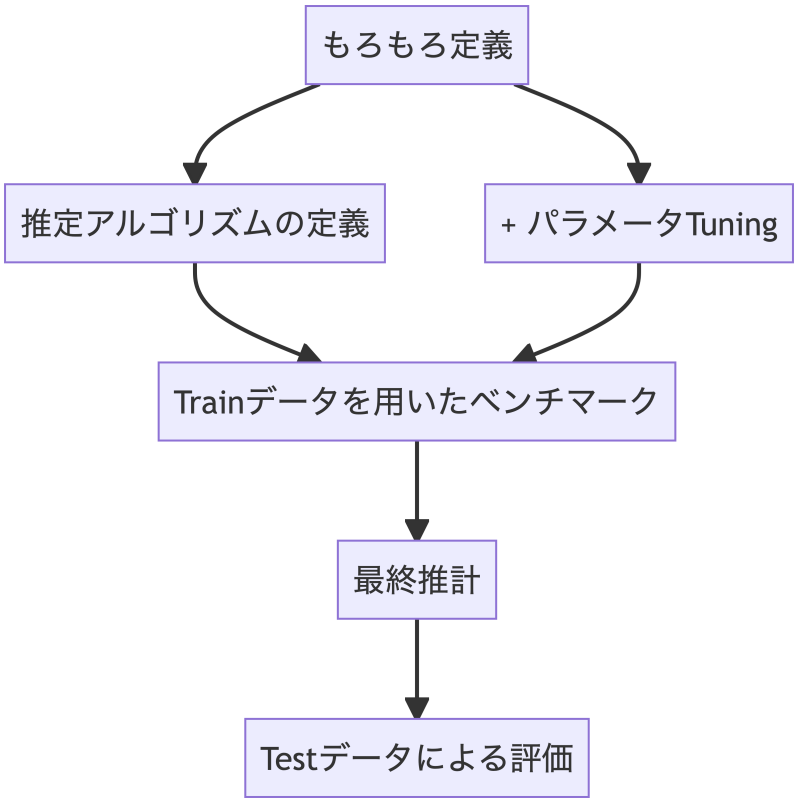
A bit Advanced Prediction Task with mlr3

川田恵介

Purpose

- 予測問題についての包括的な作業工程を例示
  - ただし Stacking は除く
- TuningParameter の推定と本推定を同時に行う AutoTuner を活用
- モデル比較を簡便かつ柔軟に行える benchmark を活用

RoadMap



SetUp

```
library(mlr3verse)
library(tidyverse)

lgr::get_logger("mlr3")$set_threshold("error") # Error のみを表示
lgr::get_logger("bbotk")$set_threshold("error") # Error のみを表示
future::plan("multisession") # 並列処理

set.seed(1)

Data <- read_csv("ExampleData/Example.csv")

Task <- as_task_regr(Data,
                     "Price") # Define Task

Subgroup <- partition(Task, ratio = 0.8)

R2 <- msr("regr.rsq") # Define Evaluation with R2

Mean <- lrn("regr.featureless") # Define SimpleMean
OLS <- lrn("regr.lm") # Define OLS
RandomForest <- lrn("regr.ranger") # Define Random Forest
```

Tuning

- TuningParameter の推定を行うアルゴリズムを定義

```
CV <- rsmp("cv",folds = 2) # Define CrossValidation with 2 folds

Tuner <- tnr("random_search") # Define search method

Terminator <- trm("evals", n_evals = 100) # Define Terminal condition
```

Pruned Tree

- Prune Tree の推定
- lts 関数 (mlr3tuningpsace パッケージ) が提供するおすすめ範囲内で探索 (Bischl et al., n.d.)

```
Tree <- lrn("regr.rpart") |> lts() # Define AdaptiveTree

Tree <- AutoTuner$new(
  learner = Tree,
  resampling = CV,
  measure = R2,
  tuner = Tuner,
  terminator = Terminator,
  store_models = TRUE
)

Tree$id <- "Tree"
```

ElasticNet

- ElasticNet を推定
- lts 関数 (mlr3tuningpsace パッケージ) が提供するおすすめ範囲内で探索 (Bischl et al., n.d.)

```
LASSO <- lrn("regr.glmnet") |> lts() # Define LASSO

LASSO <- AutoTuner$new(
  learner = LASSO,
  resampling = CV,
  measure = R2,
  tuner = Tuner,
  terminator = Terminator,
  store_models = TRUE
)


LASSO$id <- "LASSO"
```

BenchMaking

- 訓練データのみを用いて、アルゴリズムを比較 (TrainingData 内の CrossValidation)

```
Design <- benchmark_grid(
  tasks = Task$clone()$filter(Subgroup$train),
```

```
  learners = list(OLS,Tree,RandomForest,Tree,Mean,LASSO),
  resamplings = CV
)

Result <- benchmark(Design)

Result$aggregate(R2)
```

```
   nr      resample_result task_id       learner_id resampling_id iters
1:  1 <ResampleResult[21]>    Data          regr.lm            cv     2
2:  2 <ResampleResult[21]>    Data             Tree            cv     2
3:  3 <ResampleResult[21]>    Data      regr.ranger            cv     2
4:  4 <ResampleResult[21]>    Data             Tree            cv     2
5:  5 <ResampleResult[21]>    Data regr.featureless            cv     2
6:  6 <ResampleResult[21]>    Data            LASSO            cv     2
        regr.rsq
1:  4.193565e-01
2:  4.281312e-01
3:  4.820819e-01
4:  4.365090e-01
5: -4.377844e-05
6:  4.193529e-01
```

Final Model

- 最善のアルゴリムであった RandomForest、及び全訓練データを用いて最終予測モデルを推定

```
RandomForest$train(Task,Subgroup$train)

RandomForest$predict(Task,Subgroup$test)$score(R2) # Peformance in TestData
```

```
 regr.rsq
0.6043542
```

Reference

Bischl, Bernd, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, et al. n.d. “Hyperparameter Optimization: Foundations, Algorithms, Best Practices and Open Challenges.” https://doi.org/10.48550/arXiv.2107.05847.