Geographic Weighted Regression

Dan N.

2024-03-07

Data

This analysis uses home sales data and spatial polygon data for Duval County to demonstrate geographic weighted regression. This approach is used when there is variation due to geographic location. Coefficients are allowed to vary accordingly in their space; this is determined by a kernel that computes a local regression model at a central space, giving more weight to observations closer to the center and less weight to observations in the periphial. In contrast, an OLS model will output regression coefficents that are assumed to be static across the entire sample space.

Import Feature Data

The first 5 rows of feature data are shown below. Home sale price (inf_price) adjusted for inflation to the year2020 will be our dependent variable. Variables prefixed with *D* are distances to a particular neighborhood feature (i.e. DEDUCATION is distance to schools).

```
kable(head(sale.data2, n = 5), format = "html") %>%
  kable_styling(bootstrap_options = "striped", full_width = F)
```

Parcelid	inf_price	DRECREATION	DSUPERMARKET	DRESTAURANT	DENTERNAINTMENT	DHEAVYINDUSTRIAL	DEDUCATION	DOPENSPACE	DTRAN
0272760000R	25312	10843.782	9000.995	8974.337	8403.069	6651.202	7629.827	7277.925	95
0272760010R	81992	9491.421	7680.179	7626.712	7964.223	6038.195	7366.604	6917.978	99
0272770000R	106099	10795.694	8934.807	8797.160	8319.761	6647.899	7608.048	7194.450	95
0272800000R	63758	9668.117	8452.373	7925.953	8094.325	6250.699	7497.291	6933.084	92
0272870000R	67049	9638.318	8441.248	7907.433	8081.578	6247.630	7481.600	6926.124	92

Import Spatial Data

Next, we can import the spatial data and merge it with our feature data. We will also subset to a small neighborhood for computational purposes.

```
duval.area <- as(st_read("d:/lisc/GWR Oct 2021/map.shx"), "Spatial")
```

```
## Reading layer `map' from data source `d:\Lisc\GWR Oct 2021\map.shx' using driver `ESRI Shapefile'
## Simple feature collection with 12564 features and 25 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 425658.9 ymin: 2152585 xmax: 462415.8 ymax: 2197913
## Projected CRS: NAD83 / Florida East (ftUS)
```

```
duval.sales <- merge(duval.area, sale.data2, by.x="STRAP", by.y="Parcelid", all=F)
riverside <- subset(duval.sales, Area %in% c("Riverside"))
```

Model Building

In the first step, we need to figure out the kernel which will determine how many observations will be used in each model. After, we can fit the model and graph results.

Bandwith Selection

If we do not have a fixed kernel value in mind, we can allow R to determine using an adaptive approach (adapt = T).

```
# riverside@data[!complete.cases(riverside@data), ]
gwr.kernel <- gwr.sel(inf_price ~ DEDUCATION + DHEAVYINDUSTRIAL + DSUPERMARKET,
                  data=riverside,
                  adapt=T)
```

```
## Adaptive q: 0.381966 CV score: 1.262266e+13
## Adaptive q: 0.618034 CV score: 1.295813e+13
## Adaptive q: 0.236068 CV score: 1.221239e+13
## Adaptive q: 0.145898 CV score: 1.173653e+13
## Adaptive q: 0.09016994 CV score: 1.115993e+13
## Adaptive q: 0.05572809 CV score: 1.049057e+13
## Adaptive q: 0.03444185 CV score: 9.930893e+12
## Adaptive q: 0.02128624 CV score: 9.542116e+12
## Adaptive q: 0.01315562 CV score: 9.222354e+12
## Adaptive q: 0.008130619 CV score: 9.009788e+12
## Adaptive q: 0.005024999 CV score: 9.463262e+12
## Adaptive q: 0.009729937 CV score: 9.048531e+12
## Adaptive q: 0.006634959 CV score: 9.06894e+12
## Adaptive q: 0.00834246 CV score: 9.013044e+12
## Adaptive q: 0.007559327 CV score: 9.03404e+12
## Adaptive q: 0.008171309 CV score: 9.010349e+12
## Adaptive q: 0.007912405 CV score: 9.017529e+12
## Adaptive q: 0.008047268 CV score: 9.01247e+12
## Adaptive q: 0.008089929 CV score: 9.010925e+12
## Adaptive q: 0.008130619 CV score: 9.009788e+12
```

