

DESCRIBING THE GWR-GLMNET PROCESS

WESLEY BROOKS

1. PRELIMINARIES

This document describes the method that I've been using to produce Geographically Weighted Regression (GWR) models that use the LASSO for variable selection and shrinkage. It covers the cases of gaussian and binomial data.

2. DATA

Data should include the location of each observation in terms of an X (longitude) and Y (latitude) variable. If they correspond to actual longitude and latitude, then it is possible to call the functions with the option `longlat=TRUE`, which will correct for the the Earth's curvature.

Otherwise, the data is pretty typical - a row for each observation and a column for each variable.

3. GWR

The basic operation of GWR is to build a regression model at each of a set of pre-specified locations. At each model location, all of the observations in the data set are weighted based on their distance from the model location (the weights are uniquely specified by the combination of a kernel function and a bandwidth). The weighted observations are then used to build a regression model that is only meaningful at that location. The regression model can be quite generic - for gaussian data something like R's `lm` function is used, and for the binomial case, something like the `glm` function is used. Prior weights can be specified in case over-dispersion (or heteroskedasticity) is expected. This came into play with the poverty data: each county's entry was weighted by its population.

3.1. **Kernels.** At this time there are two kernel functions working: the bisquare kernel:

$$W(\text{dist}, \text{bw}) = (1 - (\frac{\text{dist}}{\text{bw}})^2)^2$$

...And the gaussian kernel:

$$W(\text{dist}, \text{bw}) = \phi(\frac{\text{dist}}{\text{bw}})$$

3.2. Bandwidth selection. The bandwidth is selected by minimizing the sum of absolute cross-validation (CV) errors from the model-fitting. Each of the unique observation locations is used as a model location, and the observations from that point are reserved as the test set. The model is built over the other data and used to predict the test set. The sum of the absolute errors from this prediction step are added to the total. The `optimize` function in R uses a golden section search to find the minimum of the total absolute CV error.

3.3. Gaussian case (linear model). test

3.4. Generalized linear model. test

4. VARIABLE SELECTION

Differences between gaussian and binomial cases. The R function `lars` is used to fit a LASSO regression model to gaussian data; `glmnet` is used to fit the model for binomial data. The function `lars` allows the one to specify how the LASSO penalty is specified: the choices are to specify the penalty itself (`mode=lambda`), the fraction of the total LASSO path (`mode=fraction`), or the number of steps on the least-angle regression (LAR) path (`mode=step`). I've always used `mode=step` for gaussian models. The only choice for a `glmnet` model is `mode=lambda` so that's what I use. Also, since `lars` doesn't allow the user to specify weights, the first step in making each `gwLars` model is to multiply the design matrix and the vector of observed outputs by the diagonal matrix of the square roots of the geographical weights. In the binomial case, the `glmnet` function allows one to specify weights, so that is how the geographic weights are incorporated. The matrix multiplication in the gaussian case happens before the centering and scaling for the adaptive LASSO, discussed below.

When each (location-specific) model is built, each column of the design matrix is centered and scaled to have unit norm. Then the basic regression step (`lm` or `glm`) is used (with the scaled-to-unit-norm design matrix) to get the adaptive weights (the adaptive weight for each predictor variable is its coefficient in the basic model); each column is multiplied by its adaptive weight. This centered and adaptively-scaled design matrix is fed into the `lars` or `glmnet` function. These models also require the user to choose a tuning parameter. Here, the tuning parameter is chosen in the cross-validation step: the tuning parameter at each model location is set to the choice that minimizes the total absolute cross-validation error at that location (the "total" CV error refers to the possibility that there are multiple observations at that location).

Remember, the tuning parameters are different between the gaussian and binomial cases: in the gaussian case, the tuning parameter is the number of steps along the LAR path, which ranges from zero to the number of predictor variables. In the binomial case, the tuning parameter is the LASSO penalty (i.e., λ in the penalty term $\lambda \times \sum_{i=1}^p \beta_i$). I've been allowing λ to range from zero to five in steps of 0.001.

4.1. Gaussian case.

4.2. **Generalized linear model case.**