Pyneapple-L: Scalable Expressive Learning-based Spatial Analysis

Yongyi Liu[1,2] Nicolas Lee[4] Yunfan Kang[5] Mohammad Reza Shahneh[1]
Ahmed Mahmood[3] Vishal Rohith Chinnam[1] Aparna Vivek Sarawadekar[1]
Samet Oymak[6] Ibrahim Sabek[4] Amr Magdy[1,2*]
[1] Department of Computer Science and Engineering, University of California, Riverside
[2] Center for Geospatial Sciences, University of California, Riverside [3] Google LLC.
[4] Thomas Lord Department of Computer Science, University of Southern California
[5] Department of Geography & Geographic Information Science, University of Illinois Urbana-Champaign
[6] Electrical Engineering and Computer Science Department, University of Michigan-Ann Arbor
yliu786@ucr.edu nml@usc.edu yfkang@illinois.edu mzare008@ucr.edu amahmoo@google.com
{vchin014,asara068}@ucr.edu oymak@umich.edu sabek@usc.edu amr@cs.ucr.edu

Abstract

This paper demonstrates *Pyneapple-L*, an open-source library designed to enhance scalable spatial analysis through learning-based techniques. Through collaboration with social scientists and domain experts, we identify scalability challenges inherent in conventional spatial analysis methods, particularly as the data size increases. *Pyneapple-L* addresses these challenges by leveraging learning-based models to offer scalable solutions. We demonstrate two modules: scalable learning of spatial hotspots along spatial networks and augmented geographically weighted regression. To showcase *Pyneapple-L*, we have developed a user-friendly frontend web application to interact with different datasets, algorithms, model configurations, and visualize outcomes on interactive maps that support both broad and analytical views.

CCS Concepts

• **Information systems → Geographic information systems**; • **Location based services**;

Keywords

Hotspot Detection, Spatial Network, Geographically Weighted Regression, Machine Learning

ACM Reference Format:
Yongyi Liu[1,2] Nicolas Lee[4] Yunfan Kang[5] Mohammad Reza Shahneh[1], Ahmed Mahmood[3] Vishal Rohith Chinnam[1] Aparna Vivek Sarawadekar[1], and Samet Oymak[6] Ibrahim Sabek[4] Amr Magdy[1,2]. 2024. Pyneapple-L: Scalable Expressive Learning-based Spatial Analysis. In *The 32nd ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL '24), October 29-November 1, 2024, Atlanta, GA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3678717.3691228

*This work is partially supported by the National Science Foundation, USA, under grants IIS-2237348 and CNS-2031418, the Google-CAHSI research grant, and Microsoft unrestricted gift.

SIGSPATIAL '24, October 29-November 1, 2024, Atlanta, GA, USA

ACM ISBN 979-8-4007-1107-7/24/10
https://doi.org/10.1145/3678717.3691228

1 Introduction

The widespread use of location-based services has led to an abundance of spatial data, empowering domain experts, such as social scientists, to discover valuable patterns and insights from the data. In spatial analysis, hotspot detection over spatial networks [8] and geographically weighted regression (GWR) [11] stand out as two critical analyses with broad implications and applications, including traffic management and transportation [2, 10], public health [12], housing price modeling [4], and crime analysis [7]. However, due to the rapid growth of data sizes, traditional methods are often constrained by their limited scalability, preventing the potential use of such methods on large-scale spatial data. In contrast, learning-based methods have arisen as powerful solutions due to their scalability and adeptness at uncovering complex spatial patterns. These models have found extensive applications across various fields, including urban planning, traffic management, and public health.

