Scalable geographically weighted regression for big data

D. Murakami*[1], N. Tsutsumida[2], T. Yoshida[3], T. Nakaya[4], and B. Lu[5]

[1]Department of Data Science, Institute of Mathematics Statistics, Tachikawa, Japan
[2]Graduate School of Global Environmental Studies, Kyoto University, Kyoto, Japan
[3]Center for Global Environmental Studies, National Institute for Environmental Studies, Tsukuba, Japan
[4]Graduate School of Environmental Studies, Tohoku University, Sendai, Japan
[5]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China

*Email: dmuraka@ism.ac.jp

Abstract

This study develops a scalable GWR (ScaGWR) for large samples. Unlike existing GWR algorithms, the ScaGWR achieves a linear time estimation through pre-compression of large matrices and vectors before the leave-one-out cross validation (LOOCV), which is the heaviest part in the standard GWR. Our proposed algorithm enables us to estimate a regularized GWR from one million samples even without parallelization. The R code is available in the R package **scgwr**. Besides, for faster computation, the code is embedded into C++ code and implemented in the **GWmodel** package.

Keywords: geographically weighted regression, large spatial data, fast computation, scalability, pre-processing.

1. Introduction

Geographically weighted regression (GWR; Brunsdon et al., 1996) and other spatial modeling approaches for big data is increasingly important. Although computationally efficient GWR algorithms have been developed (e.g., Li et al., 2019), the computational complexity of existing GWR algorithms is $O(N^2 logN)$ at best, meaning that the computational burden grows in a quasi-quadratic manner with sample size N. Unfortunately, modelling in $O(N^2 logN)$ time is not fast enough as linear-time modelling (i.e., $O(N)$) is considered as needed in the era of big data in relevant fields (see, Liu et al., 2018). Against this backdrop, we develop a scalable GWR (ScaGWR), a linear-time GWR for large spatial dataset.

2. Geographically weighted regression (GWR)

The basic GWR describes the explained variable y_i at i-th sample site using the following model:

$$y_i = \sum_{k=1}^{K} x_{i,k}\beta_{i,k} + \varepsilon_i \quad \varepsilon_i \sim N(0, \sigma^2), \qquad \text{Equation 1}$$

where $x_{i,k}$ is the k-th covariate, and $\beta_{i,k}$ is the regression coefficient for the k-th covariate at i-th site. The estimator for the coefficients vector $\boldsymbol{\beta}_i = [\beta_{i,0}, \beta_{i,1} \cdots \beta_{i,K}]'$ is given by

$$\widehat{\boldsymbol{\beta}}_i = [\mathbf{X}'\mathbf{G}_i(b)\mathbf{X}]^{-1}\mathbf{X}'\mathbf{G}_i(b)\mathbf{y}, \qquad \text{Equation 2}$$

where $\mathbf{y}$ is a vector of the explained variables and $\mathbf{X}$ is a matrix of covariates. $\mathbf{G}_i(b)$ is a diagonal matrix whose j-th element is the geographical weight $g_{i,j}(b)$ for the j-th sample, and can be specified via a distance-decaying function. We use the exponential kernel later.

The bandwidth b determining the scale of the spatially varying coefficients can be optimized by a leave-one-out cross-validation (LOOCV) minimizing the cross-validation (CV) error (Eq.3).

$$\text{CV score} = \sum_{i=1}^{N}\left(y_i - \sum_{k=1}^{K} x_{i,k}\hat{\beta}_{-i,k}\right)^2 \qquad \text{Equation 3}$$

$\hat{\beta}_{-i,k}$ represents the coefficient at the i-th sample site that is estimated using all the other samples. This estimator yields

$$\widehat{\boldsymbol{\beta}}_{-i} = [\mathbf{X}'\mathbf{G}_{-i}(b)\mathbf{X}]^{-1}\mathbf{X}'\mathbf{G}_{-i}(b)\mathbf{y}, \qquad \text{Equation 4}$$

where $\widehat{\boldsymbol{\beta}}_{-i} = [\hat{\beta}_{-i,0}, \hat{\beta}_{-i,1} \cdots \hat{\beta}_{-i,K}]'$. and $\mathbf{G}_{-i}(b)$ equals $\mathbf{G}_i(b)$ with its i-th diagonal being replaced with zero (zero weight is assigned on the i-th site).

