2023-08-01

Single-Index Multinomial Model for Analyzing Crime Data

Kwabena Gyamfi Duodu
University of Texas at El Paso

SINGLE-INDEX MULTINOMIAL MODEL FOR ANALYZING CRIME DATA

KWABENA GYAMFI DUODU

Master's Program in Statistics and Data Science

APPROVED:

Suneel Babu Chatla, Ph.D. Chair

Abhijit Mandal, Ph.D

Deepak K. Tosh, Ph.D

Stephen Crites, Ph.D
Dean of the Graduate School

I want to sincerely thank my cherished family and friends for their constant support, motivation, and inspiration during my academic career. I am appreciative of my parents' unwavering love, support, and confidence in me. Additionally, I would like to extend my profound gratitude to Dr. Suneel Babu Chatla, who served as my thesis adviser, for his priceless counsel, direction, patience, and constructive criticism. Without the support, love, and contributions of each and every one of these extraordinary people, this thesis would not have been possible. I sincerely appreciate each and every one of you, and I say thank you.

SINGLE-INDEX MULTINOMIAL MODEL FOR ANALYZING CRIME DATA

by

KWABENA GYAMFI DUODU

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Mathematical Sciences

THE UNIVERSITY OF TEXAS AT EL PASO

August 2023

Acknowledgement

The distinguished members of my thesis committee, Drs. Suneel Babu Chatla, Deepak K. Tosh, and Abhijit Mandal, have my sincere gratitude for their invaluable advice, insightful criticism, and unflinching support during the whole research process. Their extensive knowledge and experience have been important in determining the course of this thesis.

Abstract

We develop a flexible single-index multinomial model for analyzing crime data. In addition to the number of crimes reported, the data also includes covariates such as location, time of day, weather, and other demographic factors. We provide an estimation algorithm and develop R code for the single-index multinomial model. Using simulations, we evaluate the performance of the proposed estimation algorithm. When applied to crime data, the single-index multinomial model provides important insights into crime trends and risk variables, assisting in the development of tailored crime prevention programs. Policymakers and law enforcement organizations can use the model's projections to more efficiently allocate resources and design preemptive strategies to solve crime-related concerns. Finally, the single-index multinomial model demonstrates itself to be a reliable tool for assessing crime data and improving knowledge and management of crime occurrences in varied areas.

Keywords: Single-index model, High-dimensional data, Generalized additive model, Crime data

Contents

Page

List of Tables

List of Figures

Chapter 1

Introduction

Crime analysis plays a vital role in understanding and combating criminal activities in society. Law enforcement agencies, policymakers, and researchers strive to develop effective strategies to address the complex dynamics of crime. Statistical models have grown in popularity in recent years, and they have proven to be effective tools for assessing crime data and finding major elements that contribute to criminal behavior. One such model, the Single-Index Model (SIM), offers a promising approach to uncover the underlying relationships between various socioeconomic factors and crime rates.

Single-Index Model is a popular regression technique that assumes a nonlinear relationship between a response variable and a single-index variable. This index variable is a weighted mixture of predictor variables which, for example, represent different socioeconomic factors related to crime in the crime data. Researchers can use the SIM to analyze many predictor variables at the same time and capture their cumulative influence on the response variable. In single-index model, the object of interest depends on X through $g(X'\alpha)$ where α and $g : \Re \to \Re$ are unknown and ϵ is a random noise. A link function is the function $g(\cdot)$. The semi-parametric single-index model (Han et al. 2020) is defined as

$$Y = g(X'\alpha) + \epsilon$$

which, in other words, can be expressed as

$$E(Y|X) = g(X'\alpha)$$

where Y is a continuous response variable. Additional details are provided in Chapter 2.

The goal of this study is to propose an estimation algorithm when Y is multinomial response variable. The proposed model is more flexible as it includes additive component functions along with the single-index function. Using simulations, we evaluate the performance of the proposed estimation approach.

Finally to illustrate the usefulness of the proposed method, we apply it on a dataset containing crime data, demographic data, economic indicators. The single-index model will serve as the primary analytical framework for this study. By estimating the model parameters, we can quantify the impact of various socioeconomic factors on crime rates, while controlling for confounding variables. Moreover, the SIM will facilitate the development of a comprehensive crime prediction model, allowing for the identification of areas at higher risk and the formulation of targeted intervention strategies.

Last but not least, this thesis will add to the existing body of literature on crime analysis by providing insights into the complex dynamics of criminal behavior and its relationship to socioeconomic issues. The research findings will help legislators, law enforcement agencies, and urban planners make informed decisions to reduce crime rates and promote safer cities.

Chapter 2

Background

This section provides a brief background to the methods discussed in the study. First, it starts with the single-index model and then provide details about the generalized additive model and multinomial generalized additive model.

2.1 Single-Index Model

The single-index model is an extension of the generalized linear model that depicts the connection between the response variable and a single-index (Hastie & Tibshirani 1990). It has received a lot of attention in the field of dimensionality reduction and efficient parameter estimation (Fan & Li 2001).

Let Y be a scalar random variable and X be a $p \times 1$ random vector. The single model has the form

$$E(Y|X) = g(X'\alpha) \tag{2.1}$$

where α is a parameter vector of $p \times 1$ and $g(\cdot)$ is an unknown function. The amount $X'\alpha$ is known as an index. If $g(\cdot)$ is cumulative normal or logistic distribution function then the model becomes a binary probit or logit model. When g is unknown, the above model provides a specification that is more flexible than a parametric model but retains many of the desirable features of a parametric model. Flexibility is important in applications because there is usually little justification for assuming that g is known a *priori*, and seriously misleading results can be obtained if one makes an incorrect g specification. Use of semi-parametric single-index model reduces the risk of obtaining misleading results.

A single-index model achieves dimension reduction and avoids the curse of dimensionality because, the index $X'\alpha$ aggregates the dimension of X. As a result, g in a single-index model can be estimated with the same rate of probability convergence as if the one-dimensional quantity $X'\alpha$ were observable. Furthermore, α can be estimated with the same rate of convergence, $n^{-1/2}$, as in a parametric model. In terms of probability rate of convergence, the single-index model is as accurate as a parametric model for estimating α and as accurate as a one-dimensional non-parametric mean regression for estimating g. This dimension-reduction feature of single-index models gives them a considerable advantage over non-parametric methods in applications where X is multidimensional and the single-index structure is plausible.

