# Feature-Based Spatio-Temporal Modeling

A Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

in partial fulfillment

of the requirements for the degree

Doctor of Philosophy

by

Xiaofeng Wang

May

2012

APPROVAL SHEET

The dissertation

is submitted in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

AUTHOR

The dissertation has been read and approved by the examining committee:

Advisor

Stephanie Guerlain

Accepted for the School of Engineering and Applied Science:

James H. Aylor

Dean, School of Engineering and Applied Science

May

2012

# ABSTRACT

Dimensions of data are expanding. An increasing number of spatio-temporal data are available with numerous features, including ordinary numerical and categorical features as well as unstructured features like text. Although those high dimensional data can help improve predictions, efficient methods of processing spatio-temporal data with many different types of features are limited.

This dissertation formalized an important class of problems related to spatio-temporal data. In the dissertation, an effective mathematical model, the local spatio-temporal generalized additive model(LSTGAM), was developed to predict and classify spatio-temporal data. This model can fully utilize many different types of data, such as spatial and temporal data, geographic data, demographic data, textual data, etc. The model can be easily estimated by available algorithms and has good interpretability. To assist the building of LSTGAM, a randomized least angle regression (RLAR) method was used to select features for non-linear regression models. Tests with simulated data and real data showed RLAR performed well. In addition, a new method, the semantic role labeling-based latent Dirichlet allocation (SRL-LDA) model, was developed to extract key information from text. This method is based on the automatic semantic analysis and understanding of natural language, combined with dimensionality reduction via latent Dirichlet allocation. The above two models, LSTGAM and SRL-LDA, can be applied together to applications where unstructured textual data contains indicators relevant to the spatio-temporal properties of events.

The newly developed models have been applied to four real problems, including predictions of criminal incidents and analysis of train accidents. Results showed the

LSTGAM outperformed several previous models, such as spatial generalized linear models and hot spot models, in evaluations with the spatio-temporal classification problem. It also showed that SRL-LDA can effectively extract useful information from unstructured textual data like Twitter posts. Information extracted by SRL-LDA showed the ability to improve the prediction performance in different cases. Those applications also revealed interesting sources of data for criminal prediction: social media services like Twitter. As discussed at the end of the dissertation, a large scale text analysis system with modeling techniques developed in this dissertation can provide solutions for many areas where predictions are important.

## ACKNOWLEDGMENTS

I feel extremely lucky to have my PhD journey at University of Virginia with so many talented and friendly faculty, colleagues and friends. First, I would like to thank my advisor Prof. Donald Brown. Thank you for introducing me to such interesting research on knowledge discovery and data mining. Thank you for providing me with many great opportunities. Thank you for your guidance on my study. Thank you for your understanding and support during my difficult time. I have learned so much from you. You are a wonderful advisor. Second, I would like to thank my PhD advisory committee members Prof. Stephanie Guerlain, Prof. Stephen Patek and Prof. Theodore Chang. Thank you for your suggestion to help me improving my dissertation work. Especially, thank you Prof. Stephanie Guerlain for editing my final dissertation. Third, I would like to thank Jamie Conklin and Dr. Matthew Gerber, who worked closely with me during my PhD study. Without your help, the completion of this dissertation would be impossible. Thank you Jamie for helping me with programming and getting data. Thank you Matt for the insightful discussion on text modeling, especially on the semantic role labeling technique, which is the foundation for the semantic role labeling-based latent Dirichlet allocation model presented in this dissertation. Forth, I would like to thank my friends at UVa who have made my PhD life enjoyable: Xiaohuan, Tiantian, Yonghang, Ruwei, Zhenyu, Hui Hua, Haiyan, Zhang Nan, Yiyi, Jian Kang, Dandan, Mingyi, among others.

Last but the most importantly, I would like to thank my family. I thank my parents Jianhua Xu and Zhongwen Wang, for the lifelong love and support. I thank my parents-in-law Yuhuan Duan and Fuxue Shen, for your help during the first few

months of our first baby. And finally, to my wife Lei and our son Aaron: thank you Lei for everything you have done for our family. Thank you Aaron for your arrival. Thank you for being the most important part of my life!

# CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

## LIST OF SYMBOLS

| | |
|---|---|
| $\|\cdot\|$ | $L_2$ norm. |
| $\alpha$,$\beta$, $\theta$, $\eta$ | Parameters in mathematical models. |
| $\hat{\beta}$ | The estimation of $\beta$. |
| $\delta(\cdot)$ | A decision function mapping probabilities to discrete values. |
| $\delta$ | A threshold for decision functions. |
| $\varepsilon$ | Random noise. |
| $\epsilon$ | A tolerance threshold for loss functions. |
| $\kappa_t$ | A temporal dummy variable indicating the continuos zeros before time $t$ for a binary variable with the values of zero or one. |
| $\Omega(\cdot)$ | A complex penalty function. |
| $b(\cdot)$ | A basis function for generalized additive models. |
| $E(\cdot)$, $\mu$ | Expectation of random variables. |
| $f(\cdot)$ | A mathematical function. |
| $\mathbb{G}$ | A spatial grid index set. |
| $\mathcal{G}(\cdot)$ | A link function. |
| $I(C)$ | An indicator function. If condition $C$ is true, $I(C) = 1$; otherwise, $I(C) = 0$. |
| $L(\cdot)$ | A loss function. |
| $NN(C)$ | The nearest neighbor of $C$. |
| $Pr(\cdot)$, $\Pi(\cdot)$ | A probability function. |
| $s_g$ | A spatial grid. |

| | |
|---|---|
| $S$ | A space of interest. |
| $\mathbb{S}$ | A spatial data set defined by grids. |
| $t$ | Time. |
| $\mathbb{T}$ | A temporal data set. |
| $Var(\cdot)$ | Variance of random variables. |
| $w.$ | A word, token, or term. |
| $W$, $\mathbf{w}$ | A sequence of words, tokens and terms. |
| $x_{[g,t]}$ | A vector of features at grid $g$ and time $t$. |
| $x_{[g,t,p]}$ | The $p^{\text{th}}$ feature at grid $g$ and time $t$. |
| $X_p$ | The $p^{\text{th}}$ feature at all the locations and time. |
| $\mathbb{X}$ | A feature set. |
| $\mathbb{X}_{cat}$ | A categorical feature set. |
| $\mathbb{X}_{num}$ | A numerical feature set. |
| $\mathbb{X}_{txt}$ | A textual feature set. Each element is a document. |
| $\mathbb{X}_{t' \leqslant t^c}$ | The set of observed features at time $t^c$. |
| $Y$,$y$ | Response variables. |

# CHAPTER 1

# INTRODUCTION

This chapter first introduces the background of the dissertation research. Then the problem is defined and formulated mathematically. At last, the chapter overviews each chapter in the dissertation.

## 1.1 Background

In many important problems data are associated with spatial and temporal information. For example, criminal rates are linked with the specific cities and time. Textual data like news describe events which happened in the certain locations and time. Every photo is taken at the exact geographic location and time. However, those spatial and temporal information is ignored in many data analysis methods and applications, because it is relatively not important or there lacks a good model. For example, cross-sectional data analysis methods, like regression models, assume that observations are collected at the same time without regard to differences in space.

There is an increasing need for models which take spatial and temporal information into account, especially the models which can make accurate spatio-temporal predictions. Law enforcement is such an example. As reported by Bureau of Justice Statistics [67], there were 22,879,720 personal and property crimes in 2008. The total economic loss to victims of all crimes was 18,075 million dollars. To prevent crimes, law enforcement agencies need to study the spatio-temporal patterns of crimes. With a spatio-temporal model of criminal incidents, they can discover the relationships between demographic, economic, and geographic factors and crimes as well as possible causality of crimes. In addition, they can predict the locations and time of future criminal activities. If the model can predict future crimes accurately, law enforcement agencies can deploy limited resources, such as walking and driving patrols, surveillance systems, and neighborhood watch programs, to improve security and reduce threats.

To meet the above needs, spatio-temporal models have been studied. In classic spatio-temporal models, the variable of interest, such as criminal incidents, pollution levels and epidemic diseases, is modeled as a random process defined on the space and time. Other features[1] associated with the variable are not well considered. For example, spatial hot spot models [2,11,29] are widely applied by law enforcement. In this type of models, current crime clusters are so called hot spots. Methods to generate hot spots include spatial histograms, clustering, mixture models, scan statistics, and density estimation. Future criminal incidents are predicted to occur in these high risk areas. These models do not show the insight into behaviors of actual criminals and cannot indicate the changes of crime patterns. More sophisticated statistical models [60,84,115] have been developed by researchers to account for those problems. The major limitations with these models include that they considered only a few number of numerical features and they did not incorporate the temporal information very well.

The simple spatio-temporal models mentioned above are not enough. Nowadays dimensions of data are expanding. There are more and more spatio-temporal data with numerous features, especially high dimensional data like text. For example, law enforcement has incident data including locations (e.g. longitudes and latitudes),

[1] In this dissertation, a "feature" means a variable which describes a certain attribute related to the variable of interest. Sometimes, it is called predictor or explanatory variable.

time (e.g. years, months and time), location features (e.g. distances to the nearest highways), criminal features (e.g. types of crimes, numbers of victims, and types of weapons), and other multimodal data (e.g. narratives). Another example of such data is the train accident data from Federal Railroad Administration [34], which record locations and times of train collisions, attributes of trains, weather conditions, damages, and narrative descriptions. In addition, social media such as Facebook[2] and Twitter[3] allow users to instantly create, disseminate, and consume information about the events which happened in any location and time.

These spatio-temporal data and associated features provide more information than traditional numerical and categorical data. Especially the textual data from social media account for the rich and rapidly context that surrounds incidents of interest. Utilizing these data can help improve spatio-temporal predictions. For example, Google used search terms with locations to predict H1N1 outbreaks [42]. Scholars used news articles to predict stock market behaviors [70, 83, 87] and used Twitter posts to predict weekend box office results [3], election results [5], and stock market trends [12].

However, efficient methods of modeling spatio-temporal data with many features, especially textual features, are limited. The objective of this dissertation is to develop an efficient method to model such spatio-temporal data with different types of features, including numerical features, categorical features and textual features.

## 1.2 Problem Definition

This dissertation develops an efficient method to perform predictions and classifications over spatio-temporal data with the consideration of ordinary features as well as textual features. As discussed in Section 1.1, there are various features available

[2] http://www.facebook.com

[3] http://www.twitter.com

about an area of interest. Generally, the objective is to model the patterns of a certain variable (e.g. the probability of criminal incidents) of this area with these features and apply the model to predict.

