lagr: Local Adaptive Grouped Regularization in R

Whereas the coefficients in traditional linear regression are constant, the coefficients in a varying coefficient regression (VCR) model are functions - often smooth functions - of some effect-modifying parameter (Cleveland and Grosse, 1991; Hastie and Tibshirani, 1993). 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. This paper introduces **lagr**, an R package for estimating a VCR model via LAGR. The method of LAGR is described, some features of the software are explained, and the results are illustrated with an example application.

Method

A brief overview of LAGR follows. It can be shown that under some mild conditions the method achieves an oracle property, asymptotically identifying exactly the locally relevant covariates and estimating them as accurately as if their identities were known in advance (Brooks et al., 2014).

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 $s_1, \ldots, s_n$ 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

$$\begin{aligned}
E[Y(s)|\boldsymbol{X}(s)] &= \mu(s) \\
\eta(s) &= g(\mu(s)) = \boldsymbol{X}'(s)\boldsymbol{\beta}(s) \\
\text{var}[Y(s)|\boldsymbol{X}(s)] &= \phi V(\mu(s))
\end{aligned}$$

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 (McCullagh and Nelder, 1989).

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

$$\boldsymbol{\beta}(s) = \boldsymbol{\beta}(s_i) + \nabla\boldsymbol{\beta}(s_i)(s - s_i) + o(|s - s_i|). \tag{1}$$

The locally linear approximation is implemented by augmenting the matrix $\boldsymbol{X}(s)$ with interactions between the covariates and the location parameter. Letting $\boldsymbol{Z}(s) = (\boldsymbol{X}_i \; (s - s_i) \cdot \boldsymbol{X}_i)$ and $\boldsymbol{\zeta}(s) = (\boldsymbol{\beta}(s)^T, \; \nabla \boldsymbol{\beta}(s)^T)^T$, the linear predictor of the local GLM is $\hat{\eta}_i = \boldsymbol{Z}_i \boldsymbol{\zeta}_i$.

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 (Samiuddin and el Sayyad, 1990):

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

Because the kernel gives zero weight to observations farther than h from the estimation location, the error term in (1) is small.

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

Penalized local quasi-likelihood

The quasi-likelihood $Q(\mu, Y)$ is an approximation to the full log likelihood. Its derivative the quasi-score function is $q(\mu, Y) = (y - \mu)\{V(\mu)\}^{-1}$, which is a function of the linear predictor η through the link and variance functions. The method of LAGR estimates the coefficients of a VCR model by maximizing the penalized local quasi-likelihood $\mathcal{J}(\boldsymbol{\beta}_i; \lambda_i)$, which incorporates the local kernel weights and an adaptive group lasso penalty.

The penalized local quasi-likelihood is

$$\begin{aligned} \mathcal{J}(\boldsymbol{\beta}_i; \lambda_i) &= \ell(\boldsymbol{\beta}_i) - \mathcal{P}(\boldsymbol{\beta}_i; \lambda_i) \\ &= \sum_{j=1}^{n} w_{ij} Q(\hat{\mu}_i, Y_j) - \lambda_i \sum_{k=1}^{p} \gamma_{ik} \|\boldsymbol{\zeta}_{i(k)}\| \end{aligned}$$

where $\|\boldsymbol{\zeta}_{i(k)}\| = \{(\beta_k(s_i), \; \nabla \beta_k(s_i))^T \, (\beta_k(s_i), \; \nabla \beta_k(s_i))\}^{1/2}$ is the norm of the kth coefficient group, which consists of the kth coefficient and its interaction with the location parameter. The penalty on the norm of the kth covariate grouping is a combination of $\gamma_{ik} = \|\tilde{\zeta}_k(s_i)\|^{-1}$, an adaptive penalty based on the unpenalized local coefficients $\tilde{\zeta}_k(s_i)$; and λ_i, a local tuning parameter estimated by the Akaike Information Criterion (AIC) (Akaike, 1973).

Package

The R package **lagr** (`https://github.com/wrbrooks/lagr`) estimates a VCR model via LAGR. The primary functions of the package 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.

Coefficient estimation

Estimation is carried out by blockwise coordinate descent. Each block is a covariate group, consisting of a raw covariate and its interaction with the location parameter. Coordinate descent is an iterative algorithm; a compiled library can achieve a significant speedup over pure R. The **lagr** package makes use of a coordinate descent algorithm written in C++ and integrated via the **Rcpp11** package.

