

# Learning Curve Analysis

To analyse how the increase of observations in the training set improves the performance of a learner the *learning curve* is an appropriate visual tool. The experiment is conducted with an increasing subsample size and the performance is measured. In the plot the x-axis represents the relative subsample size whereas the y-axis represents the performance.

Note that this function internally uses `benchmark`

(<http://rpackages.ianhowson.com/cran/mlr/man/benchmark.html>) in combination with `makeDownsampleWrapper`

(<http://rpackages.ianhowson.com/cran/mlr/man/makeDownsampleWrapper.html>), so for every run new observations are drawn. Thus the results are noisy. To reduce noise increase the number of resampling iterations. You can define the resampling method in the `resampling` argument of `generateLearningCurveData`

(<http://rpackages.ianhowson.com/cran/mlr/man/generateLearningCurveData.html>). It is also possible to pass a `ResampleInstance`

(<http://rpackages.ianhowson.com/cran/mlr/man/makeResampleInstance.html>) (which is a result of `makeResampleInstance`

(<http://rpackages.ianhowson.com/cran/mlr/man/makeResampleInstance.html>)) to make resampling consistent for all passed learners and each step of increasing the number of observations.

## Plotting the learning curve

The `mlr` (<http://rpackages.ianhowson.com/cran/mlr/>) function `generateLearningCurveData` (<http://rpackages.ianhowson.com/cran/mlr/man/generateLearningCurveData.html>) can generate the data for *learning curves* for multiple learners ([../integrated\\_learners/index.html](http://rpackages.ianhowson.com/cran/mlr/man/integrated_learners/index.html)) and multiple performance measures ([../measures/index.html](http://rpackages.ianhowson.com/cran/mlr/man/measures/index.html)) at once. With `plotLearningCurve`

(<http://rpackages.ianhowson.com/cran/mlr/man/plotLearningCurve.html>) the result of `generateLearningCurveData`

(<http://rpackages.ianhowson.com/cran/mlr/man/generateLearningCurveData.html>) can be plotted using `ggplot2` (<http://rpackages.ianhowson.com/cran/ggplot2/>). `plotLearningCurve`

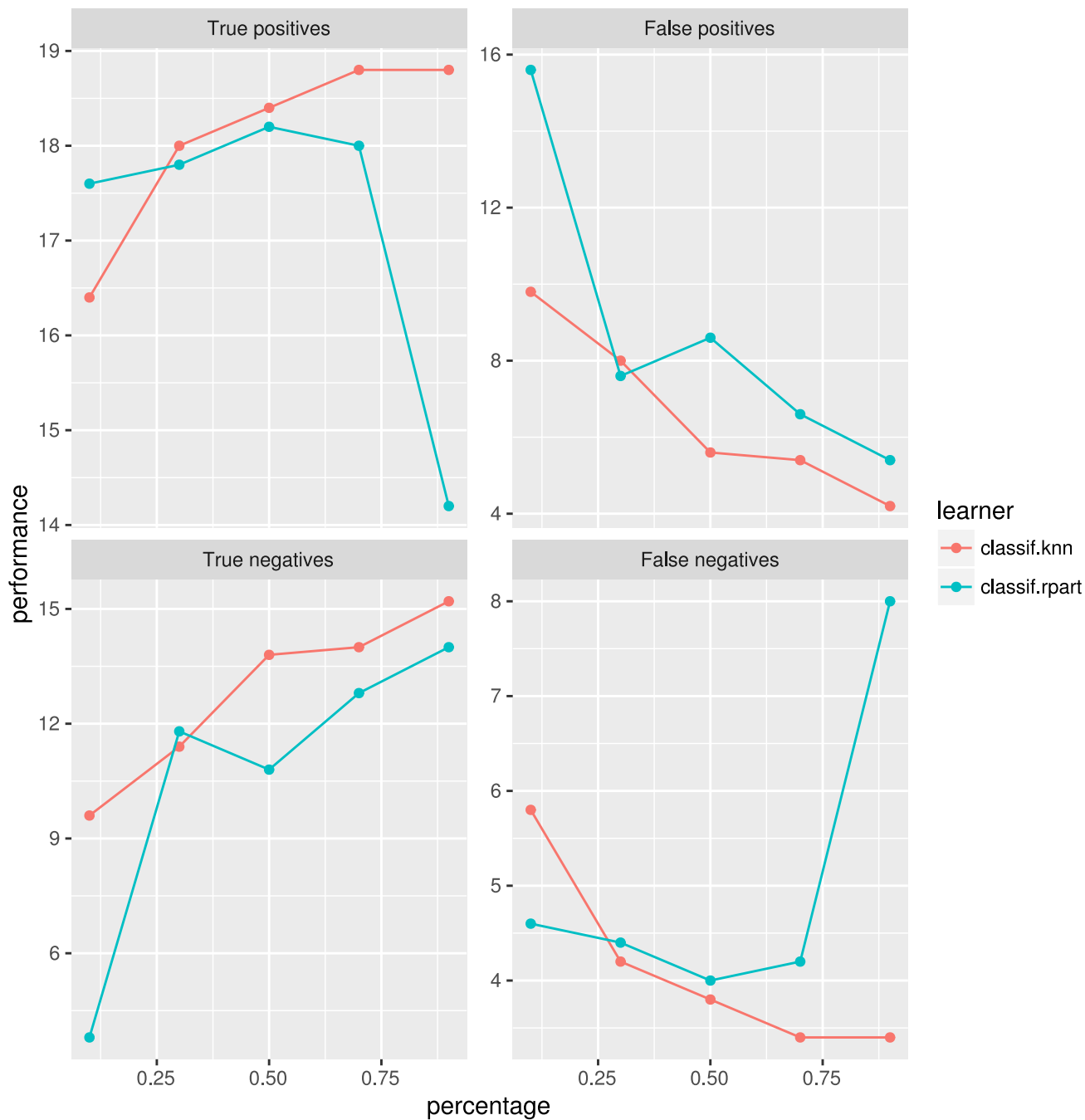
(<http://rpackages.ianhowson.com/cran/mlr/man/plotLearningCurve.html>) has an argument `facet` which can be either "measure" or "learner". By default `facet = "measure"` and faceted subplots are created for each measure input to `generateLearningCurveData`