The assumptions of a single-index model are weaker than those of a parametric model but stronger than those of a fully non-parametric model. In comparison to a parametric model, a single-index model reduces the chance of misspecification. While avoiding some of the disadvantages of fully non-parametric techniques, such as the curse of dimensionality, interpretation difficulty, and lack of extrapolation capabilities. There is a significant exception to classifying a single-index model as intermediate or as having lesser assumptions than a non-parametric model. This exception happens when structural economic models are estimated. A structural model is one in which the components have a clear relationship to economic theory. It turns out that the constraints required to allow for a structural interpretation of a non-parametric model can reduce the non-parametric model's generality to that of a single-index model.

Before estimating α and g, restrictions must be imposed to ensure their identification. That is, α and g must be uniquely determined by the population distribution of (Y, X). Identification of single-index models has been investigated by Ichimura (1993) and, for the special case of binary-response models, by Manski (1988). Interpreting SIMs and making statistical inferences pose unique challenges. Multicollinearity can affect the estimation of the index coefficients, leading to unstable parameter estimates (Wang et al. 2013). When the linearity assumption is violated, model misspecification can arise, necessitating the use of diagnostic techniques to assess model fit. Simulation studies and comparative analysis have been used to assess SIMs' performance in contrast to other statistical models. These research have demonstrated that SIMs can deliver competitive predicted accuracy while also providing interpretability and simplicity when compared to more complicated models such as neural networks. Despite the advancements in SIM methodology, challenges remain. Nonlinearity, high-dimensional data, and missing data present ongoing challenges in SIM modeling. Future research directions should focus on developing robust estimation techniques, addressing model assumptions, and exploring applications in emerging areas such as network analysis and spatio-temporal modeling.

2.2 Generalized Additive Model

The Generalized Additive Model (GAM) is a versatile statistical modeling framework that supports non-linear connections between predictors and responses. Hastie (1986) introduced GAMs as an extension of the Generalized Linear Model (GLM) that integrates non-parametric smooth functions of the predictors.The multiple linear regression model can be extended using a generalized additive model (GAM) [James et al., 2021]. The general form of a GAM is:

$$E(Y) = g(\beta_0 + f_1(X_1) + f_2(X_2) + ... + f_d(X_d)) \tag{2.2}$$

where $g(\cdot)$ is a specified link function. $E(Y)$ represents the expected value of the response variable Y, β_0 is the intercept term, and $f_j(\cdot), j = 1, \ldots, d$, are smooth function of the predictor variables of X_j. An exponential family distribution is specified for Y (for example normal, binomial or Poisson distributions), where Y relates a univariate response variable to some predictor variables. The flexibility of GAMs lies in the choice of smoothing functions. These smooth functions allow for capturing complex and non-linear relationships between predictors and the response variable without imposing rigid assumptions. Commonly used smoothing techniques in GAMs include splines, local regression, kernel smoothing, and smoothing splines.

2.2.1 Multinomial Generalized Additive Model:

This can be written in a similar form to the GAM, but with the inclusion of multiple outcome categories for the response variable. The response variable represents a categorical outcome with more than two levels. The purpose of the MGAM is to model the probabilities of each category of the response variable as a function of predictor variables.Each category represents a distinct outcome or class.

For the response variable which stems from the exponential family, it has K categories ($K > 2$), denoted as Y, and p predictor variables. The MGAM can be expressed as:

$$logit(p(Y = j)) = \beta_{0j} + f_1(X_1) + f_2(X_2) + ... + f_d(X_d) \tag{2.3}$$

for $j = 1, 2, ..., K - 1$, where $P(Y = j)$ represents the probability of the response variable taking the $j - th$ category. The logit transformation (log-odds) is commonly used to link the predictor functions with the probabilities.

The goal of the MGAM is to model the probabilities of each category for a given set of predictor variables. The probabilities are usually modeled using a logit link function, which transforms the linear combination of predictors into the logarithm of the odds of each category. The logit transformation (log-odds) is commonly used to link the predictor functions with the probabilities. The logit transformation of the response variable can be denoted as $log\left(\frac{p_j}{1-p_j}\right)$ where p_j is the probability for jth each category.

The fitting procedure for the Multinomial Generalized Additive Model (MGAM) involves estimating the model parameters that maximize the likelihood or minimize a penalized likelihood criterion specific to multinomial responses. While there is no universally standardized approach for fitting MGAMs, several methods and algorithms have been proposed. The estimation procedure for MGAMs typically involves two steps: estimation of smooth functions and estimation of category probabilities.

Estimation of smooth functions $f_j(\cdot)$:

- Specify the form of the smooth functions,$f_1(X_1)$, $f_2(X_2)$, ..., $f_d(X_d)$ for each predictor variable, $X_1, X_2, ..., X_d$ (e.g., splines, smoothing splines, or other basis functions).

- Estimate the smooth functions separately for each category of Y(K categories) while holding other terms fixed.
- Choose an appropriate estimation method based on the chosen smooth function type and available software packages (e.g., penalized regression, backfitting algorithm, or other optimization techniques).

Estimation of category probabilities:

- Use the estimated smooth functions to calculate the category probabilities for each observation.
- Employ appropriate multinomial regression techniques, such as maximum likelihood estimation, to estimate the category probabilities based on the observed data.

It should be noted that the specifics of fitting a GAM may differ based on the program or modeling framework employed. To estimate the smooth functions and model parameters, many software packages provide different algorithms and optimization techniques. Overall, the GAM fitting method entails iteratively estimating the smooth functions and updating the linear terms until a sufficient model fit is obtained.

Chapter 3

Single-Index Multinomial Model

We propose the single-index multinomial model as a statistical model method for analyzing a categorical response variables with more than two categories. It is an extension of the single-index model to the multinomial setting. Suppose we have a categorical response variable Y that can take K different levels or categories, labeled as $1, 2, ..., K$. Additionally, we have a set of predictor variables or covariates represented as $X_1, X_2, ..., X_p$.

The proposed method can be thought as a combination of additive model and single-index model. Generally, estimating a nonparametric function with a small dimension, say 2 or 3, is not a problem as it is done my most the of the existing software programs. However, if X is of higher dimension, most of those methods will breakdown. This problem is called curse of dimensionality. On the other hand, if the data includes many categorical variables, nonparametric model is not adequate. To make our model more flexible, we propose a semiparametric model blending the parametric function with the nonparametric function. Here, the parametric functions uses the single-index approach whiles the non-parametric functions uses the GAM model with the help of the mgcv package in R.