Mathematically, the problem can be defined as follows[4]:

Suppose we are interested in a space $S \subset \mathbb{R}^d$, where $d$ is the dimension of the space. Usually, $d = 2$ or 3. We can represent $S$ by discrete grids: $S = \bigcup_{s_g \in \mathbb{S}} s_g$, where $\mathbb{S} = \{s_g | s_g \subset \mathbb{R}^d, g \in \mathbb{G}\}$ is a spatial data set, and $\mathbb{G} = \{g | g \in \mathbb{N}^+\}$ is a spatial grid index set. We have a temporal data set $\mathbb{T} = \{t | t \in \mathbb{R}^+\}$ and a feature set $\mathbb{X} = \{x_{[g,t]} | x_{[g,t]} \in X_1 \times X_2 \times \cdots X_P, g \in \mathbb{G}, t \in \mathbb{T}\}$ associated with $\mathbb{S}$. For feature data, $X_p (p \in N^+, p \leq P)$ can be numerical ($X_p \subset \mathbb{X}_{num}$), categorical ($X_p \subset \mathbb{X}_{cat}$), and textual ($X_p \subset \mathbb{X}_{txt}$), where $\mathbb{X}_{num}, \mathbb{X}_{cat}, \mathbb{X}_{txt}$ are defined in Table 1.1.

Table 1.1: Mathematical Notation

| Notation | Definition | Explanation |
|---|---|---|
| $\mathbb{G}$ | $\mathbb{G} = \{g \| g \in \mathbb{N}^+\}$ | A spatial grid index set. |
| $s_g$ | $s_g \subset \mathbb{R}^d$ | A spatial grid. |
| $S$ | $S \subset \mathbb{R}^d$ | A space of interest with dimension $d$. |
| $\mathbb{S}$ | $\mathbb{S} = \{s_g \| g \in \mathbb{G}\}$ | The spatial data set such that $S = \bigcup_{s_g \in \mathbb{S}} s_g$. |
| $\mathbb{T}$ | $\mathbb{T} = \{t \| t \in \mathbb{R}^+\}$ | A temporal data set. |
| $x_{[g,t]}$ | $x_{[g,t]} \in X_1 \times X_2 \times \cdots \times X_P$ | A vector of features at grid $g$ and time $t$. A feature $X_p$ can be numerical ($X_p \subset \mathbb{X}_{num}$), categorical ($X_p \subset \mathbb{X}_{cat}$), and textual set ($X_p \subset \mathbb{X}_{txt}$). |
| $x_{[g,t,p]}$ | $x_{[g,t,p]} \in X_p$ | The $p^{\text{th}}$ feature at grid $g$ and time $t$. |

Continued on the next page

[4] Table 1.1 defines all notations used in the following problem definition.

Table 1.1: Mathematical Notations (*continued*)

| Notation | Definition | Explanation |
|---|---|---|
| $\mathbb{X}$ | $\mathbb{X} = \{x_{[g,t]} \mid x_{[g,t]} \in X_1 \times X_2 \times \cdots \times X_P, g \in \mathbb{G}, t \in \mathbb{T}\}$ | A feature set. |
| $\mathbb{X}_{cat}$ | $\mathbb{X}_{cat} = \{x_j \mid x_j \in \{C_1, C_2, \cdots\}, j \in \mathbb{N}^+\}$ | A categorical feature set, where $C.$ is any categorical value. |
| $\mathbb{X}_{num}$ | $\mathbb{X}_{num} = \{x_j \mid x_j \in \mathbb{R}, j \in \mathbb{N}^+\}$ | A numerical feature set. |
| $\mathbb{X}_{txt}$ | $\mathbb{X}_{txt} = \{x_j \mid x_j =< word_{j1}, word_{j2}, \cdots, word_{jn_j} >, word. \in Dictionary, n_j \in \mathbb{N}^+, j \in \mathbb{N}^+\}$ | A textual feature set. Each element is a document. *word.* has the value of a word or phrase and *Dictionary* is the collection of all words and phrases. |
| $\mathbb{X}_{t' \leqslant t^c}$ | $\mathbb{X}_{t' \leqslant t^c} = \{x_{[g',t']} \mid x_{[g',t']} \in \mathbb{X}, g' \in \mathbb{G}, t' \in \{t \mid t \leqslant t^c, t \in \mathbb{T}\}\}$ | The set of observed features at time $t^c$. |

Two types of problems can be defined based on the type of the variable of interest: the classification problem and the regression problem.

- **The classification problem**:

  At time $t^c$, we are interested in the values of the $P^{th}$ feature $X_P = \{0, 1\}$ at locations $\mathbb{S}^* = \{s_g \mid s_g \in \mathbb{S}, g \in \mathbb{G}^* \subset \mathbb{G}\}$ and time $\mathbb{T}^* = \{t \mid t > t^c\}$ in the future. The objective is to find a probability function:

$$\Pi_{[g,t]} = Pr[x_{[g,t,P]} = 1 \mid \mathbb{X}_{t' \leqslant t^c}, \mathbb{S}, \mathbb{T}] \tag{1.1}$$

and a decision function:

$$\delta_{[g,t]} = \delta(\Pi_{[g,t]}) : [0,1] \rightarrow \{0,1\} \tag{1.2}$$

such that:

$$\begin{aligned} L(\Pi_{[g,t]}, \delta_{[g,t]}) = & \sum_{g \in \mathbb{G}^*, t \in \mathbb{T}^*} weight_0 \cdot I(\delta_{[g,t]} = 0 | x_{[g,t,P]} = 1) \\ & + \sum_{g \in \mathbb{G}^*, t \in \mathbb{T}^*} weight_1 \cdot I(\delta_{[g,t]} = 1 | x_{[g,t,P]} = 0) < \epsilon \end{aligned} \tag{1.3}$$

where $\mathbb{X}_{t' \leqslant t^c} = \{x_{[g',t']} | x_{[g',t']} \in \mathbb{X}, g' \in \mathbb{G}, t' \in \{t | t \leqslant t^c, t \in \mathbb{T}\}\}$, $g \in \mathbb{G}^*$, $t \in \mathbb{T}^*$, $x_{[g,t,P]}$ is the value of the $P^{th}$ feature at grid $g$ and time $t$, $L(\cdot)$ is a loss function, $I(\cdot)$ is an indicator function, $weight_0, weight_1$ are weights of different types of errors, and $\epsilon$ is a tolerance threshold.

The major difficulty of this problem is to find an accurate probability function $\Pi_{[g,t]} = Pr[x_{[g,t,P]} = 1 | \mathbb{X}_{t' \leqslant t^c}, \mathbb{S}, \mathbb{T}]$, such that $\Pi_{[g,t]}$ has high values for the locations and time when $x_{[g,t,P]} = 1$ (e.g. criminal incidents will happen) and low values for the locations and time when $x_{[g,t,P]} = 0$ (e.g. criminal incidents will not happen). Given a good probability function $\Pi_{[g,t]}$, different thresholds can be run to choose the best decision function which minimizes different types of errors. Alternatively, users can choose their own decision functions based on their resources and risk preferences. For example, if modeling the probability of criminal incidents, law enforcement agencies can choose a cutoff value $\delta^* = 0.8$ to classify the locations and time with predicted probabilities higher than $\delta^*$ as the high risk area.

This dissertation focuses on the development of $\Pi_{[g,t]}$.

- **The regression problem**:

At time $t^c$, we want to predict the values of the $P^{th}$ feature $X_P \subset \mathbb{X}_{num}$ at locations $\mathbb{S}^*$ and time $\mathbb{T}^*$. ($\mathbb{S}^*$ and $\mathbb{T}^*$ are the same as defined in the classification problem). The objective is to find a function:

$$\hat{x}_{[g,t,P]} = f(\mathbb{X}_{t' \leqslant t^*}, \mathbb{S}, \mathbb{T}) + \varepsilon \tag{1.4}$$

such that:

$$L(\hat{x}_{[g,t,P]}) = \sum_{g \in \mathbb{G}^*, t \in \mathbb{T}^*} \|\hat{x}_{[g,t,P]} - x_{[g,t,P]}\| < \epsilon \tag{1.5}$$

where $g \in \mathbb{G}^*$, $t \in \mathbb{T}^*$, $\mathbb{X}_{t' \leqslant t^c} = \{x_{[g',t']} | x_{[g',t']} \in \mathbb{X}, g' \in \mathbb{G}, t' \in \{t | t \leqslant t^c, t \in \mathbb{T}\}\}$, $L(\cdot)$ is a loss function, $\hat{x}_{[g,t,P]}$ is the predicted value, $x_{[g,t,P]}$ is the true value, $\varepsilon$ is random noise, and $\epsilon$ is a tolerance threshold.

## 1.3 Overview of the Dissertation

This dissertation is organized as follows.

Chapter 2 reviews methods related to the dissertation research in four research fields, data mining, spatio-temporal modeling, text mining and natural language processing. Chapter 3 defines the overall local spatio-temporal generalized additive models (LSTGAM) to model the spatio-temporal data with features. A feature selection algorithm, model estimation methods and a model evaluation method are also discussed in this chapter. Chapter 4 describes the semantic role labeling-based latent Dirichlet allocation model (SRL-LDA) to extract textual information and how to incorporate these extracted information into LSTGAM. Chapter 5 applies the models developed in this dissertation to four real problems and evaluates the performance of the models based on real data. Chapter 6 concludes the dissertation and suggests future work.

# CHAPTER 2

# LITERATURE REVIEW

As discussed in Chapter 1, the objective of this dissertation is to develop models of spatio-temporal data with the consideration of different types of features, especially textual features. There are four research fields related to this problem: data mining, spatio-temporal modeling, text mining and natural language processing. This chapter reviews the most related models and techniques in these fields.

## 2.1 Data Mining

Data mining or knowledge discovery is a process to extract useful information from data. Most data mining techniques apply statistical models to learn patterns from data with the assistance of computers. Therefore, data mining is closely related to statistical learning, machine learning, and pattern recognition. Data mining has been broadly applied to different areas, including business, economics, medicine, and engineering.

Based on the types of problems to solve, data mining can be classified into supervised learning and unsupervised learning [36]. The problem of this dissertation is a supervised learning problem. Therefore, this section reviews supervised learning methods in details and unsupervised learning methods briefly.

### 2.1.1 Supervised Learning Methods

Supervised learning attempts to learn patterns from data by training sets. Each observation in a training set consists of both inputs (or predictors, explanatory variables, features) and an output (or response variable). Supervised learning models relationships between inputs and outputs, and predicts outputs when inputs are given.

*Linear Regression*

The most fundamental supervised learning method is multiple linear regression [36,89]. Linear regression models are simple and efficient solutions for many problems. A linear regression model assumes additive linear correlations between inputs and outputs. The model has the following form:

$$Y = \beta_0 + \sum_{i=1}^{p} X_i \cdot \beta_i + \varepsilon \tag{2.1}$$

where $Y$ is the output variable, $X_i$ are input vectors, $\beta_i$ are parameters to be estimated and $\varepsilon$ satisfies $E(\varepsilon) = 0$ and $Var(\varepsilon) = \sigma^2$.

