AutoML - Benefits, Reality, Future

Koen van der Blom

PGMO Days 2022

2022-11-30





Machine learning

- Widespread
- Automates human work
- Supports human experts

AutoML

Automatically construct (parts of) the machine learning pipeline

Includes, e.g.

- Data processing + cleaning
- Feature construction + selection
- Model selection
- Hyperparameter optimisation

What do we mean when we say AutoML?

Most widespread AutoML tools use meta-algorithms

- Automated algorithm configuration (including HPO + NAS)
- Automated algorithm selection

Many other ideas, but so far quite rare in practice

• Automated data processing, cleaning, etc.

Automated algorithm configuration

Improve performance by finding the best settings/parameters

- Systematic search over the parameter space
- Tries out unconventional parameter combinations
- Usually finds better performing algorithms [Fawcett et al. 2011, KhudaBukhsh et al. 2016, Rook et al. 2022]
- Runs while you do other things

Automated algorithm selection

Improve performance by choosing the best algorithm for the job

- Predict which algorithm to use for each problem instance
- Algorithm selection usually outperforms the single best solver

Benefits of AutoML

Performance improvement over hand designed systems

Democratise ML

- Reduce workload for ML experts
- Reduce required ML expertise

Dream: Create an ML system with one press on the button

Great benefits, great adoption?

Study AutoML adoption in software engineering for ML

• Systems where ML is used in real applications

Questionnaire [Serban et al. 2020 + 2021]

• Measure adoption levels of various AutoML techniques

Interviews [KvdB et al. 2021]

• Insights into reasons for adoption and benefits in practice

Joint work with Alex Serban, Joost Visser & Holger Hoos

AutoML adoption is not as high as expected!

- 20-30% do not adopt AutoML at all
- Another 50-60% do not completely adopt AutoML



Why is AutoML adoption lower than expected?

Points mentioned by two interviewees

- High initial cost to adoption (missing expertise)
- Difficult to predict good run length for AutoML
- Unclear what is wrong when AutoML systems fail
- Limited availability of computational resources

Literature also suggests usability, interpretability, interactivity

What to do?

Improved AutoML systems are needed

Start with the basics

- AutoML tools for computer scientists without AutoML expertise
- Meta algorithms
 - Automated algorithm configuration
 - Automated algorithm selection

Sparkle: Accessible meta-algorithms

Lower the bar to use algorithm selection and configuration

- Ease of use: Automate where possible
- Correctness: Implement best practices, avoid pitfalls
- Explanation: Clear commands, detailed (but concise) reporting

Sparkle platform: https://bitbucket.org/sparkle-ai/sparkle/ [KvdB et al. 2022]

Joint work with Holger Hoos, Chuan Luo, Jeroen Rook

Algorithm selector construction



AutoML - Benefits, Reality, Future – Koen van der Blom

Algorithm configuration

Configuration procedure Prepare algorithm, wrappers, Result & execution parameter space, instances, interpretation & analysis configurator, . . . 4. Problem instances 1. Target algorithm a. Assemble homogeneous a. Provide executable set of instances b. Split into train and test 2. Solver wrapper 7. Configuration protocol c. Create file with instance paths 8. Interpret and write up results a. Handle configurator call Sparkle generates these for provided a. Set configuration budget (per run) a. Interpret raw output format for: Instance. directories containing instance sets b. Set number of configuration runs Sparkle extracts the most important parameter settings. Sparkle defaults to multiple runs information from the output files and random seed presents it in a LaTeX/PDF report 5. Configurator c. Create configuration scenario file b. Output (compute) Sparkle generates this based on input b. Compare optimised configuration cost metric in a. Choose which configurator to use with default Validate on training set for each run configurator format b. Install configurator Sparkle includes a visual comparison Sparkle ensures validation is performed Sparkle installation already includes in its report, in addition to the e. Choose best over all runs installation of the configurator performance values 3. Configuration space Sparkle ensures the best configuration c. Write up results (after validation on train) is selected a. Set parameter values/ranges Sparkle describes the experimental 6 Scenario b. Set parameter default values f. Validate on the testing set procedure and provides references Sparkle ensures only the final configua. Set performance measure c. Indicate conditional and invalid for used tools in its report ration is compared on the testing set parameter combinations b. Set target algorithm cutoff time Analyse results Prepare configuration scenario Run configuration

AutoML - Benefits, Reality, Future – Koen van der Blom

Command example

- 1: Commands/initialise.py
- 2: Commands/add_instances.py Resources/PTN/
- 3: Commands/add_instances.py Resources/PTN2/
- 4: Commands/add_solver.py --deterministic 0 Resources/Pb0-CSCCSAT/
- 5: Commands/configure_solver.py --solver PbO-CSCCSAT
- --instance-set-train PTN
- 6: Commands/sparkle wait.py
- 7: Commands/validate_configured_vs_default.py
- --solver Solvers/PbO-CCSAT-Generic/ --instance-set-train Instances/PTN/
- --instance-set-test Instances/PTN2/
- 8: Commands/generate_report.py

Reporting

Often very basic in meta-algorithm tools (even just performance)

Detailed but concise report in Sparkle

- Used instance sets, target algorithm, configurator/selector
- Experiment description (protocol, budgets, . . .)
- Performance values + plots

Future of AutoML

No 'one button' magic system

Instead AutoML tools that support ML experts (eventually laymen)

- Correctness
- Understandability
- Interpretability
- Interaction

A lot of work needed beyond Sparkle!

Take away

- AutoML has the potential to democratise ML
- Only 20-30% adopt AutoML, another 50-60% only partially
- Adoption held back by missing expertise, usability, etc.
- Sparkle: Make meta-algorithms (core AutoML tools) accessible
- Future: Need tools for other audiences + AutoML components

Sparkle platform: https://bitbucket.org/sparkle-ai/sparkle/

koen.vdblom@lip6.fr

References

[Fawcett et al. 2011] Chris Fawcett, Malte Helmert, Holger Hoos, Erez Karpas, Gabriele Röger, and Jendrik Seipp, FD-Autotune: Domain-specific configuration using fast downward, PAL '11, 2011, pp. 13–20.

[KhudaBukhsh et al. 2016] Ashiqur R. KhudaBukhsh, Lin Xu, Holger H. Hoos, and Kevin Leyton-Brown, SATenstein: Automatically building local search SAT solvers from components, Artificial Intelligence 232 (2016), 20–42.

[Rook et al. 2022] Jeroen Rook, Heike Trautmann, Jakob Bossek, and Christian Grimme, On the potential of automated algorithm configuration on multi-modal multi-objective optimization problems, GECCO '22 Companion, 2022, p. 356–359.

[Serban et al. 2020] Alex Serban, Koen van der Blom, Holger Hoos & Joost Visser, Adoption and Effects of Software Engineering Best Practices in Machine Learning, Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM) 2020.

[Serban et al. 2021] Alex Serban, Koen van der Blom, Holger Hoos & Joost Visser, Practices for Engineering Trustworthy Machine Learning Applications. In: 2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN), 2021, 97-100.

[KvdB et al. 2021] Koen van der Blom, Alex Serban, Holger Hoos & Joost Visser, AutoML Adoption in ML Software, 8th ICML Workshop on Automated Machine Learning, 2021.

[KvdB et al. 2022] Koen van der Blom, Holger H. Hoos, Chuan Luo & Jeroen G. Rook, Sparkle: Towards Accessible Meta-Algorithmics for Improving the State of the Art in Solving Challenging Problems, IEEE Transactions on Evolutionary Computation 2022.

AutoML - Benefits, Reality, Future – Koen van der Blom