 'src' contains the original SUNNY-AS scripts from Amadini and Mauro (2015), 'oasc' contains scripts that coordinate those in 'src' for training and testing, the folder 'main' contains the scripts that automatically call 'oasc' for the different execution modalities.

The program runs training and testing in sequence. Let us take `autok` approach as execution example. To run it, in the folder 'main' the command "sh make_oasc_tasks.sh < tasks.txt" must be used to create the tasks. Then the train cab be done by running "sh oasc_train.sh run_autok tasks.txt". After training, the test is run by "sh make_oasc_tasks.sh < tasks.txt" and later by "sh oasc_test.sh autok tasks.txt".

To run `fkvar` it is sufficient to replace literally `autok` by `fkvar` in the previous commands.

References

Roberto Amadini and Jacopo Mauro. SUNNY-AS, 2015. Available at https://github.com/CP-Unibo/sunny-as.

Roberto Amadini, Fabio Biselli, Maurizio Gabbrielli, Tong Liu, and Jacopo Mauro. Feature selection for SUNNY: A study on the algorithm selection library. In *ICTAI*, pages 25–32. IEEE Computer Society, 2015.

Ron Kohavi and George H. John. Wrappers for feature subset selection. *ARTIFICIAL INTELLIGENCE*, 97(1):273–324, 1997.

Marius Lindauer, Rolf-David Bergdoll, and Frank Hutter. An Empirical Study of Per-instance Algorithm Scheduling. In *LION*, volume 10079 of *LNCS*, pages 253–259. Springer, 2016.

Tong Liu. SUNNY-OASC, 2017. Available at https://github.com/lteu/oasc.