Inverse Distance Weighted Random Forests: Modeling Unevenly Distributed Non-Stationary Geographic Data

Liangdong Deng, Malek Adjouadi and Naphtali Rishe
School of Computing and Information Sciences
Florida International University

***Abstract*—Recent years saw explosive growth of Human Geography Data, in which spatial non-stationarity is often observed, i.e., relationships between features depend on the location. For these datasets, a single global model cannot accurately describe the relationships among features that vary across space. To address this problem, a viable solution – that has been adopted by many studies – is to create multiple local models instead of a global one, with each local model representing a subregion of the space. However, the challenge with this approach is that the local models are only fitted to nearby observations. For sparsely sampled regions, the data could be too few to generate any high-quality model. This is especially true for Human Geography datasets, as human activities tend to cluster at a few locations. In this paper, we present a modeling method that addresses this problem by letting local models operate within relatively large subregions, where overlapping is allowed. Results from all local models are then fused using an inverse distance weighted approach, to minimize the impact brought by overlapping. Experiments showed that this method handles non-stationary geographic data very well, even when they are unevenly distributed.**

***Index Terms*—spatial non-stationarity, human Geography, random forests**

I. Introduction

Geographic datasets are usually categorized by the phenomena they describe. Data collected about the natural processes of the Earth are categorized as Physical Geography datasets, such as mineral resources, hydrology, weather, and climate [1]. In contrast, data generated about activities of people are called Human Geography datasets: housing, culture, traffic, disease, war, crime, etc.. Historically, researchers were more interested in Physical Geography datasets in order to learn how to survive and mitigate the damage of natural disasters [2], discover and utilize mineral resources, understand environmental damage [3], and so on. But recent years also witnessed an explosive growth of Human Geography data, as GPS-enabled devices are omnipresent in everyday life [4].

With new data come new challenges. Many of the tools and theories that worked well with Physical Geography data are incompatible with the new datasets, which have significantly different characteristics. Spatial non-stationarity is one of them. For Physical Geography data, researchers usually assume stationarity, meaning the relationships among features remain unchanged across space. This makes perfect sense because Earth's natural processes, such as the distribution of mineral deposits, will only be relevant to various environmental factors. Location doesn't matter as long as all the environmental factors have been sampled. However, such an assumption is not necessarily valid for Human Geography data due to the complicated nature of human activities. For instance, a house's sale price is affected by numerous factors like the number of bedrooms, garage space, square footage, and so on. When stationarity is assumed, the increase of garage space will cause the same amount of sale price increase everywhere, which is not the case, as garage space can be much more valuable in cities than in rural areas. Even for urban areas, it's unlikely that Paris shares the same pricing model with New York. Thus, it makes more sense to think that the sale price model is affected by "local knowledge" [5], which shifts over space. This local knowledge is not directly included in the dataset because it's difficult to collect or measure, but its influence on the data is real and observable.

A viable solution to the non-stationarity problem is to build multiple local models instead of a single average global model. Each local model represents a subregion of the space within which the data is relatively stationarity. Many studies took this route (e.g., [5], [6] and [7]) and obtained significant improvements. However, the fact that local models are only trained from nearby observations within a certain area (called the kernel) can be a double-edged sword, and the size of kernels (called the bandwidth) must be chosen carefully [8]. According to [9], modeling accuracy will be severely impacted if the sample size drops below 1000. Thus, a smaller bandwidth will generate fine-grained models, but each of them is less accurate. In contrast, a larger bandwidth will be less sensitive to non-stationarity and tends to generate local models similar to each other. When data is extremely unevenly distributed, the majority of kernels will contain so few data that it's pointless to adopt the multiple-local-model approach. Figure 1 shows the Melbourne housing data (ranging from 2016 to 2018), which illustrates how unevenly distributed data could be for Human Geography datasets.

To solve this problem, we propose the IDW-RF (Inverse Distance Weighted Random Forests) algorithm, which adopts the multiple-local-model approach but allows kernels to overlap. Experiments show that IDW-RF performs as well as other state-of-the-art methods when data is evenly distributed and outperforms in uneven distribution.