(<http://rpackages.ianhowson.com/cran/mlr/man/generateLearningCurveData.html>). If `facet = "measure"` learners are mapped to color, and vice versa.

```

r = generateLearningCurveData(
  learners = list("classif.rpart", "classif.knn"),
  task = sonar.task,
  percs = seq(0.1, 1, by = 0.2),
  measures = list(tp, fp, tn, fn),
  resampling = makeResampleDesc(method = "CV", iters = 5),
  show.info = FALSE)
plotLearningCurve(r)

```



What happens in `generateLearningCurveData`

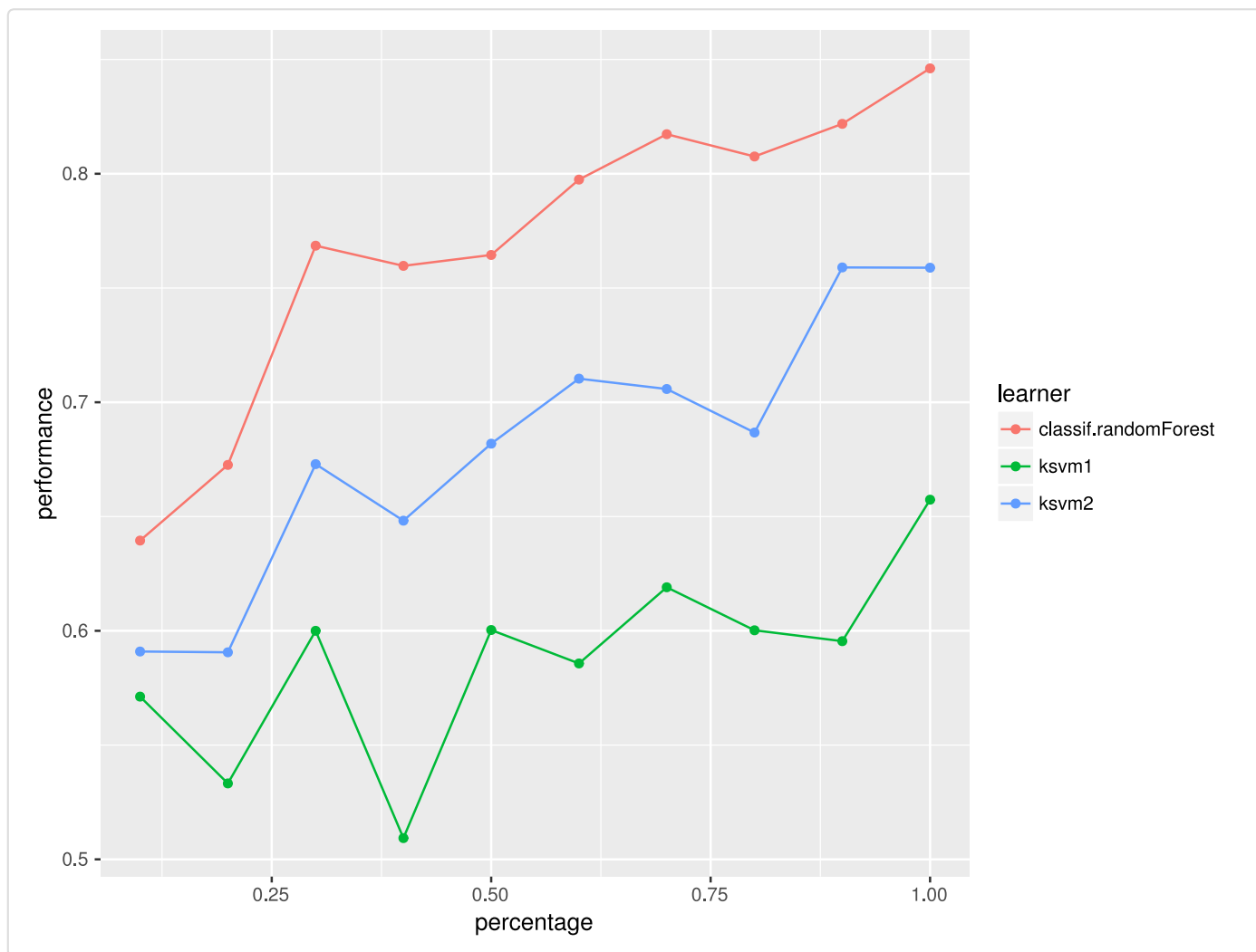
(<http://rpackages.ianhowson.com/cran/mlr/man/generateLearningCurveData.html>) is the following:

Each learner will be internally wrapped in a `DownsampleWrapper`

(<http://rpackages.ianhowson.com/cran/mlr/man/makeDownsampleWrapper.html>). To measure the performance at the first step of `percs`, say `0.1`, first the data will be split into a *training* and a *test set* according to the given *resampling strategy*. Then a random sample containing 10% of the observations of the *training set* will be drawn and used to train the learner. The performance will be measured on the *complete test set*. These steps will be repeated as defined by the given *resampling method* and for each value of `percs`.

In the first example a simplified usage of the `learners` argument was used, so that it's sufficient to give the *name*. It is also possible to create a learner the usual way and even to mix it. Make sure to give different `id`s in this case.