3.1 Model Formulation

We define the single-index multinomial model(SIMM) as

$$P(y = j|X, z_1, z_2, \dots, z_d) = g\left(f_{0j}(X^T\alpha) + f_{1j}(z_1) + f_{2j}(z_2) + ... + f_{dj}(z_d)\right), \qquad (3.1)$$

for $j = 1, \dots, K$, where

- y: This is the categorical response variable, which takes values from a set of K categories (labeled as 1, 2, ..., K). The variable y represents the outcome we want to predict or model.
- $g(\cdot)$ is a multinomial logistic function, $g(w_j) = \exp(w_j)/\sum_{j=1}^{K}\exp(w_j)$.
- X : This is a matrix of predictor variables, with each row representing a data point and each column representing a different predictor. X' denotes the transpose of the matrix X.
- α : This is a vector of coefficients corresponding to the predictor variables in X.
- $f_{0j}(\cdot)$: This term represents a smooth function of the single-index $X'\alpha$. This function is modeled nonparametrically.
- $z_1, z_2, ..., z_d$: These are additional covariates or predictors that are not part of the matrix X. They could be continuous or categorical variables that influence the response variable y but are not part of the linear predictor $X'\alpha$.
- $f_{1j}(z_1), f_{2j}(z_2), ..., f_{dj}(z_d)$: These terms represent smooth functions of the additional covariates $z_1, z_2, ..., z_d$. Similar to $f_0(\cdot)$, these functions are smooth and modeled nonparametrically.

Overall, the equation represents a flexible model that allows for nonlinear relationships between the predictors and the multinomial response. The goal of fitting this single-index multinomial model is to estimate the single-index coefficients α and the smooth functions $f_{0j}(\cdot), f_{1j}(\cdot), ..., f_{dj}(\cdot)$ that best describe the relationship between the predictors and the categorical response variable y. In the following, we provide a pseudo algorithm for model estimation.

3.2 Algorithm for Model Estimation

The estimation of the single-index multinomial model includes the following steps:

- Start with an initial $\alpha^{(0)}$.

- Use optim function in R to find $\widehat{\alpha}$ based on the grid search for α. The solution minimizes the gcv.ubre criterion when additive model is fitted with $X'\alpha^{grid}$, z_1, …, z_d using mgcv package in R. Denote the solution as $\alpha^{(final)}$.

- Compute $X'\alpha^{(final)}$ and fit an additive model to compute $\widehat{f}_{0j}, \ldots, \widehat{f}_{dj}$, $j = 1, \ldots, K-1$. We take $\widehat{\alpha} = \alpha^{(final)}$.

The R code for estimation is provided in the Appendix.

Chapter 4

Simulation Study

4.1 Simulation Design and Results

We conduct a simulation study to assess the performance of the proposed method. It is aimed at evaluating the accuracy and efficiency of the algorithm in estimating the parameters under different scenarios. The single-index covariates X_1, X_2, X_3 are generated from the standard uniform distribution. Another covariate Z is generated from the standard normal distribution. Let $\alpha = (0.1, -0.2, 0.1)$ and fix the number of categories in the response variable Y as three. The data generation includes the following steps:

- Compute two linear predictors η_1 and η_2 such that the ith element

  $$\eta_{i1} = sf_{i1} + f_2(Z_i), \qquad \eta_{i2} = sf_{i2} + f_3(Z_i)$$

  where $sf_{i1} = f_1((\sum_{j=1}^{3} X_{ij}\alpha_j + 0.41)/4)$, $sf_{i2} = f_4(\sum_{j=1}^{3} X_{ij}\alpha_j)$ with

  $$\begin{aligned} f_1(x) &= 0.2x^{11}(10(1-x))^6 + 10(10x)^3(1-x)^{10}, \\ f_2(x) &= \sin(3\pi x)\exp(-x), \\ f_3(x) &= x^3 \\ f_4(x) &= \sin(2\pi x). \end{aligned}$$

- Compute the probability matrix $p = \exp(0, \eta_1, \eta_2)$ and divide each row by their row sum. Calculate the cumulative values for each row and denote it by matrix cp.

- Generate a uniform random number and consider $Y_i = index - 1$ where $index$ denotes which ever the first value it exceeds in each row.

- Finally, the simulated response variable Y with $K = (0, 1, 2)$ levels of categories.

The simulation process is repeated 20 times for different sample sizes 200, 500, and 1000. Each iteration of the simulation involves fitting the multinomial single-index Model and calculating the MSE. The resulting MSE values are then aggregated to calculate the average MSE for each smooth term across the 20 iterations. The Mean Square Error (MSE) is defined as

$$MSE = n^{-1} \sum_{i=1}^{n} (m_i - \widehat{m}_i)^2,$$

where m_i represents each of sf_{i1}, sf_{i2}, $f_2(Z_i)$ and $f_3(Z_i)$. For this purpose, the model was applied to predict the smooth terms, and the differences between the predicted and true values were computed. The R code use for this simulation can be found in the appendix. The MSE values are provided in the following table:

Mean Square Error(MSE)	**N=200**	**N=500**	**N=1000**
MSE.1	1.5520623	1.2145308	1.4299910
MSE.2	0.7022928	0.3172879	0.3379416
MSE.3	4.8810909	3.9086439	3.3457132
MSE.4	3.4953102	3.4301909	3.7679205

Table 4.1: MSE values for the single-index model for n = 200,500 and 1000

Now we also provide three performance measures, the accuracy, κ statistic, and the *Pabak* statistic to evaluate the performance of the model. The accuracy is defined as

$$Accuracy = n^{-1} \sum_{i=1}^{n} 1(Y_i == \hat{Y}_i),$$

where $1(\cdot)$ is the indicator function, from the proposed single-index model in Table 5.3.

The κ statistic, which denotes the agreement beyond chance alone, is defined as

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

where

- P_o is the observed agreement (proportion of cases where both raters or classifiers agree).
- P_e is the expected agreement by chance, which is calculated as the product of the marginal proportions of agreement for each category (Fleiss et al. 2013).

In general, κ values ranges from -1 to 1.