978-1-7281-9279-6/20/$31.00

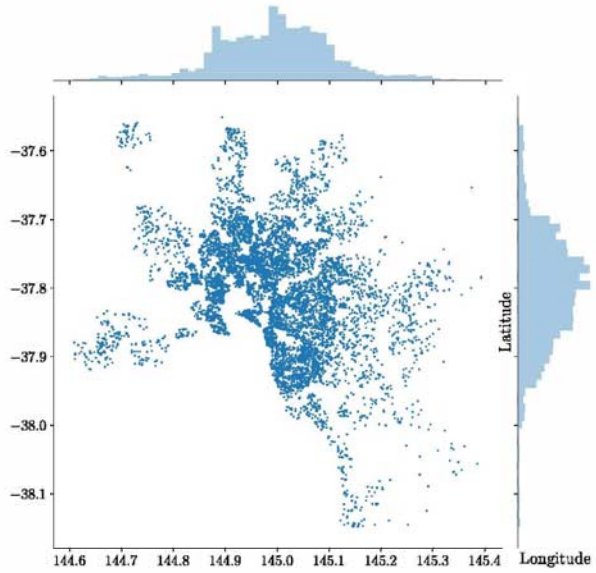

Fig. 1. Spatial distribution of Melbourne housing data.

II. Background

Researchers had been studying how to model non-stationary spatial data for a long time. Brunsdon, Fotheringham, and Charlton performed the first well-known research on this topic in 1996 [5]. In the paper, an algorithm called Geographically Weighted Regression (GWR) was proposed, whose "main characteristic is that it allows regression coefficients to vary across space, so the values of the parameters can vary between locations" [10]. GWR solves non-stationarity by making it possible for relationships between features and labels to differ across spaces, rather than generating an average global model. Many later studies are based on this multiple-local-model idea. For example, [6] improves GWR by replacing Ordinary Least Squares (OLS) – which is used in GWR to generate local models – with Random Forests and see substantial improvements, as the Random Forests algorithm is naturally superior to OLS at modeling spatial data. And [7] allows different local models to operate at different spatial scales, thus building more flexible and scalable regression models.

These studies also discussed the problem mentioned above that the bandwidth of kernels must be carefully chosen to get any useful models. [5] uses cross-validation to find the optimal bandwidth. [7] has a complicated weighing method and invented a Back-Fitting algorithm to solve the bandwidth selection problem. [6] fuses prediction results from global and local models, to improve overall accuracy when the local models are not good enough. However, these methods fail to discuss the case in which data are so unevenly distributed that it's impossible to find a bandwidth that achieves both high accuracy (favors larger bandwidth) and non-stationarity (prefers smaller bandwidth). Here, we present a new approach that allows local models to be trained from very large kernels (typically greater than 10% of the entire dataset) so that bandwidth no longer plays a critical role in the success of the algorithm. The impact brought by letting kernels overlap each other is then minimized by fusing all prediction results with the IDW method.

III. Inverse Distance Weighted Random Forests

IDW-RF's design includes five key steps in training and predicting process, which we detail below.

A. Training: Select Kernel Centroids

As previously mentioned, a kernel is defined as an area in which a local model operates. The number, location, and radius of kernels will have a direct impact on the model performance and, thus, must be carefully chosen. A common way to do this – as adopted by many other researchers – is to use the locations of all the data points as kernel centroids, which also implies that the number of local models will be the same as the number of data points. The main disadvantage of this method is that it's computationally expensive and doesn't scale well when the data size increases. In our case, another drawback of this method is most of the calculations would be unnecessary and even harmful because the adoption of large kernels means most kernels will significantly overlap with each other if there are too many of them.

Here, we use a simple grid-based method to generate kernel centroids. Within the dataset's boundaries, space is evenly divided to grid cells, whose geometric centers are then used as kernel centroids. The value of G – the size of grid cells – can be determined by either prior knowledge of the data, an exhaustive grid search process, or a combination of both. As a general rule of thumb, if prior knowledge is to be used, G should be the best guess on the average range within which data points remain relatively stationary. For example, when predicting house sale prices, homes within the same community are generally believed to follow the same pricing model. In other words, data is stationary on the scale of communities. So for this case, G can be set as the average size of communities. However, in most scenarios, when prior knowledge doesn't exist or cannot be precisely determined, an exhaustive grid search will be used to find the optimal value of G, which will be explained in detail later.

