Day 2, Practical 2

Thomas Alexander Gerds

September 28, 2021

Super learner

1. Load the following libraries (install them if necessary).
2. Download the `pph` data from the course website. The `pph` data are saved in `rds` (R Data Single) format.
   - Load the data into R (`readRDS`).
   - Split the data into two data sets:
     - `pph09` contains only data from calendar year 2009 (`Year==2009`)
     - `pph` contains data from the previous years (`Year!=2009`).
   - The outcome variable is planned cesarian section at second delivery: `plannedCS` (1="yes", 0="no"). Calculate the probability of a planned cesarian section at second delivery by calendar year. Is there a calendar time trend?
   - The predictor variables are the following characteristics of the first delivery:
     - `MotherAge`
     - `PrevPPHbin`
     - `PrevArgumented`
     - `PrevPraeecl`
     - `PrevAbruptio`
     - `PrevCS`
     - `PrevRetained`
     - `PrevInduced`

     Calculate the median age and the probabilities of the other risk factors at second delivery by calendar year. Are there calendar time trends?
3. Fit the following models in the data `pph` (before 2009):
   - logistic regression (`glm`) with additive effects of all predictor variables.

- generalized additive model (`gam`) with additive effects of all predictor variables where the effect of `MotherAge` is modeled by a smoothing spline.
- Random forest (`rfsrc`) with 200 trees (to make the fit faster) and otherwise default parameters.

   Then predict the probabilities of a planned cesarian section for the 2009 data. Scatterplot the predicted probabilities against each other. Calculate the average Brier scores in the 2009 data.

4. Create level-one data

   - data splitting: 10-fold cross-validation
     - split the data `pph` (before 2009) at random into 10 non-overlapping subsets (called folds in what follows) with roughly same size.
   - in a `foreach::foreach` loop (with argument `.combine` set to `"rbind"`) across the folds do:
     - fit all models (i.e., train all learners) from step 3 in the data which excludes the current fold
     - predict the probabilities of planned cesarian section for the subjects in the current fold
     - combine the resulting matrix with the observed outcome values (`plannedCS`) of the current fold into a data frame: the level-1 data corresponding to this fold.
   - the result of the `foreach` loop are the level-1 data obtained with 10-fold cross-validation. There are 4 columns where the first is the observed outcome (`plannedCS`) and the remaining are the cross-validated predicted probabilities of the three learners. The level-1 data are used in the following to construct super learners.

5. Calculate the following super learners:

   - discrete super learner (manual programming)
   - Polley's default: non-negative least squares (package `nnls`)
   - Breiman's suggestion: shrinkage (e.g., package: `rms::lrm`, set penalty to `0.2`)

   Then scatterplot the predicted probabilities of the 3 super learners against each other.

6. Use the `SuperLearner` package:

   - from the learning data extract the outcome column (argument `Y`) and construct the design matrix which contains the "dummy coding" of the predictor values (argument `X`), e.g., with `model.matrix`.
   - specify the following super learner libraries:

    - SL.mean
    - SL.glm
    - SL.gam
    - SL.glmnet
  - Consider the coefficients of the `nnls` fit: `object$coef`
  - Plot the predicted values: `object$SL.predict` against the manually computed super learner from step 5.

7. In the 2009 data `pph09` compare the average Brier scores of:
   - all single models from step 3
   - the super learner model from step 6.
   - the single library results of the super learner from step 6.

8. Monte-Carlo error
   - Check the program and mark all lines with random seed dependence.
   - Set the seed (`set.seed`) to control the randomness and re-run the whole program.
   - Visualize the seed dependence of the super learner obtained with the `SuperLearner` package (step 6) by scatterplotting the predicted values in the 2009 data for two different seeds.

9. Tune the random forest parameters
   - create a list of strong random forest learners by varying the `ranger` parameters `mtry` (values `1,3,7` and minimum node size (values `20,50,100`). See section 10 of the R package vignette: `Guide-to-SuperLearner`.
   - run the SuperLearner by adding the tuned forest to the libraries of step 7, check the coefficients . . .

References

1. David H Wolpert. Stacked generalization. Neural Networks, 5(2):241–259, 1992.

2. Leo Breiman. Stacked regressions. Machine Learning, 24(1):49–64, 1996.

3. Mark J Van der Laan, Eric C Polley, and Alan E Hubbard. Super learner. Statistical Applications in Genetics and Molecular Biology, 6(1), 2007.