A linear regression model is usually estimated by the least squares estimation method, which finds a linear fit to the data that minimizes the sum of square errors. The estimates of $\beta_i$ can be efficiently computed by $\hat{\beta} = (X^T X)^{-1} X^T Y$. The common assumptions of linear regression include that the data for the input variables are known; the input variables are linearly independent; the output variable is quantitative and $\varepsilon$ are independent, identical distributed with zero mean and constant variance.

To better discover patterns in data and relax the assumptions of linear regression, many non-linear supervised learning models have been developed.

*Generalized Linear Models (GLM)*

The generalized linear model (GLM) relates response variables to linear models of inputs via link functions. The distribution of the response variable $y$ in a GLM is assumed to be from the exponential family, where the probability density function can be expressed as [109]:

$$f_\theta(y) = exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\} \tag{2.2}$$

where $b, a$ and $c$ are arbitrary functions, $\phi$ is the dispersion parameter and $\theta$ is the location parameter. It can be shown that $\mu = E[Y] = b'(\theta)$.

A typical GLM assumes the response mean $\mu$ can be modeled as follows:

$$\mathcal{G}(\mu) = \beta_0 + \sum_{i=1}^{p} X_i\beta_i \tag{2.3}$$

where $\mathcal{G}(\cdot)$ is a link function (usually non-linear), $\beta_i$ are parameters and $X_i$ are inputs. Table 2.1 shows the link functions for the common distributions of response variables.

Table 2.1: Link Functions for Different Distributions

| Distribution of $y$ | Link Function $\mathcal{G}(\mu)$ |
|---|---|
| Gaussian | $\mu$ |
| Binomial | $\ln\left(\frac{\mu}{1-\mu}\right)$ |
| Poisson | $\ln(\mu)$ |
| Exponential | $\mu^{-1}$ |

Different from linear regression, GLM cannot be estimated by the least squares estimates. Instead, GLM is usually estimated by the maximum likelihood method, which finds the parameters $\beta$ to maximize the probability of $y$ being observed. This optimization problem can be solved by applying the Newton-Raphson algorithm to log-likelihood functions.

*Nearest Neighbor Methods*

The nearest neighbor method makes predictions by using outputs with similar inputs. For example, the k-nearest-neighbor method fits $\mu(\mathbf{X})$ as follows:

$$\mu(\mathbf{X}) = \frac{1}{k} \sum_{\mathbf{X}' \in NN(\mathbf{X})} \mu(\mathbf{X}') \tag{2.4}$$

where $NN(\mathbf{X})$ is the neighborhood of $\mathbf{X}$ including the $k$ closest points to $\mathbf{X}'$. As we can see, this method gives weight 1 to the observations in the neighborhood and weight 0 to the other observations in the training set. Then it averages the outputs with weight 1.

The k-nearest-neighbor method has low bias but high variance. To enhance the performance, several methods have been developed [36]. Instead of giving weights 1 and 0 to observations, kernel methods give weights to observations based on their distances to the point of interest smoothly. Rather than averaging the outputs, local regression methods fit linear models by locally weighted least squares.

*Generalized Additive Models (GAM)*

The generalized additive model (GAM) [49, 50] is a generalization of GLM. GAM is more flexible in the treatment of nonlinearity than GLM. GAM assumes additivity between predictors, but allows for local nonlinearity in each predictor. The model can be formed as

$$\mathcal{G}(\mu) = \alpha + \sum_{i=1}^{p} f_i(X_i) \tag{2.5}$$

where $\mathcal{G}$ is a link function, $\mu$ is the mean of the response variable, $\alpha$ is the intercept, and $f_i(X_i)$ are nonlinear smooth functions.

To estimate GAM, $f_i(X_i)$ are first represented by scatterplot smoothers. Then the smoothers are updated by the backfitting algorithm [50] until the changes of functions $\hat{f}_i$ are less than a certain threshold. Alternatively, $f_i(X_i)$ are first expressed as sums of spline basis functions. Then GAM can be reviewed as GLM and estimated by the penalized iteratively reweighted least squares algorithm [109].

Similar to linear regression, GAM can also be fitted locally [75].

### *Tree-based Models*

A tree model partitions the data set using a sequential series of linear surfaces, and fits a simple model in each smallest rectangle space. It's among the most commonly used data mining techniques because of easy interpretability and easy implementability. However, in most tests, its accuracy is worse than other methods.

The tree size is an important tuning parameter controlling the model's complexity and performance. A small tree might not be able to capture the structure of data while a large tree might over fit data. The optimal tree size should be adaptively chosen from data. To build a tree model, a greedy algorithm is applied. A tree is first grown to a large size and then this large tree is pruned to a smaller size using cost-complexity pruning, such as the weakest link pruning [13].

Tree methods are unstable, which means a small change in the training data can result in a very different series of splits. The major reason for this problem is that the feature space is partitioned hierarchically in the tree building process. To improve the performance of trees, the random forests method [14] was developed by Breiman and Cutler. In the random forests model, multiple $N$ trees are built. Each tree uses a small subset of predictors and a bootstrapping sample of the training set to calculate the best split. Each tree is grown to the largest size without pruning. Then the prediction is made by voting from all trees. Tests show the random forests model has lower generalization error than tree models [14].

*Support Vector Machines (SVM)*

For the binary classification problem with training data $\{(X_i, y_i)|y_i \in \{-1, 1\}\}$, the support vector machine (SVM) [99] maps the input vectors $X$ into a high-dimensional feature space $Z = \{(h_1(X), h_2(X), \cdots, h_M(X))\}$, where $h_m$ are basis functions. Then SVM finds the optimal separating hyperplane in this new space $Z$. The optimal separating hyperplane is defined as the one which generates the largest margin between the observations from different classes 0 or 1.

Mathematically, a hyperplane can be defined by $\{x : f(x) = h(x)^T\beta + \beta_0 = 0\}$, where $\|\beta\| = 1$. The classification rule by $f(x)$ is then $G(x) = sign(f(x))$. SVM solves the following optimization problem [36]:

$$\max_{\beta,\beta_0,\|beta\|=1} C \tag{2.6}$$

$$s.t.\ y_i f(x_i) \geq C \text{ for all i} \tag{2.7}$$

The above optimization can be solved by quadratic programming using Lagrange multipliers. SVM can also be adapted for regression problems where the response variables are quantitative.

*Neural Networks*

The neural network model mimics biological neural networks by constructing hidden layers of nodes or neurons. As discussed in [36], a neural network model can be considered as a two-stage regression or classification model. Each node in the hidden layer can be expressed as:

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T X), m = 1, \cdots, M \tag{2.8}$$

where $\sigma(v)$ is the activation function, usually chosen to be the sigmoid function $\sigma(v) = \frac{1}{1+e^{-v}}$. The final output of the model is determined by:

$$T_k = \beta_{0k} + \beta_k^T Z, k = 1, \cdots, K \tag{2.9}$$

$$f_k(X) = g_k(T), k = 1, \cdots, K \tag{2.10}$$

where $Z$ and $T$ are vectors of $Z_m, m = 1, \cdots, M$ and $T_k, k = 1, \cdots, K$ respectively. Here $g_k$ are final transformation of the outputs $T$.

The back-propagation method was developed to estimate the parameters in neural network models [99].

*Boosting*

Boosting is a method to improve the prediction performance of various supervised learning models, especially tree models. The intuitive idea of boosting is to change the distribution over the feature space $X$ in a way that increases the probability of the harder parts of the space, thus forcing the model to learn more and make less mistakes on these parts.

The most popular boosting algorithm is called the AdaBoost algorithm [35]. In the $m^{th}$ iteration of the algorithm, a model $G_m$ is fitted to the training data with weights $\{w_i\}$. Then the weights $\{w_i\}$ are updated according to the performance of $G_m$ so that $w_i$ increase for poorly performed observations.

The problem with boosting is that it suffers poor performance when applied to noisy data [61].

*Comparison of Different Supervised Methods*

Among the above supervised learning models, GAM and SVM usually perform best in prediction accuracy. Nearest neighbor methods do not perform well on high dimensional data because of the curse of dimensionality. GLM are not as flexible as GAM. SVM and neural networks models are usually hard to interpret [36].

### 2.1.2 Unsupervised Learning Methods

Unsupervised learning attempts to learn properties of data without training sets. Each observation has features (like inputs in supervised learning), and unsupervised learning builds models to describe certain patterns of observations. For example, clustering methods measure similarities between observations and classify them into different groups such that observations from the same group have similar properties. One of the most popular clustering methods is K-means, which iteratively adds observations into the closest group [47]. A similar unsupervised learning problem is data association. Given an incident of interest, data association models find other incidents with similar properties [16]. Association rule analysis is another widely applied unsupervised learning method, which tries to discover sets of joint features appearing frequently [36]. These models play an important role in marketing data mining.

### 2.1.3 Feature Selection

Both supervised learning models and unsupervised learning models are built on features of observations. Therefore, the feature selection is critical in data mining, especially with a large number of features. Well-selected features not only simplify models, but also improve prediction accuracy.

If the total number of features $p$ is small, it is possible to get all the subsets of features and compare the model performance like mean squared errors with all possible combinations of features. For $p$ features, there are $2^p - 1$ possible subsets.

However, if the total number of features $p$ is large, this method is not feasible. For example, if we have a problem with 20 features, there will be more than 1 million possible subsets.

Generally, there is no feature selection method which can output the optimal subset of features when $p$ is arbitrarily large. In practice, the following methods are popular to choose good subsets of features.

### *Stepwise Selection*

A simple algorithm to select features is the forward selection. This greedy algorithm begins with no feature and adds one feature each time to the subset. In each step, the algorithm selects the feature which can most improve the model performance. The algorithm stops when no feature can be added or some preset criteria are satisfied. A closely related algorithm is backward selection, which begins with all features and drops one feature each time.

The problem with forward selection is that the features added in the early steps might be no longer important when other features are available. This problem also exists for backward selection. A simple solution to this problem is stepwise selection. In each step of the algorithm, a feature can be added or dropped based on changes of model performance. Stepwise selection performs better than forward selection and backward selection in practice. However, this algorithm is not optimal because features are added or dropped sequentially.

### *Feature Selection by Clustering*

When a large number of features are collected for a problem, some of them might describe similar attributes of the problem or even exactly the same attribute. For example, the communities and crime data set [82] includes 122 features related to violent crimes in communities. There are two features in the data set “agePct12t21” and “agePct12t29” describing the percentage of population in similar age groups. It is reasonable to use just one of them in modeling.

