mlr3automl - Automated Machine Learning in R

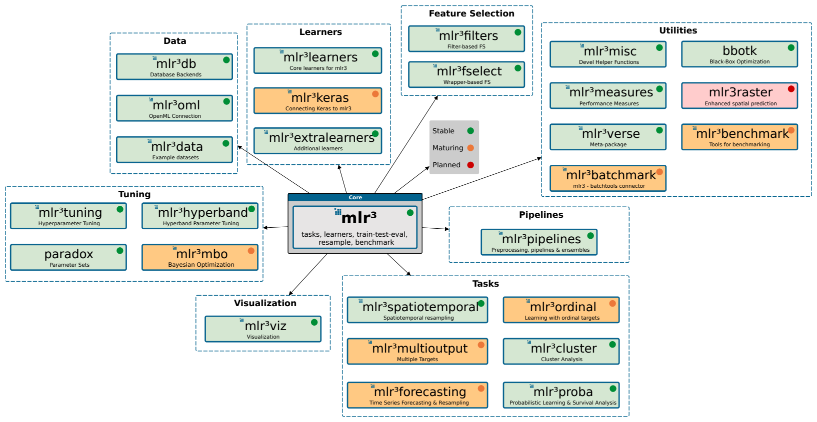
Martin Binder, Bernd Bischl, Alexander Hanf

2021-07-06

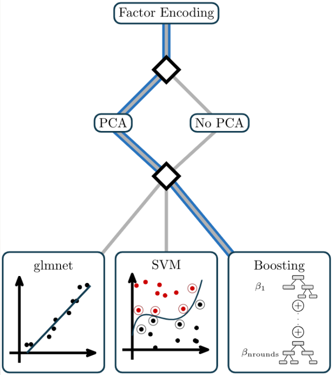
https://github.com/a-hanf/mlr3automl

mlr3 [5] - Machine Learning in R

- powerful, object-oriented and extensible framework for ML
- rich package ecosystem providing general-purpose ML tools



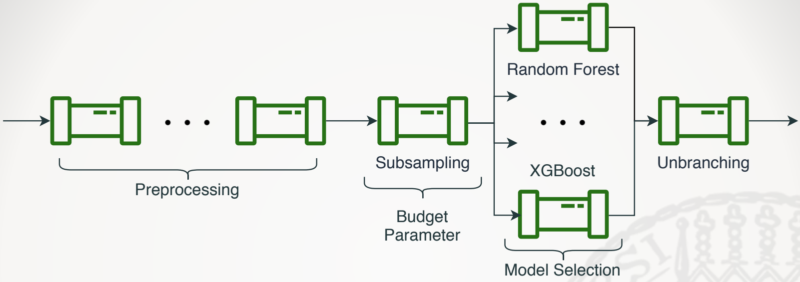
- Automating machine learning workflows (preprocessing, model selection, hyperparameter tuning)
- Provides baseline models with little effort or expertise
- Many approaches, e.g. Combined Algorithm Selection and Hyperparameter Optimisation [9]



Search Space $\tilde{\Lambda}$

Name	Type	Bounds/Values	Trafo
`encoding`	C	one-hot, impact	
◇ `pca`	C	PCA, no PCA	
◇ `learner`	C	glmnet, SVM, Boosting	
`if learner = glmnet`			
`s`	R	$[-12, 12]$	2^x
`alpha`	R	$[0, 1]$	–
`if learner = SVM`			
`cost`	R	$[-12, 12]$	2^x
`gamma`	R	$[-12, 12]$	2^x
`if learner = Boosting`			
`eta`	R	$[-4, 0]$	10^x
`nrounds`	I	$\{1, \ldots, 5000\}$	–
`max_depth`	I	$\{1, \ldots, 20\}$	–

mlr3automl - AutoML in mlr3



AutoML package for regression and classification based on mlr3

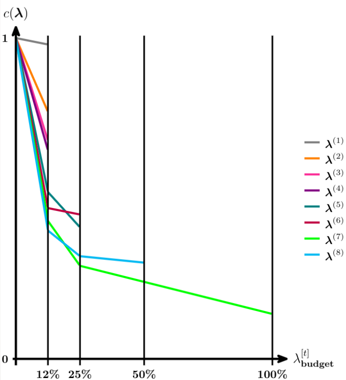
- Automatic preprocessing using mlr3pipelines [1],
- Few, but well-tested learning algorithms,
- Joint optimisation of pipeline and model hyperparameters with Hyperband,
- Static portfolio of known good pipelines

Tuning in mlr3automl

First step: evaluate 8 fixed hyperparameter configurations

Second step: continue tuning with Hyperband [7]:

