Weekly Exercises Week 10

Marjolein Fokkema

The data used for this exercise are described in the document with the exercises for this week. So first check out those exercises, the current weekly exercise builds on these, most specifically on the gradient boosting ensemble (Exercise 3).

Cross validation of gradient boosting parameters

In this exercise, you tune the parameters of a gradient-boosted ensemble using cross validation (CV). To this end, we use function `train` from package `caret` (short for Classification And REgression Testing):

```r
library("caret")
```

Each prediction method that can be tuned with function `train` has a pre-specified set of parameter values. These are rarely an exhaustive set of the parameters, but you can generally trust that these are the main parameters driving predictive accuracy. For each parameter, you need to specify a set of values. If you specify only a single value, the parameter will not be varied and tuned but kept fixed.

To inspect the pre-specified set of parameter values for the `gbm` method, type `modelLookup("gbm")`.

To specify the parameter values to test, provide a grid that has parameters on the columns and each row providing the set of values to try. Other approaches are also available (e.g., random search), but we will not use it in this course. Use the following grid (if you feel confident or adventurous, you may of course adjust this grid in any way you like):

```r
grid <- expand.grid(shrinkage = c(.1, .01, .001),
                    n.trees = c(10, 100, 1000, 2000, 2500),
                    interaction.depth = 1:4,
                    n.minobsinnode = 10)
```

a) Next, apply function `train`. Make sure to set the random seed, first. Pass `grid` to the `tuneGrid` argument. Note that unlike function `gbm`, function `train` wants a factor as the response in classification problems, so it is best to supply the original `qsar` dataset to function `train` (but make sure to only specify the training observations).

Also make sure to specify `distribution = "bernoulli"` (it will be passed to function `gbm` through the ellipsis `...`). Specify `trControl = trainControl(number = 10)`. Type `?trainControl` to see what this does.

Applying function `train` will take quite some time to run. Cross validating over the parameter grid involves fitting `nrow(grid)*10` models. You can additionally pass `verboseIter = TRUE` to function `trainControl` to have the fitting progress printed to the command line.

b) Once function `train` has completed, print the result. Use `plot` to visualize the result.

c) Describe the main effects of the `shrinkage`, `n.trees` and `interaction.depth` parameters on the accuracy of the model. Also describe their possible interactions.

d) If you look at the effect of `interaction.depth`, what would you conclude about the possible presence of interactions in the QSAR dataset?

e) Refit a gradient boosted ensemble on the training data using function `gbm`. This time, use the optimal set of parameter values returned by function `train` (you can extract this from the result of `train` as `$bestTune`).

f) Use function `predict` to compute predicted probabilities for the test observations. Compute the Brier score and misclassification rate. Compare accuracy of this boosted ensemble with that of all earlier models (single trees, bagged and random forest ensemble, boosted ensemble with default settings). Which approach yielded the most accurate predictions?