lagr: Local Adaptive Grouped Regularization in R

by Wesley Brooks

Abstract An abstract of less than 150 words.

Whereas the coefficients in traditional linear regression are scalar constants, the coefficients in a varying coefficient regression (VCR) model are functions - often smooth functions - of some effect-modifying parameter (**??**). A basic fact of varying coefficient regression (VCR) models is that the coefficients vary over the model's domain. It is natural, then, to allow that a coefficient function may be nonzero in part of the domain and exactly zero elsewhere. To date, methods of estimating VCR models could allow only global variable selection, where a variable is either in or out of the model over the entire domain. The method of local adaptive grouped regularization (LAGR) is an estimation method for VCR models that allows for local variable selection (Brooks et al., 2014).

Some R packages that estimate VCR models are **spgwr**, **mgcv**, **np**, **dlm**.

This paper introduces **lagr**, an R package for estimating a VCR model via LAGR.

Method

Varying coefficient regression

The response $Y(s)$, covariates $\boldsymbol{X}(s)$, and coefficients $\boldsymbol{\beta}(s)$ in a VCR model are indexed by the location parameter s. Assume n observations of the response and the covariates at locations $s_1, \dots, s_n$ and let $y_i = y(s_i)$, $\boldsymbol{x}_i = \boldsymbol{x}(s_i)$, and $\boldsymbol{\beta}_i = \boldsymbol{\beta}(s_i)$. The generalized linear model with varying coefficients is written

$$E[y(s)|\boldsymbol{x}(s)] = \mu(s) \tag{1}$$
$$\eta(s) = g(\mu(s)) = \boldsymbol{x}'(s)\boldsymbol{\beta}(s) \tag{2}$$
$$\text{var}[y(s)|\boldsymbol{x}(s)] = \phi V(\mu(s)) \tag{3}$$

where $g(\cdot)$, $V(\cdot)$, and ϕ are, respectively, the link function, variance function, and dispersion parameter of the response family. For simplicity of notation, assume the canonical link function (**?**).

Local polynomial regression

Local adaptive grouped regularization is in the class of local polynomial regression methods. At each estimation location s_i, the coefficient functions are approximated by Taylor's expansion as locally linear functions of the location parameter

$$\boldsymbol{\beta}(s) = \boldsymbol{\beta}(s_i) + \nabla\boldsymbol{\beta}(s_i)(s - s_i) + o(|s - s_i|) \tag{4}$$

and because the approximation is local, $|s - s_i|$ is small.

Weights are calculated according to a kernel function that gives more weight to nearby observations than to distant ones. The popular Epanechnikov kernel is defined as (**?**):

$$K(x) = (3/4)(1 - x^2) \text{ if } x < 1, \text{ and } 0 \text{ otherwise.} \tag{5}$$

Let q be the dimension of the location parameter (e.g., $q = 1$ for a coefficients that vary with time). For estimating the coefficient functions at location s_i with kernel function $K(\cdot)$ and bandwidth h, the weights are $w_{ij} = h^{-q} K(|s_i - s_j|)$ for $j = 1, \ldots, n$.

Local penalized log likelihood

Coefficients are estimated locally by maximizing the penalized local likelihood $\mathcal{J}(\cdot)$.

$$\begin{aligned} \mathcal{J}(\boldsymbol{\beta}_i) &= \ell(\boldsymbol{\beta}_i) - \mathcal{P}(\boldsymbol{\beta}_i) & (6)\\ &= \sum_{j=1}^{n} w_{ij} Q(\boldsymbol{\beta}_i, Y_j) + \sum_{k=1}^{p} \phi_k \|\boldsymbol{\beta}_{ik}\| & (7) \end{aligned}$$

Regularization

$$\ell_i = \sum_{j=1}^{n} w_{ij} \log L(\boldsymbol{\beta}_i, Y_j) + \lambda_i \sum_{k=1}^{p} \phi_k \|\boldsymbol{\beta}_k\| \tag{8}$$

Local degrees of freedom

In the context of Stein's unbiased risk estimation, the degrees of freedom used in fitting a model are defined as

$$df = \sum_{i=1}^{n} \frac{\text{cov}(y_i, \hat{y}_i)}{\sigma^2} \tag{9}$$

where y_i is the observed value of the ith response, $\hat{y}_i$ is the corresponding fitted value, and σ^2 is the exponential family dispersion parameter (Efron, 1986). The degrees of freedom for an adaptive group lasso estimate (Vaiter et al., 2012).

Model averaging

A weighted average of the candidate models is used to acknowledge uncertainty in the variable selection.

$$\begin{aligned} \hat{g} &= \operatorname{argmin} w_j g_j & (10)\\ w_j &\geq 0 \forall j \in 1, \ldots, m & (11) \end{aligned}$$

Bandwidth parameter

In order to estimate a VCR model by a local polynomial method like LAGR, we need to set the bandwidth parameter. In the `lagr` function, the bandwidth can be specified in terms of distance or k-nearest neighbors. The k-nearest neighbors method is a type of adaptive bandwidth that specifies a value for $\sum_{j=1}^{n} w_{ij}/n$, while the distance method specifies an identical h .