- Multi-fidelity approach to speed up random search
- Budget parameter: subsampling rate $\in [10\%, 100\%]$
- mlr3hyperband provides implementation for any tuning scenario in mlr3



bracket 3

t	$\lambda_{\text{budget}}^{[t]}$	$p_3^{[t]}$
0	1	8
1	2	4
2	4	2
3	8	1

bracket 2

t	$\lambda_{\text{budget}}^{[t]}$	$p_2^{[t]}$
0	2	6
1	4	3
2	8	1

bracket 1

t	$\lambda_{\text{budget}}^{[t]}$	$p_1^{[t]}$
0	4	4
1	8	2

bracket 0

t	$\lambda_{\text{budget}}^{[t]}$	$p_0^{[t]}$
0	8	4

Basic usage

```
automl_model = AutoML(task = train_tsk)
automl_model$train()
predictions = automl_model$predict(predict_tsk)
```

Only required argument: regression or classification Task

AutoML() customisation options:

- runtime: Time budget
- measure: Performance measure to optimise for
- learner_list: Learners to choose from
- preprocessing: Type of preprocessing
- additional_params: Additional parameters for tuning

Example 1 - custom learner & runtime budget

```
automl_model = AutoML(
  task=tsk("mtcars"),
  learner_list=c("regr.ranger", "regr.lm"),
  learner_timeout=10,
  runtime=300)

automl_model$train()
```

- default learners:
  - Random Forest (ranger)
  - Gradient Boosting (xgboost)
  - Logistic / Support Vector Regression (LiblineaR)
- accepts any learner from mlr3 or extension packages
- timeouts for individual learners and overall runtime
- learners are encapsulated in separate R sessions: failing learners do not tear down main session

Example 2 - custom parameters & transformation

```
new_params = ParamSet$new(list(
   ParamInt$new("classif.kknn.k",
   lower = 1, upper = 5, default = 3, tags = "kknn")))

my_trafo = function(x, param_set) {
   if ("classif.kknn.k" %in% names(x)) {
      x[["classif.kknn.k"]] = 2^x[["classif.kknn.k"]]
   }
   return(x)
}

automl_model = AutoML(task=tsk("iris"),
  learner_list="classif.kknn",
  additional_params=new_params,
  custom_trafo=my_trafo)
```

- predefined parameter spaces for integrated learners
- support for custom parameter spaces via paradox package
- parameters can be transformed with user-defined functions

Example 3 - custom preprocessing

```
library(mlr3pipelines)
imbalanced_preproc = po("imputemean") %>>%
  po("smote") %>>%
  po("classweights", minor_weight=2)

automl_model = AutoML(task=tsk("pima"),
  preprocessing = imbalanced_preproc)
```

- three pre-defined preprocessing settings
  - "none" - no preprocessing
  - "stability" - for missing data, numerical and high-cardinality features
  - "full" - tunable imputation, factor encoding and PCA
- alternative: custom Graph object from mlr3pipelines

Evaluation

We evaluated mlr3automl in the AutoML Benchmark [4]:

- 39 challenging classification tasks
- Time budget: 10 minutes for small tasks, 1 hour otherwise[1]
- Competitors: AutoGluon-Tabular[2], auto-sklearn[3], H2O AutoML[6], TPOT[8]

Comparison to winning framework (AutoGluon-Tabular):

- binary classification: 1.1% worse in mean AUC
- multi-class: 2.8% worse in mean ACC
- only other library to finish all tasks without failures

[1] A more extensive benchmark by the OpenML team is currently under way

Acknowledgements

Thanks to everyone in the open source community!

Give it a try: github.com/a-hanf/mlr3automl

Keep in touch: linkedin.com/in/alexander-hanf

References

[1] Martin Binder et al. "mlr3pipelines – Flexible Machine Learning Pipelines in R". In: JMLR MLOSS (accepted, not yet published) (June 2021).

[2] Nick Erickson et al. "AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data". In: arXiv preprint arXiv:2003.06505 (2020).

[3] Matthias Feurer et al. "Efficient and robust automated machine learning". In: Advances in neural information processing systems. 2015, pp. 2962–2970.

[4] P. Gijsbers et al. "An Open Source AutoML Benchmark". In: arXiv preprint arXiv:1907.00909 [cs.LG] (2019). Accepted at AutoML Workshop at ICML 2019. URL: https://arxiv.org/abs/1907.00909.

[5] Michel Lang et al. "mlr3: A modern object-oriented machine learning framework in R". In: Journal of Open Source Software 4.44 (2019), p. 1903.

[6] Erin LeDell and Sebastien Poirier. "H2O AutoML: Scalable Automatic Machine Learning". In: 7th ICML Workshop on Automated Machine Learning (AutoML) (July 2020). URL: https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf.

[7] Lisha Li et al. "Hyperband: A novel bandit-based approach to hyperparameter optimization". In: The Journal of Machine Learning Research 18.1 (2017), pp. 6765–6816.

[8] Randal S. Olson et al. "Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science". In: Proceedings of the Genetic and Evolutionary Computation Conference 2016. GECCO '16. Denver, Colorado, USA: ACM, 2016, pp. 485–492. ISBN: 978-1-4503-4206-3. DOI: 10.1145/2908812.2908918. URL: http://doi.acm.org/10.1145/2908812.2908918.

[9] Chris Thornton et al. "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms". In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. 2013, pp. 847–855.