By default, a local model is estimated at each observation location. Since each local model is estimated independently of the others, estimation of a VCR model by LAGR is an embarrassingly parallel problem. The package is written to take advantage of multicore parallelism via the R packages **foreach** and **doMC**.

Bandwidth parameter estimation

Two types of bandwidth parameter are supported by the **lagr** package. The user may specify the bandwidth directly, in which case the `bw` function argument is used as h in the kernel function. Alternatively, k-nearest neighbors (KNN) is a type of adaptive bandwidth that allows h to differ among the local models. Specifically, if a KNN bandwidth is specified as α, then the bandwidth for the local model at s_i is set so that $\alpha = \sum_{j=1}^{n} w_{ij}$.

Both types of bandwidth provide a tuning parameter than must be estimated. In the case of the raw bandwidth, that parameter is h, and in the case of k-nearest-neighbors, that parameter is α. In either case, the function `lagr.tune` calls R's `optim` function to find the bandwidth parameter that minimizes a selection criterion (AIC, BIC, or GCV).

Response family

R provides several `family` objects representing exponential family distributions (e.g., `gaussian()`, `binomial()`, `poisson()`). These objects supply the link and variance functions for estimating the local models. Because the **Rcpp11** package provides seamless R and C++ integration we can use objects of type `Function` within the C++ code to represent either R or C++ functions. This capability allows us to call the `link()` and `varfun()` functions of any `family` object from within compiled C++ code, even a user-provided `family` object written in pure R.

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 poly-

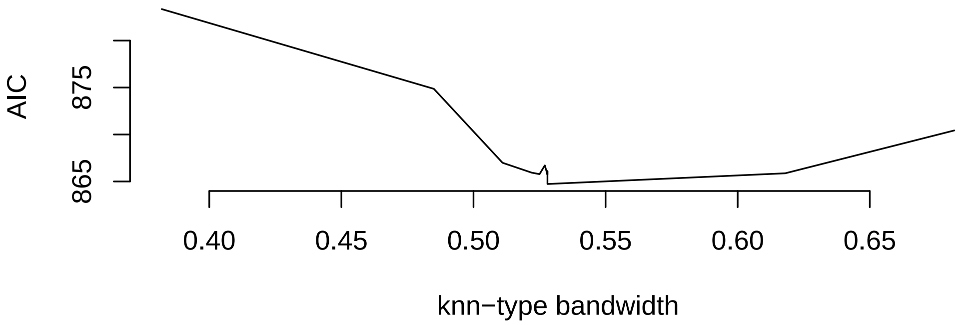

Figure 1: Profile AIC for estimating the bandwidth parameter in the Georgia educational attainment model. The AIC is minimized for k-nearest-neighbors bandwidth parameter 0.525.

gons are used to produce a choropleth (Figure 2). 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.

Example - Georgia educational attainment data

We illustrate the functionality of **lagr** with an example. The dataset `georgia` is attached to the package **lagr**. Here we produce a model for how some demographic covariates are related to educational attainment in Georgia, based on data from the 1990 U.S. Census (Fotheringham et al., 2002). The variables in the data set are listed in Table 1. In this example, the response is the PctBach, the percentage of residents in each county who have at least a bachelor's degree.

Name	Description
Latitude, Longitude	Latitude and longitude of each county's center of mass
PctBach	Percentage of residents with at least a bachelor's degree
PctRural	Percentage of residents living in a rural area
PctEld	Percentage of residents age 65 or older
PctFB	Percentage of residents who are foreign-born
PctPov	Percentage of residents living in poverty

Table 1: Listing of the variables for each county of Georgia in the example data set. The response for our model is PctBach and the coordinates are given by Longitude and Latitude.