To estimate the bandwidth parameter, we profile it with our favorite model selection criterion. The optimal value of the bandwidth parameter is the one that minimizes the selection criterion. However, selecting this bandwidth and treating it as known truth would introduce model-selection bias (**?**). We average over the implicit distribution of the bandwidth parameter based on the profile AIC that was calculated in

Total degrees of freedom

In the typical local polynomial regression model, the degrees of freedom are calculated as the trace of the projection matrix. Because LAGR is an $\mathcal{L}_1$ regularization method, though, it is nonlinear and thus generates no projection matrix. Recall the definition of degrees of freedom (**??**).

For estimating the bandwidth, each observation is estimated with a local model. Only observations colocated with the local model are affected by the local fit, so the total degrees of freedom are the sum of the colocated degrees of freedom from each local model. An unbiased(?) approximation of the colocated degrees of freedom is $df_i / sum_{j=1}^{n} w_{ij}$.

Code, explained

Package

The R package **lagr** (`https://github.com/wrbrooks/lagr`) . Its primary functions are `lagr` and `lagr.tune`. The `lagr` function estimates a model by LAGR, while the `lagr.tune` function estimates profiles the bandwidth parameter with respect to a model selection criterion (AIC, BIC, or GCV).

Estimation of a model

Estimation is carried out by blockwise coordinate descent. This is an iterative process and carrying out the coordinate descent algorithm in a compiled C++ library is considerably faster than doing so in R. The **Rcpp11** is used to integrate C++ code into the **lagr** package.

The model weighting is a constrained quadratic programming problem. The solution is found via the **quadprog** package.

Response family

R provides several `family` objects representing exponential family distributions (e.g., `gaussian()`, `binomial()`, `poisson()`). In the **lagr** package, these objects supply the link and variance functions for fitting the local GLM models. Because the **Rcpp11** package provides seamless R and C++ integration we can use objects of type `function` within the C++ code that can represent either R or C++ functions. This capability allows us to call the `link()` and `varfun()` functions of any `family` object. As a result, the user can write their own `family` object in either R or C/C++ and use it as the response distribution of a VCR model estimated by LAGR.

Plotting

The function `lagr.plot` is used to plot a `lagr` object. In the case of a one-dimensional location parameter (e.g., time), the function produces line plots. If the location parameter is two-dimensional, then the plotting behavior depends on how the data was specified. If data was provided as a `SpatialPolygonDataFrame` (defined in package **sp**), then the spatial polygons are used to produce a choropleth (Figure XX). Otherwise the estimation locations are used to generate a Voronoi diagram (**?**).

The content of the plot depends on two parameters. The parameter `type` is one of `raw`, `coef`, or `is.zero`; these respectively output the raw covariate, the estimated coefficient, or the confidence that a coefficient is locally zero. The parameter `target` indicates which covariate is plotted.

Examples

Boston house price data

Here is a model for how some demographic covariates are related to house prices in Boston, MA, based on data from the 1970 U.S. Census (**?**). The data is available in the R package **spgwr**; the variables are listed in Table 1.

Name	Description
LON, LAT	Longitude and latitude of each county's center of mass
MEDV	Median home price
CRIM	Per-capita crime rate
RM	Mean number of rooms per dwelling
RAD	An index of access to radial roads
TAX	Property tax per $10,000 of value
LSTAT	Percentage of residents considered "Lower status"

Table 1: Listing of the variables in the Boston house price data set. The response for our model is MEDV and the coordinates are given by LON and LAT.

Here, we import the data and estimate the bandwidth by AIC, then fit a model using the estimated bandwidth:

```
library(lagr)
data(boston)

bw = lagr(MEDV~CRIM+RM+RAD+TAX+LSTAT, data=boston,
    bwselect.method="AIC", longlat=TRUE, bw.type="knn",
    kernel=epanechnikov, varselect.method="wAIC")
model = lagr(MEDV~CRIM+RM+RAD+TAX+LSTAT, data=boston, bw=bw,
    longlat=TRUE, kernel=epanechnikov, varselect.method="wAIC")

summary(model)
plot(model, target="RM", type="coef", id="ID")
```

Summary

Bibliography

W. Brooks, J. Zhu, and Z. Lu. Local adaptive grouped regularization and its oracle properties for varying coefficient regression. In prep, 2014. [p]

B. Efron. How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, pages 461–470, 1986. [p]

A. Fotheringham, C. Brunsdon, and M. Charlton. *Geographically Weighted Regression: The Analysis of Spatially Varying Relationships*. Wiley, West Sussex, England, 2002. [p]

S. Vaiter, C. Deledalle, G. Peyre, J. Fadili, and C. Dossal. The degrees of freedom of the group lasso for a general design. *arXiv*, 2012. [p]

Wesley Brooks
Department of Statistics, University of Wisconsin-Madison
1300 University Ave. Madison, WI 53706
USA wrbrooks@uwalumni.com