- $\kappa < 0$ indicates less agreement than expected by chance
- $\kappa = 0$ indicates agreement equal to what would be expected.
- $0 < \kappa < 1$ indicates agreement beyond what would be expected by chance.
- $\kappa = 1$ indicates perfect agreement

We also provided the prevalence adjusted and bias adjusted κ statistic (*pabak*), this corrects for imbalances induced by variances in prevalence and bias in the data (Byrt et al. 1993). The *pabak* can be denoted as :

$$pabak = \frac{P_o - P_e}{1 - P_e} + \frac{P_e(1 - P_e)}{P_c(1 - P_c)}$$

where

- P_o and P_e are same as in κ statistic
- P_c denotes the positive category's prevalence (the proportion of positive examples in the data).

For comparison, we also fit a regular multinomial GAM for the data and provide its accuracy, Kappa, and pabak values. The GAM is fitted with the following model

$$E(Y = j|Z, X_1, X_2, X_3) = g(\alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X_3 + f(Z)).$$

The results in Table 5.3 indicate that the proposed single-index method is more flexible compared to regular GAM.

Model	**Accuracy**	**Kappa**	**pabak**
Single-Index	0.755	0.62634	0.51
GAM	0.76	0.63396	0.52

Table 4.2: Average accuracy, Kappa and pabak values for the single-index and the regular additive models for samples sizes n = 200

Model	**Accuracy**	**Kappa**	**pabak**
Single-Index	0.736	0.59442	0.472
GAM	0.732	0.58612	0.464

Table 4.3: Average accuracy, Kappa and pabak values for the single-index and the regular additive models for samples sizes n = 500

Model	**Accuracy**	**Kappa**	**Pabak**
Single-Index	0.731	0.59132	0.462
GAM	0.719	0.57272	0.438

Table 4.4: Average accuracy, Kappa and pabak values for the single-index and the regular additive models for samples sizes n = 1000

Chapter 5

Chicago Crime Data

5.1 Data source and description

The data for this study is compiled from two sources: kaggle and data.cityofchicago.org. It includes information about the crimes of Chicago along with some weather related information. To include the spatial information, the longitude and latitude of 277 police beat are extracted from the city of Chicago data portal and mapped them to the various crimes happening in each beat. Since the coordinates of each beat comprises a polygon, the median of the coordinates of each polygon police beat is calculated to get one single longitude and latitude. The combined data has 26 columns and 187989 observation for the year 2021. Among the 26 columns, primary-type is the response variable. While there are 31 different crimes recorded under primary-type, the the number of categories is reduced to 6 levels for simplifying the analysis.

5.1.1 Multinomial Response Variable

Table 5.1 shows the grouping of 31 different crimes into 6 groups. Each group is categorized based on general understanding of the crime in its entirety. Meaning, similar crimes are grouped under one umbrella. Hence, our new response variable has only 6 levels.

Group1	**Group2**	**Group3**	**Group4**	**Group5**	**Group6**
Weapon Violation	Obscernity	Deceptive Practice	Burglary	Assault	Offense Involving Children
Conceal Carry License Violation	Criminal Sexual Assault	Gambling	Motor Vehicle Theft	Battery	Human Trafficking
Liquor Law Violation	Prostitution	Non-Criminal	Robbery	Intimidation	Kidnapping
Other Narcotic Violation	Public Indecency	Interference with Public Officer	Theft	Homicide	Other Offense
Public Peace Violation	Sex Offense	None	Criminal Damage	Narcotics	None
Criminal Trespass	Stalking	None	Arson	None	None

Table 5.1: categorization of the response variable into 6 levels

5.1.2 Variables Description

variable	**description**
'Primary-type'	type of crime
'Date'	Date when the incident occurred
'Hour'	Hour when the incident occurred
Month'	month
' DayOfWeek'	day of the week
'Dholiday'	dummy of official us holiday
'Location-description'	Description of the location where the incident occurred
'Beat'	police geographic area where the incident occurred
'Ward'	City Council district where the incident occurred
'HubDist'	distance between the location of the incident and the nearest police station
'PRCP'	Precipitation
'SNOW'	Snowfall
'SNWD'	Snow depth
'TMAX'	Maximum temperature
'TMIN'	Minimum temperature
'WDF2'	Direction of fastest 2-minute wind
'WSF2'	Fastest 2-minute wind speed
'WT01' -	dummy of Fog, ice fog, or freezing fog
'WT02'	dummy of Heavy fog or heaving freezing fog
'WT03'	dummy of Thunder
'WT04'	dummy of Ice pellets, sleet, snow pellets, or small hail
'WT06'	dummy of Glaze or rime
'WT08'	dummy of Smoke or haze
'WT09'	dummy of Blowing or drifting snow
'Lati'	latitude of police beat where crime happened
'Longi'	Longitude of police beat where crime happened

Table 5.2: variable description of the dataset

Table 5.2 describes the available variables in the combined dataset. This will ensure that anyone working with the dataset understands the variables' meanings and can make informed decisions during data analysis and modeling.

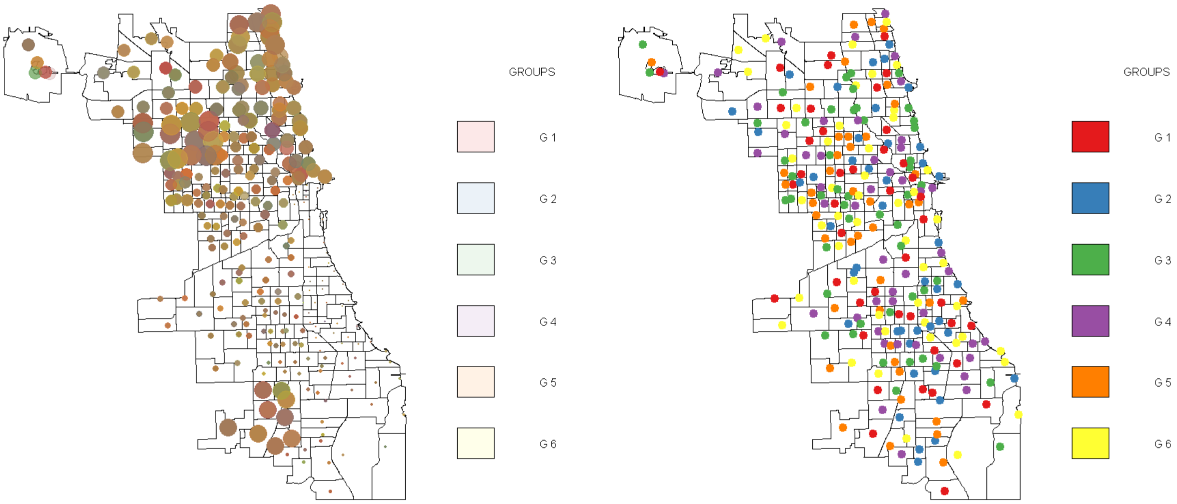

Figure 5.1: Map of Chicago city and group of crimes within the 277 police beat

Figure 5.1 shows the crimes for each type of crime in the city of Chicago. The left side figure shows the frequency of these crimes in their respective location and the figure on the right side shows the category of crimes in the city of Chicago in the month of January 2021. In other words, the left side plot is shedding more light, in the sense of total number of occurrences of the category of crimes.

