Adaptive Non-linear Intensity Modeling in Spatial Point Processes with GAMs

Zahra Maleki[1]

[1]MPI-CBG-dresden

Maleki@mpi-cbg.de

Abstract. *This study presents an efficient method using Generalized Additive Models (GAMs) to estimate complex intensity functions in spatial data in addition to a cross-validation technique that was employed to select optimal smoothing parameters, ensuring a balance between model accuracy and generalization. This approach was applied to both synthetic and real-world datasets. The results demonstrate the effectiveness of GAMs for uncovering hidden patterns and capturing spatial dependencies in ecological and biological data.*

1. Instruction

In this research, I focused on developing and optimizing Generalized Additive Models (GAMs) for estimating complex intensity functions in spatial data, specifically Poisson and Cox point processes. GAMs offer flexibility over traditional models, such as Generalized Linear Models (GLMs), by allowing for the estimation of non-linear effects and unknown factors through the use of smoothing parameters and basis functions.

To demonstrate the utility of GAMs, I applied the method to both simulated and real-world ecological datasets. For example, I analyzed gorilla nesting patterns in relation to environmental covariates such as vegetation type, heat levels, and proximity to water. The results revealed key insights into gorilla habitat preferences, which were not as easily captured using simpler models.

2. Chapter one

In this chapter, we discuss the foundational aspects of the model used in this work. In statistics, a generalized additive model (GAM) is a generalized linear model in which the linear response variable depends linearly on unknown smooth functions of some predictor variables, and interest focuses on inference about these smooth functions.

Generalized Additive Models (GAMs) provide more flexibility than traditional linear or generalized linear models (GLMs) by not assuming a fixed relationship between covariates and the dependent variable. Instead, GAMs allow for the automatic detection of non-linear effects, making them useful for revealing complex patterns in the data. However, with this flexibility comes the risk of overfitting, requiring careful application and interpretation.

GAMs are particularly valuable in fields like ecology, where relationships between environmental factors and outcomes are often non-linear. Their ability to model complex, non-parametric relationships makes them powerful tools, but they require careful tuning to avoid overfitting.

2.1. Generalized additive model

The model relates a univariate response variable, Y, to some predictor variables, xi. An exponential family distribution is specified for Y (for example normal, binomial or Poisson distributions) along with a link function g (for example the identity or log functions) relating the expected value of Y to the predictor variables via a structure such as

$$g[\mu(X)] = \alpha + f_1(X_1) + \cdots + f_p(X_p)$$

The additive logistic regression model replaces each linear term by a more general functional form. Examples of classical link functions are the following:

- $g(\mu) = \mu$: linear and additive models for Gaussian response data.
- $g(\mu) = \text{logit}(\mu)$: modeling binomial probabilities.
- $g(\mu) = \log(\mu)$: Poisson count data.

In the generalized additive model (GAM), we assume that the functions $f_i(x_i)$ are unknown smooth functions. At its simplest, the idea is to replace the unknown smooth functions in the model with basis expansions, such that:

$$f_j(x_j) = \sum_{k=1}^{K_j} \beta_{jk} b_{jk}(x_j)$$

where the $b_{jk}(x_j)$ are known basis functions, typically chosen for their approximation-theoretic properties. Examples of such basis functions includes B-splines. The coefficients β_{jk} are parameters to be estimated as part of the model fitting process. The basis dimension K_j is selected to be sufficiently large, ensuring that the model can overfit the training data (thereby avoiding bias from oversimplification). However, K_j is kept small enough to ensure computational efficiency during model fitting.

2.2. B-splines

A B-spline of order $p + 1$ is a collection of piecewise polynomial functions $B_{i,p}(t)$ of degree p in a variable t. The values of t, where the pieces of the polynomial meet, are known as *knots*, denoted as $t_0, t_1, t_2, \ldots, t_m$, and sorted into non-decreasing order.

for all t between the knots t_p and t_{m-p}, then the scaling factor of $B_{i,p}(t)$ becomes fixed. The knots in-between (and not including) t_p and t_{m-p} are called the *internal knots*. B-splines can be constructed using the Cox–de Boor recursion formula. We start with the B-splines of degree $p = 0$, i.e., piecewise constant polynomials:

$$B_{i,p}(t) = \begin{cases} \text{non-zero} & \text{if } t_i \le t < t_{i+p+1}, \\ 0 & \text{otherwise.} \end{cases}$$

The higher $(p + 1)$-degree B-splines are defined by recursion:

$$B_{i,p}(t) := \frac{t - t_i}{t_{i+p} - t_i} B_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} B_{i+1,p-1}(t).$$

For B-splines of degree 2, we can visualize both 1D and 2D examples of the basis functions. Below are the figures demonstrating these examples.