Feature selection by clustering is a sophisticated method to pick representative features for the above situation [15, 46, 60]. This method first applies unsupervised clustering algorithms such as K-means to all the features. Then it chooses a certain number of features in each cluster to form the final subset of features.

*Penalized Regression Models*

For regression models, one of the most popular feature selection methods is the penalized regression model. This method estimates parameters of regression models by solving the optimization problem:

$$\hat{\theta} = \arg\min \frac{1}{n}\sum_{i=1}^{n} L(f_\theta(\mathbf{x}_i), y_i) + \Omega(\theta) \tag{2.11}$$

where $\theta$ is the parameter in the regression model $f_\theta$, $L(\cdot)$ is a loss function, and $\Omega(\theta)$ is a penalized function for the complexity of $f_\theta$. The more complex $f_\theta$ is, the larger the value of $\Omega(\theta)$ is. If we define $\Omega(\theta)$ as a function of the number of features, the optimization will try to find the best subset of features with a given size which has the best performance measured by the loss function.

For linear regression models, the complexity of models is determined by the coefficients $\beta$. Therefore, penalized linear models can be expressed as follows:

$$\hat{\beta} = \arg\min \frac{1}{n}\sum_{i=1}^{n} \|y_i - \hat{y}_i\| + \lambda \sum_{j=1}^{p} \|\hat{\beta}_j\|^\gamma \tag{2.12}$$

where $\|y_i - \hat{y}_i\|$ measures the errors of the regression models, $\lambda$ adjusts how much the complexity should be penalized and $\gamma$ decides how to compute the complexity of $\beta$. When $\gamma = 1$, the above model 2.12 is the $L_1$ regularized regression. As shown in [97], the solution to the model 2.12 has some of the $\beta$'s equal to zero. Therefore, it can be used to select features.

Many methods have been developed to solve penalized linear models, such as Lasso [97], Bridge [38], LARS [30], gradient descent methods [77], interior-point methods [58], and cyclical coordinate descent methods [37]. Among these methods, Lasso is the most successful and widely applied method for feature selection. Least angle regression (LARS) is a computationally efficient method to get the solution equivalent to the Lasso. It only needs $p$ steps to compute the order of variables entering the regression model, where $p$ is the number of features.

There are also studies on penalized generalized additive models [55,66,110]. However, these models focus on choosing the smoothness of the function $f_\theta$ rather than selecting features.

## 2.2 Spatio-Temporal Modeling

Spatio-temporal modeling analyzes data with both spatial and temporal information. Spatio-temporal observations usually correlate with each other, which violates assumptions of many classic data mining models. Therefore, a class of models focusing on spatio-temporal data have been developed. Time series analysis models one dimensional temporal data. Spatial analysis models spatial data with two or three dimensions. Spatio-temporal analysis combines methods from time series analysis and spatial analysis to model spatio-temporal data.

### 2.2.1 Time Series Analysis

Time series analysis studies temporal data in the frequency domain and in the time domain [90].

### *Frequency Domain Methods*

The frequency domain methods or spectral analysis methods treat temporal data as continuous signals. The objective is to identify the dominant frequencies in a series. For example, periodogram [90] is developed to identify significant frequencies. After dominant frequencies are identified, a time series is modeled by periodic functions:

$$x_t = \sum_{k=1}^{q} [U_{k1} cos(2\pi\omega_k t) + U_{k2} sin(2\pi\omega_k t)] \tag{2.13}$$

where $x_t$ is the value of the time series at time $t$, $U_{k1}, U_{k2}$ are zero-mean random variables defining amplitudes and phases of triangular functions, and $\omega$ is a frequency index defining cycles per unit time.

### *Time Domain Methods*

The time domain methods model the time series $x_t$ by time $t$ and previous observed values. A time series $x_t$ is usually decomposed into four components: trend, seasonality, cycle and random fluctuations. The first three components can be modeled by traditional data models such as linear regression. The last component is autocorrelated and is the focus of time series analysis. A widely studied model in the time domain to model the autocorrelated weak stationary time series is the integrated autoregressive moving average (ARIMA) model. The ARIMA model with order $(p, d, q)$ can be formulated as

$$\phi(B)(1 - B)^d x_t = \theta(B) w_t \tag{2.14}$$

where $B$ is a backshift operator defined as $Bx_t = x_{t-1}$, $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$ is the autoregressive operator, $\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q$ is the moving average operator and $w_t$ is white noise. If $w_t$ is not white noise, advanced time series models like seasonal ARIMA and autoregressive conditional heteroskedasticity

(ARCH) models can be applied.

### 2.2.2 Spatial Modeling

Spatial models study the variables about spaces. Spatial modeling has three branches, geostatistics, lattice process, and spatial point process, identified by Cressie [24].

*Geostatistics*

Geostatistics studies a random process $\{Z(\mathbf{s})|\mathbf{s} \in D\}$ defined on a continuous space $D \subset \mathbb{R}^d$. A typical geostatistical model has the form of

$$Z(\mathbf{s}) = \mu(\mathbf{s}) + W(\mathbf{s}) + \eta(\mathbf{s}) + \varepsilon(\mathbf{s}) \tag{2.15}$$

where $Z(\mathbf{s})$ is the value at location $\mathbf{s}$, $\mu(\mathbf{s})$ is the large-scale trend, $W(\mathbf{s})$ is the smooth-scale variation, $\eta(\mathbf{s})$ is the micro-scale variation, and $\varepsilon(\mathbf{s})$ is the random error [86].

In the above model, $\mu(\mathbf{s})$ is related to the features at location $\mathbf{s}$. It can be modeled by linear and non-linear regression models. $W(\mathbf{s})$ is assumed to be a stationary process with properties

$$E[W(\mathbf{s})] \equiv c \tag{2.16}$$

$$var(W(\mathbf{s} + \mathbf{h}) - W(\mathbf{s})) = 2\gamma(\mathbf{h}) \tag{2.17}$$

where $c$ is a constant and $2\gamma(\mathbf{h})$ is the so called variogram. $\eta(\mathbf{s})$ is usually ignored because it cannot be estimated from data. $\varepsilon(\mathbf{s})$ is commonly assumed to be white noise. Given a number of observations $\{Z(\mathbf{s}_i)\}$, the large scale trend and variogram can be estimated. Then, the above function can predict values at any location $\mathbf{s}$. This process is called spatial prediction or kriging.

*Lattice Process*

Lattice data modeling investigates a random process $\{Z(\mathbf{s}_i)|\mathbf{s}_i \in D\}$ defined on a countable set of spatial grids $D = \{\mathbf{s}_i|\mathbf{s}_i \in \mathbb{R}^d, i \in N^+\}$, indexed by $i$. Grids can have regular shapes like squares or irregular shapes like county territories.

Lattice data analysis models $Z(\mathbf{s}_i)$ by neighborhood grids and features associated with grid $i$. For example, the Markov random field model assumes the conditional distribution of $Z(\mathbf{s}_i)$, given $\{Z(\mathbf{s}_i)|\mathbf{s}_i \in D\}$, depends only on $\{Z(\mathbf{s}_k)|k \in N_i\}$, where $N_i$ consists of neighbors of $i$. Brown and his colleagues [60,73,115] studied conditional density models on the lattice data with spatial features like distances to the nearest buildings or roads. Another popular method builds regression models on grids [15]. A recent study [118] also applied the penalized regression model LASSO to choose the neighborhood size.

*Spatial Point Process*

Spatial point process modeling studies patterns of spatial points, such as locations of diseases and crimes. It concerns how points are distributed over spaces: completed randomly, clustered, or regulated.

Spatial point process models usually assume points are from the Poisson processes. For example, for a finite region $A$, $N(A)$, the number of points on $A$ follows a Poisson distribution with mean $\lambda_A|A|$ $(\lambda_A > 0)$. $\lambda_A$ can be a constant, $\lambda_A \equiv c$, or a function of features associated with the region $A$, $\lambda_A = \lambda(x_A)$. Diggle [28] conducted a comprehensive study on spatial point process modeling. Recent research [84] applied the GAM to improve classic spatial point process models.

### 2.2.3 Spatio-Temporal Modeling

Three common approaches directly apply time series models and spatial data models to analyze spatio-temporal data [86]:

1. Given each time period, build separate spatial models.

2. Given each location, build separate temporal models.

3. Build models in $\mathbb{R}^{d+1}$, where $d$ is the dimension of space.

The first two methods model inadequately correlations between locations and time, while the last method treats locations and time the same [86].

Recent developments in spatial-temporal modeling use the GAM to combine spatial features, temporal features and other ordinary features. A general form of these models is:

$$\mathcal{G}(Z(\mathbf{s}_i, t)) = f_{1t}(\mathbf{s}_i) + \sum_{k=2} f_{kt}(X_{ikt}) + T_t(i) + \varepsilon \tag{2.18}$$

where $\mathcal{G}(\cdot)$ is a link function, $f_{1t}$ models spatial variance, $f_{kt}(k > 2)$ accounts for variance caused by features, and $T_t(\cdot)$ models time-varying residuals. All of $f_{1t}$,$f_{kt}(k > 2)$ and $T_t(\cdot)$ are non-linear smooth functions. The spatio-temporal GAM have been applied successfully to different fields, including epidemiology, ecology and security informatics [27, 74, 100].

## 2.3 Text Mining

Text, such as documents and speeches, is one of the most common unstructured data. Unlike structured data (including numerical and categorical data), text cannot be fully modeled by mathematical equations or algorithms, but can usually store more information than structured data. Text mining, or text data mining, is a process of deriving important and useful information for such unstructured text data. Because text is a part of human language, text mining is closely related to natural language processing, which aims to understand human natural language automatically. This section reviews methods to process text into structured data and how to use the derived structured data. Advanced methods to model and understand language will be discussed in the next section about natural language processing.

### 2.3.1 Overview of Text Mining Process

A traditional text mining method includes three steps: structuring input text, performing data mining on structured data, and evaluating and interpreting outputs.

The first step focuses on obtaining numerical representation of text. Research interests focus on how to represent text with only a small number of features and speeding up this process. The second step usually applies well-developed techniques from the data mining field. All methods reviewed in Section 2.1 can fit this need. The last step interprets the results from data mining models and links them to the text problems.

Among the three steps, the first step, how to structure and model text, is critical in text mining.

### 2.3.2 Structuring Text Data

The mostly widely applied method to structure text data as numerical features is the vector space model. Documents are represented by feature vectors. The dimension of each vector is the number of different words in the whole collection of documents, usually more than one thousand.

