Homework Assignment 2: The Advantages of Backwardness

36-402, Data Analysis, Spring 2011

Due 25 January 2011

Many theories of economic growth say that it's easier for poor countries to grow faster than rich countries — "catching up", or the "advantages of backwardness". One argument for this is that poor countries can grow by copying existing, successful technologies and ways of doing business from rich ones. But rich countries are already using those technologies, so they can only grow by finding new ones, and copying is faster than innovation. So, all else being equal, poor countries should grow faster than rich ones. One way to check this is to look at how growth rates are related to other economic variables.

We will use the `np` package on CRAN to do kernel regression.[1] Install it, and load its `oecdpanel` data set. This contains growth data for many countries for 1960–1995, collected by the Organization for Economic Cooperation and Development (the OECD). We won't use all the variables this time.

GDP is "gross domestic product", the total value of all economic production. It's usually reported per capita and per year. Call it $Y_{i,t}$, since it depends on the country i and the year t. GDP isn't perfect[2], but it is standard.

In `oecdpanel`, the variable `growth` is the logarithmic growth rate of GDP, $= \log Y_{i,t+1}/Y_{i,t}$. We look at logarithms because economic models suggest that the factors which affect growth should multiply together, rather than adding. What's actually recorded here is the average growth rate over a five-year period, reducing year-to-year accidents.

`initgdp` is $\log Y_{i,t}$, the logarithm of per-capita GDP at the start of each five-year period.

A country's investment rate is the fraction of its GDP that goes into building or repairing productive assets (roads, harbors, power plants, factory machines, buildings, etc.). `inv` is the logarithm of the investment rate, so `inv=-2.26` means 10.4% of output was invested.

`popgro`, similarly, is the logarithm of the population growth rate.

[1] The package has good help files, if you want to know more. Or see `http://www.jstatsoft.org/v27/i05`.

[2] If everyone gets worried about being robbed, GDP goes *up* by the amount we spend on extra locks, alarms, guards, etc., none of which would be needed if we just didn't have so many burglars.

1. (5 points) Fit a linear model of `growth` on `initgdp`. What is the coefficient? What does it suggest about catching-up?

2. (20 points) The `npreg` function in the `np` package does kernel regression. By default, it uses a combination of cross-validation and sophisticated but very slow optimization to pick the best bandwidth. In this problem, though, we will force it to use fixed bandwidths, and do the cross-validation ourselves.

   ```
   oecd.0.1 <- npreg(growth~initgdp,bws=0.1,data=oecdpanel)
   ```

   does a kernel regression of `growth` on `initgdp`, using the default kernel (which is Gaussian) and bandwidth 0.1. You can run `fitted`, `predict`, etc., on the output of `npreg` just as you can on the output of `lm`.

   The code at the end of this assignment (also online) uses five-fold cross-validation to estimate the mean-squared error for the five bandwidths $0.1, 0.2, 0.3, 0.4, 0.5$. Use it to create a plot of MSE versus bandwidth. Add to the same plot the MSEs of the five bandwidths on the *whole* data. What bandwidth predicts best?

3. (10 points) Make a scatterplot of `initgdp` versus `growth`. Add the line for the linear model. Add the fitted values for the kernel curve with the best bandwidth (according to the previous problem). Does the kernel regression curve suggest that poorer countries tend to grow faster?

   (There are at least two ways to get the fitted values for the kernel regression, using `fitted` or `predict`.)

4. (5 points) If we want to check whether poorer countries tend to grow faster, all else being equal, it seems reasonable to try to keep all else equal. Do a linear regression of `growth` on `initgdp`, along with `popgro` and `inv`. What are the new regression coefficients? Does the coefficient of `initgdp` have the same sign as before? What does it suggest about catching-up?

5. (10 points) `npreg` will also do kernel regressions with multiple input variables. This time, use the built-in bandwidth selector:

   ```
   oecd.npr <- npreg(growth ~ initgdp + popgro + inv, data=oecdpanel, tol=0.1, ftol=0.1)
   ```

   (The last two arguments tell the bandwidth selector to not be very hard to optimize — which in this case saves a *lot* of time, and works out well.) What are the selected bandwidths? (Use `summary`.)

6. (15 points) What are the median values of `popgro` and `inv`? For countries with those median values, plot the predicted growth rate versus initial GDP, under both the linear model from problem 4 and the kernel regression from problem 5. (One way to do this is to use `predict`, but there are probably others.) Describe what each curve suggests about catching-up.

7. (15 points) To chose between the linear model and the kernel regression, use cross-validation again. Modify the code from problem 2 to use five-fold cross-validation to get CV MSEs for both models. Which predicts better?

8. (10 points) Based on your analysis, does the data support the idea of catching up, undermine it, or provide no evidence either way? (As always, explain your answer.)

```
# Compare predictive ability using five-fold CV
nfolds <- 5
case.folds <- rep(1:nfolds,length.out=nrow(oecdpanel))
    # divide the cases as evenly as possible
case.folds <- sample(case.folds) # randomly permute the order
bandwidths <- (1:5)/10 # Evenly space bandwidths from 0.1 to 0.5
fold.mses <- matrix(0,nrow=nfolds,ncol=length(bandwidths))
colnames(fold.mses) = as.character(bandwidths)
   # By naming the columns, we'll won't have to keep track of which bandwidth
   # is in which position
for (fold in 1:nfolds) {
  # What are the training cases and what are the test cases?
  train <- oecdpanel[case.folds!=fold,]
  test <- oecdpanel[case.folds==fold,]
  for (bw in bandwidths) {
    # Fit to the training set
    current.npr <- npreg(growth ~ initgdp, data=train,bws=bw)
    # Predict on the test set
    predictions <- predict(current.npr, newdata=test)
    # What's the mean-squared error?
    fold.mses[fold,paste(bw)] <- mean((test$growth - predictions)^2)
    # Using paste() here lets us access the column with the right name...
  }
}
# Average the MSEs
bandwidths.cv.mses <- colMeans(fold.mses)
```