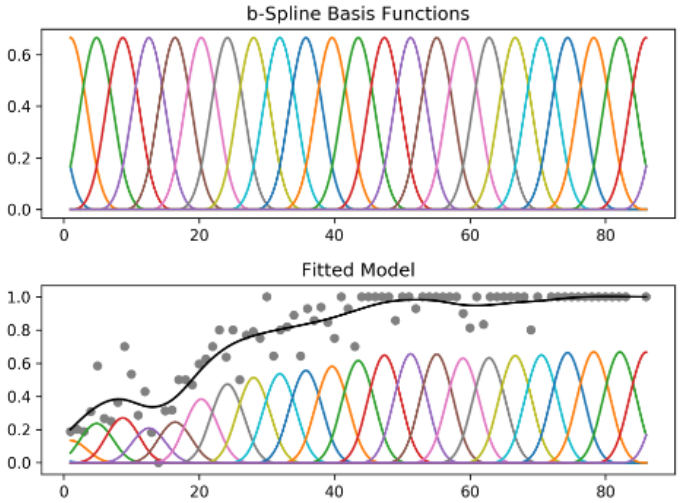

Figure 1. 1D B-spline Basis Functions

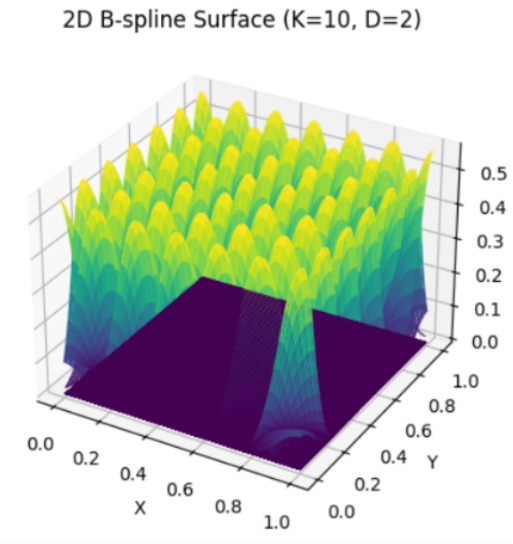

Figure 2. 2D B-spline Surface

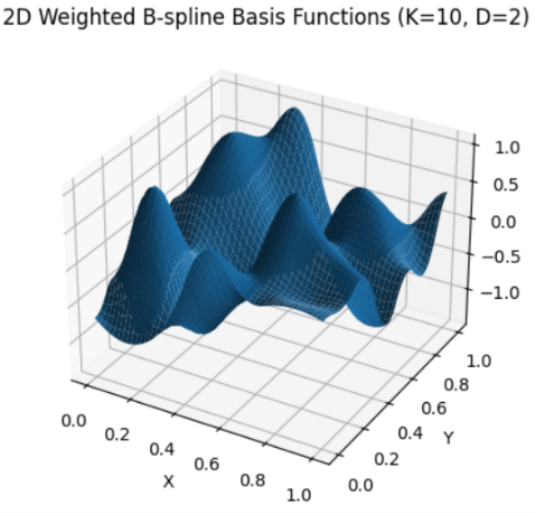

Figure 3. 2D Weighted B-spline Basis Functions

2.3. Fitting the model

Generalized Additive Models (GAMs) are flexible because they allow for non-linear relationships between the covariates and the response. However, due to the large number of basis functions K_j, the model can become over-parameterized, which may lead to overfitting if fitted using standard methods like Generalized Linear Models (GLMs).

To address this, GAMs introduce a penalty on the smoothness of the estimated functions. These smoothing penalties help control the complexity of the fitted model. The balance between model fit and smoothness is controlled by smoothing parameters λ_j, which penalize departure from smoothness.

Model Fitting with Penalties

For a GAM with univariate smooths, the parameters can be written as a vector $\boldsymbol{\beta}$, and the deviance $D(\boldsymbol{\beta})$ represents the goodness of fit. To prevent overfitting, the fitting procedure

minimizes a penalized version of the deviance:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\{ D(\boldsymbol{\beta}) + \sum_j \lambda_j \int \left(f_j''(x)\right)^2 dx \right\}$$

Here, the term $\int \left(f_j''(x)\right)^2 dx$ penalizes wiggliness, ensuring smoothness of the function $f_j(x_j)$. The smoothing parameters λ_j control the trade-off between model fit and smoothness: - When $\lambda_j \to \infty$, $f_j(x_j)$ becomes a straight line, leading to a linear relationship between the covariate and the response.

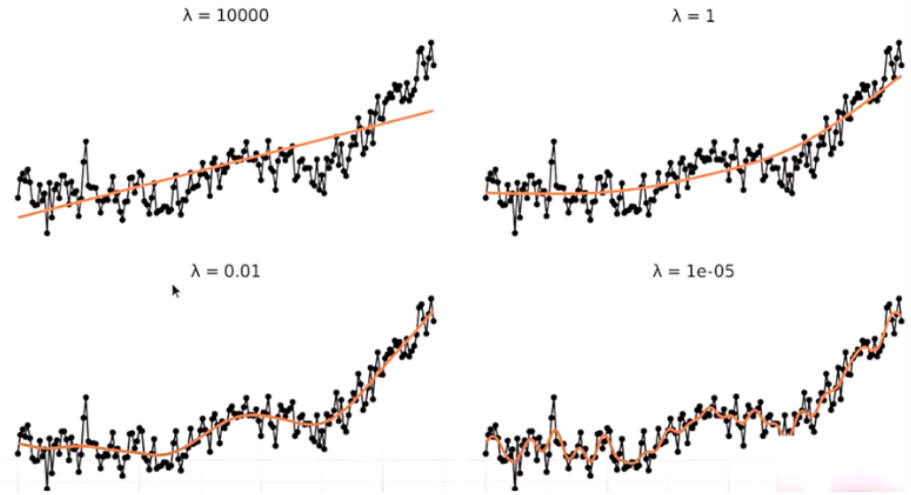

Figure 4. expressing the effect of λ for Hadcrut global temperature

Expressing Wiggliness as a Quadratic Form

Given the basis expansion for each smooth function f_j, the wiggliness penalties can be expressed as a quadratic form in the model coefficients:

$$\int \left(f_j''(x)\right)^2 dx = \boldsymbol{\beta}^T S_j \boldsymbol{\beta}$$

WhereS_j is a known matrix computable from the penalty and basis. This allows us to rewrite the penalty in terms of the full coefficient vector $\boldsymbol{\beta}$. S_j is computable from the penalty and basis functions.

$$\{S_j\}_{kl} = \int b_j''(t) b_k''(t) dt$$

where b_j and b_k are the basis functions, and b_j'' denotes the second derivative of the basis function b_j. This penalty ensures that the smoothness of the functions $f_j(x_j)$ is maintained during the fitting process.

Minimizing the Penalized Deviance

The model fitting problem now becomes minimizing the following objective function:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\{ D(\boldsymbol{\beta}) + \sum_j \lambda_j \boldsymbol{\beta}^T S_j \boldsymbol{\beta} \right\}$$

Smoothing parameter estimation

Smoothing parameter estimation is a crucial part of the model, as discussed earlier. The original approach, described in the work by Wood (2011), involves maximizing the restricted log-likelihood using gradient-based numerical methods, such as quasi-Newton algorithms. This method employs the Laplace approximation, requiring complex calculations involving the inverse of the covariance matrix and the log determinant of $\mathbf{S}$. The restricted log-likelihood is given by:

$$\ell_R(\rho) = \ell_p(\hat{\beta}) + \frac{1}{2}\log|\mathbf{S}^{-1}| - \frac{1}{2}\log|\mathbf{H}| + \text{constant}$$

Although effective, this approach involves multiple iterations and is computationally expensive.

To simplify the process and reduce computation time, I decided to use cross-validation. By plotting the negative log-likelihood (NLL) of the validation set against various smoothing parameters, I can efficiently determine the optimal smoothing parameter without the high computational cost of the original method.

To determine the optimal smoothing parameter, I split the data into two equal halves. One half is used as the training set ($p = 0.5$) and the other half as the validation set. For the training set, I use the negative log-likelihood (NLL) of the GAM to learn the matrix of coefficients β.

The NLL for the training set is defined as:

$$\log \ell_{\text{train}}(\beta) = \sum_{u \in X_{\text{train}}} \log \lambda(u) - p \int \lambda(u) du - \frac{1}{2}\sum_{k=1}^{K} \rho_k \boldsymbol{\beta}_k^\top \mathbf{S}_k \boldsymbol{\beta}_k$$

where $\lambda(u)$ is the intensity function, p is the proportion of the data used in training, and the penalty term enforces smoothness.

For the validation set, I use the NLL of the Poisson point process without the penalty term, since this ensures that the evaluation focuses on the fit to the data alone, without imposing any additional smoothing constraints. The NLL for the validation set is given by:

$$\log \ell_{\text{val}}(\beta) = \sum_{u \in X_{\text{val}}} \log \lambda(u) - (1-p) \int \lambda(u) du$$

By plotting the validation -NLL of the validation set against the smoothing parameter, I identify the minimum of this curve, which corresponds to the best smoothing parameter for the model. This method is computationally efficient and avoids the complexity and high computational cost of other methods, such as those based on full likelihood maximization. Cross-validation ensures that the chosen smoothing parameter not only fits the training data well but also generalizes effectively to new data.