```
lrns = list(
  makeLearner(cl = "classif.ksvm", id = "ksvm1", sigma = 0.2, C = 2),
  makeLearner(cl = "classif.ksvm", id = "ksvm2", sigma = 0.1, C = 1),
  "classif.randomForest"
)
rin = makeResampleDesc(method = "CV", iters = 5)
lc = generateLearningCurveData(learners = lrns, task = sonar.task,
                              percs = seq(0.1, 1, by = 0.1), measures = acc,
                              resampling = rin, show.info = FALSE)
plotLearningCurve(lc)
```



There is also an experimental ggvis (<http://rpackages.ianhowson.com/cran/ggvis/>) plotting function, `plotLearningCurveGGVIS` (<http://rpackages.ianhowson.com/cran/mlr/man/plotLearningCurveGGVIS.html>). Instead of the `facet` argument to `plotLearningCurve` (<http://rpackages.ianhowson.com/cran/mlr/man/plotLearningCurve.html>) there is an argument `interactive` which plays a similar role. As subplots are not available in ggvis (<http://rpackages.ianhowson.com/cran/ggvis/>), measures or learners are mapped to an interactive sidebar which allows selection of the displayed measures or learners. The other feature is mapped to color.

```
plotLearningCurveGGVIS(r, interactive = "measure")
```

Documentation built with MkDocs (<http://www.mkdocs.org/>).