In the code listed below, we import the data and estimate the bandwidth by AIC, then fit a model using the estimated bandwidth. The profile AIC of

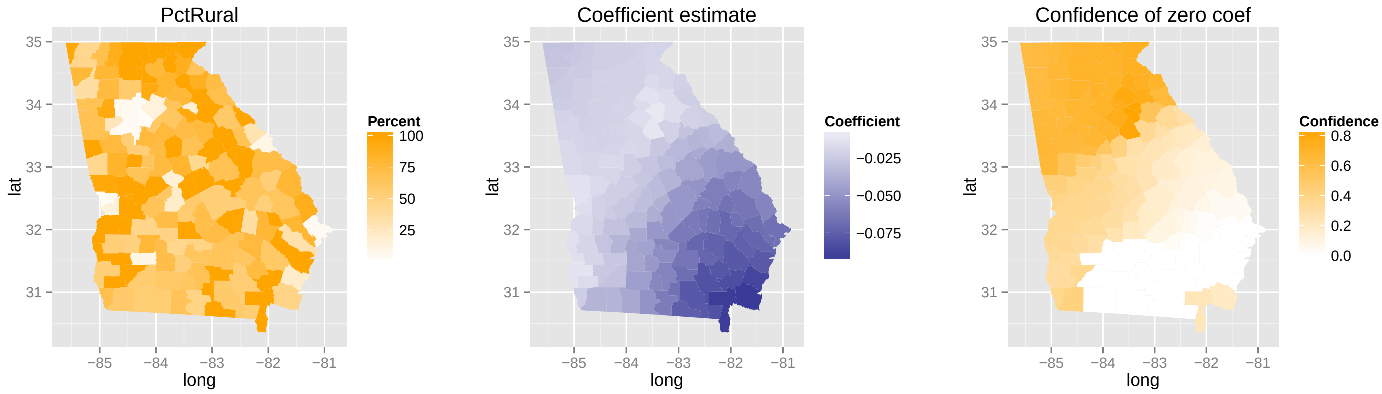

Figure 2: An example produced by the `plot.lagr()` function. Plots are based on a model for educational attainment by county in Georgia. The response is the PctBach, the percentage of residents in each county who have at least a bachelor's degree. Left: PctRural, the percentage of people in each county who live in a rural area. Middle: estimated coefficient of PctRural. Right: confidence that the coefficient of PctRural is exactly zero.

the bandwidth estimation is shown in Figure 1, showing that the optimal *k*-nearest-neighbors bandwidth for the example is 0.525.

```
library(lagr)
data(georgia)

bw = lagr.tune(PctBach~PctRural+PctEld+PctFB+PctPov, data=georgia,
    bw.type='knn', bwselect.method='AIC', longlat=TRUE,
    kernel=epanechnikov)

model = lagr(PctBach~PctRural+PctEld+PctFB+PctPov, data=georgia, bw=bw,
     longlat=TRUE, kernel=epanechnikov, varselect.method="AIC")

plot(model, target="PctRural", type="coef")
```

Summary

This paper has outlined the varying coefficient regression model, and shown how local adaptive grouped regularization is used for local variable selection and estimation. The R package **lagr** was introduced and illustrated by estimating a model for educational attainment in Georgia. In particular, the `lagr.tune` function was used to estimate the model's bandwidth parameter, the `lagr` function was used to fit a model using the estimated bandwidth, and the `plot.lagr` function was used to produce choropleths of the results of analysis.

Bibliography

H. Akaike. Information theory and an extension of the maximum likelihood principle. In B. Petrov and F. Csaki, editors, *2nd International Symposium on Information Theory*, pages 267–281, 1973. [p2]

W. Brooks, J. Zhu, and Z. Lu. Local adaptive grouped regularization and its oracle properties for varying coefficient regression. In prep, 2014. [p1]

W. Cleveland and E. Grosse. Local regression models. In J. Chambers and T. Hastie, editors, *Statistical models in S*. Wadsworth and Brooks/Cole, 1991. [p1]

A. Fotheringham, C. Brunsdon, and M. Charlton. *Geographically Weighted Regression: The Analysis of Spatially Varying Relationships*. Wiley, West Sussex, England, 2002. [p4]

T. Hastie and R. Tibshirani. Varying-coefficient models. *Journal of the Royal Statistical Society Series B*, 55:757–796, 1993. ISSN 00359246. URL http://www.jstor.org/stable/2345993. [p1]

P. McCullagh and J. Nelder. *Generalized Linear Models, Second Edition*. Taylor and Francis, 1989. ISBN 9780412317606. URL http://books.google.com/books?id=h9kFH2_FfBkC. [p1]

M. Samiuddin and G. M. el Sayyad. On nonparametric kernel density estimates. *Biometrika*, 77:865–874, 1990. [p2]