3. Chapter two

In this chapter, I will describe the data used in this study and the methods applied to generate additional datasets for model training. The chapter covers both real-world datasets and

simulated point processes, including in-homogeneous Poisson and Cox point processes. These datasets are used for training and evaluating the performance of the Generalized Additive Models (GAMs) used in this work.

3.1. Poisson point process

A Poisson point process describes random events occurring independently over space or time, with no interaction between points. This randomness makes it useful for modeling events like star appearances or phone call arrivals.

homogeneous Poisson point process

In a homogeneous Poisson point process, events occur at a constant rate across the space. The likelihood of an event happening is the same everywhere, with the rate (intensity) λ representing the average number of points per unit area or time.

in-homogeneous Poisson point process

An in-homogeneous Poisson point process allows the event rate to vary across space or time. Some regions or times may have more events than others, depending on local conditions or factors.

Generating the Data

In this work, I am generating in-homogeneous Poisson point process data to train my model. The intensity function for the point process is based on a Gaussian distribution, where the intensity decreases as the distance from the center increases. To generate the data, I first sample points from a homogeneous Poisson process, which assumes a constant rate across the space. I then apply a thinning algorithm, which adjusts the point densities based on the location-specific intensity function. Points with higher intensities are more likely to be accepted, while points in areas with lower intensity are rejected.

I generate 100 sets of data, each containing 10,000 points, using this method. All the sets share the same underlying intensity function, which is governed by a Gaussian distribution with a defined spread and maximum intensity. This dataset is then used to train and evaluate the performance of my model on in-homogeneous spatial data.

GAM Forms for the Intensity Function

$$\Lambda(V;\boldsymbol{\theta},\mathbf{z}(V)) = \int_V \lambda(\mathbf{v};\boldsymbol{\theta},\mathbf{z}(\mathbf{v}))d\mathbf{v}$$

The above equation is the mean of the Poisson-distributed events. Given a dataset of hazard events indexed by $i = 1,\ldots,n$, occurring on V, the likelihood of the Poisson process model is:

$$L(\boldsymbol{\theta}) = \exp\left(|V| - \Lambda(V;\boldsymbol{\theta},\mathbf{z}(V))\right)\prod_{i=1}^{n}\lambda(\mathbf{v}_i;\boldsymbol{\theta},\mathbf{z}(\mathbf{v}_i))$$

To fit the model, we maximize a penalized log-likelihood function:

$$\ell_p(\boldsymbol{\theta}) = \log L(\boldsymbol{\theta}) - \frac{1}{2}\sum_{k=1}^{K}\rho_k\boldsymbol{\beta}_k^\top\mathbf{S}_k\boldsymbol{\beta}_k$$

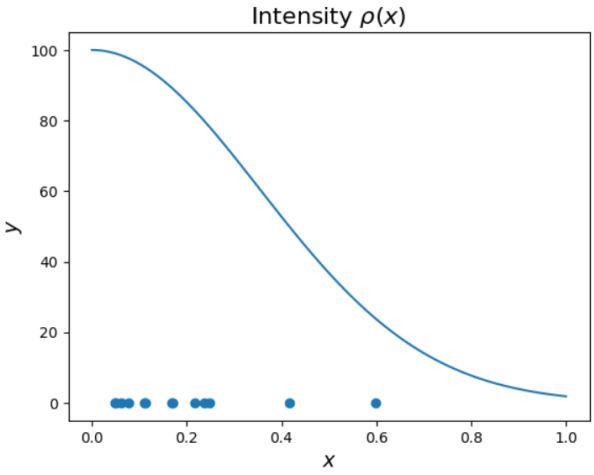

Figure 5. 1D Intensity Profile

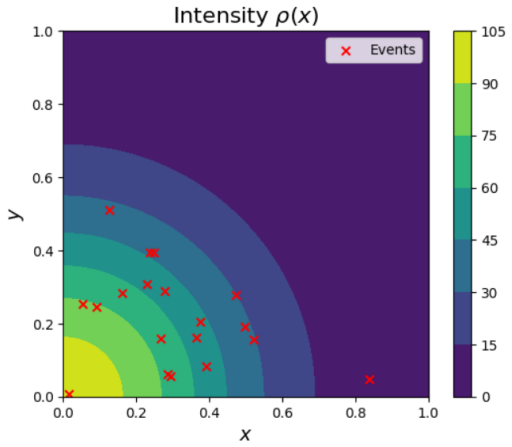

Figure 6. 2D Intensity Map with Events

where ρ_k are the smoothing parameters, S_k are the penalty matrices for smoothness, and $\boldsymbol{\beta}_k$ are the model coefficients.

3.2. Cox point process

A Cox point process is an extension of the Poisson point process where the intensity function is a realization of a random process, often modeled as a Gaussian Process (GP). This introduces clustering, making it useful for modeling spatial data with dependencies.

Generating Cox Point Processes

In this work, I generate Cox point processes by simulating a Log-Gaussian Cox Process (LGCP). The steps are as follows:

- Simulate a Gaussian random field (GRF) over the spatial domain with mean μ, variance var, and spatial scale to control correlation.
- Exponentiate the GRF to obtain a spatially varying intensity function.
- Sample points from the LGCP using this intensity, leading to spatial clustering in areas of higher intensity.

This process is repeated over multiple simulations, and I compute the average number of points for each realization.

Log-Likelihood for Cox Processes

The log-likelihood for the Cox process is the same as for the in-homogeneous Poisson point process since, conditioned on the GP, the points are distributed like a Poisson process. The intensity $\lambda(x)$ is a realization of the Gaussian process, and the log-likelihood uses this intensity function.

3.3. Gorilla Nesting Sites Dataset

To demonstrate the application of GAMs, I used the gorillas dataset, which contains the locations of nesting sites of gorilla groups in the Kagwene Gorilla Sanctuary, Cameroon. The dataset includes 647 spatial points representing nesting sites, along with several associated covariates such as group identifier, season, and observation date.

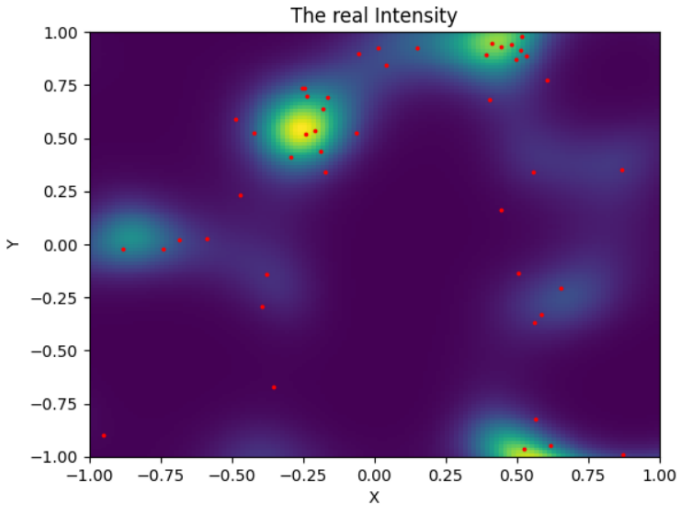

Figure 7. a sample of Cox generated data

The spatial covariates, include terrain attributes such as elevation, slope angle, slope type, vegetation type, and distance to the nearest water body. These covariates are important for modeling the habitat preferences of the gorillas.

