Applied Machine Learning

Feature Engineering: mlr3pipelines - Linear Pipelines

Learning goals

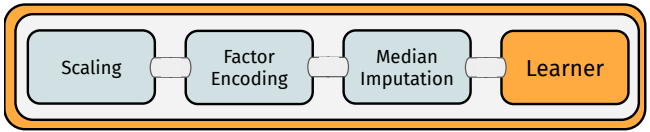
- Linear pipelines in action

mlr3pipelines: Linear Pipelines

MLR3PIPELINES IN ACTION

Linear Preprocessing Pipeline

```
graph = po("scale") %>>%
  po("encode") %>>%
  po("imputemedian") %>>%
  lrn("classif.rpart")
```

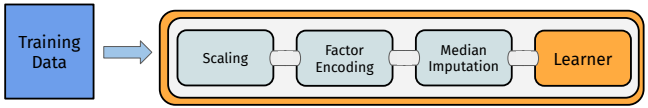


MLR3PIPELINES IN ACTION

Linear Preprocessing Pipeline

- `train()`ing: Data propagates and creates `$states`

```
glrn = as_learner(graph) # or: GraphLearner$new(graph)
glrn$train(task)
```

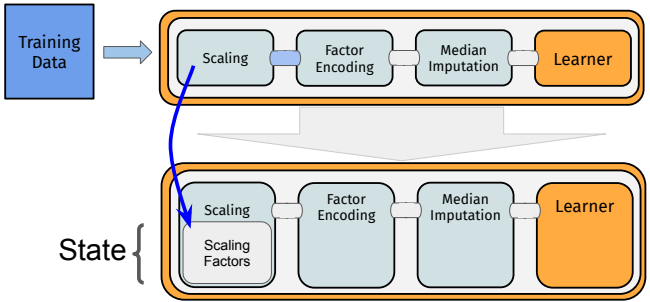


MLR3PIPELINES IN ACTION

Linear Preprocessing Pipeline

- `train()`ing: Data propagates and creates `$states`

```
glrn = as_learner(graph) # or: GraphLearner$new(graph)
glrn$train(task)
```

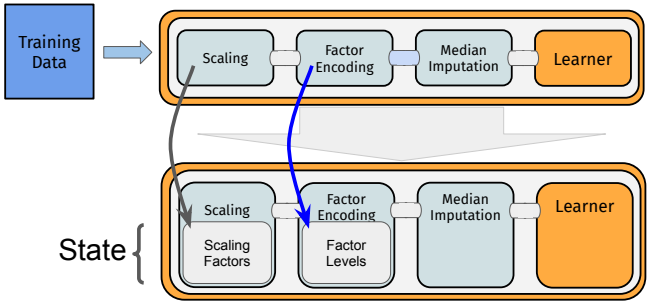


MLR3PIPELINES IN ACTION

Linear Preprocessing Pipeline

- `train()`ing: Data propagates and creates `$states`

```
glrn = as_learner(graph) # or: GraphLearner$new(graph)
glrn$train(task)
```

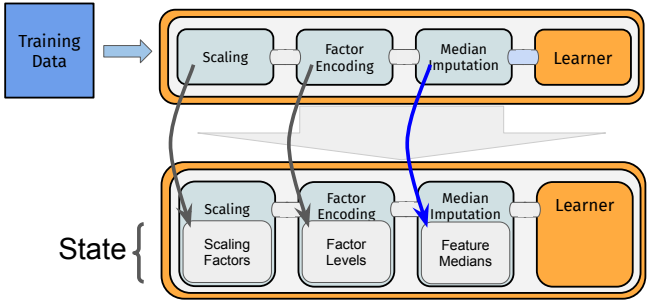


MLR3PIPELINES IN ACTION

Linear Preprocessing Pipeline

- `train()`ing: Data propagates and creates `$states`

```
glrn = as_learner(graph) # or: GraphLearner$new(graph)
glrn$train(task)
```

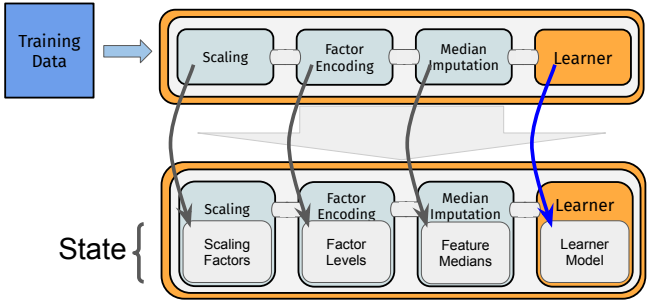


MLR3PIPELINES IN ACTION

Linear Preprocessing Pipeline

- `train()`ing: Data propagates and creates `$states`

```
glrn = as_learner(graph) # or: GraphLearner$new(graph)
glrn$train(task)
```

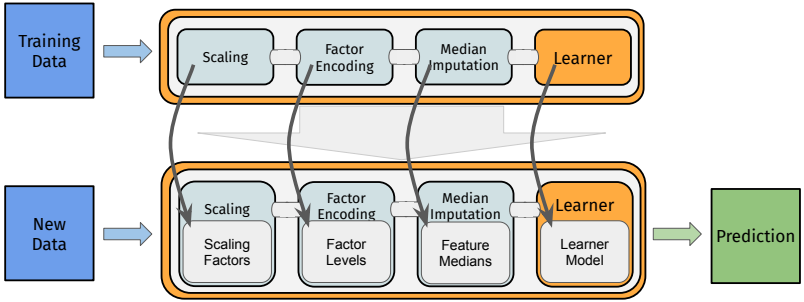


MLR3PIPELINES IN ACTION

Linear Preprocessing Pipeline

- `train()`ing: Data propagates and creates `$states`
- `predict()`tion: Data propagates, uses `$states`

```
glrn$predict(task)
```



MLR3PIPELINES IN ACTION

Linear Preprocessing Pipeline `scale %>>% encode %>>% impute %>>% rpart`

- Setting / retrieving parameters: `$param_set`

```
graph$pipeops$scale$param_set$values$center = FALSE
```

- Retrieving state: `$state` of individual `PipeOps` (*after* `$train()`)

```
graph$pipeops$scale$state$scale
#> Petal.Length  Petal.Width Sepal.Length  Sepal.Width
#>          4.2          1.4          5.9          3.1
```

- Retrieving intermediate results: `$.result` (set debug option before)

```
graph$pipeops$scale$.result[[1]]$head(3)
#>    Species Petal.Length Petal.Width Sepal.Length Sepal.Width
#>     <fctr>        <num>       <num>        <num>       <num>
#> 1:  setosa         0.34        0.14         0.86        1.13
#> 2:  setosa         0.34        0.14         0.83        0.97
#> 3:  setosa         0.31        0.14         0.79        1.03
```