Unfortunately, $\widehat{\boldsymbol{\beta}}_{-i}$ includes large matrices $\mathbf{X}, \mathbf{y}, \mathbf{G}_{-i}(b)$, whose size depend on N, that must be repeatedly evaluated for all $i \in \{1, \cdots, N\}$ during the LOOCV. This is the main reason for why GWR is slow for large N. Our objective is to overcome this limitation by eliminating the repeated evaluation of these large matrices.

3. Scalable geographically weighted regression (ScaGWR)

3.1. Model

GWR is accelerated if we let the large matrices out of the LOOCV iteration. However, it is not possible because b is not separable from $\mathbf{X}'\mathbf{G}_{-i}(b)\mathbf{X}$ and $\mathbf{X}'\mathbf{G}_{-i}(b)\mathbf{y}$ if the non-linear kernel is used as usual. We overcome this bottleneck using a linear multiscale kernel that allows us to linearly separate the internal parameters from the kernel function and pre-process the large matrices and vectors before the LOOCV.

Specifically, we use the following linear polynomial kernel:

$$\tilde{g}^0_{i,j}(\tilde{b}) = \sum_{p=1}^{P} \tilde{b}^p \, g_{i,j}(b_0)^{\frac{2^{P/2}}{2^p}}, \qquad \text{Equation 5}$$

where $g_{i,j}(b_0)$ is a base kernel with known bandwidth b_0. Later, $g_{i,j}(b_0)$ is defined by the exponential kernel $\exp(-d_{i,j}/b_0)$ where $d_{i,j}$ is the distance between sites i and j. Instead of the known b_0, we estimate $\tilde{b}$. For instance, when $P = 3$, Eq. (5) becomes $\sum_{p=1}^{3} \tilde{b}^p \, g_{i,j}(b_0)^{4/2^p}$ with known kernels $\{g_{i,j}(b_0)^2, g_{i,j}(b_0), g_{i,j}(b_0)^{1/2}\}$ and their coefficients $\{\tilde{b}^1, \tilde{b}^2, \tilde{b}^3\}$. Because $g_{i,j}(b_0)$ is a decreasing function, $g_{i,j}(b_0)^2$ describes a faster decay while $g_{i,j}(b_0)^{1/2}$ describes a slowest decay. By weighing these three kernels using the coefficients $\{\tilde{b}^1, \tilde{b}^2, \tilde{b}^3\}$, we can estimate the decay pattern behind samples. Figure 1 shows that the $\tilde{b}$ parameter substitutes the usual bandwidth parameter.

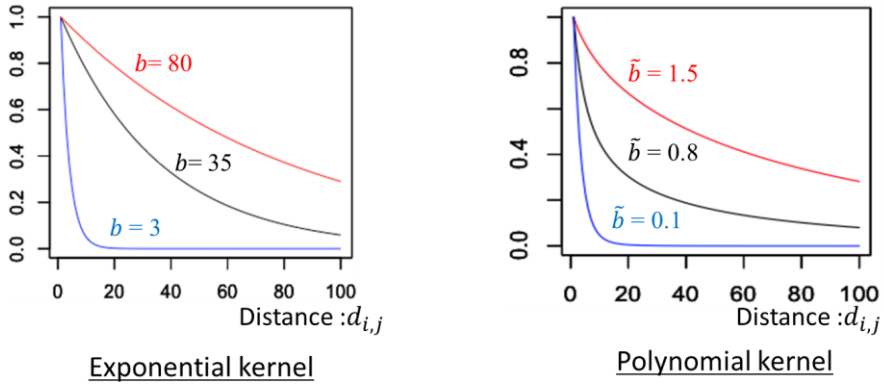

Figure 1. Examples of the exponential kernel (left) and the linear polynomial kernel with exponential base kernel and $b_0 = 20$ (right)