Pyneapple-L demonstrates two *scalable* and *expressive* techniques that were recently developed to tackle the problems of learning hotspots from large spatial networks [8] and boosting the scalability and expressiveness of the popular geographically weighted regression (GWR) technique through augmenting it with general-purpose machine learning models [11]. The objective of hotspot detection in spatial networks is to pinpoint areas within a network that exhibit a significantly higher concentration of objects than surrounding regions. Hotspot detection is applied in diverse fields such as traffic management [10], public health [12], and crime analysis [7]. Existing spatial network hotspot detection methods can be classified as either clustering-based methods or statistical-based methods. While clustering methods have efficient runtime, they might result in false-positive results. On the other hand, statistical methods offer rigorous statistical validation for the detected hotspots, e.g., Monte Carlo trials, and log-likelihood scores, ensuring the detected hotspots are statistically robust. However, such methods do not scale to large-sized datasets due to the unacceptable execution time in statistical validation [13], which takes tens of minutes to run on tens of thousands of objects. This is particularly problematic in applications requiring fast response times, such as traffic hotspot detection. While social scientists prefer statistical-based hotspot detection methods due to their reliability, scalability becomes the limitation. For instance, the Chicago crime dataset includes 7 million objects, exceeding the capacity of existing statistical methods.

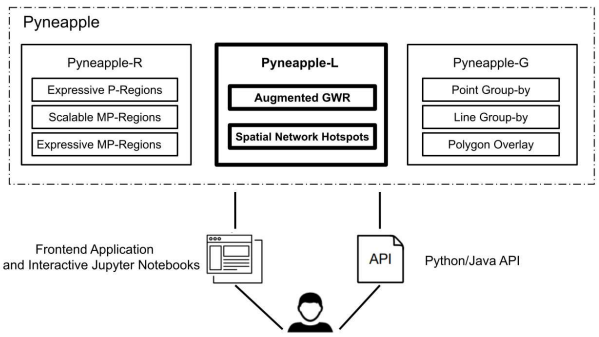

Figure 1: Pyneapple Overview

On another hand, geographically weighted regression (GWR) extends traditional regression analysis by modeling the spatial variability in the relationships among variables. It assigns a local regression equation to each observation in the dataset across different geographical locations. GWR has been widely applied in diverse areas including transportation science [2] and housing price modeling [4]. Nonetheless, GWR encounters two primary challenges: expressiveness and scalability. The issue of limited expressiveness arises from the fact that many GWR models utilize a uniform weighting factor (bandwidth) across all features, neglecting the fact that different features may exhibit unique spatial scales. On the scalability front, the efficiency of GWR is constrained by its quadratic time complexity in training, which restricts the applicability of GWR to large spatial datasets.

This paper presents a system demonstration of *Pyneapple-L*, an open-source library for scalable expressive learning-based spatial analysis based on our research work in [8, 11]. Our techniques achieve orders of magnitude in runtime improvement as outlined below. Attendees at our demonstration can interact with *Pyneapple-L* from a frontend application to visualize hotspots and employ geographically weighted regression algorithms in different use cases. Subsequent sections offer an overview of the *Pyneapple-L* library (Section 2) and the demonstration scenarios (Section 3).

2 *Pyneapple-L* Overview

Pyneapple-L is an integral sub-package of the more extensive *Pyneapple* library [9]. Figure 1 shows an overview of the *Pyneapple* ecosystem, which is currently under development with more features being added. The current version of *Pyneapple* comprises three main sub-packages, regionalization queries (*Pyneapple-R* [6]), group-by-aggregation queries (*Pyneapple-G* [1]), and learning-based queries (*Pyneapple-L*). *Pyneapple-L* consists of two modules to demonstrate our work in [8, 11]: (a) scalable learning of hotspots over large spatial networks [8], and (b) scalable expressive augmented geographically weighted regression (A-GWR) [11]. Each module in *Pyneapple-L* is equipped with Python API documentation, facilitating a seamless integration into the broader data science landscape. The rest of this section outlines *Pyneapple-L* modules.