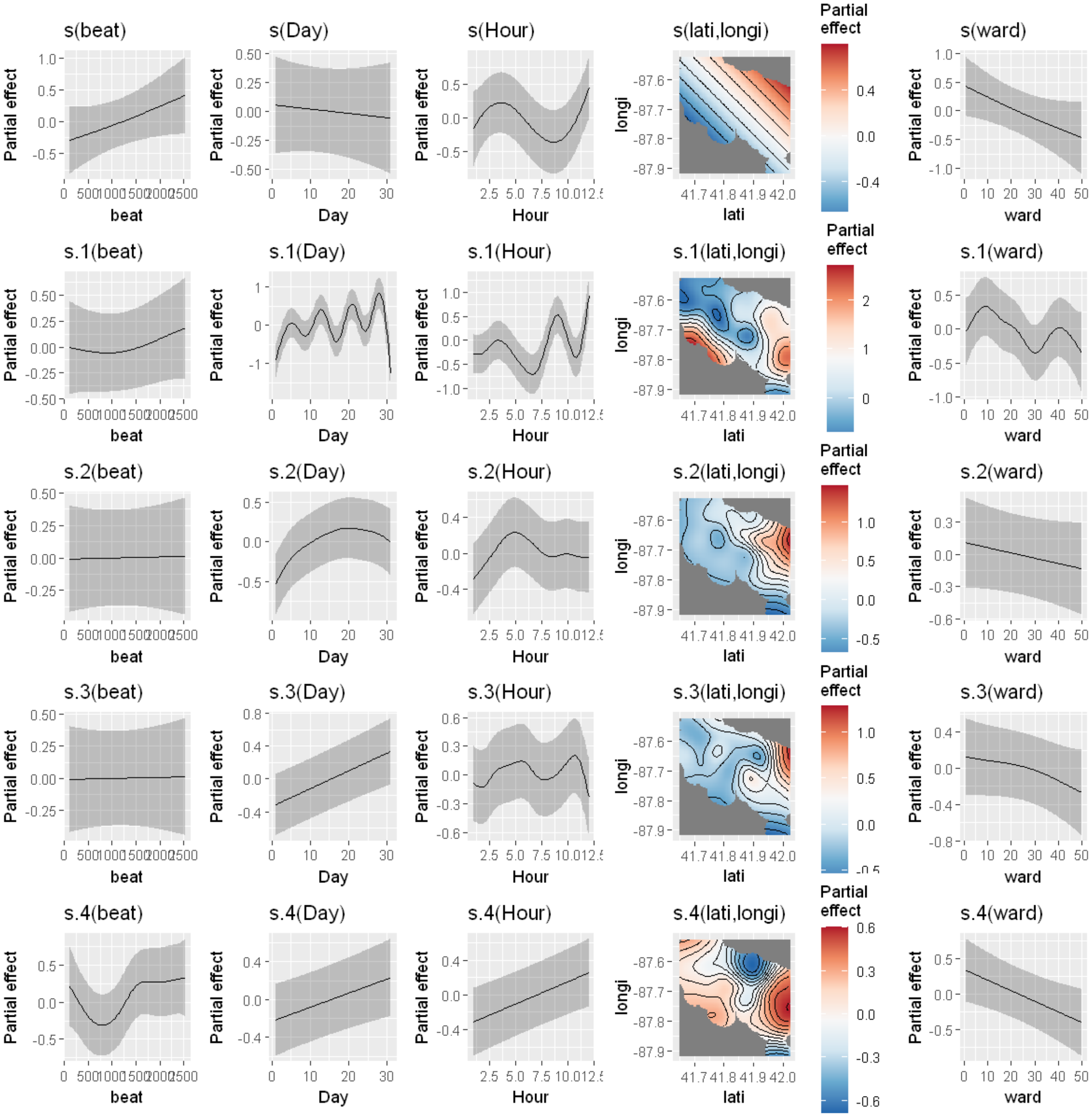

Figure 5.2: Crime Trends

Figure 5.2 contains the output from the additive model. The covariates beat, day, hour, latitude and longitude and ward are modeled as nonparametric effects. We note that each of these variables exhibit different behavior for each crime type.

5.2 Application of Real Data on Models

We fit a single-index multinomial model to the Chicago city crime and weather data, where our response variable is primary-type. As shown in Chapter 3, the proposed method comprises of both the single-index part and the GAM part. The goal of this method is to understand the relationship between the weather conditions and the crime type. Since the the data is large and the model is computationally heavy, only partial data (1000 observations) from week 1 is used to train the model. Further, the number of crime types has been restricted to only 4.

To evaluate the performance of the proposed model, another 500 observations from week 2 are considered as test data. The accuracy, Kappa and Pabak values of the predictions generated from the test data are used in the evaluation. For comparison, the accuracy, Kappa and Pabak estimates of the regular additive is also computed. The predictions from both the proposed and the regular additive model on test data are presented as a bar chart in Figure 5.3.