Unfortunately, evaluation of $\tilde{g}^0_{i,j}(\tilde{b})$ for all the sample pairs requires the computational complexity of $O(N^2)$ that is heavy for large samples. To lighten the cost, we use a linear multiscale kernel Eq.(6) that applies the local weight Eq.(5) to the Q-nearest neighbours and a global weight to all the samples:

$$\tilde{g}_{i,j}(\tilde{b},\alpha) = \alpha + \sum_{p=1}^{P} \tilde{b}^p \, g^{(Q)}_{i,j}(b_0)^{4/2^p}, \qquad \text{Equation 6}$$

where $g^{(Q)}_{i,j}(b_0) = g_{i,j}(b_0)$ for the Q-nearest neighbours while $g^{(Q)}_{i,j}(b_0) = 0$ for the other samples. $\tilde{b}$ and α are unknown parameters determining local and global weights, respectively.

3.2. LOOCV

Given Eq.(6), the leave-one-out coefficient estimator $\widehat{\boldsymbol{\beta}}_{-i}$ yields

$$\widehat{\boldsymbol{\beta}}_{-i} = \left[\mathbf{X}'\widetilde{\mathbf{G}}_{-i}(\tilde{b},\alpha)\mathbf{X}\right]^{-1}\mathbf{X}'\widetilde{\mathbf{G}}_{-i}(\tilde{b},\alpha)\mathbf{y}, \qquad \text{Equation 7}$$

where $\widetilde{\mathbf{G}}_{-i}(\tilde{b},\alpha)$ is a diagonal matrix whose j-th entry equals $\tilde{g}_{i,j}(\tilde{b},\alpha)$ if $j \neq i$, and 0 if $j = i$.

Owing to the linearity of our kernel, the (k,k')-th element of the $\mathbf{X}'\widetilde{\mathbf{G}}_{-i}(\tilde{b},\alpha)\mathbf{X}$ matrix is analytically obtained as $\sum_{j\neq i}\tilde{g}_{i,j}(\tilde{b},\alpha)x_{j,k}x_{j,k'}$; it is further expanded using Eq.(6) as follows:

$$\sum_{j\neq i}\sum_{p=1}^{P}\left(\alpha + \tilde{b}^p g^{(Q)}_{i,j}(b_0)^{4/2^p}\right)x_{j,k}x_{j,k'} = \alpha m^{(0)}_{-i,k,k'} + \sum_{p=1}^{P}\tilde{b}^p \, m^{(p)}_{-i,k,k'} \qquad \text{Equation 8}$$

where $m^{(0)}_{-i,k,k'} = \sum_{j\neq i} x_{j,k}x_{j,k'}$ and $m^{(p)}_{-i,k,k'} = \sum_{j\neq i} g^{(Q)}_{i,j}(b_0)^{4/2^p}x_{j,k}x_{j,k'}$. Likewise, the k-th element of $\mathbf{X}'\widetilde{\mathbf{G}}_{-i}(\tilde{b},\alpha)\mathbf{y}$ has the following expression:

$$\sum_{j\neq i}\tilde{g}_{i,j}(\tilde{b},\alpha)x_{j,k}x_{j,k'} = \alpha m^{(0)}_{-i,k,\mathbf{y}} + \sum_{p=1}^{P}\tilde{b}^p \, m^{(p)}_{-i,k,\mathbf{y}} \qquad \text{Equation 9}$$

