AutoML: Hyperparameter Optimization

Practical Problems

Bernd Bischl   Frank Hutter   Lars Kotthoff
Marius Lindauer

Practical Problems: Stability

AutoML system should:

- never fail to return a result
- terminate within a given time
- save intermediate results and allow to continue

Practical Problems: Stability

AutoML system should:

- never fail to return a result
- terminate within a given time
- save intermediate results and allow to continue

Failure points:

- optimizer can crash (e.g. GP in BO)
- training can crash (e.g. segfault)
- training of a single model can run "forever"

Practical Problems: Stability

AutoML system should:

- never fail to return a result
- terminate within a given time
- save intermediate results and allow to continue

Failure points:

- optimizer can crash (e.g. GP in BO)
- training can crash (e.g. segfault)
- training of a single model can run "forever"

Ways out:

- Encapsulate train/predict in separate process from HPO
- Ressource limit time and memory of that process by OS
- If learner crashes, run robust fallback (constant predictor)
- Use "robust" HPO, run random config as last resort if proposal fails (exploration)

Practical Problems: Parallelization

Parallelization should allow:

- multiple cores
- multiple nodes

Possible parallelization levels:

- training of learner (threading / GPU)
- resampling
- evaluations of configurations (batch proposal of HPO)

Possible problems:

- Sequential nature of HPO algorithms (e.g. BO)
- Heterogeneous runtimes cause idling $\longrightarrow$ asynchronous HPO attractive, but more complex
- Main memory or CPU-cache becomes bottleneck
- Communication between workers