This dataset is ideal for illustrating how GAMs can be used to model spatial patterns in ecological data, with a focus on understanding how environmental factors influence the distribution of gorilla nesting sites.

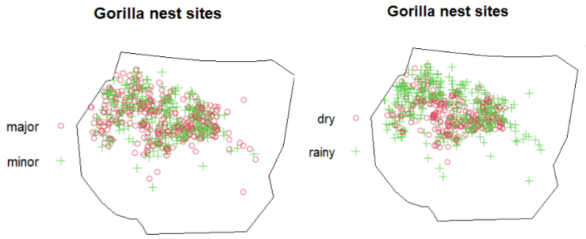

Figure 8. classes of the datasets

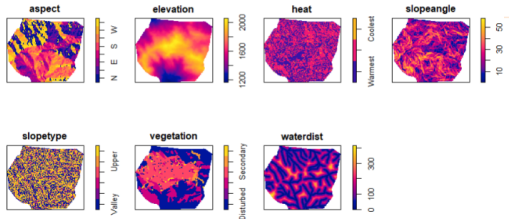

Figure 9. The spatial covariates of the dataset

4. Chapter three

4.1. 1D Poisson point process

Consider a 1D domain x from 0 to 1. After applying the Generalized Additive Model (GAM) and learning the vector of coefficients β on the generated Poisson data, the results are shown in the figure below. The blue line represents the learned intensity using GAM, and the orange line represents the true intensity function that was initially used to generate the data.

I further compared the learned intensity and the true intensity using two metrics:

- **Mean Squared Error (MSE)**: This metric measures the average of the squared differences between the predicted and true intensity values.the MSE is:

$$\text{MSE} = 35.8116$$

- **R-squared ($\mathbf{R}^2$) Score**: This metric represents the proportion of variance in the true intensity that is explained by the learned intensity. The R^2 score is:

$$R^2 = 0.9507$$

which indicates that the GAM learned intensity explains about 95% of the variance in the true intensity.

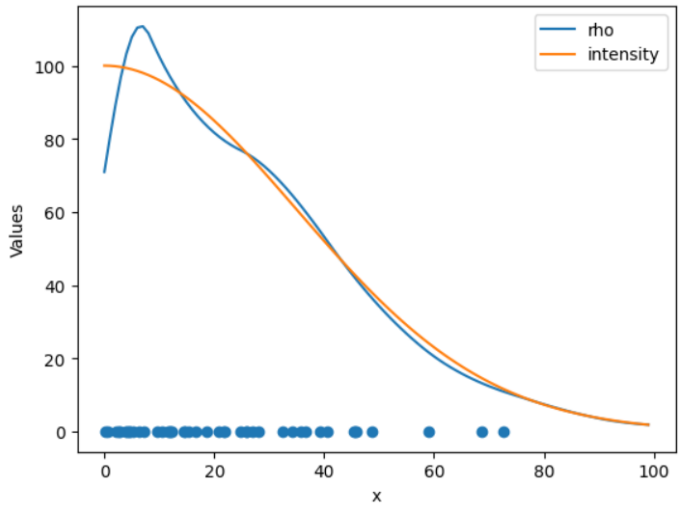

Figure 10. Comparison of learned intensity vs true intensity for a 1D domain.

One significant advantage of Generalized Additive Models (GAMs) compared to Generalized Linear Models (GLMs) is their ability to capture the variance and local distribution of the data more effectively. GAMs can model non-linear relationships and spatial variations, making them better suited for understanding patterns that vary across different regions of the data.

To test this, I generated data with different variances and means to see how well GAMs adapt to these changes. Specifically, I combined three datasets, each with a different variance and mean:

- Variance: [0.5, 0.1, 0.3]
- Mean: [0, 0.5, 1]

The results show how GAMs adjust to varying data distributions. Unlike GLMs, which might struggle with regions of high or low variance.

- MSE = 99.336
- R^2 = 0.87

Expanding the Application of Our Approach

We can extend the application of our approach beyond data within a fixed domain to explore how the distribution of data varies depending on the distance to a specific location. This has potential applications in fields such as systems biology and ecology, where understanding how the presence of a particular data point influences the surrounding spatial distribution is crucial.

Below is a figure demonstrating the results of this experiment, where we assess the impact of a single data point on the background intensity.

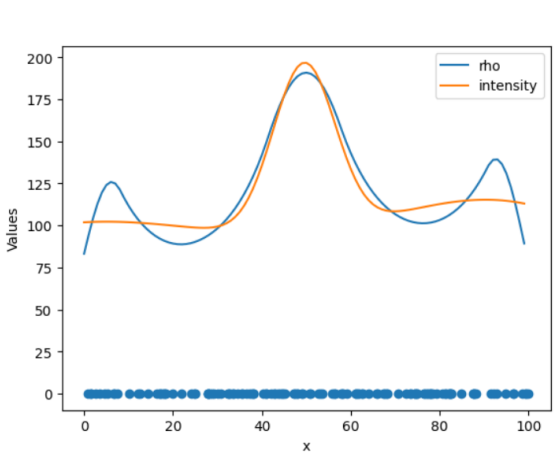

Figure 11. GAM results for a combination of datasets

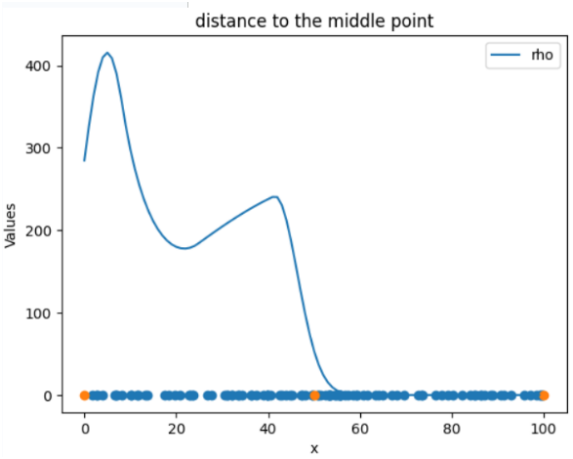

Figure 12. Intensity based on the distance to the middle point

4.2. 2D Poisson point process

In this experiment, I generated data on a 2D domain where $x \in [0, 1]$ and $y \in [0, 1]$. The evaluated intensity for the generated data is shown in the figure below. This plot shows the learned function using the GAM approach, which is compared to the ground truth shown in Figure 6.