B. Training: Determine Kernel Sizes

There are two types of kernels. Adaptive kernels are defined by n nearest neighbors, whereas fixed kernels have a predetermined radius r [8]. In the present study, adaptive kernels are used, since they perform much better when data density varies across space, which is the main challenge this paper tries to solve. Instead of n, we use $\alpha = \frac{n}{N}$, where N is the total number of data points, to get a better idea of what percentage of data are used to train each of the local models. Similarly to G, this parameter is also tuned by the grid search process.

978-1-7281-9279-6/20/$31.00 © 2020 IEEE

C. Training: Create Local Models

After kernels are chosen, local models can then be trained. Here we use the Random Forests [11] algorithm to create local models. As the name suggests, Random Forests use multiple randomly generated decision trees to predict unknown observations. Each of the trees is trained by only part (about two-thirds) of the data points available. Moreover, during the feature selection process, each decision tree node only chooses from a random subset of features. The final prediction result is either a majority vote (for classification) or an average (for regression) of results from all the trees. The theory behind RF is that the bagging process will decrease the variance of the model without increasing the bias, leading to better overall model performance.

We choose RF to create local models for several reasons. First, RF will not over-fit no matter how the number of trees is increased. According to a study of the Random Forests [12], expected generalization error of ensembling decision trees has a variance of:

$$var(x) = \rho(x) \cdot \sigma^2_{\mathcal{L},\theta}(x) + \frac{1-\rho(x)}{M} \cdot \sigma^2_{\mathcal{L},\theta}(x)$$

in which M is the size of the ensemble and $\rho(x)$ is Pearson's correlation coefficient between two randomized models trained from the same data. Thus if $\rho(x)$ is smaller than 1 (which is always the case for RF), increasing M (number of decision trees) will always cause $var(x)$ to decrease. This characteristic of RF is critical to the success of our algorithm, which is de facto an ensemble of decision trees that trained on the same or partially different data.

Second, RF is based on decision trees, which is naturally good at handling coordinates in geographic datasets. Many other models have trouble with them because latitude and longitude will be treated as independent variables if fed directly into the model. In such a case, loss of spatial information would be unavoidable, and the model's effectiveness would be undermined. As a workaround, researchers often run a feature engineering process on geographic datasets before training, to capture the information embedded within data locations and convert them into additional features. But this method is often unreliable, and the results largely depend on the skill of the person who performs the feature engineering process. But decision trees do not suffer from such complications. If a leaf node of the decision tree is examined, its criteria are determined only by the set of ancestor nodes all the way up to the root, for which the order doesn't matter. If latitude or longitude appears in any of its ancestor nodes, like in the example shown in Figure 2, the leaf node can be considered as operating within the area defined collectively by all its ancestor latitude/longitude nodes. This is precisely how we expect location information to be accounted for.

Additionally, Random Forests is one of the best machine learning algorithms available and often shows excellent potential when dealing with spatial data, as observed by many studies, including [13] and [14]. Using RF as the underlying local model will enable us to inherit all of these advantages.

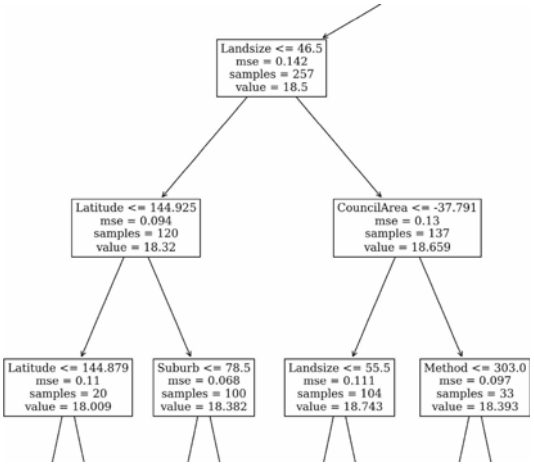

Fig. 2. A branch of decision tree trained from Melbourne housing data.

