Applied Machine Learning

MLR3 Pipelines: Part 1

Learning goals

- Overview of PipeOps
- Nonlinear Pipelines
- Pipeline Construction

“PIPELINES” DICTIONARY & SHORT FORM

Many frequently used *patterns* for pipelines

- Making Learners robust to bad data (imputation + feature encoding + . . .)
- Bagging
- Branching

Collection of these is in mlr3pipelines

```
head(as.data.table(mlr_pipeops), 5)[, list(key, input.num, output.num)]

#> Key: <key>
#>                key input.num output.num
#>             <char>     <int>      <int>
#> 1:         boxcox         1          1
#> 2:         branch         1         NA
#> 3:          chunk         1         NA
#> 4: classbalancing         1          1
#> 5:     classifavg        NA          1
```

“PIPELINES” DICTIONARY & SHORT FORM

Many frequently used *patterns* for pipelines

- Making Learners robust to bad data (imputation + feature encoding + . . .)
- Bagging
- Branching

Collection of these is in mlr3pipelines

`po()` accesses the `mlr_pipeops` “Dictionary”.

```
pca = po("pca")
pca

#> PipeOp: <pca> (not trained)
#> values: <list()>
#> Input channels <name [train type, predict type]>:
#>   input [Task,Task]
#> Output channels <name [train type, predict type]>:
#>   output [Task,Task]
```

PIPEOPS SO FAR

```
mlr_pipeops$keys()
```

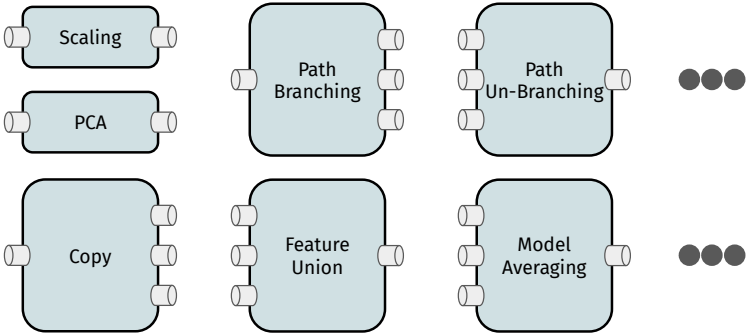
```
#>  [1] "boxcox"                "branch"                "chunk"
#>  [4] "classbalancing"        "classifavg"            "classweights"
#>  [7] "colapply"              "collapsefactors"       "colroles"
#> [10] "copy"                  "datefeatures"          "encode"
#> [13] "encodeimpact"          "encodelmer"            "featureunion"
#> [16] "filter"                "fixfactors"            "histbin"
#> [19] "ica"                   "imputeconstant"        "imputehist"
#> [22] "imputelearner"         "imputemean"            "imputemedian"
#> [25] "imputemode"            "imputeoor"             "imputesample"
#> [28] "kernelpca"             "learner"               "learner_cv"
#> [31] "missind"               "modelmatrix"           "multiplicityexply"
#> [34] "multiplicityimply"     "mutate"                "nmf"
#> [37] "nop"                   "ovrsplit"              "ovrunite"
#> [40] "pca"                   "proxy"                 "quantilebin"
#> [43] "randomprojection"      "randomresponse"        "regravg"
#> [46] "removeconstants"       "renamecolumns"         "replicate"
#> [49] "scale"                 "scalemaxabs"           "scalerange"
#> [52] "select"                "smote"                 "spatialsign"
#> [55] "subsample"             "targetinvert"          "targetmutate"
#> [58] "targettrafoscalerange" "textvectorizer"        "threshold"
#> [...]
```

PIPEOPS SO FAR AND PLANNED

- Simple data preprocessing operations (scaling, Box Cox, Yeo Johnson, PCA, ICA)
- Missing value imputation (sampling, mean, median, mode, new level, ...)
- Feature selection (by name, by type, using filter methods)
- Categorical data encoding (one-hot, treatment, impact)
- Sampling (subsampling for speed, sampling for class balance)
- Ensemble methods on Predictions (weighted average, possibly learned weights)
- Branching (simultaneous branching, alternative branching)
- Combination of data: `featureunion`
- Text processing
- Date processing
- Time series and spatio-temporal data *(planned)*
- Multi-output and ordinal targets *(planned)*
- Outlier detection *(planned)*