*Term Frequency-Inverse Document Frequency (TF-DF) Model*

The term frequency- inverse document frequency (TF-IDF) model is a vector space model developed by Buckley and Salton [85]. The TF-IDF method models a document by a feature vector, consisting of numerical weights of different words in the document. The weight $w_{ij}$ of word $i$ in the document $j$ is computed by a TF-IDF function:

$$w_{ij} = TF_{ij} \times IDF_i \tag{2.19}$$

$$TF_{ij} = \frac{tf_{ij}}{max_i tf_{ij}} \tag{2.20}$$

$$IDF_i = log_2(\frac{N}{n_i}) \tag{2.21}$$

where $tf_{ij}$ counts the frequency of the $i^{th}$ word in the $j^{th}$ document, $N$ is the total number of candidate documents available for retrieval, and $n_i$ is the number of documents that contain the $i^{th}$ word. $w_{ij}$ is a trade-off between high frequency terms within a document (represented by $TF$) and the high distinctiveness of term frequency within the whole collection of documents (represented by $IDF$). As discussed in [1], $IDF$ can also be interpreted as the change of the amount of information within a collection of documents after observing a specific word (or term). Therefore, the $IDF$ is a good measure of distinguishability of words.

TF-IDF model is popular in text mining research, especially for information retrieval applications [63]. For example, after representing a collection of documents by a TF-IDF matrix, the similarity between any two documents can be measured as the Euclidean distance between two feature vectors. Given a search term, the most similar documents can be easily found.

The major problem with the TF-IDF model is that it regards words as independent tokens without the consideration of semantics. As a result, the size of feature vectors is usually large and the feature vectors are too rigid to compare the meanings of documents.

To reduce the size of feature vectors, the latent semantic index method [25], principal component method [51], keyword method [117] and high information content words method [103] were developed. These methods can significantly reduce dimensions of feature vectors from thousands to less than one hundred without losing much information.

### *Latent Semantic Indexing*

Latent Semantic Indexing (LSI) is a simple but effective method to improve the TF-IDF model. Suppose we have a TF-IDF matrix $M_{t\times d}$ where $t$ is the number of total terms and $d$ is the total number of documents, LSI first applies the singular value decomposition method to $M_{t\times d}$:

$$M_{t\times d} = T_{t\times r} S_{r\times r} D_{r\times d} \tag{2.22}$$

where $T$ and $D$ are two matrices with orthogonal and unit-length columns, $S$ is the diagonal matrix with $S_{1,1} \geq S_{2,2} \geq \cdots \geq Sr, r \geq 0$, and $r$ is the rank of $M$.

Then, the collection of documents is approximated by $\hat{M}_{t\times d}$ defined as:

$$\hat{M}_{t\times d} = T_{t\times k} S_{k\times k} D_{k\times d} \tag{2.23}$$

where $k \leq r$ is the selected number of dimensions, and $T$ and $D$ are defined as before but with smaller dimensions determined by $k$.

As we can see, this approximation by LSI removes some information from the original representation of the documents. In this way, common meaning components of the documents are kept while random effects of individual documents are removed. Especially, each feature vector no longer describes a specific document as a collection of independent words but indicates its strength with each underlying concept or latent variable. Each latent variable can be regarded as a collection of words commonly appearing in the same context. As discussed in [25], LSI can partly deal with the synonymy and polysemy problems and thus can perform better than the simple TF-IDF method.

*Probabilistic Latent Semantic Indexing*

In the above LSI model 2.23, if $S_{k\times k}$ is absorbed into $T_{t\times k}$ or $D_{k\times d}$, the term-document matrix $\hat{M}_{t\times d}$ can be reviewed as a product of two matrices $T^*_{t\times k}$ and $D^*_{k\times d}$. The first matrix $T^*_{t\times k}$ might be thought as a projection of the original high dimensional term space (with the dimension of $t$) into the latent space (with the dimension of $k$). In this case, rows of $T^*_{t\times k}$ define the coordinates of documents in the latent space and columns of $D^*_{k\times d}$ define relationships between the documents and each individual latent variable [52].

The above two matrices $T^*_{t\times k}$ and $D^*_{k\times d}$ describe a possible method of how a document is generated. A document $d$ is observed with some probability $Pr(d)$. Document $d$ belongs to a certain latent topic $z$ with probability $Pr(z|d)$. Each latent class consists of certain words $\{w_i\}$. To generate $d$, the words of the document $d$ are picked according to probability $Pr(w|z)$. This process describes the probability latent semantic indexing (pLSI) model, which is a probability generative model. Mathematically, it can be expressed as follows:

$$Pr(d,w) = Pr(d)Pr(w|d)Pr(w|d) = \sum_{z\in Z} Pr(w|z)Pr(z|d) \tag{2.24}$$

where $Pr(\cdot)$ is the probability function, $d \in D = \{d_1, \cdots, d_n\}$ is a document, $w \in W = \{w_1, \cdots, w_t\}$ is a word or term observed in $D$, and $z \in Z = \{z_1, \cdots, z_k\}$ is a latent variable or topic. Given a corpus[1], the above model can be estimated by the maximum likelihood estimation using the expectation maximization (EM) method [26].

In addition to the advantage of the LSI model, the pLSI model results in meaningful outputs like the probability that a document belongs to a latent topic. In practice,

[1] In text mining, a corpus is referred as a collection of documents.

it also performs better than the LSI model [52].

### *Latent Dirichlet Allocation Models*

Similar to pLSI, the latent Dirichlet allocation (LDA) model [10] is a popular probabilistic topic model to model documents. It is a three-level hierarchical Bayesian model to extract latent variables (or topics) from a corpus. LDA assumes a document can be considered as a bag of words, where orders between words are negligible. LDA can be described by the following generative process for each document $w$ in a corpus $D$:

1. Draw $K$ topics[2] from a Dirichlet distribution: $\beta_k \sim Dir_V(\eta)$;
2. For each document $d$, draw topic proportions from another Dirichlet distribution: $\theta_d \sim Dir_K(\alpha)$;
3. For each word $n$ in the document $d$,
   (a) Draw a topic: $z_{d,n}|\theta_d \sim Multinomial(\theta_d)$
   (b) Draw a word: $w_{d,n}|z_{d,n}, \beta_{1:K} \sim Multinomial(\beta_{z_{d,n}})$

By assuming the Dirichlet prior distributions, the resulting estimated topic distribution $\theta_d$ and word distribution $\beta_k$ would be sparse vectors. This is a desired property, because a document only relates to a small number of topics and a topic only includes a small number of unique words in reality. Another good property of LDA is that it can partly solve the word-sense disambiguation problem by grouping the frequently co-occurred words. For example, consider the word "bank" in the following two sentences:

(2.25) I went to the **bank** to cash the check.

[2] Here a topic is a distribution over $V$ terms

(2.26) I walked along the river **bank**.

LDA can associate the above two sentences with two different topics. The first topic consists of words “bank”, “cash” and “check”. The second includes words “bank” and “river”. This can be achieved by training the LDA model with a large amount of documents where each set of words occurred together frequently.

Algorithms for LDA inference include EM algorithm, variational inference, and Markov chain Monte Carlo (MCMC). The Gibbs sampling algorithm is the most popular one [95]. There are several open source software programs [44,93] implementing the LDA model.

Several improvements have been made to LDA. The hierarchical Dirichlet process (HDP) model [96] is developed to estimate the best number of topics from data. The nested Chinese restaurant process model (nCRP) [8] can organize topics into trees. The dynamic topic model (DTM) was developed to analyze the time evolution of topics [7,9]. Similar to LDA, these methods still do not fully incorporate semantic information into the modeling process.

### 2.3.3 Text Mining Applications

With the increasing availability of text data, the above text mining methods along with other methods have been widely applied.

*Information Retrieval*

One of the fundamental problems in text mining is to find the most related text given search terms, or information retrieval.

Broadly speaking, information retrieval is the science to efficiently search information from documents or databases. This section only considers information retrieval on text. Given a search term, information retrieval methods return the most interesting and related documents [91]. Google[3] is an example of information retrieval: users enter words, phrases, or sentences in Google, and it returns related information from the Internet.

Today, computers perform most information retrieval tasks automatically with different algorithms. Most of these algorithms are based on vector space models, probabilistic models, inference network models [91], and ranking models [19, 113]. For example, the TF- IDF model, LSI method and pLSI method can all be applied to information retrieval applications.

*Automatic Document Organization*

With more and more documents stored electronically, we need efficient methods to organize them automatically. Text mining provides the following tools for this need: information extraction, text categorization, document summarization, sentiment analysis and so on [59].

Information extraction attempts to extract certain types of information from text. For example, information extraction algorithms can highlight peoples' names, times, vehicle types, and events from a document. Information extraction methods can be rule-based [22] or statistics-based [106]. Text categorization usually applies unsupervised learning clustering methods to text and then searches the best clusters to organize documents [32, 53]. Document summarization tries to build automatic systems to summarize documents based on linguistic rules, statistics, or both [62]. Sentiment analysis focuses on the classification of a document into either positive attitude or negative attitude based on extracted subjective information from the documents [72]. For example, sentiment analysis tries to automatically label movie reviews on the

[3] `http://www.google.com`

Internet Movie Database[4] and hotel reviews on Tripadvisor[5] as positive and negative accurately.

*Prediction Using Text*

Recent text mining research focuses on utilizing the rich information within text for prediction. After structuring text data into numerical vectors, various predictive statistical models discussed in Section 2.1 can be applied to predict. For example, Yang, Spasic and their colleagues [116] predicted disease status using clinical discharge summaries. Schumaker and Chen [87] used breaking financial news to predict stock market performance.

## 2.4 Natural Language Processing

Natural language processing (NLP) develops techniques to understand and process natural human language automatically. Because text is a part of human language, all the techniques described in Section 2.3 are applicable to NLP problems. In addition to document processing and understanding, NLP studies speech recognition, machine translation, video understanding and so on. Similar to text mining, most NLP methods transform unstructured natural languages into structured data, then analyze these structured data with mathematical models or algorithms [64]. In this section, two types of advanced NLP models related to the scope of this dissertation are described.