D. Predicting: An Inverse Distance Weighted Approach

To predict an unknown observation, results from all local models are combined with the following formula:

$$f(x) = \frac{\sum_{i=1}^{N} w_i(x) f_i(x)}{\sum_{i=1}^{N} w_i(x)},$$

where

$$w_i(x) = \begin{cases} d(x,x_i)^{-p}, & \text{if } d(x,x_i) \geq L \\ L^{-p}, & \text{if } d(x,x_i) < L \end{cases}$$

Here, $f_i(x)$ is the prediction result for unknown observation x, given by the ith local model. $w_i(x)$ is the weight, which decreases as distance $d(x,x_i)$ increases between x and the local model's kernel centroid. p is a positive real number, called the power parameter. L applies a lower bound to the distance function, to avoid the situation in which x is so close to a local model that renders all others useless. This situation is generally not a problem when IDW is used for interpolation purposes but is harmful in our case. Being close to the centroid of a kernel doesn't make the data point a better fit for the local model than others. As a general rule of thumb, L should be smaller than G (grid cell size), and here we use $L = G/2$ in our model. Its value can be fine-tuned. But experiments show that as long as L stays close to $G/2$, its value doesn't have an observable impact on overall accuracy.

There are multiple reasons why this inverse distance weighted approach works. For one, the idea is in accordance with the first law of geography "everything is related to everything else, but near things are more related than distant things", which was proposed by Tobler in 1970 [15]. Spatial heterogeneity is accounted for in this method by assigning larger weights to closer local models. Another reason is that the adoption of large kernels significantly improves the

local model's accuracy, which then benefits the entire model. Although ensembling local models trained from the same data could increase the variance of the generalization error, this possibility is eliminated by adopting Random Forests as the underlying local model, which doesn't over-fit no matter how many decision trees are trained, as discussed in the previous section.

E. Exhaustive Grid Search with Cross-Validation

So far, the algorithm is almost complete except several parameters are not yet determined: grid cell size G, kernel size α, and the power parameter p in the IDW formula. Best values for these parameters depend on the dataset and have to be fine-tuned on a case-by-case basis. As a general idea, datasets with larger scales of non-stationarity tend to favor a large G value. The smaller the dataset is, the larger α should be to offset the impact on accuracy brought by small training data size. And large p values should be used when the data is highly non-stationary.

To get the best performance, our research utilizes the Grid Search method to determine the best G, α, and p. The grid search method runs an exhaustive search on a predetermined hyper-parameter space. Each parameter has a lower bound, an upper bound, and the number of steps. The method will attempt all parameter combinations to find the best one. This process is considered to be computationally expensive. However, since we only run this process during the training stage, it is generally not a problem for most applications that are not sensitive to long training time.

Still, Grid Search alone is not effective enough as it is prone to variance problems. Model performance obtained from one test may differ from the others due to randomness in the tests performed. If not dealt with, this variance may propagate further down the line and cause the parameters learned from the Grid Search process to be biased. For this reason, we add Cross-Validation [16] to the evaluation process of the Grid Search. A straight-forward K-Fold Cross-Validation would be sufficient for a general problem, but the story is very different for a geographic dataset. A randomly generated training set from the regular K-Fold algorithm is not necessarily equally random at all locations. Many researches have noticed that this could lead to potential issues and proposed different cross-validation strategies [17]. In our study, we adopt the Block Cross-Validation method, which splits data into blocks from which samples are equally withdrawn. For this method, there is no set rule on how large the blocks should be and how many folds (i.e., the value of K) work the best. Generally speaking, the blocks should be of the same scale on which the data remains stationary. And although a higher K gives better results, it would significantly extend the Grid Search process. Thus, K should be set as high as the computing time limitations allow.

IV. A Case Study: Melbourne Housing Market

In this section, the IDW-RF algorithm will be applied to the Melbourne Housing Market data (downloaded from the Kaggle website [18]). The dataset contains 8,841 (records with missing fields are stripped) real estate transaction records in the city of Melbourne in Australia from 2016 to 2018, during which the area was experiencing a housing bubble. As shown in Figure 1, this is a very typical non-stationary human geography dataset in which data is extremely unevenly distributed. In densely populated areas, there are more than enough data points to outline the Port Phillip Bay's coast. But rural areas only see sparse data points scattered all over the space. Thus, this is a perfect dataset to test the capability of the IDW-RF algorithm.