where $m^{(0)}_{-i,k,\mathbf{y}} = \sum_{j\neq i} x_{j,k}y_j$ and $m^{(p)}_{-i,k,\mathbf{y}} = \sum_{j\neq i} g_{i,j}(b_0)^{4/2^p}x_{j,k}y_j$. The estimator $\widehat{\boldsymbol{\beta}}_{-i}$ is expanded using Eqs.(8) and (9) as follows:

$$\widehat{\boldsymbol{\beta}}_{-i} = \left[\alpha\mathbf{M}^{(0)}_{-i} + \sum_{p=1}^{P}\tilde{b}^p\mathbf{M}^{(p)}_{-i}\right]^{-1}\left[\alpha\mathbf{m}^{(0)}_{-i} + \sum_{p=1}^{P}\tilde{b}^p\mathbf{m}^{(p)}_{-i}\right]. \qquad \text{Equation 10}$$

where $\mathbf{M}_{-i}^{(0)}$ and $\mathbf{M}_{-i}^{(p)}$ are $K \times K$ matrices whose (k, k')-th elements are $m_{-i,k,k'}^{(0)}$ and $m_{-i,k,k'}^{(p)}$, respectively. $\mathbf{m}_{-i}^{(0)}$ and $\mathbf{m}_{-i}^{(p)}$ are $K \times 1$ vectors whose k-th elements are $m_{-i,k,\mathbf{y}}^{(0)}$ and $m_{i,k,\mathbf{y}}^{(p)}$.

Importantly, $\mathbf{M}_{-i}^{(0)}$, $\mathbf{M}_{-i}^{(p)}$, $\mathbf{m}_{-i}^{(0)}$, $\mathbf{m}_{-i}^{(p)}$ do not include any parameter. Using this property, (i) our ScaGWR evaluates these matrices a priori, and (ii) performs the LOOCV minimizing the CV error (Eq.3) after that. The resulting computational complexity to evaluate $\widehat{\boldsymbol{\beta}}_{-i}$ in step (ii) is only $O(K^3)$. Thus, the complexity for our CV score evaluation is $O(NK^3)$. If the golden section search (expected number of iterations: $O(\log N)$ is employed, the complexity for our LOOCV loop is $O(NK^3 \log N)$. Thus, our developed ScaGWR achieves a quasi-linear computational time with respect to N.

3.3. Comparison with the basic GWR

Unlike the usual GWR, the ScaGWR estimator Eq.(10) is interpretable as an empirical Bayes estimator that shrinks the local estimator $\left[\sum_{p=1}^{P} \tilde{b}^p \mathbf{M}_{-i}^{(p)}\right]^{-1} \sum_{p=1}^{P} \tilde{b}^p \mathbf{m}_{-i}^{(p)}$, which equals the standard GWR estimator with the kernel $\tilde{g}_{i,j}^0(\tilde{b})$, towards the OLS estimator $\left[\alpha \mathbf{M}_{-i}^{(0)}\right]^{-1} \alpha \mathbf{m}_{-i}^{(0)} = \left[\mathbf{M}_{-i}^{(0)}\right]^{-1} \mathbf{m}_{-i}^{(0)} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$. Thus, the ScaGWR estimator is more stable than the basic GWR. Another difference is that while GWR estimates one parameter b in the LOOCV, the ScaGWR estimates two parameters $\{\tilde{b}, \alpha\}$. For these reasons, ScaGWR can be more accurate than GWR. Still, ScaGWR with CP cost of $O(N \log N)$ is much faster than GWR with the cost of $O(N^2 \log N)$ (K is assumed fixed here).

