Homework 10

Advanced Methods for Data Analysis (36-402)

Due Friday April 19, 2019, at 3:00 PM

See the syllabus for general instructions regarding homework. Note that you should **always show all your work** and submit **exactly two files** (Rmd or R file as *code*, and knitted or scanned/merged pdf or html as *writeup*). Make sure that everything you submit is readable.

Gross Metropolitan Product

Economists believe that the economic output per person (called the *per-capita gross metropolitan product* or `pcgmp`) of a city or region tends to increase with the population. Recently, there has been some controversy over the exact form of the relationship, and over its explanation. One claim is that `pcgmp` (Y) is related to population N by a power law:

$$Y \approx \beta N^{\gamma}, \tag{1}$$

where β and γ are positive constants. Another claim is the *urban hierarchy* model in which larger regions have different mixtures of industries and the different industries have different levels of economic output. In this problem, you will compare the power-law and urban-hierachy models of *pcgmp*.

The data file `gmp.csv` is posted on the Canvas web site with this assignment. There are 133 observations with the following variables:

- the name of each metropolitan area;
- its per-capita gross metropolitan product, in dollars;
- its population;
- the share of its economy derived from finance;
- the share of its economy derived from "professional and technical services";
- the share of its economy derived from "information, communication and technology" (ICT); and
- the share of its economy derived from "management of firms and enterprises."

To fit models in this problem, use the package `mgcv` which fits additive models and allows the degrees of freedom to be chosen by generalized cross-validation (GCV), unlike the package `gam`. The `gam` function in the `mgcv` package has a similar syntax to the function with the same name in the `gam` package, but there are several differences. The most important difference for this problem is the following: If you specify a component of the additive model as `s(ict)`, for example, the effective degrees of freedom (edf) for this spline will be chosen by GCV. If you want to fix the edf at `d`, specify `s(ict,k=d+1,fx=T)`. By default, the `k` parameter sets an upper bound for the edf, and it counts an intercept for each spline, which gets removed before the fitting is finished. The `fx=T` argument forces the specified edf to be the value of `k` (minus 1) rather than using `k` as an upper bound.

Look at the `summary` for the fit. The chosen edf for each (non-linear) predictor is reported in `summary`.

Part a

Fit the linear version of the power law (1):

$$\log(Y) = \alpha + \gamma \log(N) + \varepsilon.$$

Call this ModelA. Look at residuals to see if there are any assumptions of a linear model that appear to be violated.

Part b

Fit an additive model for $\log(Y)$ in which $\log(N)$ enters linearly as in ModelA, and the logarithms of the four "share" variables enter non-linearly as splines with 4 degrees of freedom. For example, the term for `finance` would be `s(log(finance),k=5,fx=T)`, and similarly for the other three. Call this ModelB. Look at the residuals and model summary. Comment on what they tell you.

Part c

Use an F test to test the null hypothesis that ModelA is correct against the alternative that the larger ModelB is correct. Is there evidence that the $\log(N)$ term in ModelB is really needed? For each of the four "share" variables, comment on the extent to which a non-linear fit is really needed. Use the `plot` function to draw the partial response functions, and see if these might contain some information.

Part d

There arises the question of whether the F test that was used in part (c) is really appropriate for the models being tested. ModelA is a simple linear regression model with response $\log(Y)$ and predictor $\log(N)$. But the alternative ModelB is a non-linear model. You can bootstrap the null distribution of the F statistic[1] in two different ways. Recall from previous lectures:

- A parametric bootstrap would assume that the data arise from ModelA, and the noise terms are i.i.d. normal random variables with mean 0 and common variance σ^2. One would need to estimate σ using the data. For each bootstrap sample, one would simulate new noise terms that would then be added to the fitted values from ModelA.
- A "resample residuals" bootstrap would assume that the data arise from ModelA, and that the noise terms are i.i.d. random variables with a common distribution H. One would use the empirical distribution $\hat{H}$ of the residuals from ModelA to stand in for that common distribution H. For each bootstrap sample, one would simulate new noise terms by sampling from $\hat{H}$ with replacement. The new noise terms would then be added to the fitted values from ModelA.

Perform both bootstraps described above with $B = 1000$ bootstrap samples each. Comment on what these analyses suggest about the null distribution of the F statistic used to test the hypotheses in part (c).

Part e

Explain why it is not appropriate to do a "resample cases" bootstrap to estimate the null distribution of the F statistic. If one did perform a "resample cases" bootstrap and computed the F statistic for each bootstrap sample, what distribution of the F statistic would one be estimating?

[1] The *null distribution* of a test statistic is the distribution of the test statistic under the assumption that the null hypothesis is true.

Part f

The size of the sample that we are using is not large enough to get a good estimate of prediction MSE from five-fold cross-validation. Instead, do leave-one-out cross-validation (LOOCV) to compare ModelA, ModelB, and ModelC, where ModelC is the simplification of ModelB in which all contributions of `log(pop)`, `log(finance)` and `log(prof.tech)` are removed from the model. The shortcut that we learned for computing LOOCV MSE applies to ModelA, but not the other two. Compute the LOOCV MSE for each of the three models by refitting the models once for each observation let out. Also, to see that the shortcut applies only to ModelA, use the shortcut based on the diagonal entries of the smoothing matrix S. The diagonal entries of the smoothing matrix S of Modelx (for x=A,B,C) are stored in `Modelx$hat`. Based on LOOCV and all of the other analyses you have done, which of the three models looks best? Mention at least one further improvement that is suggested by some of the analyses done so far.

Part g

Compute coefficient 0.9 confidence intervals for the response function from ModelC at the predictor vectors corresponding to the observations numbered 10, 34, and 70 in the data set. The `predict` function, when applied to an additive model object will compute the standard errors that are the square-roots of what we called $\hat{s}^2(\hat{Y}_i)$ in the lecture notes. (Use `help(predict.gam)`) in R to learn the syntax for the `predict` function that works with `gam` objects.) Compare these confidence intervals to pivotal bootstrap confidence intervals based on resampling cases. Use 1000 bootstrap samples. Use the same bootstrap samples to estimate the bias and standard deviation of the three $\hat{r}(x_i)$ values. Do the biases appear large? Are the `se.fit` values produced by the `predict` function close to the standard deviations estimated by the bootstrap? Do the three sets of bootstrap values of $\hat{r}(x_i)$ appear to have approximately the t distributions that correspond to confidence intervals that the were computed without the bootstrap?