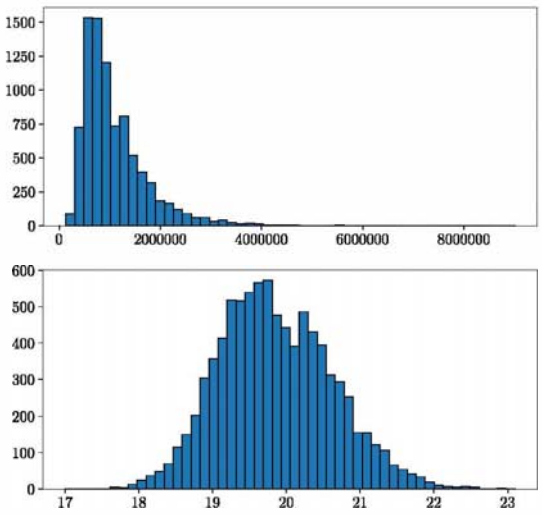

Fig. 3. Histogram of Price vs. log(Price).

A. Data Cleaning and Exploratory Analysis

The original dataset has 21 features which can be categorized into these groups:

- Location related features: latitude/longitude, zip code, suburb, region, etc.
- Transaction related features: sale price, date, seller and sale method, etc.
- House related features: the number of bedrooms and bathrooms, garage space, land size, etc.

Of all these features, the sale price is the target variable we would like to model and predict. After plotting the house price value as a histogram, we immediately realize that it spans over a broad range with a long tail, as illustrated in Figure 3. Thus we adopt log(Price) – which has a normal distribution – instead of using Price directly as the target variable. Among the rest of the features, Latitude and Longitude are the most important ones giving us the precise location of the house. The address field is unnecessary as it's inferior to the coordinates and cannot provide any other useful information. However, other location-related features, like zip code and suburb, are kept even though they are derivable form the coordinates, as they have sharp boundaries affecting the tax and school district

of the house. And all transaction/house related features are also useful.

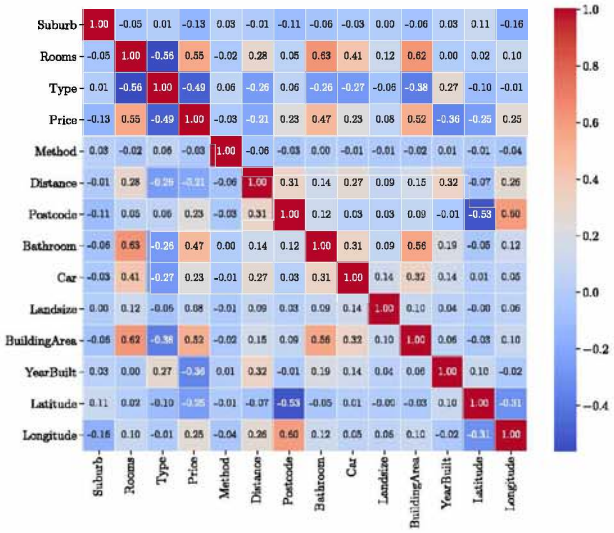

Fig. 4. Pearson correlation of all important features.

Figure 4 shows the Pearson correlation heatmap among the most important features. From the figure, the sale price has an obvious positive correlation with the number of rooms/bathrooms and the building's area (square meters of the living area). It also has a solid negative correlation with year built and house type. The sale method (how the house was sold) is not correlated with anything; thus, it can be removed from the modeling process. None of the rest of the feature pairs show a strong positive or negative correlation. Therefore, it's safe to retain all of them.

However, Fig. 4 only gives us the global average correlation among features, which doesn't tell anything about how it could vary across space. Therefore, we split the entire coordinate space into 80 grids ($10 * 8$). To reduce randomness introduced by the small sample, grids with too few data points are removed from the rest of the calculation. In each of the grids left, the Pearson correlation coefficient is calculated between all regular features and the target variable log(Price). Then, all the results are summarized to form Table I. Here, we can see that Landsize has the largest standard deviation among all features, which means it's probably "more non-stationary" than the others. Nevertheless, almost all features show a great difference between the minimum and maximum correlation, which is an indication that the level of non-stationarity cannot be overlooked in this dataset.

B. Assessment Measurements

Before proceeding, we still need to decide how to measure the accuracy of our models. The choice of measuring method would affect the creation of Random Forests and the Grid Search process during which parameters are optimized. The most commonly used error measurements are: mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE). Here, we prefer MAE (see the equation below) as the latter two methods tend to penalize large errors, which makes them unfavorable in our situation.