2.1 Learning Hotspots Over Spatial Networks

We learn hotspots by computing a localized network K-function that reveals the inherent statistical distribution of objects within a spatial network [8]. Unlike training an off-shelf machine-learning model, our learning paradigm fine-tunes the computation scalability of the K-function method to scale it up on large datasets. We address two problems for learning hotspots within spatial networks. The first problem, known as Hotspot Detection with Predefined Radius (HDPR), requires inputs of a radius distance threshold and a statistical confidence threshold. It identifies hotspots that exceed the given confidence at the specified radius. The second problem is Hotspot Detection Without Predefined Radius (HDWPR), which only requires a statistical confidence threshold as input and autonomously determines the optimal radius for each identified hotspot.

To address Hotspot Detection with Predefined Radius (HDPR), Incremental Batched Traversal (IBT) has been introduced. IBT proposes a batch-processing strategy that processes all objects located on the same edge collectively in a single batch. This approach stems from the principle that identifying a hotspot requires the statistical analysis of adjacent network areas, such as counting the objects and measuring the size of the isodistance subnetwork. Since objects on the same edge are close to each other and have shared neighboring objects, IBT utilizes this spatial proximity to streamline calculations for these objects collectively. Furthermore, IBT introduces optimizations to reduce unnecessary calculations. For example, it prunes the exploration of objects far from a certain center, utilizing precomputed distances at a high level.

Approximate Hotspot Identification via Incremental Batched Traversal (AH-IBT) is proposed to address Hotspot Detection Without a Predefined Radius (HDWPR). AH-IBT introduces a unique strategy that involves incrementally expanding the hotspot radius until it encounters the first local maximum of statistical confidence, at which point this hotspot is selected for its high degree of localization. This method ensures that the hotspot identified represents a truly dense cluster that is as localized as possible. Given the vast range of possible radius sizes, AH-IBT achieves a trade-off between efficiency and effectiveness by gradually enlarging the radius in larger steps to incorporate surrounding locations. This allows for the inclusion of neighboring locations in an incremental manner. Through this method, AH-IBT identifies hotspots efficiently while maintaining practical effectiveness.

The experimental results on large spatial road network datasets show that, IBT achieves up to 28 times faster compared to the state-of-the-art methods [3] in solving Hotspot Detection with Predefined Radius (HDPR). AH-IBT achieves more than four orders of magnitude faster in solving Hotspot Detection Without Predefined Radius (HDWPR). The significant runtime improvements stem from the optimization in sharing computation and the effective reduction of unnecessary exploration space.

2.2 Augmented Geographical Regression

Geographically Weighted Regression (GWR) extends traditional regression analysis by integrating spatial geography into its framework. This approach considers not only the variables of interest but also the spatial coordinates of each data point, enabling a nuanced examination of how relationships vary across geographical space. The inputs for GWR include spatial coordinates, predictor variables, and a bandwidth type parameter that defines the extent of spatial variation being modeled. This bandwidth may be fixed, which applies the same scale of influence across all locations as in the traditional GWR, or adaptive, which allows different scales

for different features as in the multiscale GWR (MGWR). The primary output of GWR and its variants is a series of local regression coefficients for each geographical location, illustrating the spatial variability in the relationships among variables.

We introduce Augmented Geographical Weighted Regression (A-GWR) [11] as an advanced GWR variant to handle large-sized spatial datasets with more expressive capabilities. First, to improve scalability, A-GWR incorporates a novel spatial regression component known as Stateless-MGWR (S-MGWR), an adaptation of the MGWR model based on directly fitting a set of bandwidths, which eliminates the need to store historical bandwidth values. This design also enhances flexibility and efficiency in optimizing bandwidth parameters through the use of black-box optimization techniques. To address the challenge of scaling with large datasets during training, A-GWR employs a divide-and-conquer strategy. This method divides the dataset into smaller, more manageable chunks without losing the spatial relationships among the data points. By doing so, A-GWR can handle large datasets effectively, even with limited computing resources.

Second, to improve expressiveness, A-GWR combines the S-MGWR spatial regression model with general-purpose machine learning models, such as random forests, to analyze complex non-spatial relationships within the data. This integration allows A-GWR to uncover intricate patterns and dependencies in the input data, ensuring high accuracy and efficiency in its results. A major strength of our framework is that the integration of spatial-aware regression models (such as GWR variants) with traditional machine learning models is seamless. This allows the spatial models to make use of the new advances in machine learning models without the need to tailor every new model for spatial data.