### 2.4.1 Markov Models

One major limitation with the classic text mining models is that the orders between words are ignored. To solve this problem, NLP researchers developed statistical

[4] `http://www.imdb.com`

[5] `http://www.tripadvisor.com`

language models to compute the probability of a word sequence [31, 56]. Mathematically, let $W =< w_1, \cdots, w_n >$ [6] be a sequence of terms, a language model computes $Pr(W)$ by the following equation:

$$Pr(W) = \prod_{i=1}^{n} Pr(w_i|w_1, \cdots, w_{i-1}) \tag{2.27}$$

One of the most successful language models is the N-gram language model, which is developed from the Markov chain source model first investigated by Shannon [88]. This model computes the probability of a certain term occurring after observing previous $N$ terms in the sequence. Therefore, probabilities of sentences can be modeled by Markov functions with the moving windows of $N$ terms. Formally, it is shown in Equation 2.28.

$$Pr(W) = \prod_{i=1}^{n} Pr(w_i|w_1, \cdots, w_{i-1}) \approx \prod_{i=1}^{n} Pr(w_i|w_{i-n+1}, \cdots, w_{i-1}) \tag{2.28}$$

Another popular Markov model considering the orders between terms is the hidden Markov model (HMM). Figure 2.1 shows a general process of HMM. A sequence of observations $O =< O_1, \cdots, O_T >$ is generated as follows. First, choose the initial state $q_1 = State_i, i = \{1, \cdots, m\}$ by an initial state distribution $\boldsymbol{\pi} = \{\pi_i\}$ and set $t = 1$. Second, choose observation $O_t = v_k$ according to some probability based on the state $State_t$: $b_{state_t}(k)$. Third, transit to a new sate $q_{t+1} = State_j$ according to the state transition probability for state $State_i$: $a_{ij} = Pr(State_j|State_i)$. Fourth, set $t = t + 1$ and repeat the second and third steps until the end state achieves.

HMM can be applied to several important NLP problems. In the speech recognition problem, $O_t$ are voice signals and $q_t$ are corresponding words. In the information extraction problem, $O_t$ are terms in the documents and $q_t$ are corresponding topics.

[6] Here $W$ can be a sentence, paragraph, or document. $w_i$ can be a word, phrase, or punctuation mark. This section refers $w_i$ as “term”.

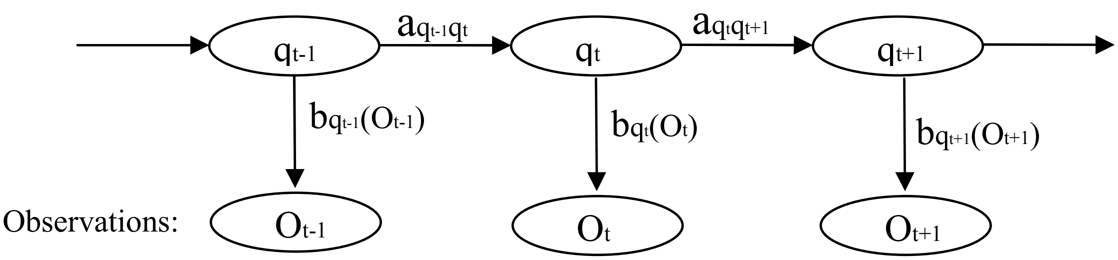


Figure 2.1: Hidden Markov Model

In the part-of-speech tagging problem, $O_t$ are terms in the documents and $q_t$ are corresponding tags.

### 2.4.2 Semantic Role Labeling

Humans read sentences by understanding not only meanings of words but also syntactic structures as well as other semantic information. We usually focus on the predicate words in sentences and related subjects and arguments surrounding them. For example, when we read the sentence in Example 2.29, we know an event "sentencing" happened to the subject "Thompson". This event was about a violent crime that happened in the October.

(2.29) Thompson was sentenced this morning in the October death of Daniel Neumeister, a 31-year-old local winemaker.

One approach to analyze text semantically is called semantic role labeling (SRL) [39]. SRL methods try to identify the verbal and nominal predicates in sentences and label the roles of surrounding entities. The most recent SRL systems [39, 79] use statistical learning models to achieve this objective. The overall process of these SRL systems includes two phases. In the first phase, sentences are parsed syntactically. The outputs of this phase are parse trees. In the second phase, nodes in the parse trees are classified into multiple classes using machine learning models. Each class represents a different semantic role. To estimate parameters of the classification models, manually labeled corpora such as Penn Treebank [65] and NomBank [68] are used. The outputs from SRL systems are labeled sentences. For example, applying the two SRL systems developed by Punyakanok et al. [79] and Gerber and Chai [40], the sentence in Example 2.29 is labeled as follows.

(2.30) $[_{e1:criminal}$ Thompson$]$ $[_{e1}$ was sentenced$]$ $[_{e1:temporal}$ this morning$]$ $[_{e1:crime}$ in the October death of Daniel Neumeister , a 31-year-old local winemaker.$]$

As we can see, the SRL systems correctly labeled the event of "sentencing" and its corresponding criminal "Thompson" as well as the criminal incident.

SRL systems provide us with important semantic information about text. Incorporating these information should be able to improve the performance of available text mining and NLP models.

# CHAPTER 3

# THE LOCAL SPATIO-TEMPORAL GENERALIZED ADDITIVE MODELS

This chapter describes a feature-based approach to model the spatio-temporal data and solve the problem described in Section 1.2. There are two models. The first one, the spatio-temporal generalized additive model (STGAM), is based on generalized additive models and built on spatio-temporal grids. It can take into account various types of features. It also has good interpretability. The second one, the local spatio-temporal generalized additive model (LSTGAM), improves STGAM by building models conditioned on different regions. In addition, this chapter addresses three issues related to the modeling process. After describing STGAM and LSTGAM, a randomized least angle regression method (RLAR) is studied to select features for nonlinear models. Then model estimation methods are described to estimate model parameters efficiently. Finally, a new method, high risk percentage versus true incident percentage (HRP-TIP) plot, is developed to evaluate the prediction performance of spatio-temporal models.

## 3.1 The Spatio-Temporal Generalized Additive Model (STGAM)

Section 1.2 defined the objective of this dissertation: to develop an efficient method to perform predictions and classifications over spatio-temporal grids with the consideration of ordinary features as well as textual features. The overall model to solve this problem is based on a generalized additive model (GAM). GAM is chosen because it has the following desired properties.

As discussed in Section 2.1, GAM assumes additivity between predictors and nonlinear relationships between individual predictors and the response variable. The additivity makes GAM able to take into account many predictors without the problem of the curse of dimensionality[1]. As shown in [15, 92], the spatial generalized linear model (GLM)[2] had better predictability than other probabilistic models built on high dimensions. The nonlinearity makes GAM more flexible in the treatment of the relationships between predictors and responses. This flexibility is helpful for modeling certain types of features. For example, consider the spatio-temporal modeling of criminal incidents. Criminals may prefer to burgle richer houses. However, they might not choose expensive houses because these houses often have security systems. This effect cannot be modeled by linear functions as used in GLM. Because of those two properties, GAM usually performs well for problems having many predictors.

GAM can also easily incorporate interactions between predictors. In addition to smooth functions $f_i(X_i)$ with individual predictors $X_i$, GAM (as shown in Equation 2.5) can include smooth functions $f_j(X_{j_1}, X_{j_2})$ with multiple predictors to model the interaction of $X_{j_1}$ and $X_{j_2}$. For spatial modeling, the smooth function on the multi-dimensional space can account for the overall spatial trend of the response variable.

Another good property of GAM is its interpretability. The outputs of GAM include estimated smooth functions, each of which describes the relationship between a predictor (or predictors) and the response variable when all the other effects from other predictors are controlled. This is helpful for applications where users are interested in studying the underlying mechanism of spatio-temporal processes. For

[1] The curse of dimensionality refers to the problem that when the dimensionality increases, the distance between any two points increases very fast. Therefore, the data points become sparse in the high dimensional space.

[2] GLM can be considered as a special case of GAM.

example, law enforcement agencies are interested in not only where and when the future crimes will be, but also how a specific feature, such as the distance to the nearest gas station, affects the criminal activity. By plotting the smooth function of the distance versus the crime probability, law enforcement agencies can study the effect easily. For instance, they can find out at what distance the crimes are most likely to occur. Another example is the modeling of the spatio-temporal pattern of a disease. The interpretable outputs can help doctors to find out the possible causation of the disease and to prevent the disease in the future.

Furthermore, GAM can be estimated with well developed algorithms. Since formally developed by Hastie and Tibshirani [49,50] in the 1980s, GAM has been studied thoroughly in statistical sciences. Efficient estimation algorithms have been developed theoretically [50, 109] and implemented in statistical software [48, 112]. As long as a model can be formulated as GAM, it can be estimated directly with available algorithms.

The overall model, the spatio-temporal generalized additive model (STGAM), developed in this dissertation is formulated as follows.

*STGAM for Classification*

For the classification problem defined in Section 1.2, the major objective is to find a probability function:

$$\Pi_{[g,t]} = Pr[x_{[g,t,P]} = 1 | \mathbb{X}_{t' \leqslant t^c}, \mathbb{S}, \mathbb{T}] \tag{3.1}$$

such that $\Pi_{[g,t]}$ has high values the for the locations and time where $x_{[g,t,P]} = 1$ and low values for the locations and time where $x_{[g,t,P]} = 0$. Here $x_{[g,t,P]}$ is the feature of interest $X_P$ (or the response variable) at the spatial grid $s_g$ and time interval $t$.

Let $Y$ denote $X_P$. The STGAM for the classification problem has the following form:

$$\text{logit}[Pr(y_{g,t} = 1)] = f_0(s_g) + \sum_{n=1}^{N} f_n(x_{[g,t,n]}) + \kappa_{g,t} \tag{3.2}$$

where:

1. $\text{logit}(p) = \log(\frac{p}{1-p})$ is a logit link function as used in GLM.

2. $f_0(s_g)$ is a smooth function built on the entire space $\mathbb{S}$. It accounts for the overall properties across the space.

3. $N$ is the total number of useful features for the modeling. If all the available $P - 1$ features are used, $N = P - 1$.

4. $x_{[g,t,n]}$ is the $n^{\text{th}}$ feature associated with location $s_g$ and time $t$.

5. $f_n$ is the smooth function of the $n^{\text{th}}$ feature to be estimated from data.

6. To include temporal information of previous incidents[3], STGAM applies the idea of the binary time-series-cross-sectional data (BTSCS) model[4] from Beck, Katz, and Tucker [4], because at a given spatial grid $s_g$, $Y_g$ and the corresponding features are essentially BTSCS data. In the above Equation 3.2, $\kappa_{g,t}$ is the dummy variable indicating the length of the continuous zeros (no incident happened) that precede the current observation of $Y$ at location $s_g$ and time $t$. An example of the values of $\kappa_{g,t}$ is shown in Table 3.1. Notice that $\kappa_{g,t}$ is a dummy

[3] In this dissertation, an incident means the occurrence of the event of interest or $Y = 1$.

[4] The BTSCS model is developed based on the observation that BTSCS data are equivalent to grouped duration data, which can be modeled by the proportional hazards model [23]. In the BTSCS model, a temporal dummy variable $\kappa$ is used to indicate when the last incident happened.

variable, and its values are factors instead of integers. Usually, $\kappa_{g,t} = 1, \cdots, K$. Here $K$ is the maximum length of the continuous zeros considered. For example, if the last incident happened before $K$ time intervals at location $s_g$ and time $t$, then $\kappa_{g,t} = K$.