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

TABLE I
STATISTICS OF CORRELATIONS ACROSS SPACE

	mean	std	min	max
Suburb	-0.01	0.18	-0.28	0.43
Rooms	0.68	0.11	0.47	0.87
Type	-0.67	0.12	-0.84	-0.38
Distance	-0.09	0.19	-0.39	0.22
Postcode	0.03	0.15	-0.27	0.27
Bathroom	0.48	0.12	0.24	0.74
Car	0.35	0.11	0.14	0.55
Landsize	0.31	0.28	-0.16	0.80
BuildingArea	0.61	0.13	0.28	0.83
YearBuilt	-0.28	0.17	-0.57	-0.02

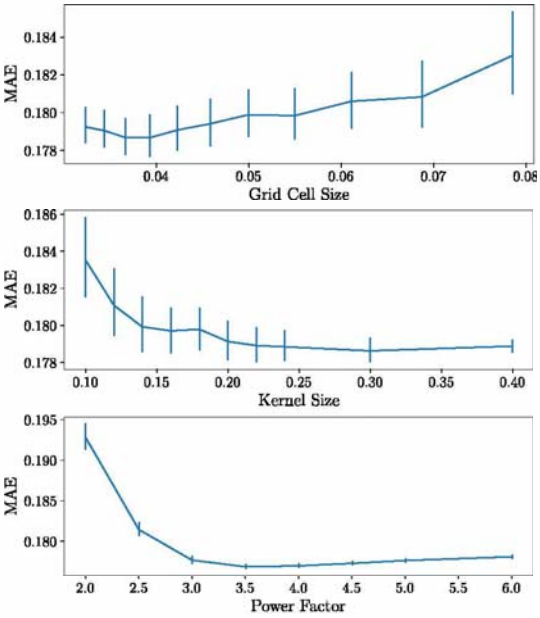

Fig. 5. Calibrate Parameters with Grid Search.

C. Results

Now that everything is ready, we apply the IDW-RF algorithm to the data. Results returned by the Grid Search are plotted as an Average-Max-Min chart, as shown in Figure 5. We can see that a valley is present in all the three graphs, where MAE is the lowest. According to the results, a kernel size of 0.22 (meaning 22% of all data points are used to form the kernel) is optimal for this particular dataset. This observation matches our theory that when data points are sparse in most areas, a large kernel will produce better overall

results. But if the kernel becomes too big, the model will fail to capture non-stationarity, as all local models tend to behave the same.

Additionally, the optimal grid cell size is 0.039 (the difference in the longitude), which translates to about 3.4 kilometers in the city of Melbourne. And our assumption was that grid cell size should be roughly on the same scale with which the data remain stationary. Although for this dataset there is no way to know on what distance would the house price model remain stationary, it is reasonable to believe it's of the same scale of 3.4 kilometers. As for the power factor, its range of MAE is way smaller than the others. The implication of this is that the optimization of power factor p takes priority vs. the others. This makes sense as the local models' training areas overlap each other heavily, and the power factor must be carefully tuned to fuse them correctly.

TABLE II
RESULTS FROM DIFFERENT ALGORITHMS.

	MAE
Linear regression	0.308
Neural Network	0.317
Random Forests	0.187
MGWR	0.186
RFsp	0.191
IDW-RF	0.174

As a comparison, we run several other algorithms on this dataset and list the results in Table II. Unsurprisingly, the non-geographic algorithms (Linear Regression and Neural Network) perform poorly, as they are not capable of handling non-stationary geographic data. We also tested two state-of-the-art geographic machine learning algorithms RFsp [19] and MGWR [7], using their R and Python implementations. Both of them adopt the multiple-local-model method and use RF as the underlying local model. Results show that their performance will degrade to that similar to the original RF for such an unevenly distributed dataset.

V. CONCLUSION

This paper presents IDW-RF, which models unevenly distributed non-stationary data very well. IDW-RF first split the entire coordinate space into multiple grids from which kernel centroids are chosen. It then establishes a kernel for every centroid by including the nearest data points. Next, local Random Forest models are trained from these kernels. All the local models will perform predictions together by fusing their results with an Inverse Distance Weighted approach. Finally, an Exhaustive Grid Search with Cross-Validation will be performed to calibrate the parameters.