Fit the Model

Now we can fit the model using the kernel found by R, or we can tune the model ourselves (lets try 0.05).

```
gwr.mod <- gwr(inf_price ~ DEDUCATION + DHEAVYINDUSTRIAL + DSUPERMARKET,
               data=riverside,
               adapt=0.05, # adapt=gwr.kernel
               hatmatrix=T, se.fit=T)
gwr.mod
```

```
## Call:
## gwr(formula = inf_price ~ DEDUCATION + DHEAVYINDUSTRIAL + DSUPERMARKET,
##     data = riverside, adapt = 0.05, hatmatrix = T, se.fit = T)
## Kernel function: gwr.Gauss
## Adaptive quantile: 0.05 (about 67 of 1354 data points)
## Summary of GWR coefficient estimates at data points:
##                        Min.     1st Qu.      Median     3rd Qu.        Max.
## X.Intercept.     -3.7087e+06 -1.2742e+06 -5.6591e+05  1.2872e+05  1.6410e+06
## DEDUCATION       -3.5246e+02 -1.2371e+02  5.6881e+01  2.2957e+02  6.0112e+02
## DHEAVYINDUSTRIAL -3.1793e+02 -2.0897e+02 -5.7271e+01  7.3778e+01  2.9140e+02
## DSUPERMARKET     -1.3746e+02  2.3393e+01  1.2011e+02  1.5018e+02  2.8552e+02
##                      Global
## X.Intercept.     -629219.622
## DEDUCATION           -19.714
## DHEAVYINDUSTRIAL      50.211
## DSUPERMARKET          66.943
## Number of data points: 1354
## Effective number of parameters (residual: 2traceS - traceS'S): 49.42678
## Effective degrees of freedom (residual: 2traceS - traceS'S): 1304.573
## Sigma (residual: 2traceS - traceS'S): 86566.03
## Effective number of parameters (model: traceS): 34.6396
## Effective degrees of freedom (model: traceS): 1319.36
## Sigma (model: traceS): 86079.55
## Sigma (ML): 84971.33
## AICc (GWR p. 61, eq 2.33; p. 96, eq. 4.21): 34651.73
## AIC (GWR p. 96, eq. 4.22): 34613.11
## Residual sum of squares: 9.776051e+12
## Quasi-global R2: 0.3961435
```

We used a kernel function of 0.05 which resulted in 67 observations per local model. The min, q1, med, q3 and max show us how the local models varied in their coefficient estimation. For example one model found that, for every one unit increase in distance to schools, there is a $352 decrease (Min column) in home sale price, while another found a $601 increase (Max column). Note the Global coefficients column in the output. If we run an OLS model, these would be the coefficients.While there is room for improvement with this model, we can gain an understanding of how to apply these techniques on a well defined research question.

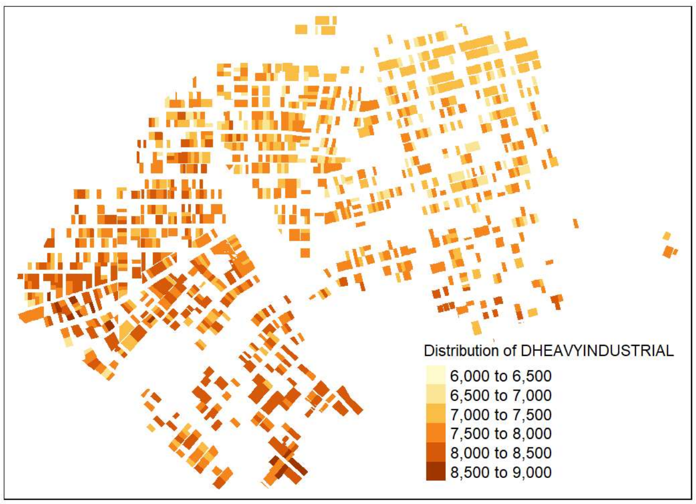
Plot Results

```
results <- as.data.frame(gwr.mod$SDF)
gwr.map <- cbind(riverside, as.matrix(results))
gwr.map2 <- st_as_sf(gwr.map)
```

Now we can see the plots for each home sale in this neighborhood. Notice how the distance to heavy industrial areas is smaller in the upper right hand corner of our map.

```
map1 <- tm_shape(gwr.map2) +
  tm_fill("DHEAVYINDUSTRIAL",
          n = 6,
          title = "Distribution of DHEAVYINDUSTRIAL") +
  tm_layout(frame = T,
            legend.text.size = 0.75,
            legend.title.size = 0.85)

map1
```



We can also get fancier with the maps by adding a street layer!

```
tmap_mode("view")
map2 <- tm_shape(gwr.map2) +
  tm_basemap(c(StreetMap = "OpenStreetMap")) +
  tm_fill("DHEAVYINDUSTRIAL.1",
          n = 6,
          title = "Coef. of DHEAVYINDUSTRIAL")  +
  tm_layout(frame = T,
            legend.text.size = 0.75,
            legend.title.size = 0.85)
map2
```