4. Monte Carlo simulation

GWR and ScaGWR were compared by fitting them on the synthetic data randomly generated from

$$y_i = \beta_{i,0} + x_{i,1}\beta_{i,1} + x_{i,2}\beta_{i,2} + \varepsilon_i, \quad \varepsilon_i \sim N(0,1), \qquad \text{Equation 11}$$

where the explanatory variables are generated from $N(0,1)$, and the coefficients are generated from spatial moving average processes $\beta_{i,k} = \sum_{j=1} g_{i,j} u_j$, $u_j \sim N(0,1)$ assuming the exponential kernel for $g_{i,j} = \exp(-d_{i,j})$. The spatial coordinates were randomly determined using two standard normal distributions. For ScaGWR, *P* = 4 and *Q* = 100 is assumed (see Murakami et al., 2019). The ScaGWR estimation was iterated 200 times for each $N \in \{500, 1000, 1500, 2000, 3000, 5000, 8000, 12000, 20000, 40000, 80000\}$. GWR is applied for cases with $N \leq 12000$.

We used a Mac Pro (3.5 GHz, 12-Core Intel Xeon E5 processor with 64 GB memory) for the computation, R (version 3.6.2) for model estimation, and the R package GWmodel (version 2.1-3) for GWR and ScaGWR calibrations. Both the calibrations are implemented using C++.

Figures 4 (left) summarizes the evaluated root mean squared errors (RMSEs) for $\widehat{\boldsymbol{\beta}}_1$. If N is small, GWR estimators are more accurate than ScaGWR. However, if N exceeds 2000, ScaGWR estimators indicate better accuracy owing to the stability. Figure 4 (right) compares computational (CP) time. While the CP time of the standard GWR rapidly grows as N increase, the CP time of our approach increase only linearly. When $N = 80{,}000$, our ScaGWR takes only 120 seconds. Even if $N = 1{,}000{,}000$, the ScaGWR took only 1,637 seconds on average in five trials without parallelization.

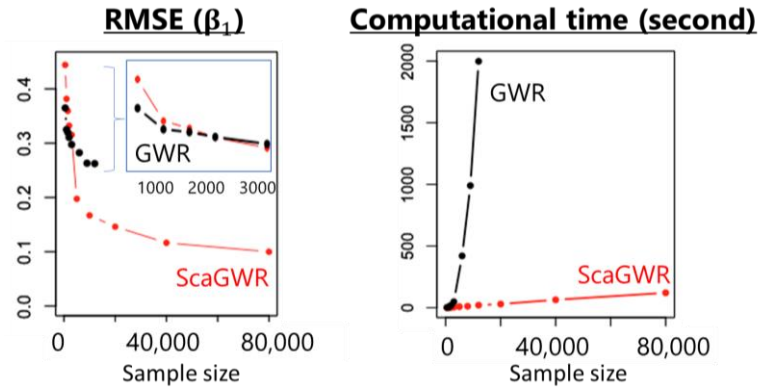

Figure 4. RMSE and CP time. RMSEs for $\widehat{\boldsymbol{\beta}}_0$ and $\widehat{\boldsymbol{\beta}}_2$ are not shown because they are very similar to $\widehat{\boldsymbol{\beta}}_1$.

5. Concluding remarks

We developed ScaGWR of large samples. See Murakami et al. (2019) for further details. ScaGWR is implemented in the R packages **GWmodel** and **scgwr**.

5. Acknowledgements

This research was supported by the Japan Society for the Promotion of Science (17K12974, 18H03628), and Joint Support-Center for Data Science Research (006RP2018, 004RP2019), Japan.

6. References

Li, Z., Fotheringham, A.S., Li, W. and Oshan, T. 2019. Fast Geographically Weighted Regression (FastGWR): a scalable algorithm to investigate spatial process heterogeneity in millions of observations. *International Journal of Geographical Information Science*, 33(1), 155-175.

Liu, H., Ong, Y.S., Shen, X., and Cai, J. 2018. When Gaussian process meets big data: A review of scalable GPs. *ArXiv*, 1807.01065.

Murakami, D., Tsutsumida, N., Yoshida, T., Nakaya, T. and Lu, B. 2019. Scalable GWR: a linear-time algorithm for large-scale geographically weighted regression with polynomial kernels. *ArXiv*, 1905.00266.