A Monte Carlo Investigation of Locally Weighted Regression

Aaron Swoboda and Sam Carruthers

October 16, 2012

This document writes up the results of the recent run of `uberScript.R`. It contains the following code:

```
# set our simulation parameters
Replications = 100
sample.size = c(50, 100, 200, 500, 1000)
error.sd = c(2, 4, 6)
B1.spatial.var = c(0, .1, .2, .3)
B2.spatial.var = c(0, .1, .2, .3)

# now march through the different parameter combinations running the simulations

for( i in 1:meta.sim.num) {
  start = Sys.time()
  simRepOut = simulationReplicator(Replications, sim.parameters[i, ], MC = TRUE)
  simOut = simRepReorganizer(simRepOut)

  R2Output[as.character(sim.parameters[i, "sample.size"]),
           as.character(sim.parameters[i, "error.sd"]),
           as.character(sim.parameters[i, "B1.spatial.var"]),
           as.character(sim.parameters[i, "B2.spatial.var"]), , ] = simOut[[1]]

  MetricOutput[as.character(sim.parameters[i, "sample.size"]),
               as.character(sim.parameters[i, "error.sd"]),
               as.character(sim.parameters[i, "B1.spatial.var"]),
               as.character(sim.parameters[i, "B2.spatial.var"]), , , ] = simOut[[2]]
  end = Sys.time()

  print(paste("For loop", i,"of", meta.sim.num))
  print(round(difftime(end, start, units = "m"), 2))
  save(R2Output, MetricOutput, file = "SpecificationSims/uberScriptOutput.RData")
}
```

I'm not going to run that code here (it took almost a month to run on the R Server), but let's load up the results and start to look at them. Or at least come up with some questions to ask of the data and a plan for the future.

```
load("../Data/uberScriptOutput20120919.RData")
dimnames(MetricOutput)
```

```
## $ss
## [1] "50"   "100"  "200"  "500"  "1000"
##
## $error.sd
## [1] "2" "4" "6"
##
## $B1sv
## [1] "0"   "0.1" "0.2" "0.3"
##
## $B2sv
## [1] "0"   "0.1" "0.2" "0.3"
##
## $simNum
##   [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11"
##  [12] "12"  "13"  "14"  "15"  "16"  "17"  "18"  "19"  "20"  "21"  "22"
##  [23] "23"  "24"  "25"  "26"  "27"  "28"  "29"  "30"  "31"  "32"  "33"
##  [34] "34"  "35"  "36"  "37"  "38"  "39"  "40"  "41"  "42"  "43"  "44"
##  [45] "45"  "46"  "47"  "48"  "49"  "50"  "51"  "52"  "53"  "54"  "55"
##  [56] "56"  "57"  "58"  "59"  "60"  "61"  "62"  "63"  "64"  "65"  "66"
##  [67] "67"  "68"  "69"  "70"  "71"  "72"  "73"  "74"  "75"  "76"  "77"
##  [78] "78"  "79"  "80"  "81"  "82"  "83"  "84"  "85"  "86"  "87"  "88"
##  [89] "89"  "90"  "91"  "92"  "93"  "94"  "95"  "96"  "97"  "98"  "99"
## [100] "100"
##
## $optimized
##  [1] "AICc"     "corB0"    "corB1"    "corB2"    "CV"       "GCV"
##  [7] "R2"       "RMSE.B0"  "RMSE.B1"  "RMSE.B2"  "SCV"      "ttest%B0"
## [13] "ttest%B1" "ttest%B2"
##
## $metric
##  [1] "bandwidths" "B0.cor"     "B1.cor"     "B2.cor"     "B0.RMSE"
##  [6] "B1.RMSE"    "B2.RMSE"    "B0.t.perc"  "B1.t.perc"  "B2.t.perc"
## [11] "GCV"        "SCV"        "CV"         "AICc"       "R2"
##
```

```
dimnames(R2Output)
```

```
## $ss
## [1] "50"   "100"  "200"  "500"  "1000"
##
## $error.sd
## [1] "2" "4" "6"
##
## $B1sv
## [1] "0"   "0.1" "0.2" "0.3"
##
## $B2sv
## [1] "0"   "0.1" "0.2" "0.3"
##
## $simNum
##   [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11"
```

```
##  [12] "12"  "13"  "14"  "15"  "16"  "17"  "18"  "19"  "20"  "21"  "22"
##  [23] "23"  "24"  "25"  "26"  "27"  "28"  "29"  "30"  "31"  "32"  "33"
##  [34] "34"  "35"  "36"  "37"  "38"  "39"  "40"  "41"  "42"  "43"  "44"
##  [45] "45"  "46"  "47"  "48"  "49"  "50"  "51"  "52"  "53"  "54"  "55"
##  [56] "56"  "57"  "58"  "59"  "60"  "61"  "62"  "63"  "64"  "65"  "66"
##  [67] "67"  "68"  "69"  "70"  "71"  "72"  "73"  "74"  "75"  "76"  "77"
##  [78] "78"  "79"  "80"  "81"  "82"  "83"  "84"  "85"  "86"  "87"  "88"
##  [89] "89"  "90"  "91"  "92"  "93"  "94"  "95"  "96"  "97"  "98"  "99"
## [100] "100"
##
## $R2
## [1] "OLS" "LWR"
##
```