The insight of IDW-RF is that by choosing a relatively large kernel size, local models' accuracy will be significantly improved (especially when data are unevenly distributed over space). On the other hand, the expected generalization errors brought by allowing kernels to overlap each other are minimized by choosing Random Forest – which doesn't suffer to over-fitting no matter how many decision trees are trained – as the algorithm to build local models. This method is capable of handling non-stationary datasets and outperform others when data is spatially unevenly distributed.

ACKNOWLEDGMENT

This material is based in part upon work supported by the National Science Foundation under Grant Nos. MRI CNS-2018611, I/UCRC IIP-1338922, MRI CNS-1532061, MRI CNS-1920182.

REFERENCES

[1] D. Massey, "Space-time, 'science' and the relationship between physical geography and human geography," *Transactions of the Institute of British Geographers*, vol. 24, no. 3, pp. 261–276, 1999.
[2] E. Spyrou and Y. Avrithis, "A region thesaurus approach for high-level concept detection in the natural disaster domain," vol. 4816, pp. 74–77, 12 2007.
[3] E. Figueiredo, C. Farrar, K. Worden, and J. Figueiras, "Machine learning algorithms for damage detection under operational and environmental variability," *Structural Health Monitoring*, vol. 10, 03 2010.
[4] C. Beath, I. Becerra-Fernandez, J. Ross, and J. Short, "Finding value in the information explosion," *MIT Sloan Management Review*, vol. 53, pp. 18–20, 06 2012.
[5] C. Brunsdon, A. S. Fotheringham, and M. E. Charlton, "Geographically weighted regression: A method for exploring spatial nonstationarity," *Geographical Analysis*, vol. 28, no. 4, pp. 281–298, 1996.
[6] S. Georganos, T. Grippa, A. N. Gadiaga, C. Linard, M. Lennert, S. Vanhuysse, N. Mboga, E. Wolff, and S. Kalogirou, "Geographical random forests: a spatial extension of the random forest algorithm to address spatial heterogeneity in remote sensing and population modelling," *Geocarto International*, pp. 1–16, 2019.
[7] A. S. Fotheringham, W. Yang, and W. Kang, "Multiscale geographically weighted regression (mgwr)," *Annals of the American Association of Geographers*, vol. 107, no. 6, pp. 1247–1265, 2017.
[8] S. Kalogirou, "Destination choice of athenians: An application of geographically weighted versions of standard and zero inflated poisson spatial interaction models," *Geographical Analysis*, vol. 48, no. 2, pp. 191–230, 2016.
[9] J. Morgan, R. Dougherty, A. Hilchie, and B. Carey, "Sample size and modeling accuracy with decision tree based data mining tools," *Acad Inf Manag Sci J*, vol. 6, 01 2003.
[10] J. Mateu, "Comments on: A general science-based framework for dynamical spatio-temporal models," *Test*, vol. 19, pp. 452–455, 11 2010.
[11] L. Breiman, "Random forests," in *Machine Learning*, pp. 5–32, 2001.
[12] G. Louppe, "Understanding random forests: From theory to practice," 2014.
[13] A.-L. Boulesteix, S. Janitza, J. Kruppa, and I. R. König, "Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics," *WIREs Data Mining and Knowledge Discovery*, vol. 2, no. 6, pp. 493–507, 2012.
[14] M. Nussbaum, K. Spiess, A. Baltensweiler, U. Grob, A. Keller, L. Greiner, M. E. Schaepman, and A. Papritz, "Evaluation of digital soil mapping approaches with large sets of environmental covariates," *SOIL*, vol. 4, no. 1, pp. 1–22, 2018.
[15] W. R. Tobler, "A computer movie simulating urban growth in the detroit region," *Economic Geography*, vol. 46, pp. 234–240, 1970.
[16] D. M. Allen, "The relationship between variable selection and data agumentation and a method for prediction," *Technometrics*, vol. 16, no. 1, pp. 125–127, 1974.
[17] D. Lieske, "A robust test of spatial predictive models: Geographic cross-validation," *Journal of Environmental Informatics*, vol. 17, pp. 91–101, 06 2011.
[18] T. Pino, "Melbourne housing market data." https://www.kaggle.com/anthonypino/melbourne-housing-market, 2018.
[19] T. Hengl, M. Nussbaum, M. N. Wright, G. B. Heuvelink, and B. Gräler, "Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables," *PeerJ*, vol. 6, p. e5518, Aug. 2018.