The performance of the model was evaluated using several metrics:

- MSE = 38.599
- R^2 = 0.8798
- Absolute Difference Between Integrals: The absolute difference between the integral of the predicted intensity and the true intensity is = 0.2038
- The calculated function integral = 19.3079

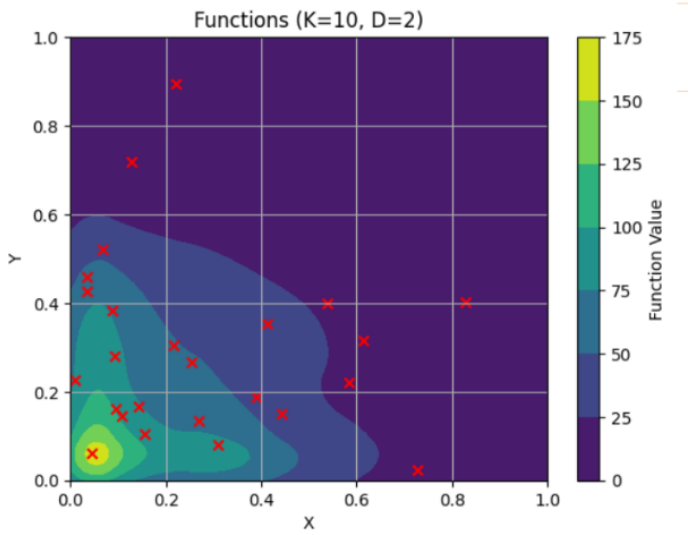

Figure 13. Evaluated intensity function for the generated data (K=10, D=2).

To estimate the best smoothing parameter, I applied the cross-validation approach described in Chapter One. The validation negative log-likelihood (NLL) was calculated

for a range of smoothing parameters. The results are shown in the figure below, where the NLL is plotted against the smoothing parameter.

From the plot, it is clear that the minimum NLL occurs at approximately 10^{-10}, which suggests that this is the optimal smoothing parameter for the model. This value of the smoothing parameter strikes the best balance between fitting the data and maintaining the smoothness of the estimated functions, without overfitting or underfitting the data.

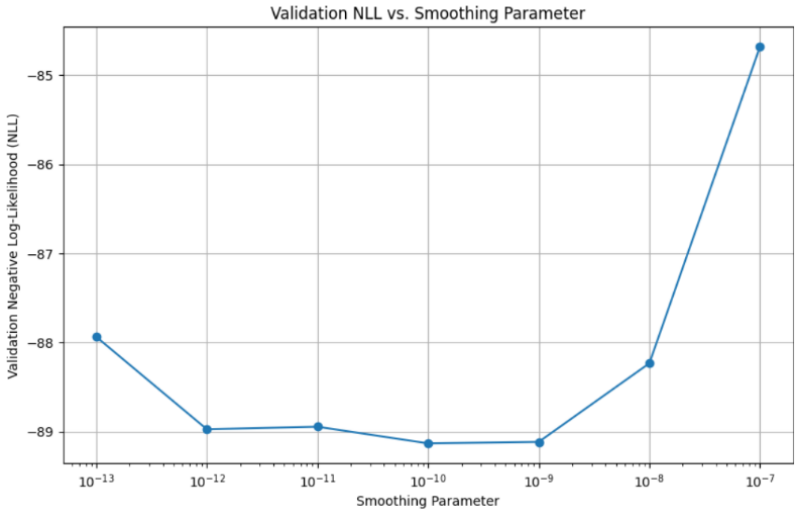

Figure 14. Validation NLL vs. Smoothing Parameter.

4.3. 2D Cox point process

Next, we extended our model to work with the Cox point process data described in Chapter Two. The results of applying the model to this data are shown in the figure below.

For one of the samples of this experiment, the best smoothing parameter was found to be smoothing_param $= 0.0088 \times 10^{-8}$. The performance of the model was evaluated using the following metrics for this sample:

- MSE = 1465.31
- R^2 = 0.4353
- Difference in integrals = 19.451
- Calculated integral of the actual intensity = 118.8932

These results suggest that the model's ability to the captures in the intensity function is evident from the visual comparison in the figure.

In the figure below, the left plot shows the actual intensity function for the Cox point process data, while the right plot shows the intensity predicted by my model.

In the figure 16, we can see the effect of the smoothing parameter on the results of the model. The plot on the right represents the actual intensity function, while the other plots correspond to the intensity learned by the model with different smoothing parameters.

To estimate the best smoothing parameter, I used the cross-validation approach described earlier. The validation negative log-likelihood (NLL) was computed for a range of smoothing parameters, as shown in the figure 17.

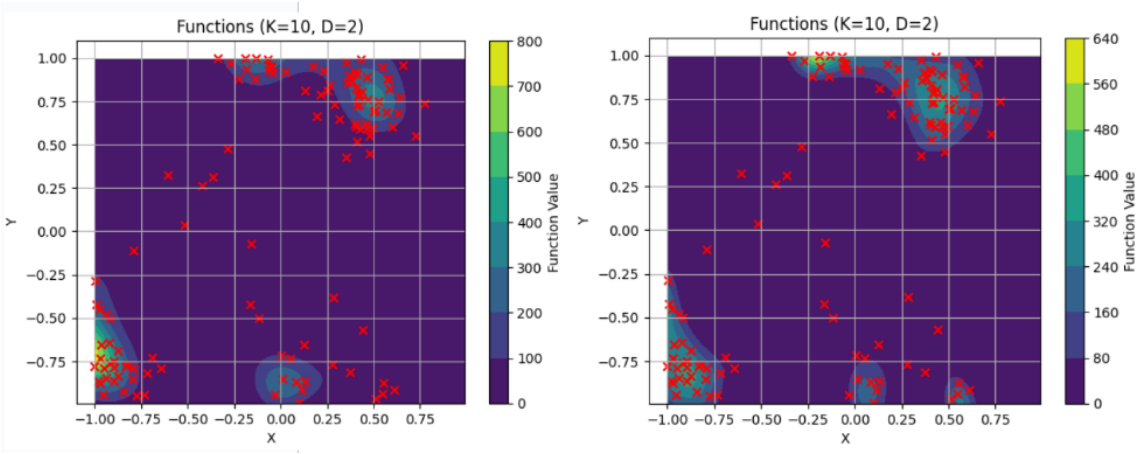

Figure 15. Left: Actual intensity. Right: Predicted intensity.

From the NLL plot, we observe that the minimum NLL occurs at around 10^{-6}, which suggests this is the optimal smoothing parameter. By comparing the results with this parameter to the actual intensity (right plot), we see that the learned intensity closely matches the actual one, confirming that this choice of smoothing parameter leads to a reasonable approximation of the true intensity.