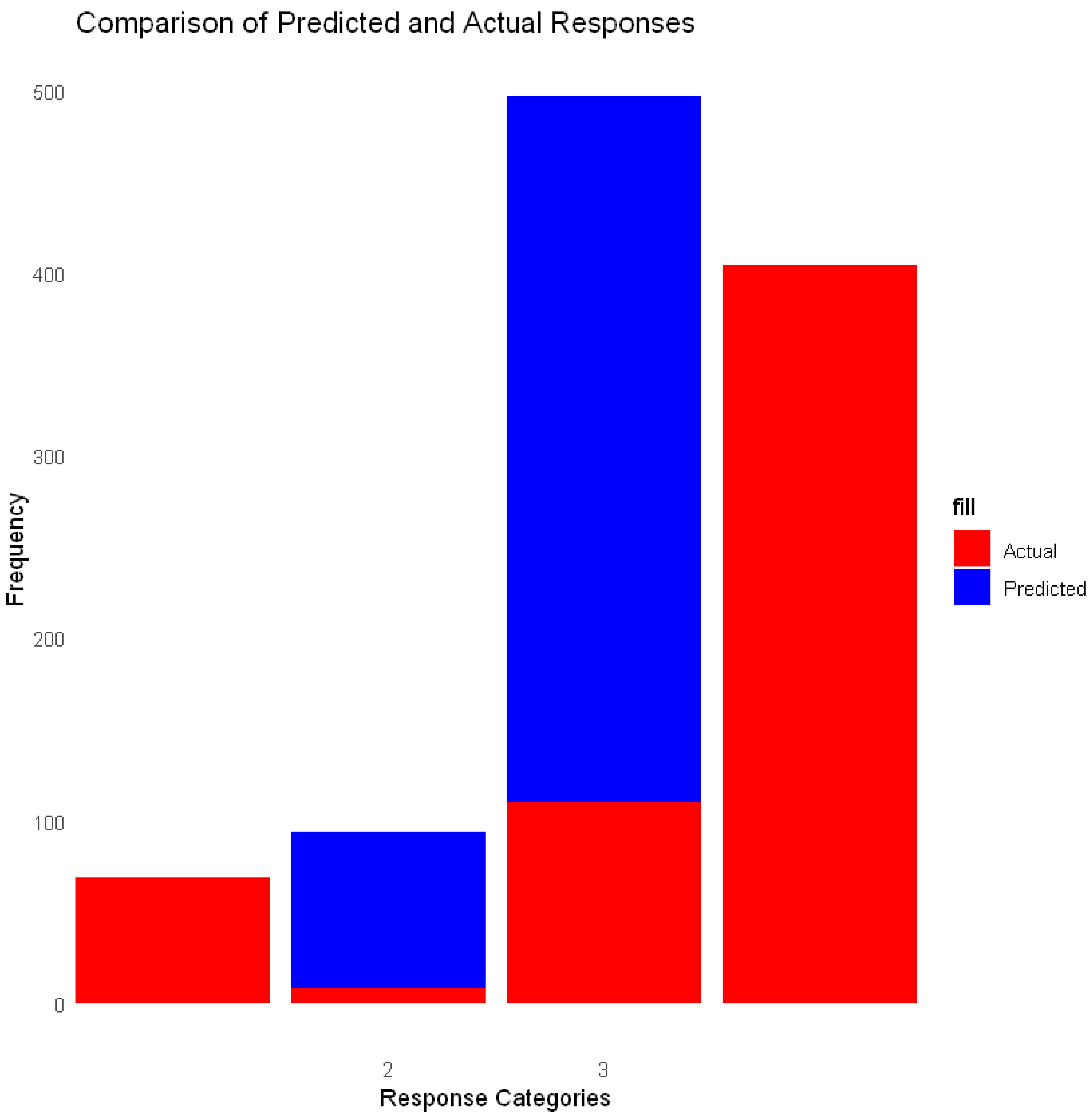

Figure 5.3: Comparison of actual and predicted response from the proposed model on the test data

Figure 5.3 shows the comparison of the actual response variable with the predicted response in respect to the four level of categories. The model accuracy, Kappa and Pabak results are summarized in the following table.

Model	**Accuracy**	**Kappa**	**pabak**
Single-Index	0.65821	0.13587	0.31641
GAM	0.63458	0.11047	0.26904

Table 5.3: Accuracy, Kappa and pabak estimats for the proposed single-index and the regular additive models for the crime data.

To sum up, the proposed multinomial single-index model provides more flexibility and achieves better predictive accuracy, a better Kappa's estimate and a better pabak estimate. On the flip side, it is computationally expensive. There is a need for computationally efficient method and we defer this for future research.

Chapter 6

Conclusion

Single-index model is a powerful model and studied well in the literature. In this study, we proposed and implemented a multinomial single-index model which is more flexible than the regular additive model and can overcome the curse of dimensionality. We evaluate its performance using simulations and a real application. In both cases, we demonstrated that it outperforms the regular additive model in terms of the predictive performance.

While the proposed method displays superior performance than the regular additive model, it certainly has some limitations. One of the most notable limitations is its computational cost. Even with 1 week data, it took some day to obtain the output. Since the model involves many parameters, the existing optimization libraries struggle to work fast. A more efficient algorithms are needed to overcome this problem. We defer this issue for future research.

Bibliography

Byrt, T., Bishop, J. & Carlin, J. B. (1993), ‘Bias, prevalence and kappa’, *Journal of clinical epidemiology* **46**(5), 423–429.

Fan, J. & Li, R. (2001), ‘Variable selection via nonconcave penalized likelihood and its oracle properties’, *Journal of the American statistical Association* **96**(456), 1348–1360.

Fleiss, J. L., Levin, B. & Paik, M. C. (2013), *Statistical methods for rates and proportions*, john wiley & sons.

Han, Z.-C., Lin, J.-G. & Zhao, Y.-Y. (2020), ‘Adaptive semiparametric estimation for single index models with jumps’, *Computational Statistics & Data Analysis* **151**, 107013.

Hastie, T. (1986), ‘Hastie t., tibshirani r’, *Generalized additive models, Statistical Science* **1**(3), 297–310.

Hastie, T. J. & Tibshirani, R. J. (1990), *Generalized additive models*, Vol. 43, CRC press.

Ichimura, H. (1993), ‘Semiparametric least squares (sls) and weighted sls estimation of single-index models’, *Journal of econometrics* **58**(1-2), 71–120.

Manski, C. F. (1988), ‘Identification of binary response models’, *Journal of the American statistical Association* **83**(403), 729–738.

Wang, D., Liang, S., He, T. & Yu, Y. (2013), ‘Direct estimation of land surface albedo from viirs data: Algorithm improvement and preliminary validation’, *Journal of Geophysical Research: Atmospheres* **118**(22), 12–577.

