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

This proposed solution is an improvement of SUNNY-AS Amadini et al. (2015); Amadini (2015) with the ideas suggested by 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 step, by studying the value K and the number of solvers, can improve SUNNY's performance significantly. In this Work, we have proposed two solutions, 'autok' and 'fkvar'. The 'autok' re-implemented TSUNNY mentioned in Lindauer et al. (2016) by considering only the value K (ignoring the number of solvers). The 'fkvar' instead trains for both value K and optimal features by using a wrapper method Kohavi and John (1997), which selects the best combination of K and features, such that, SUNNY enhances the most on training data.

Our previous experiments Amadini et al. (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]. In the end of execution, we re-run 'autok' $K \in [3, 80]$ for a backup, i.e. if SUNNY runs better with entire features, we then adopt the combination of K and the whole feature set instead. Differently, in the 'autok' version, we consider the full training set as effective training data.

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

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

Roberto Amadini, Fabio Biselli, Maurizio Gabbrielli, Tong Liu, and Jacopo Mauro. SUNNY for algorithm selection: a preliminary study. In *Proceedings of the 30th Italian Conference on Computational Logic, Genova, Italy, July 1-3, 2015.*, pages 202–206, 2015. URL http://ceur-ws.org/Vol-1459/paper3.pdf.

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 *Learning and Intelligent Optimization - 10th International Conference, LION 10, Ischia, Italy, May 29 - June 1, 2016, Revised Selected Papers*, pages 253–259, 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.