Nonlinear Pipelines

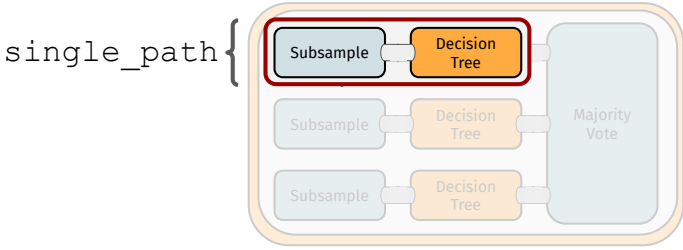
PIPEOPS WITH MULTIPLE INPUTS / OUTPUTS



MLR3PIPELINES IN ACTION

Ensemble Method: Bagging

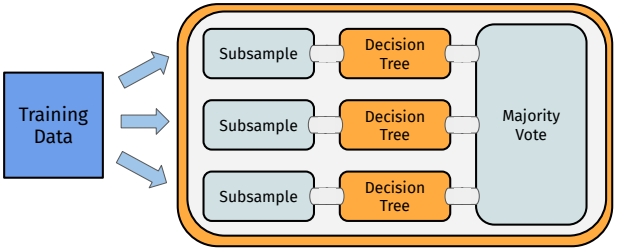
```
single_path = po("subsample") %>>% lrn("classif.rpart")
```



MLR3PIPELINES IN ACTION

Ensemble Method: Bagging

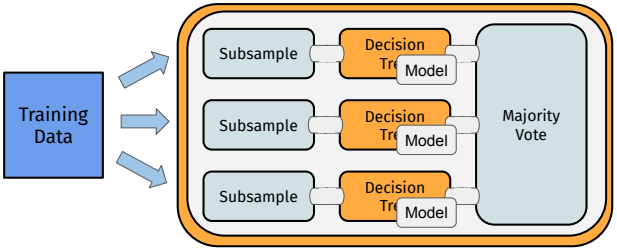
```
single_path = po("subsample") %>>% lrn("classif.rpart")

graph_bag = pipeline_greplicate(single_path, n = 3) %>>%
  po("classifavg")
```



MLR3PIPELINES IN ACTION

Ensemble Method: Bagging

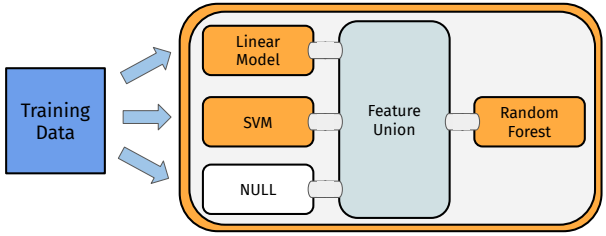
```
single_path = po("subsample") %>>% lrn("classif.rpart")

graph_bag = pipeline_greplicate(single_path, n = 3) %>>%
  po("classifavg")
```



MLR3PIPELINES IN ACTION

Ensemble Method: Stacking

```
graph_stack = gunion(list(
      po("learner_cv", learner = lrn("regr.lm")),
      po("learner_cv", learner = lrn("regr.svm")),
      po("nop"))) %>>%
  po("featureunion") %>>%
  lrn("regr.ranger")
```

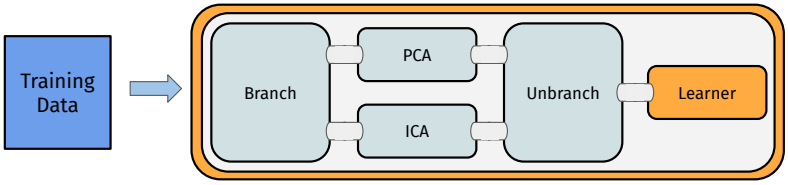


MLR3PIPELINES IN ACTION

Branching

```
graph_branch = ppl("branch", list(
  pca = po("pca"),
  ica = po("ica"))) %>>%
  lrn("classif.kknn")
```

Execute only one of several alternative paths

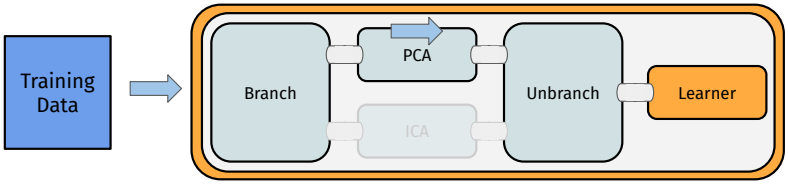


MLR3PIPELINES IN ACTION

Branching

```
graph_branch = ppl("branch", list(
  pca = po("pca"),
  ica = po("ica"))) %>>%
  lrn("classif.kknn")
```

```
> graph_branch$pipeops$branch$
    param_set$values$selection = "pca"
```



MLR3PIPELINES IN ACTION

Branching

```
graph_branch = ppl("branch", list(
  pca = po("pca"),
  ica = po("ica"))) %>>%
  lrn("classif.kknn")
```

```
> graph_branch$pipeops$branch$
    param_set$values$selection = "ica"
```