So, we ran some simulations, varying the sample size of the data set, the standard deviation of the error term in the model and the degree of spatial variation in the model coefficients.

Each simulation was conducted as follows:

1. Grab the simulation parameters.
2. Generate the data according to the model and parameters.
3. Choose a number of observations to include in the Locally Weighted Regression.
4. Run Locally Weighted Regression on the data using the chosen bandwidth for each observation within the dataset.
5. Calculate a number of model metrics for each bandwidth
6. Repeat previous two steps for a number of bandwidths, ranging from only 5 data points to a model approaching a global Ordinary Least Squares model (in our case, we still had declining weights based on distance, but all observations received positive weight in the regresssion).
7. Collect data on each metric when each metric is optimized. For instance, when we choose the bandwidth associated with the lowest GCV score, what are the other metric values (β RMSEs, etc.)

We kept track of the following model performance metrics, the pseudo R^2 of the model results, the correlation between the $\hat{\beta}$ and the true β, the percent of the observations for which we can reject the null hypothesis that $\hat{\beta} = \beta$, cross validation scores (leave one out, generalized, and standardized according to Paez), lastly the AIC score.

0.1 Data Generation Process

The Data Generation Process is achieved using the `DataGen` function, the code for which is given below.

```
source("../SimFunctions.R")
DataGen

## function (sample.size, error.sd, B1.spatial.var, B2.spatial.var)
## {
##     n = sample.size
##     east = runif(sample.size) * 10
##     north = runif(sample.size) * 10
##     indep.var1 = runif(sample.size) * 10
##     indep.var2 = runif(sample.size) * 10
```

```
##     trueB0 = 0
##     trueB1 = B1.spatial.var * north + 1 - 5 * B1.spatial.var
##     trueB2 = B2.spatial.var * east + 1 - 5 * B2.spatial.var
##     error = rnorm(sample.size, 0, error.sd)
##     dep.var = trueB0 + indep.var1 * trueB1 + indep.var2 * trueB2 +
##         error
##     output = data.frame(dep.var, north, east, indep.var1, indep.var2,
##         trueB0, trueB1, trueB2, error)
##     output
## }
```

The dependent variable is produced as follows:

$$Y = \beta_0 + \beta_1(location)X_1 + \beta_2(location)X_2 + error \tag{1}$$

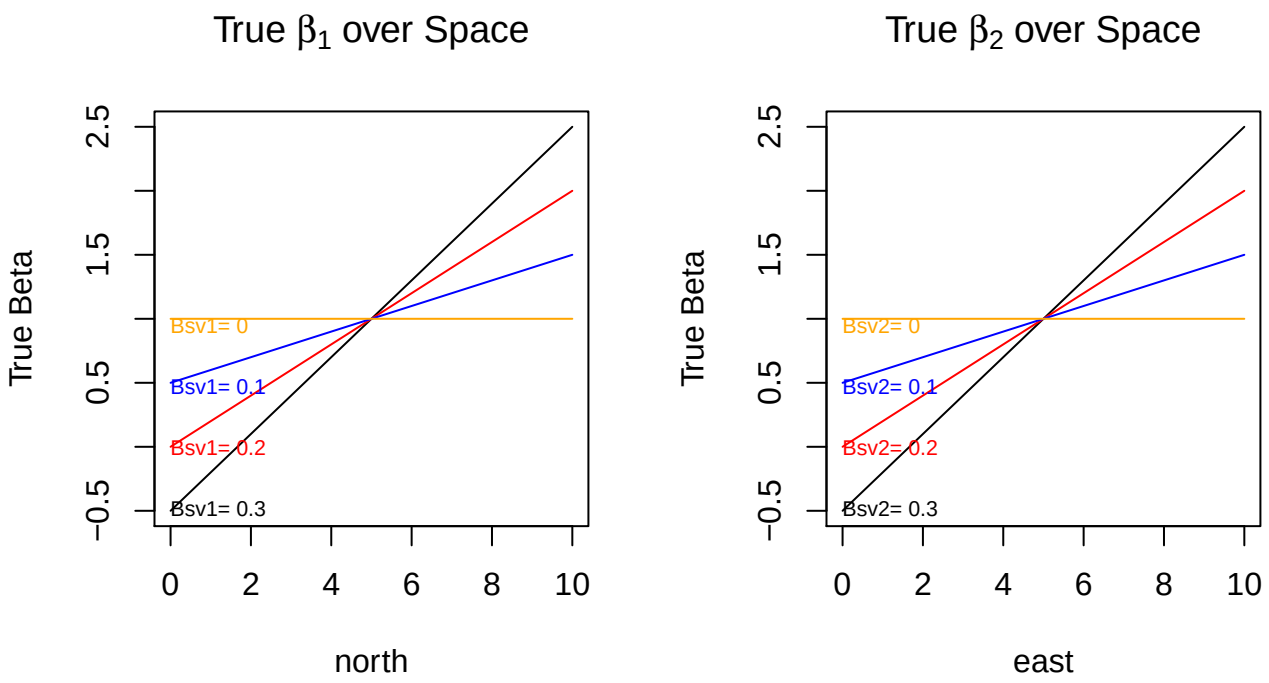
where $error \sim n(0, \sigma^2)$. Each observation is located within a geographic coordinate system $(east, north)$ where both $east$ and $north$ values are $\sim u(0, 10)$. The functions determining β_1 and β_2 are :

$$\beta_1(east, north) = 1 + Bsv1 * north - 5 * Bsv1 \tag{2}$$
$$\beta_2(east, north) = 1 + Bsv2 * east - 5 * Bsv2 \tag{3}$$

In our simulations we let Bsv_i vary $\{0, 0.1, 0.2, 0.3\}$, thus the relationship between βs and location can be visualized as:

```
east = north = 0:10
BetaFunc = function(x, Bsv) {
    1 + Bsv * x - 5 * Bsv
}
par(mfrow = c(1, 2))
plot(north, BetaFunc(north, 0.3), type = "l", xlab = "north", ylab = "True Beta",
    main = expression(paste("True ", beta[1], " over Space")))
lines(north, BetaFunc(north, 0.2), col = "red")
lines(north, BetaFunc(north, 0.1), col = "blue")
lines(north, BetaFunc(north, 0), col = "orange")
text(rep(0, 4), seq(0.925, -0.5, length = 4), paste("Bsv1=", (0:3)/10),
    pos = 4, col = c("orange", "blue", "red", "black"), cex = 0.7, offset = 0)

plot(east, BetaFunc(east, 0.3), type = "l", xlab = "east", ylab = "True Beta",
    main = expression(paste("True ", beta[2], " over Space")))
lines(east, BetaFunc(east, 0.2), col = "red")
lines(east, BetaFunc(east, 0.1), col = "blue")
lines(east, BetaFunc(east, 0), col = "orange")
text(rep(0, 4), seq(0.925, -0.5, length = 4), paste("Bsv2=", (0:3)/10),
    pos = 4, col = c("orange", "blue", "red", "black"), cex = 0.7, offset = 0)
```



Our simulations include data generation processes in which:

1. neither coefficient varies over space ($Bsv_1 = 0$ & $Bsv_2 = 0$)
2. both coefficients vary over space ($Bsv_1 \neq 0$ & $Bsv_2 \neq 0$)
3. only one coefficient varies over space ($Bsv_1 = 0$ & $Bsv_2 \neq 0$ OR $Bsv_1 \neq 0$ & $Bsv_2 = 0$)

Each simulation can be characterized by our selection of four data generation parameters,

- sample size $\{50, 200, 500, 1000\}$
- variance of the error term $\{2^2, 4^2, 6^2\}$
- degree of spatial variation in β_1 $\{0, .1, .2, .3\}$
- degree of spatial variation in β_2 $\{0, .1, .2, .3\}$

1 Research Questions

What do we want to know about LWR?