A-GWR achieves up to 14.4 times faster compared to the state-of-the-art models [5] on large spatial datasets. This improvement is due to its streamlined optimization methods, such as removing the requirement to keep the historical bandwidth data during fitting, using advanced black-box parameter optimization techniques, and dividing the data into smaller chunks for scalable training.

3 Demonstration Scenarios

To demonstrate *Pyneapple-L*, we design different scenarios for different groups of target attendees. Users will interact with *Pyneapple-L* library through user-friendly web interfaces. Attendees will be able to visualize results on interactive maps for both broad and detailed perspectives and explore different features in *Pyneapple-L*. By exploring these scenarios, the attendees will discover the full capabilities of the algorithms within the library for tackling large-scale spatial analytics challenges. The rest of this section outlines different demonstration scenarios.

3.1 AGWR - Local View

In the first scenario, the user can initiate a Geographically Weighted Regression (GWR) analysis by creating an instance of the AGWR model on a selected dataset and a set of parameters. For example, as demonstrated in Figure 2, assume the user is interested in generating a suggested rate (US dollar) for an Airbnb rental in New York City, with a set of features including room type, minimum nights, number of reviews, and location, among others. The user begins

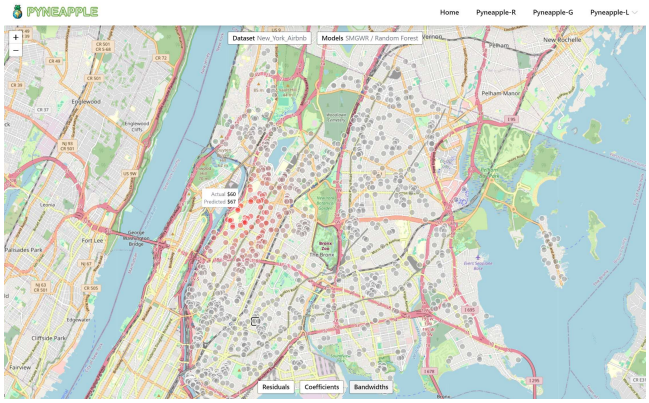

Figure 2: Local View of AGWR

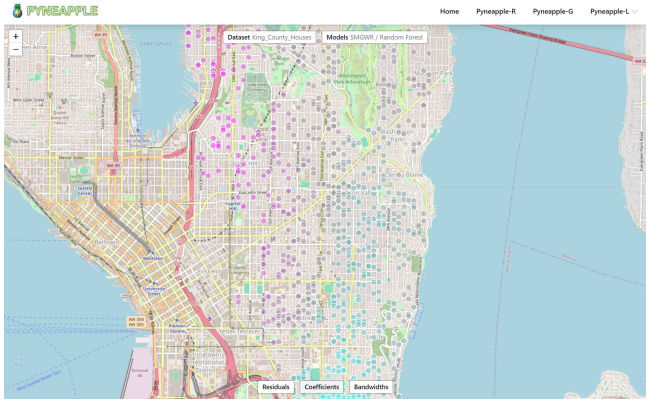

Figure 3: Global View of AGWR

by choosing the New York City Airbnb dataset from the provided examples. Next, the user selects the desired spatial and machine learning models for the analysis. For this scenario, SMGWR is chosen as the spatial model and Random Forest as the general-purpose model. Then, a request containing the specifications is sent to the backend where the corresponding module generates the model and predictions with AGWR and returns a response containing the results. Upon receiving the response from the backend, the results are displayed on an interactive map.

From the interactive map, the user can click a location to see the predicted value given by the model and the ground-truth value for the object. For instance, in Figure 2, the model's predicted rental rate is 67 dollars, compared to the actual rate of 60 dollars. Additionally, the map visually conveys the bandwidth for a specific feature through color. The user can switch the bandwidth visualization for different features by clicking the "Bandwidth" button at the bottom. In this example, "number of reviews" is selected as the feature to visualize in red. Darker shades of red indicate objects that have a larger influence on the prediction due to their proximity, whereas lighter shades denote objects further away with less influence.