Table 3.1: An Example of the Values of $\kappa$

| $t$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $Y_{g,t}$ | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| $\kappa_{g,t}$ | 1 | 2 | 3 | 1 | 1 | 2 | 1 | 2 | 1 | 2 |

The STGAM for classification can be extended to model more complex spatio-temporal patterns in the following two ways. First, higher dimensional smooth functions $f_h(x_{[g,t,n_1]}, x_{[g,t,n_2]})$ can be added to consider interactions between features $X_{n_1}$ and $X_{n_2}$. Second, $\kappa_{g,t}$ can be defined on the values of $Y_{g'}$ of the neighborhood grids $\{s_{g'}\}$. In this case, $\kappa_{g,t}$ means the time length that no incident happened within the neighborhood of location $s_g$ before time $t$.

*STGAM for Regression*

The above model 3.2 can be modified as shown in Equation 3.3 to solve the regression problem defined in Section 1.2.

$$E[y_{g,t}] = f_0(s_g) + \sum_{n=1}^{N} f_n(x_{[g,t,n]}) + f_T(Y_{g,t-1:t-K}) \tag{3.3}$$

In Equation 3.3,

1. $f_0(s_g)$, $N$, and $f_n(x_{[g,t,n]})$ are defined the same as in Equation 3.2.
2. $E[y_{g,t}]$ is the expectation of $Y$ at location $s_g$ and time $t$.

3. $f_T(Y_{g,t-1:t-K})$ models the temporal effect of the past values of $Y$ at location $s_g$. $f_T$ is a smooth function estimated from the data. $Y_{g,t-1:t-K}$ is a vector of the response values at location $s_g$ during the time period between $t-1$ and $t-K$. $K$ is the same as defined in the STGAM for classification.

Similar to the STGAM for classification, Equation 3.3 can be extended by including higher order smooth functions and utilizing the response values at neighborhood grids.

*The General STGAM*

Generally, if the response variable $Y$ is from the exponential family, STGAM can be formulated as shown in Equation 3.4.

$$\mathcal{G}\left(E[y_{g,t}]\right) = f_0(s_g) + \sum_{n=1}^{N} f_n(x_{[g,t,n]}) + f_T\left(f^{feature}(Y_{NN(g),t-1:t-K})\right) \tag{3.4}$$

This model is explained as follows:

1. $\mathcal{G}$ is a link function. For different distributions of $Y$, $\mathcal{G}$ can have the different forms as defined in Table 2.1.

2. $E[y_{g,t}]$ is the expectation of $Y$ at location $s_g$ and time $t$.

3. $f_0(s_g)$ is a smooth function built on the entire space $\mathbb{S}$ to model the overall properties across the space.

4. $N$ is the total number of useful features for the modeling.

5. $f_n$ is the smooth function of the $n^{\text{th}}$ feature to be estimated from data.

6. $x_{[g,t,n]}$ is the $n^{\text{th}}$ feature associated with location $s_g$ and time $t$.

7. $f_T$ is a smooth function to be estimated from data. It models the temporal effect of the past values of $Y$.

8. $f^{feature}$ is a predefined function to calculate temporal features from the past response values of $Y$. For example, in the STGAM for classification (Equation 3.2), $f^{feature}(\cdot) = \kappa_{g,t}$.

9. $NN(g)$ includes spatial grids close to $s_g$. It is defined by $NN(g) = \{g' | dist(s_{g'}, s_g) \leq \delta_{dist}, g' \in \mathbb{G}\}$. Here, $dist(s_{g'}, s_g)$ measures the spatial distance between two grids $s_g$ and $s_{g'}$. $\delta_{dist}$ controls the size of the neighborhood.

10. $Y_{NN(g),t-1:t-K}$ are the past values of $Y$ in the neighborhood $NN(g)$ and the time period between $t-1$ and $t-K$. Equations 3.2 and 3.3 are the special case where $\delta_{dist} = 0$ and $NN(g) = s_g$.

## 3.2 The Local Spatio-Temporal Generalized Additive Model (LSTGAM)

STGAM assumes that all grids in the area $\mathbb{S}$ have the same underlying pattern. A single model is built for the entire area. In reality, this assumption might not be satisfied. When the entire area $\mathbb{S}$ is large, it is likely to have multiple regions $\mathbb{S}_r$ where each one has its own pattern. For example, suppose we have all criminal incident data of a state, including big cities, small towns and rural counties. Different types of regions may have different criminal patterns. Even for a small region $\mathbb{S}_r$, the same feature might impact high risk areas (such as crime hot spots) differently from low risk areas.

To account for this situation, STGAM is extended to the local spatio-temporal generalized additive model (LSTGAM) as follows:

$$E[y_{g,t}] = \sum_{r=1}^{R} I(s_g \in \mathbb{S}_r) \cdot E_r[y_{g,t}] \tag{3.5}$$

$$\mathcal{G}\left(E_r[y_{g,t}]\right) = f_{r,0}(s_g) + \sum_{n=1}^{N} f_{r,n}(x_{[g,t,n]}) + f_{r,T}\left(f^{feature}(Y_{NN(g),t-1:t-K})\right)$$

$$\text{, for all } r \in \{1, \cdots, R\} \tag{3.6}$$

Here are the explanations of the above LSTGAM model:

1. Equation 3.5 models the response variable $Y$ over the entire area $\mathbb{S}$ with $R$ regions.

2. $R$ is the total number of regions in $\mathbb{S}$. $\mathbb{S}_r$ is the $r^{th}$ region, where $\{\mathbb{S}_r | \mathbb{S}_r \subset \mathbb{S}, r \in \{1, \cdots, R\}\}$ satisfies $\cup_r \mathbb{S}_r = \mathbb{S}$ and $\mathbb{S}_{r_i} \cap \mathbb{S}_{r_j} = \emptyset$ $(r_i \neq r_j)$.

3. $I(\cdot)$ is an indicator function with values of 0 and 1.

4. $E_r[y_{g,t}]$ models the expected value of $Y$ at location $s_g$ and time $t$ within the region $r$. Equation 3.6 defines $E_r[y_{g,t}]$. As we can see, it has the same form as Equation 3.4.

5. Notice that there are actually $R$ equations in the form of Equation 3.6 and each one may have different smooth functions $f_{r,\cdot}(\cdot)$.

LSTGAM can be considered as a two stage model. The first stage is to decide which region a grid $s_g$ belongs to. The second stage is to build different STGAM to model the expected value of $Y$ for each region. Clearly, STGAM is the special case of LSTGAM with $R = 1$.

### 3.3 Feature Selection

To use the above LSTGAM model, we need to decide what features the model should use. Simply using all the available features has the following problems. First, it requires too much unnecessary computation. For example, LSTGAM uses spatio-temporal grids. For an area of 16 square miles and a time period of one year, if we use the resolution of 0.02mile × 0.02mile for the space and 1 day for the time, there are about $1.46 \times 10^7$ records. This means we need to compute more than ten million data points if we include one additional feature. Second, including irrelevant features means adding noise to the model estimation process. Such noise will make predictions less accurate. Third, a model with a large number of predictors is harder to be interpreted and analyzed than a model with less predictors, because users need to spend more time to study each individual factor.

Therefore, before applying LSTGAM, we need to select features to be included in the model. This is the feature selection problem described in Section 2.1.3. As discussed in that section, among available feature selection methods, least angle regression (LAR) is a fast method to select features for linear regression models with good performance. This dissertation develops a randomized algorithm based on LAR to select features for nonlinear additive models. Before introducing the algorithm, let us first review the basic LAR method and its limitation.

### 3.3.1 Review of Least Angle Regression (LAR)

For linear regression models, the feature selection problem is equivalent to selecting the best subset of features with any given size such that the variance of the response variable is explained the most. Mathematically, it can be formulated as follows:

$$\min \frac{1}{n} \sum_{i=1}^{n} \| \sum_{i=1}^{p} \beta_i x_i - y_i \| \tag{3.7}$$

$$\text{s.t.} \sum_{j=1}^{p} |\hat{\beta}_j| < \delta \tag{3.8}$$

The above optimization problem is the so-called the Lasso[5] model [97]. It is equivalent to the penalized regression problems defined in Equation 2.12 with $\gamma = 1$ and can be solved by the convex optimization methods as used in [37, 77, 97].

LAR is a computationally efficient method to get the solution equivalent to Lasso. Algorithm 1 shows the general steps of LAR [17, 30]. As we can see, for a problem with $p$ features, LAR computes the order of features entering the regression model within $p$ steps. After getting the order, we can choose the best number $q(< p)$ of features based on criteria like mean squared errors estimated by cross validation. If $q$ is the best number, all features entering the regression model before the $q^{\text{th}}$ step are selected. Thus, the order of features entering the regression model can be considered as the priority of features. The earlier a feature enters the regression model, the more important it is.

LAR solves the problems with quantitative response variables. If the response variable is not continuous, the $L_1-$regularization path algorithm [76] can be used for generalized linear models. Algorithm 2 shows the general steps of this method. Similar to LAR, the output of this algorithm is the order of predictors entering the models.

In practice, LAR can not only reduce the size of features for modeling, but also improve the prediction performance. For example, let us consider the communities and crime data set[6] [82]. In the data set, the response variable is the total number of violent crimes per 100,000 population. The cleaned data set includes 99 features related to the incidents of violent crimes. All data were normalized into the range 0.00-1.00. Two-thirds of the data were used for training and the rest were kept as the test set to compute predicted root mean squared errors (RMSE). Three types of

[5] Lasso stands for “Least Absolute Shrinkage and Selection Operator”.

[6] In this example, the data were cleaned by removing columns with “NA”. The first five columns about community information and the fold number were also removed.

