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

Our proposed solution is an extension of SUNNY-AS DBL (2015); Amadini (2015) with the ideas borrowed from the Works Lindauer et al. (2016); Kohavi and John (1997).

SUNNY-AS is an per instance algorithm scheduling strategy based on K-NN techniques. The Work Lindauer et al. (2016) has demonstrated that how a training phase, by studying the value K and the number of solvers, can boost SUNNY performance. However, we believe that the number of solvers is a trivial option to configure, because, in our original version, SUNNY employs less solvers as reported.

1.1. Approaches

In this Work, we have proposed two solutions, 'autok' and 'fkvar'.

- The 'autok' is a variant of TSUNNY mentioned by Lindauer et al. (2016), where, we consider only the value K (ignoring the number of solvers).
- The 'fkvar' instead trains for both value K and features by using a wrapper method Kohavi and John (1997), that is, we take SUNNY as the evaluator and we then perform a greedy forward selection, which repeats a selection cycle defined as follow: in each cycle, we loop on unselected feature set, we pick one feature at time and combine it with selected features (initially empty) as 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. In the end of each cycle, it incorporates a new feature to the selected feature set. We stop further selection cycle when adding the new test feature alters SUNNY performance or the number of feature limit (cycle) is reached. At last, 'fkvar' produces a combination of features and value K which SUNNY performs the best on training data.

1.2. Representative instances

The representative instances are selected as follow, it first classifies each instance to a solver who solve it in least time, later, for each solver (class), it orders the instances from hard to easy in terms of runtime, then for each class, it picks one instance at time in order to reach the limit of instance set.

1.3. Parameters

Our previous experiments DBL (2015) suggested that a handful subset of features (eg: 5) is often enough for SUNNY to obtain a competitive performance, as such, in 'fkvar' we fixed such amount of feature to select. In order to guarantee an acceptable execution runtime, for the 'fkvar' approach, we have taken up to 1500 representative instances from training set as effective instances, and we also fixed the interval of K as [3,30] [1]. In addition, in the submitted 'fkvar' version, when evaluating feature relevance, the par10 is calculated by averaging par10 of three best k instead of a single best k. In the end of execution, we re-run 'autok' $K \in [3, 80]$ for a backup, i.e. if SUNNY runs better with the entire features, we then go for the approach of 'autok'. Differently, in the submitted 'autok' version, we consider the full training set as effective training data instead of 1500 instances.

2. Setup Instruction

The source code is available at Liu (2017) which requires Python v2.x. There are five folders, 'data' and 'results' contain oasc-challenge data and solution results respectively. 'src' contains the original SUNNY-AS scripts from Amadini (2015), 'oasc' contains scripts who coordinate those in 'src' for training and testing. In the end, in the folder 'main', there have been placed the scripts that automatically call scripts in 'oasc' for different scenarios.

The program runs training and testing in sequence, let us take 'autok' approach as execution example. In the folder 'main', launch the command "sh make_oasc_tasks.sh < tasks.txt" to create tasks. Then train scenarios with "sh oasc_train.sh run_autok tasks.txt". After training, run testing with command "sh make_oasc_tasks.sh < tasks.txt" then, "sh oasc_test.sh autok tasks.txt". Whereas, to run fkvar approach, it is sufficient to replace literally 'autok' by 'fkvar' in the previous commands.

References

27th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2015, Vietri sul Mare, Italy, November 9-11, 2015, 2015. IEEE Computer Society. ISBN 978-1-5090-0163-7. URL http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7372093.

Amadini. Sunny as, 2015. Available at https://github.com/CP-Unibo/sunny-as.

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 10, Ischia, Italy, May 29 - June 1, 2016*, 2016. doi: 10.1007/978-3-319-50349-3_20. URL https://doi.org/10.1007/978-3-319-50349-3_20.

Liu. Sunny-oasc, 2017. Available at https://github.com/lteu/oasc.

1. Several scenarios with around 5.000 instances, it may take couple of days for training. Besides, the interval [3,30] would cover most of the useful K values.