Appendix

R CODE FOR SIMULATION

```
library(mgcv)
library(mltools)

si <- function(theta, xmat,fix.formula, tdata, opt=TRUE, k=k,fx=fx, levK=levK
## Fit single index model using gam call, given theta (defines alpha).
## Return ML if opt==TRUE and fitted gam with theta added otherwise.
## Suitable for calling from 'optim' to find optimal theta/alpha.
  alpha <- c(1,theta) ## constrained alpha defined using free theta
  k=5
  kk <- sqrt(sum(alpha^2))
  alpha <- alpha/kk  ## so now ||alpha||=1
  tdata$a <- xmat%*%alpha       ## argument of smooth
  formula.updated <- eval(lapply(fix.formula, function(x) update(x, ~.+ s(a,
  e1 <- environment(alpha)
  b <- gam(formula.updated,data=tdata, family=multinom(K=levK)) ## fit model
  if (opt) return(b$gcv.ubre) else {
    b$alpha <- alpha  ## add alpha
    J <- outer(alpha,-theta/kk^2) ## compute Jacobian
    for (j in 1:length(theta)) J[j+1,j] <- J[j+1,j] + 1/kk
    b$J <- J ## dalpha_i/dtheta_j
    return(b)
  }
} ## si
```

```
sim.data.si <- function(n, seed=123,plot=FALSE)
{
  # setting random seed
  set.seed(seed)
  f1 <- function(x) 0.2 * x^11 * (10 * (1 - x))^6 + 10 *
                (10 * x)^3 * (1 - x)^10
  f2 <- function(x) sin(3*pi*x)*exp(-x)
  f3 <- function(x) x^3
  f4 <- function(x) sin(2*pi*x)

  m <- 3
  x <- matrix(runif(n*m),n,m) ## the covariates for the single index part
  z <- rnorm(n) ## another covariate
  alpha1 <- c(0.1, -0.2, 0.1); alpha1 <- alpha1/sqrt(sum(alpha1^2))
  #alpha1 <- c(0.1, -0.4, 0.1)
  # c(0.1, -0.2, 0.1)
   # 0.08, -0.2, 0.1
    #c(0.05, -0.2, 0.08)
    #c(0.05,-0.2,0.1)
  sf1 <- scale(as.numeric(f1((x%*%alpha1+.41)/4)),scale=FALSE)
  sf2 <- scale(as.numeric(f4(x%*%alpha1)), scale=FALSE)
  if(plot==TRUE)
  {
    par(mfrow=c(1,2))
    p1=plot(x%*%alpha1, sf1, ylab = "Function1", xlab = "Single Index1")
    p2=plot(x%*%alpha1, sf2, ylab = "Function2", xlab = "Single Index2")
  }
    f2f <- scale(f2(z), scale=FALSE)
    f3f <- scale(f3(z), scale=FALSE)
  eta1 <- sf1 + f2f
  eta2 <- sf2+ f3f

  p <- exp(cbind(0,eta1,eta2))
  p <- p/rowSums(p) ## prob. of each category
  cp <- t(apply(p,1,cumsum)) ## cumulative prob.
  ## simulate multinomial response with these probabilities
  ## see also ?rmultinom
  y <- apply(cp,1,function(x) min(which(x>runif(1))))-1

  #
  tdata <- data.frame(y=y,x1=x[,1], x2=x[,2], x3=x[,3],z=z)
  #
  return(list(tdata,sf1=sf1,sf2=sf2,f2=f2f,f3=f3f))
}

si.fit <- function(data, fn=si, sformula , fformula, k=5,fx=fx, levK=3)
{
  # input data arguments
  xmat <- model.matrix(sformula, data = data)[,-1]
  m <- dim(as.matrix(xmat))[2]
  th0 <- rep(0,m-1)
  ## get initial theta, using no penalization...
  f0 <- optim(th0,fn,gr=NULL, xmat, fformula, tdata=data, fx=fx,
levK=levK, k=k)
  ## now get theta/alpha with smoothing parameter selection...
  fx=FALSE
  f1 <- optim(f0$par,fn,gr=NULL, xmat, fformula, tdata=data, hessian=TRUE,fx=
levK=levK, k=k)
  theta.est <-f1$par
  ## extract and examine fitted model...
  b <- si(theta.est,xmat,fformula, tdata = data,opt=FALSE,fx=fx,
levK=levK, k=k) ## extract best fit model
  ##
  Vt <- b$J%*%solve(f1$hessian,t(b$J))
  se <- diag(Vt)^.5
  # return
  return(list(fit=b,pcoef=theta.est, se=se))
}

MSE <- function(n_samples) {
    result <- data.frame()
    for (size in n_samples) {  # Iterate over different sample sizes
        mse_per_size <- data.frame()
        for (i in 1:20) {
            set.seed(i)  # Set the seed for each iteration
            true <- data.frame(sim.data.si(size))
            sformula <- as.formula(~ x1 + x2 + x3)
            fformula <- list(y ~ s(z), ~ s(z))

            d <- si.fit(true, sformula = sformula, fformula = fformula, levK
            pred <- predict.gam(d$fit, type = 'terms')

            MSE.1 <- mse(pred[,'s(a)'], true$sf1)
            MSE.2 <- mse(pred[,'s.1(a)'], true$sf2)
            MSE.3 <- mse(pred[,'s.1(z)'], true$f3)
            MSE.4 <- mse(pred[,'s(z)'], true$f2)

            mse_per_size <- rbind(mse_per_size, data.frame(MSE.1, MSE.2, MSE
        }
        # Calculate the mean of MSE for each sample size and store in the res
        mse_mean <- colMeans(mse_per_size)
        result <- rbind(result, mse_mean)
    }
    return(result)
}

# Run the function for sample sizes 200, 500, and 1000
sample_sizes <- c(200, 500, 1000)
mean_square_errors <- MSE(sample_sizes)
rownames(mean_square_errors)=c("N=200","N=500","N=1000")
colnames(mean_square_errors)=c("MSE.1","MSE.2","MSE.3","MSE.4")
t(mean_square_errors)

# Display the mean square errors for each sample size
```

```
#print(mean_square_errors)
```

