Meta-algorithms

(Per-instance) Automated Algorithm Selection (AAS)
For a given problem instance, which algorithm from a portfolio of algorithms is most able to solve it best? [Rice,’76]

Automated Algorithm Configuration (AAC)
What are the best hyperparameters of an algorithm for a given problem?

Get the best performance out of algorithms
More accurately represent the state of the art in solving challenging problems in AI.
Adoption of meta-algorithms

- Adoption is limited, even in ML research [Bouthillier & Varoquaux, 2020]
- Meta-algorithms are complex and difficult for non-experts
- Substantial pitfalls, e.g., in AAC [Eggensperger et al., 2019]
- Errors are costly, e.g., re-running AAC is computationally expensive
Goals of Sparkle

- Simplify the use of meta-algorithms
- Increase the adoption of meta-algorithms
- Prevent common pitfalls and often-made errors
- Ensure proper experimentation pipelines

- Improve our ability to assess, access and improve the SOTA in computational problem solving
Sparkle

- Algorithms
- Instances
- Feature extractors

Sparkle
AAS
AAC

\LaTeX\ report

Set-up
Results
Analysis
Simple Command Line Interface

1: Commands/initialise.py
2: Commands/add_instances.py path/to/PTN/
3: Commands/add_solver.py --deterministic 0 path/to/PbO-CSCCSAT/
4: Commands/add_solver.py --deterministic 0 path/to/CSCCSat/
5: Commands/add_solver.py --deterministic 0 path/to/MiniSAT/
6: Commands/add_feature_extractor.py path/to/Extractor/
7: Commands/compute_features.py
8: Commands/construct_sparkle_portfolio_selector.py
9: Commands/generate_report.py
5 Comparison between Configured Version and Default Version on the Training Instance Set

In order to investigate the performance on the training instance set, Sparkle would run the configured version of PbO-CCSAT-Generic and the default version of PbO-CCSAT-Generic on the training instance set. During this phase, each version was run per instance with a cutoff time of 120 seconds. The results are reported as follows:

- **PbO-CCSAT-Generic (configured)**, PAR10: 3.2082931200663247
- **PbO-CCSAT-Generic (default)**, PAR10: 421.2084581601

The empirical comparison between the PbO-CCSAT-Generic (configured) and PbO-CCSAT-Generic (default) on the training set of PTN is presented in Figure 1.

![Graph showing comparison between configured and default versions](image-url)

Table 2: Ablation paths from PbO-CCSAT-Generic (default) to PbO-CCSAT-Generic (configured) where parameters with higher importance are ranked higher.

<table>
<thead>
<tr>
<th>Round</th>
<th>Flipped parameter</th>
<th>Source value</th>
<th>Target value</th>
<th>Validation result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>gamma_hscore2</td>
<td>1000</td>
<td>351</td>
<td>116.32313</td>
</tr>
<tr>
<td>2</td>
<td>prob_pac</td>
<td>0</td>
<td>1</td>
<td>0.005730374136488115</td>
</tr>
<tr>
<td>3</td>
<td>gamma MainForm</td>
<td>0.0</td>
<td>0.6807207179674418</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>gamma MainForm</td>
<td>0.2</td>
<td>0.6807207179674418</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>prob_pac</td>
<td>0</td>
<td>1</td>
<td>0.005730374136488115</td>
</tr>
<tr>
<td>6</td>
<td>gamma_hscore2</td>
<td>1000</td>
<td>351</td>
<td>116.32313</td>
</tr>
<tr>
<td>7</td>
<td>prob_pac</td>
<td>0</td>
<td>1</td>
<td>0.005730374136488115</td>
</tr>
</tbody>
</table>

Table 3: Number of time-outs for PbO-CCSAT-Generic (configured), PbO-CCSAT-Generic (default), and for how many instances both timed out on the training set of PTN.

<table>
<thead>
<tr>
<th>Run</th>
<th>PbO-CCSAT-Generic (configured)</th>
<th>PbO-CCSAT-Generic (default)</th>
<th>Both timed out</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

References

Conclusion

Sparkle makes meta-algorithms accessible for improving the state of the art in solving challenging problems in AI.

Try out Sparkle yourself!
bitbucket.org/sparkle-ai/sparkle

Contact
Koen van der Blom
Jeroen Rook