3.2 AGWR - Global View

In the second scenario, demonstrated in Figure 3, the user can explore the coefficient of features at different map locations, as well as evaluate the accuracy of these predictions by using the "Coefficient" and "Residual" buttons located at the bottom of the interface. This

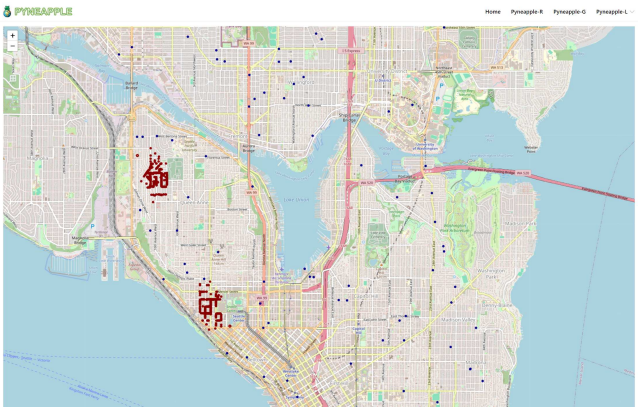

Figure 4: Basic View of Hotspot Detection

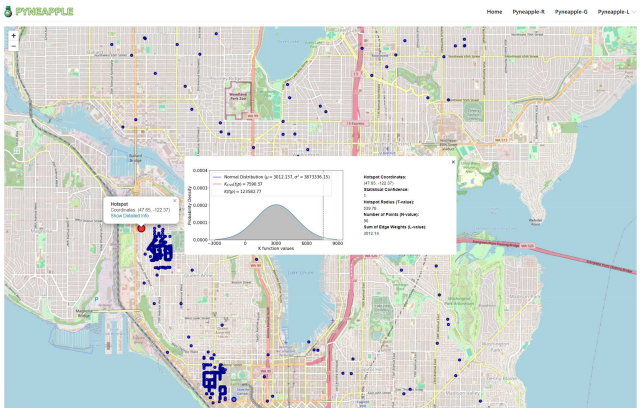

Figure 5: Analytical View of Hotspot Detection

view facilitates a broader understanding of the relationships between features and outcomes at a higher level. For example, assume the user is interested in analyzing the impact of the feature "number of reviews" on house price prediction across King County, USA. The user will first select the dataset of house sales in King County and the desired models to be used, upon which a request will be sent to the backend where the corresponding module generates the model and predictions with AGWR and returns a response containing the coefficient map and residual map.

Figure 3 shows the coefficient map on the feature "number of reviews". In this map, purple indicates a greater coefficient of this feature and the prediction outcome, while cyan and grey represent weak and average coefficients, respectively. The color gradient across the map delineates the range of coefficient strengths, revealing areas where specific features have a more pronounced or diminished impact on house prices. Similarly, the residual map illustrates the discrepancies between the predicted and actual values across different locations in King County, providing a visual assessment of prediction accuracy.

3.3 Hotspot Detection - Basic View

In the third scenario, the user can issue a hotspot query and visualize the results using a basic, color-differentiated map. For instance, as shown in Figure 4, suppose the user aims to visualize hotspots for Seattle traffic collisions. The user starts by selecting the Seattle traffic collision dataset and determining the type of hotspot analysis to be conducted, choosing between HDWPR (Hotspot Detection without Predefined Radius) and HDPR (Hotspot Detection with Predefined Radius). Following this, the user will specify the algorithm parameters and a query will be sent to the backend. In this case, the user uses HDPR with a predefined hotspot radius of 2000 meters and a statistical confidence level of 99%.

Once the analysis is complete in the backend, the user receives a map visualization where hotspot objects are marked in red. Users have the flexibility to zoom in or out to examine the hotspots with varying degrees of detail. This basic view offers users a clear and direct representation of hotspots, allowing for straightforward identification and analysis of patterns within the data.