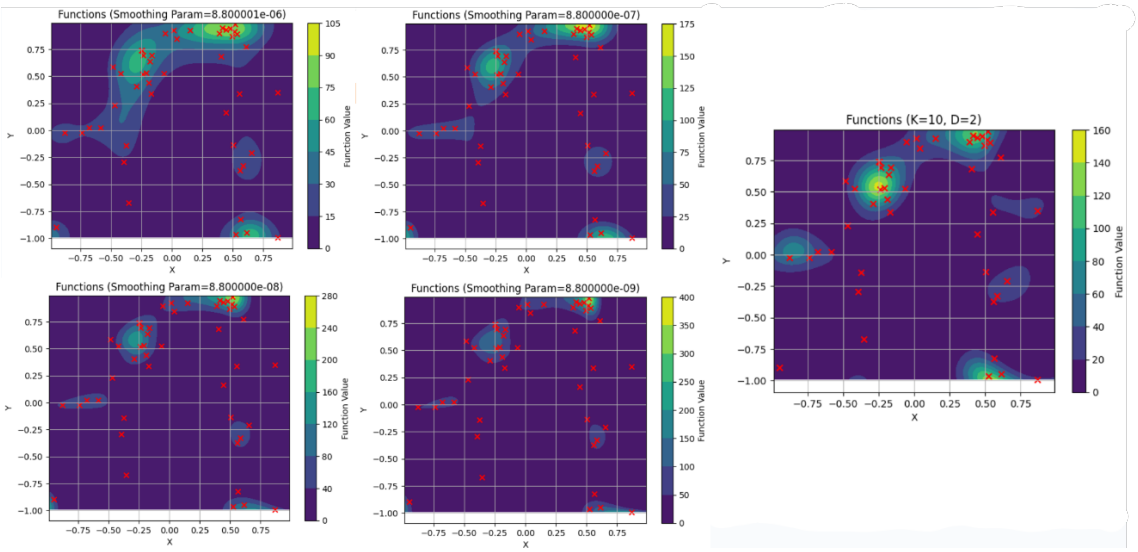

Figure 16. Effect of smoothing parameter on the learned intensity function. The right plot is the actual intensity.

4.4. Gorilla Nesting Sites Analysis

Intensity Estimation

As described in Chapter Two, I worked with the gorilla nesting dataset to extract information about the intensity and correlations between the covariates using Generalized Additive Models (GAMs). The dataset contains locations of nesting sites in a National Park in Cameroon, divided into two main groups:

- Major class: 350 data points

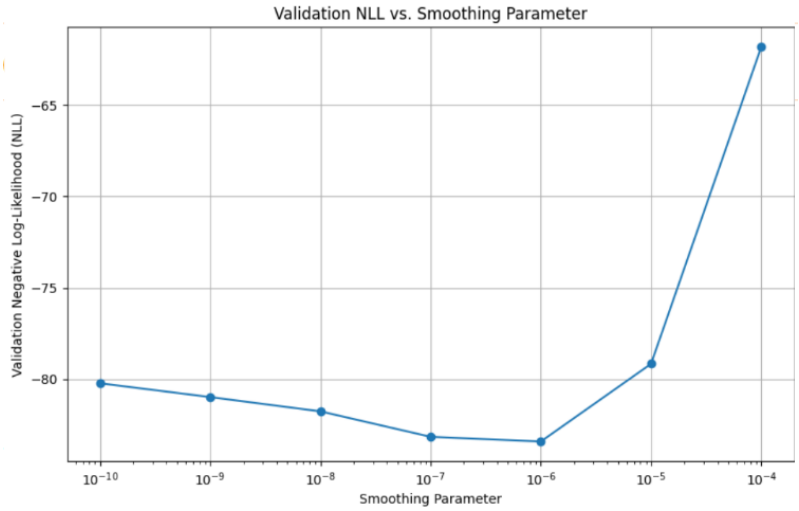

Figure 17. Validation NLL vs. Smoothing Parameter

- Minor class: 297 data points
- Rainy season: 372 data points
- Dry season: 274 data points

I estimated the intensity of nesting sites for these four cases using a GAM model on a 2D domain. The results are shown in the figure below, where the intensity of nesting sites is visualized for each group.

The figure demonstrates how the intensity of the nesting sites varies across the different classes and seasons. This analysis helps in understanding the spatial patterns of gorilla nesting behavior and the influence of various covariates on the distribution of the nests.

Heat Data Visualization

Using one of the covariates from the dataset, I visualized the heat data and aligned it with the estimated intensity function of the gorilla nesting sites.

The heat map indicates temperature variations across the National Park, where:

- Red represents the hottest regions,
- Yellow corresponds to moderate temperatures,
- Blue shows the coolest areas, and
- Dark blue represents areas with no available data.

From the visual comparison of the heat data with the estimated nesting intensity, we observe that gorillas tend to avoid the hottest areas. There are fewer nests in the red regions, suggesting that gorillas prefer moderate temperatures for nesting. This pattern holds true in both the rainy and dry seasons, as the nests are predominantly located in yellow regions, indicating moderate heat levels.