1. Are there systematic differences in the bandwidth size selected by different techniques? How do LOOCV, Standardaized CV, Generalize CV, and the AICc compare?
2. What sort of spatial variation in the coefficients is necessary relative to the error to need LWR?
3. If there is no spatial relationship, will LWR default back to global OLS?

2 Applying Locally Weighted Regression

After generating the data, we applied Locally Weighted Regression and calculated numerous diagnostics in order to measure the performance of the regression technique.

Locally Weighted Regression (LWR) is an estimation strategy allowing non-stationary model parameters. A vector of regression parameters is estimated using Equation (4) for each location within the dataset,

$$\hat{\beta}_{location_i} = (X^T W_{location_i} X)^{-1} X^T W_{location_i} Y, \tag{4}$$

where X is the standard n x m data matrix, Y the n x 1 vector of dependent variable values, and $W_{location_i}$ is an n x n weights matrix. We construct the weights matrix for a given location to give positive weights to the k-nearest data points, with weights declining according to a bi-square function as distances increase. Specifically, we create the weights matrix with zeros on the off-diagonal and calculate the jjth diagonal element as,

$$w_{ij} = \begin{cases} \left[1 - \left(\frac{d_{ij}}{d_{ik}}\right)^2\right]^2 & \text{if } d_{ij} \leq d_{ik} \\ 0 & \text{if } d_{ij} > d_{ik} \end{cases} \tag{5}$$

where d_{ij} is the distance between observations i and j, and d_{ik} is the distance to the kth nearest observation to observation i.

2.1 Cross-Validation

Theory does not provide guidance as to how many observations should receive positive weights in the local regression and must be determined by the researcher for the problem at hand. Typically, the k parameter is determined by minimizing a cross-validation metric. This research aims to systematically compare the performance of four different cross-validation metrics used in LWR research.

1. Leave-One-Out Cross-Validation
2. Generalized Cross-Validation
3. Standardized Cross-Validation
4. Akaike Information Criterion

Does choosing the optimal number of observations to include in the LWR through these four strategies yield similar results? If there are differences, are there patterns in how they are different?

2.2 Leave-One Out Cross-Validation

$$\sum (y - \hat{y}_{-i})^2 \tag{6}$$

2.3 Generalized Cross-Validation Score

$$n * \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{(n - v_1)^2}, \tag{7}$$

where y_i is the dependent variable value, $\hat{y}_i$ is the predicted dependent variable value for observation i, and v_1 is the "effective number of model parameters."[1] In an LWR model, the number of parameters to be estimated is no longer equal to the number of variables included because we allow the regression

[1] v_1 =tr($\mathbf{S}$), where the matrix $\mathbf{S}$ is the "hat matrix" which maps y onto $\hat{y}$,

$$\hat{y} = \mathbf{S}y,$$

and each row of $\mathbf{S}$, r_i is given by:

$$r_i = X_i (X^T W(location_i) X)^{-1} X^T W(location_i).$$

coefficients to vary over space. The GCV score calculates the "effective" number of model parameters, v_1, and penalizes the model for increasing the number of parameters without sufficient reduction in model accuracy. Taking the square root of Equation (7) and rearranging yields,

$$\sqrt{GCV} = \sqrt{\frac{n}{n - v_1}} \sqrt{\frac{\text{Sum of Squared Residuals}}{n - v_1}}, \tag{8}$$

which approaches $\hat{\sigma}$ as v_1 approaches m for large n. Henceforth, throughout the paper we report the square root of (7) because of its similarity to $\hat{\sigma}$.

2.4 Row Standardized Cross-Validation

Something about Paez, who wanted a CV score that was more robust to outliers.

$$\frac{\sum (y - y_{-i})^2}{\sum y} \tag{9}$$

2.5 Akaike Information Criterion

$$2 * n * ln(\hat{\sigma}) + n * ln(2 * \pi) + n * \frac{n + v_1}{n - 2 - v_1} \tag{10}$$