R CODE FOR SINGLE-INDEX AND GAM ACCURACY

```
library(mgcv)
library(mltools)

si <- function(theta, xmat,fix.formula, tdata, opt=TRUE, k=k,fx=fx, levK=levK
## Fit single index model using gam call, given theta (defines alpha).
## Return ML if opt==TRUE and fitted gam with theta added otherwise.
## Suitable for calling from 'optim' to find optimal theta/alpha.
  alpha <- c(1,theta) ## constrained alpha defined using free theta
  k=5
  kk <- sqrt(sum(alpha^2))
  alpha <- alpha/kk  ## so now ||alpha||=1
  tdata$a <- xmat%*%alpha      ## argument of smooth
  formula.updated <- eval(lapply(fix.formula, function(x) update(x, ~.+ s(a,
  e1 <- environment(alpha)
  b <- gam(formula.updated,data=tdata, family=multinom(K=levK)) ## fit model
  if (opt) return(b$gcv.ubre) else {
    b$alpha <- alpha  ## add alpha
    J <- outer(alpha,-theta/kk^2) ## compute Jacobian
    for (j in 1:length(theta)) J[j+1,j] <- J[j+1,j] + 1/kk
    b$J <- J ## dalpha_i/dtheta_j
    return(b)
  }
} ## si
```

```r
sim.data.si <- function(n, seed=123,plot=FALSE)
{
  # setting random seed
  set.seed(seed)
  f1 <- function(x) 0.2 * x^11 * (10 * (1 - x))^6 + 10 *
               (10 * x)^3 * (1 - x)^10
  f2 <- function(x) sin(3*pi*x)*exp(-x)
  f3 <- function(x) x^3
  f4 <- function(x) sin(2*pi*x)

  m <- 3
  x <- matrix(runif(n*m),n,m) ## the covariates for the single index part
  z <- rnorm(n) ## another covariate
  alpha1 <- c(0.1, -0.2, 0.1); alpha1 <- alpha1/sqrt(sum(alpha1^2))
  #alpha1 <- c(0.1, -0.4, 0.1)
  # c(0.1, -0.2, 0.1)
   # 0.08, -0.2, 0.1
    #c(0.05, -0.2, 0.08)
    #c(0.05,-0.2,0.1)
  sf1 <- scale(as.numeric(f1((x%*%alpha1+.41)/4)),scale=FALSE)
  sf2 <- scale(as.numeric(f4(x%*%alpha1)), scale=FALSE)
  if(plot==TRUE)
  {
    par(mfrow=c(1,2))
    p1=plot(x%*%alpha1, sf1, ylab = "Function1", xlab = "Single Index1")
    p2=plot(x%*%alpha1, sf2, ylab = "Function2", xlab = "Single Index2")
  }
```

```
    f2f <- scale(f2(z), scale=FALSE)
    f3f <- scale(f3(z), scale=FALSE)
  eta1 <- sf1 + f2f
  eta2 <- sf2+ f3f

  p <- exp(cbind(0,eta1,eta2))
  p <- p/rowSums(p) ## prob. of each category
  cp <- t(apply(p,1,cumsum)) ## cumulative prob.
  ## simulate multinomial response with these probabilities
  ## see also ?rmultinom
  y <- apply(cp,1,function(x) min(which(x>runif(1))))-1

  #
  tdata <- data.frame(y=y,x1=x[,1], x2=x[,2], x3=x[,3],z=z)
  #
  return(tdata)
}

si.fit <- function(data, fn=si, sformula , fformula, k=5,fx=fx, levK=3)
{
  # input data arguments




  xmat <- model.matrix(sformula, data = data)[,-1]
  m <- dim(as.matrix(xmat))[2]
  th0 <- rep(0,m-1)
  ## get initial theta, using no penalization...
  f0 <- optim(th0,fn,gr=NULL, xmat, fformula, tdata=data, fx=fx,
levK=levK, k=k)
 ## now get theta/alpha with smoothing parameter selection...
  fx=FALSE
  f1 <- optim(f0$par,fn,gr=NULL, xmat, fformula, tdata=data, hessian=TRUE,fx=
levK=levK, k=k)
  theta.est <-f1$par
 ## extract and examine fitted model...
  b <- si(theta.est,xmat,fformula, tdata = data,opt=FALSE,fx=fx,
levK=levK, k=k) ## extract best fit model
 ##
  Vt <- b$J%*%solve(f1$hessian,t(b$J))
  se <- diag(Vt)^.5
 # return
  return(list(fit=b,pcoef=theta.est, se=se))
}

sample=c(200,500,1000)
for (i in sample){
     actual1=sim.data.si(i)

     sformula <- as.formula(~ x1 + x2 + x3)
     fformula <- list(y ~ s(z), ~ s(z))

     d <- si.fit(actual1, sformula = sformula, fformula = fformula, levK = 2
     pred <- predict.gam(d$fit, type = 'response')

    colnames(pred)=c('0','1','2')
    max_columns1 <- data.frame(colnames(pred)[max.col(pred, ties.method = "fi
    colnames(max_columns1) = 'yhat'

    Accuracy1=sum(actual1$y==max_columns1)/sample # accuracy for single inde
 }




for (i in sample){
      actual=sim.data.si(i)
      gam.model=gam(list(y~ x1 +x2 +x3 + s(z),
                         ~ x1 +x2 +x3 + s(z)),family=multinom(K=2),data=actu

        gam.pred= predict.gam(gam.model,type='response')

      colnames(gam.pred)=c('0','1','2')
      max_columns <- data.frame(colnames(gam.pred)[max.col(gam.pred, ties.me
      colnames(max_columns) = 'yhat'

      Accuracy=sum(actual$y==max_columns)/sample # accuracy for gam
    }

Accuracy1
Accuracy



Accuracy1=t(Accuracy1)
colnames(Accuracy1)=c('N=200','N=500','N=1000');Accuracy1 #single index
```

```
Accuracy=t(Accuracy)
colnames(Accuracy)=c('N=200','N=500','N=1000');Accuracy  #gam
```

Curriculum Vitae

Kwame Duodu and Osei Boateng Margaret's second child, Kwabena Gyamfi Duodu, was born on December 14, 1993. He pursued higher study at the Kwame Nkrumah University of Science and Technology (KNUST) in Ghana after finishing his secondary education at Dwamena Akenten Senior High School in 2013. He registered there for a four-year bachelor's degree in statistics program. Kwabena actively participated in the Mathematical Science Student Association events while he was a student at KNUST and received commendable recognition for his efforts. Following graduating from college, he worked as a medical records clerk at Kumasi's Tafo Government Hospital. At The University of Texas at El Paso, Kwabena started his Master's program in Statistics and Data Science in the fall of 2021. He gained a solid foundation in statistics and machine learning throughout his studies, which piqued his interest in the subject of data science even more. Maxwell actively participated in the academic community as a teaching and research assistant and had the chance to work with Dr. Suneel Babu Chatla on a project. With a clear career goal of becoming a senior biostatistician and data scientist in the health sector, Kwabena plans to pursue a doctorate program in Data Science at the University of Texas at El Paso. His dedication to advancing his knowledge and skills in the field of Data science reflects his commitment to making a meaningful impact on healthcare research and data analysis.
E-mail address: *kgduodu@miners.utep.edu*