Ratio of points where intensity is above the threshold and Modest = 0.9414

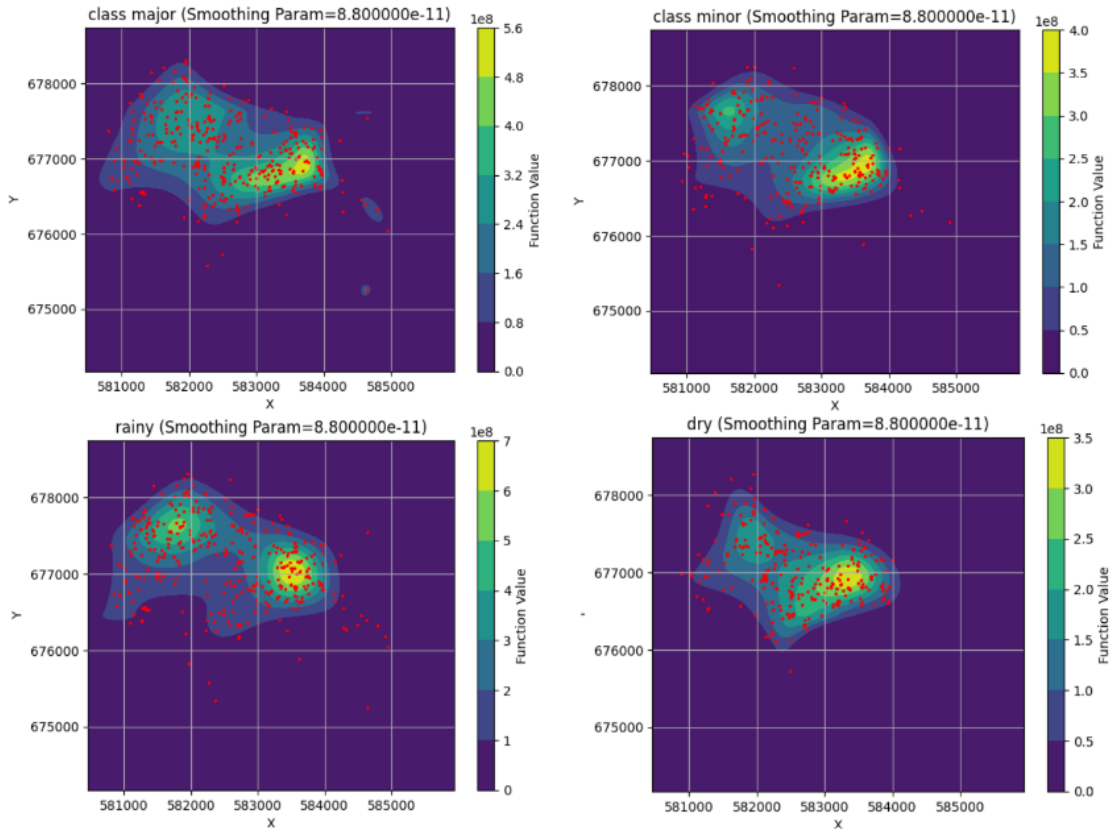

Figure 18. Intensity estimation for different classes and seasons.

Vegetation and Intensity

In addition to analyzing the heat data, I also plotted the vegetation data and aligned it with the estimated intensity function of gorilla nesting sites. The vegetation map is shown below, with various types of land cover represented by different colors.

- Yellow: Transition vegetation
- Green: Secondary forest
- Teal: Primary forest
- Blue: Grassland
- Purple: Colonising forest
- Dark Purple: Disturbed areas

By comparing the estimated intensity of gorilla nesting sites with the vegetation types, we observe that gorillas predominantly chose to nest in areas classified as **primary** and **secondary forests**. These regions provide suitable conditions for gorillas, such as ample food sources, shelter, and protection from disturbances.

Ratio of points above the threshold in 'Primary' vegetation areas: 0.94653

Distance to Nearest Water

In this analysis, I used GAMs to model the intensity of gorilla nests based on their distance to the nearest water source. The left panel shows the distance to water across the entire study region, where darker areas indicate proximity to water, and yellow areas are farther from water sources. The dark blue areas represent regions with no available data.

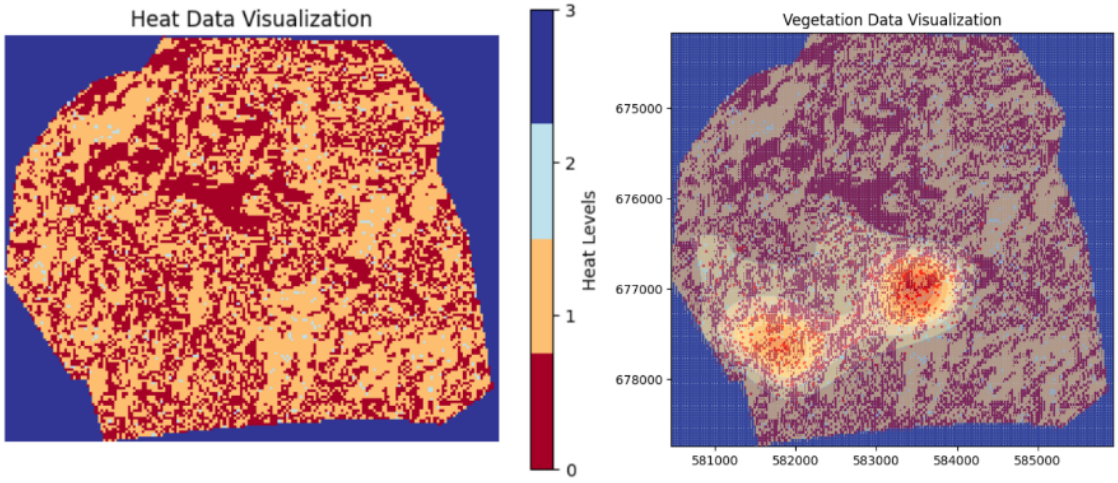

Figure 19. Heat data visualization (temperature levels) and aliened plot of intensity and heat data

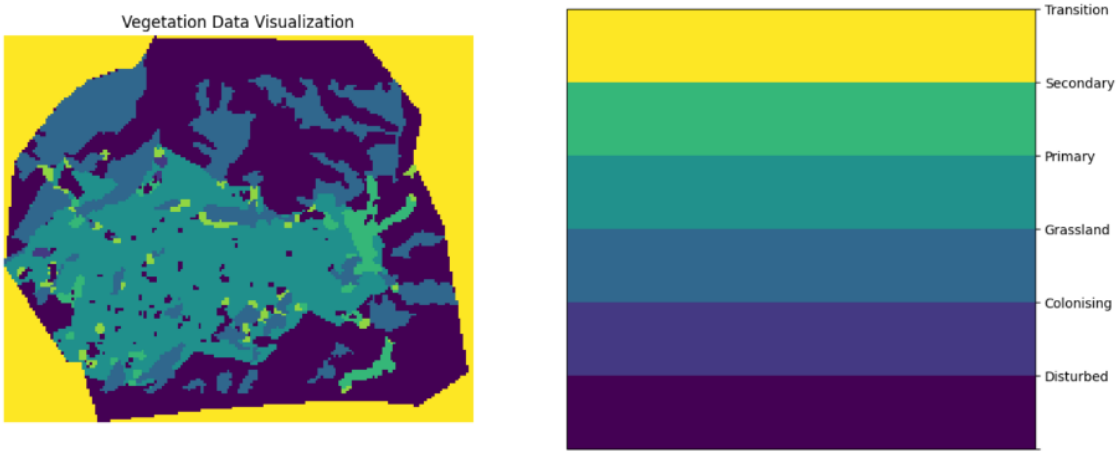

Figure 20. Vegetation Data Visualization.

On the right, the density function plot compares the distribution of nest distances to water during the rainy and dry seasons. From the plot, we can observe that gorillas generally prefer to nest closer to water sources. This behavior is especially pronounced during the dry season, where the probability of nesting near water is higher. In contrast, during the rainy season, the distance distribution is wider, indicating that the distance to water is less of a limiting factor. The wider curve indicates that gorillas are more flexible in their nesting locations when water is more readily available in the environment.

- Standard deviation in the rainy season: 79.10 meters
- Standard deviation in the dry season: 78.88 meters

The smoothing of the density plots could be further improved by selecting an optimal smoothing parameter using the cross-validation method described earlier. This would allow for a more accurate representation of nesting intensity in relation to water proximity.