3.4 Hotspot Detection - Analytical View

In this scenario, the application offers an analytical view of hotspot detection, providing users with detailed statistical insights for each hotspot, as illustrated in Figure 5. Initially, the user selects a dataset and chooses a hotspot detection method. In this case, HDWPR with a minimum statistical confidence of 99% is selected. These inputs initiate a request to the backend where the selected algorithm processes the data to identify hotspots on an interactive map.

As users navigate the interactive map, clicking on a specific object reveals detailed statistical information about that hotspot. For example, in Figure 5, when the user selects an object on the map, it is highlighted in red. The detailed statistical breakdown then appears, providing the hotspot label, coordinates, K-function plot including the corresponding normal distribution, statistical confidence level, the radius of the hotspot, the number of points, and the sum of edge weights. This analytical view fosters a deeper understanding of each hotspot, allowing users to explore and analyze the statistical characteristics and significance of each identified location.

References

[1] L. Abdelhafeez, A. Calderon-Romero, A. Magdy, and V. J. Tsotras. Pyneapple-G: Scalable Spatial Grouping Queries. In *VLDB*, 2024.

[2] O. D. Cardozo, J. C. García-Palomares, and J. Gutiérrez. Application of Geographically Weighted Regression to the Direct Forecasting of Transit Ridership at Station-level. *Applied geography*, 34:548–558, 2012.

[3] T. N. Chan, L. H. U, Y. Peng, B. Choi, and J. Xu. Fast Network K-function-based Spatial Analysis. In *VLDB*, 2022.

[4] H. Crosby, P. Davis, T. Damoulas, and S. A. Jarvis. A Spatio-temporal, Gaussian Process Regression, Real-estate Price Predictor. In *ACM SIGSPATIAL*, pages 1–4, 2016.

[5] A. S. Fotheringham, W. Yang, and W. Kang. Multiscale geographically weighted regression (mgwr). *Annals of the American Association of Geographers*, 107:1247–1265, 2017.

[6] Y. Kang, Y. Liu, H. Alrashid, A. Bilgi, S. Purohit, A. Mahmood, S. Rey, and A. Magdy. Pyneapple-R: Scalable and Expressive Spatial Regionalization. In *IEEE ICDE*, pages 5497–5500, 2024.

[7] S. Khalid, F. Shoaib, T. Qian, Y. Rui, A. I. Bari, M. Sajjad, M. Shakeel, and J. Wang. Network Constrained Spatio-temporal Hotspot Mapping of Crimes in Faisalabad. *Applied Spatial Analysis and Policy*, 11:599–622, 2018.

[8] Y. Liu, Y. Kang, A. Mahmood, and A. Magdy. Scalable Evaluation of Local K-Function for Radius-Accurate Hotspot Detection in Spatial Networks. In *ACM SIGSPATIAL*, pages 1–12, 2023.

[9] MagdyLab. Pyneapple Repo. https://github.com/MagdyLab/Pyneapple.

[10] B. Romano and Z. Jiang. Visualizing Traffic Accident Hotspots based on Spatial-temporal Network Kernel Density Estimation. In *ACM SIGSPATIAL*, pages 1–4, 2017.

[11] M. R. Shahneh, S. Oymak, and A. Magdy. A-GWR: Fast and Accurate Geospatial Inference via Augmented Geographically Weighted Regression. In *ACM SIGSPATIAL*, pages 564–575, 2021.

[12] Y. Shi, Y. Chen, M. Deng, L. Xu, and J. Xia. Discovering Source Areas of Disease Outbreaks based on Ring-shaped Hotspot Detection in Road Network Space. *International Journal of Geographical Information Science*, 36(7):1343–1363, 2022.

[13] X. Tang, E. Eftelioglu, and S. Shekhar. Detecting Isodistance Hotspots on Spatial Networks: A Summary of Results. In *SSTD*, pages 281–299, 2017.