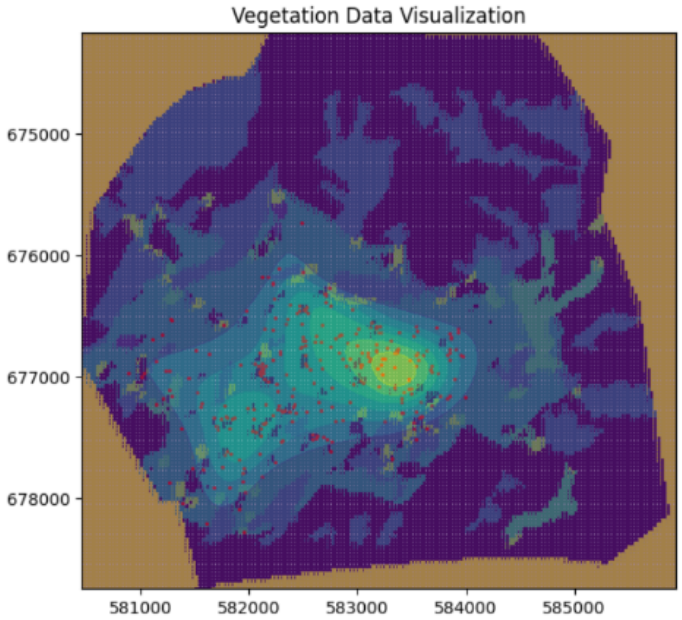

Figure 21. Vegetation Data Visualization aligned with estimated intensity of gorilla nests.

5. Future Work

Significant effort was made to optimize the method and reduce training time from 8 hours to just 2 minutes by improving code efficiency and leveraging GPU acceleration. However, further improvements can be made:

- **Faster Optimization with Gradient Descent**: Currently, training is done using PyTorch, but incorporating the gradient of the negative log-likelihood (NLL) into the optimization process could speed up training even further. This would allow for more efficient minimization of the objective function.
- **Extending GAM Applications**: GAMs provide flexibility for capturing complex data patterns. A potential direction is to train multiple GAMs on different aspects of the dataset to estimate various intensity functions, each with its own variance and mean. This approach would help uncover localized patterns and improve the interpret-ability of the model. By training and analyzing multiple GAMs, we could capture subtle variations in the data and understand the significance of different regions or features.

These enhancements could further accelerate training and extend the power of GAMs, allowing for deeper insights into the data and broader applications across various fields.

GAM Applications in Biology

Generalized Additive Models (GAMs) offer significant potential in biological research, particularly when working with complex datasets where some influencing factors are known, and others remain unidentified. For example, consider a microscopy dataset where the observed image is influenced by multiple biological factors, but we only have information about a subset of those factors.

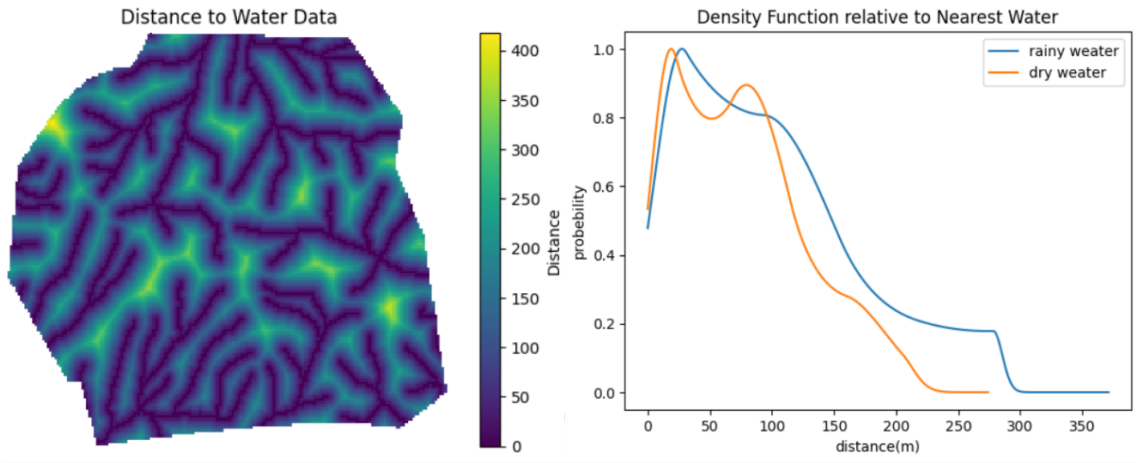

Figure 22. Left: Distance to water across the study region. Right: Density function of nest distance relative to the nearest water source during the rainy and dry seasons.

In such cases, using Generalized Linear Models (GLMs) can help estimate the effects of the known factors. However, GLMs fall short when it comes to capturing the effects of unknown or unmeasured factors. This is where GAMs come in as a powerful tool. GAMs allow for the estimation of unknown factors through their flexible structure. By incorporating smoothing parameters and the number of basis functions K, GAMs can adapt to the underlying data, allowing us to capture complex patterns that GLMs might miss. The smoothing parameter helps to balance the trade-off between fitting the data closely and maintaining generalizability, preventing over-fitting.

This flexibility makes GAMs a powerful tool for biological studies, where the interactions between factors may be intricate, and many contributing variables may remain unknown.

6. Conclusion

In this research, I developed an efficient and flexible approach using Generalized Additive Models (GAMs) to estimate complex intensity functions in spatial data. The method I developed for selecting the optimal smoothing parameter through cross-validation. This approach was shown to work effectively, ensuring that the model fits the data well while maintaining generalizability. Starting with Poisson point process data, I demonstrated how GAMs, with their smoothing parameters and basis functions, provide a powerful tool for capturing non-linear effects and uncovering hidden patterns that traditional models like GLMs cannot handle.

I applied the method to several datasets, including simulated Poisson data, Cox point processes, and real-world ecological data such as the gorilla nesting site dataset. GAMs proved effective in estimating the influence of various covariates like vegetation type, distance to water, and heat levels on gorilla nesting patterns. The models revealed meaningful insights, such as the preference of gorillas for primary and secondary forests and their tendency to nest closer to water during the dry season.

Overall, this research shows that GAMs are not only effective in modeling complex spatial data but also offer potential for further applications in fields such as ecology

and system biology, where flexibility and robustness are crucial.

References

[1] B. D. Youngman and T. Economou, "Generalised additive point process models for natural hazard occurrence," *Environmetrics*, vol. 28, no. 3, pp. e2444, Jan. 2017. DOI: 10.1002/env.2444.

[2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., Springer Series in Statistics, 2009, ISBN: 978-0-387-84857-0.

[3] S. N. Wood, N. Pya, and B. Säfken, "Smoothing Parameter and Model Selection for General Smooth Models," *Journal of the American Statistical Association*, vol. 111, no. 516, pp. 1548–1563, 2016. DOI: 10.1080/01621459.2016.1180986.

[4] J. Møller and R. P. Waagepetersen, "Modern statistics for spatial point processes," *Scandinavian Journal of Statistics*, vol. 34, pp. 643–684, 2007. DOI: 10.1111/j.1467-9469.2007.00569.x.

[5] S. N. Wood, *Generalized Additive Models: An Introduction with R*, CRC Press, 2006.

[6] A. Baddeley, R. Turner, and E. Rubak, *Datasets Provided for spatstat*, spatstat version 3.1-1, 2020.