**Algorithm 1** LAR

```
1: Initialization:
2:     standardize the variables;
3:     set residuals: $Resid = Y$;
4:     set initial parameters: $\beta_j = 0$ for all $j \in \{1, \cdots, p\}$ ($p$ is the total number of features);
5: Find $X_j$ most correlated with $Resid$ and update $\hat{\beta}_j$ as follows:
6:     increase $\hat{\beta}_j$ in the direction of sign($corr(Resid, X_j)$);
7:     let $Resid = Y - \hat{\beta}_j X_j$;
8:     stop when $corr(X_k, Resid) = corr(X_j, Resid)$ for some $X_k$;
9: Update both $\hat{\beta}_j$ and $\hat{\beta}_k$ in the joint least squares direction until some variable $X_i$ has the same amount of correlation with the current residual and enter $X_i$ into the model;
10: Continue the above step with all the entered predictors until all predictors have been entered into the regression model or stop when $corr(X_j, Resid) = 0, \forall j$.
```

**Algorithm 2** $L_1-$regularization Path for GLM

```
1: Initialization: set $\lambda_0 = \lambda_{max}$, such that the intercept $\beta_0$ is the only non-zero coefficient; ($\lambda$ is the regularization parameter as defined in Equation 2.12.)
2: For the $k^{th}$ step ($k > 0$),
3:     set $\lambda_k = \lambda_{k-1} - \Delta_k$ such that a new variable enters the model;
4:     calculate the linear approximate change in $\beta$: $\hat{\beta}^{k-} = \hat{\beta}^{k-1} + (\lambda_{k+1} - \lambda_k)\frac{\partial \beta}{\partial \lambda}$
5:     find the exact solution of the estimated parameters $\hat{\beta}^k$ using $\hat{\beta}^{k+}$ as the starting value;
6:     check whether the size of predictors must be modified; if so repeat line 5 to update $\hat{\beta}^k$;
7: repeat lines 2 to 6 until no predictor can be added.
```

models were built: a linear regression model using all 99 predictors, a linear regression model using the predictors selected by stepwise selection; and linear regression models using different numbers of predictors ranked by LAR. Each model was estimated by the training set and the predicted RMSE was computed using the test set. The variances of the predicted RMSE were estimated by 500 replicates of sampling the training and test sets. Figure 3.1 shows the results. The x-axis is the number of top-ranked predictors by LAR used in the linear models to predict violent crimes. It's only related to the third type of models. The y-axis is the average predicted RMSE. The black line shows the predicted RMSE of the full model using all predictors. The blue line shows the predicted RMSE of the stepwise model. Roughly there were 45 predictors in the stepwise model. The red line shows the predicted RMSE of the LAR models. As we can see, the full model performed slightly better than the stepwise model, but with a cost of using about 50 more predictors. The LAR model with 15 predictors performed better than the full model with 99 predictors. After 25 predictors, the performance of LAR models no longer improved.

Although LAR can select a small subset of features effectively and efficiently for linear regression models, if predictors are nonlinearly correlated with the response, it might not be able to select features correctly. For instance, consider the following simulation problem. Suppose the response variable $y$ is determined as:

$$y = (x_1 - 0.5)^2 + 0 \cdot x_2 \tag{3.9}$$

where $x_1, x_2 \sim U(0,1)$ and $U(0,1)$ denotes the uniform distribution ranging from 0 to 1. Given $D_s = \{x_1, x_2, y\}$, a good feature selection method should rank $x_1$ with higher priority than $x_2$. However, applying LAR to $D_s$, $x_2$ is ordered before $x_1$.

LAR does not work in the above example because $y$ is quadratically related to $x_1$. The linear correlation between these variables is close to zero as shown by the

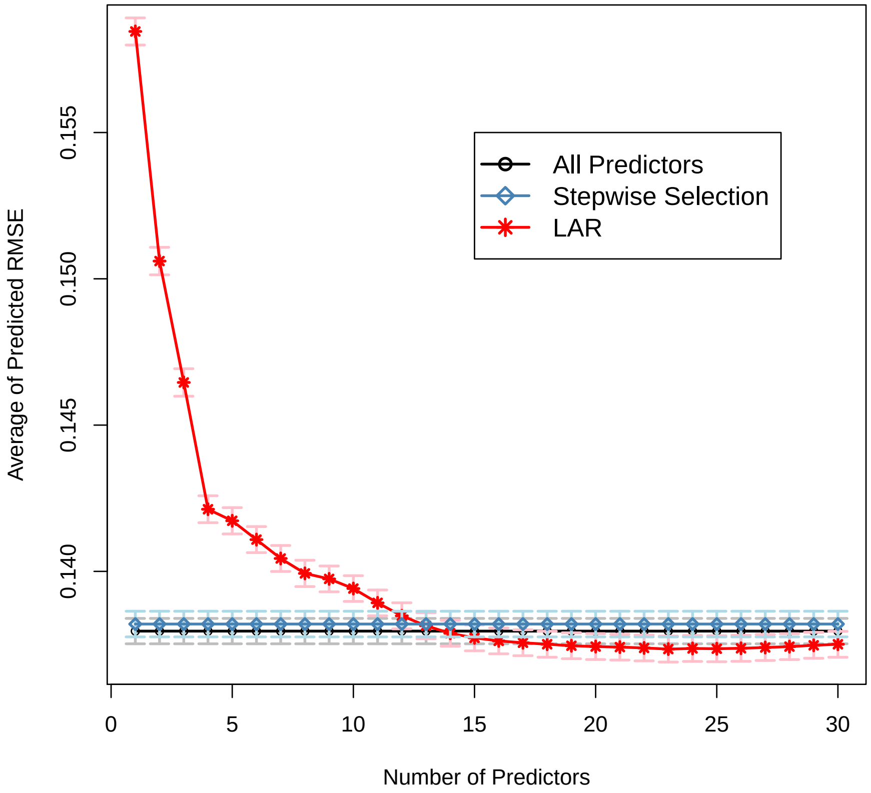
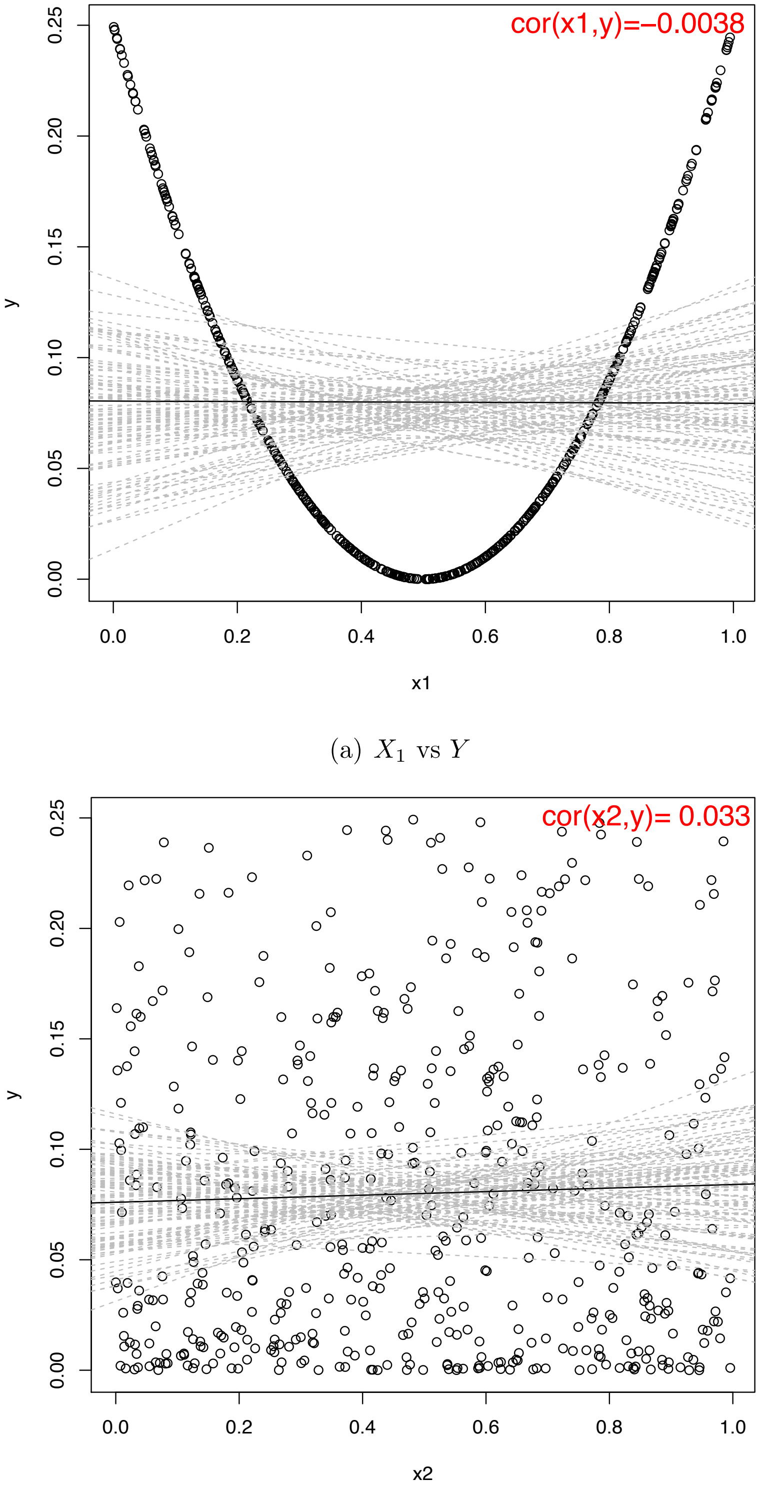


Figure 3.1: Comparison of Full Model, Stepwise Model and LAR Models

horizontal black line in Figure 3.2(a). Because $y$ is independent of $x_2$, the linear correlation between them is also close to zero. Due to the randomness of $x_1$ and $x_2$, the linear correlation between $y$ and $x_2$ was slightly higher. Since LAR measures the linear correlation between predictors and the response, $x_2$ was more important than $x_1$.

### 3.3.2 The Randomized LAR (RLAR)

This problem shown in Figure 3.2 might be solved if we sample subsets from $D_s$ randomly. For example, the grey dashed lines in Figures 3.2(a) and 3.2(b) measure



(a) $X_1$ vs $Y$

(b) $X_2$ vs $Y$

Figure 3.2: Relationships between $X_1$, $X_2$ and $Y$

the linear relationships between variables using samples of 10% of $D_s$. As shown, the slopes of the grey lines are generally greater than the slopes of black lines. Also, the variance of the coefficients of the linear models for $x_1$ and $y$ is greater than the variance for $x_2$ and $y$. Another possible approach to solve the nonlinearity problem is by transforming predictors. However, it is usually unknown whether we should transform a specific predictor or not. In addition, it is unknown what the best transformation is to perform for each individual predictor. Therefore, the sampling method is better for general data sets.

A randomized LAR (RLAR) method[7] is developed to introduce randomness into the data set to improve LAR when the relationships between variables might be nonlinear. The method is described as follows.

Instead of applying LAR directly to the data set $D_{n \times p}$[8], the method first samples $N_{sample}$ data sets $D^s_{m \times q}$ from $D_{n \times p}$, where $m < n$ and $q < p$. LAR is then applied to each $D^s$ to rank features. The feature priority for $D^s$ is $rank^s = \langle feature_{s1}, \ldots, feature_{sq} \rangle$. In the last step, feature priorities are voted on by $\{rank^s\}$. The complete algorithm is shown in Algorithm 3. In Algorithm 3, there are two hyper-parameters. The first one is the number of sampling times $N_{sample}$. The larger this number is, the more stable the final rank is. The second one is the size of each sample $D^s$ or $m$ and $q$. A large size of $D^s$ introduces too little randomness, while a small size of $D^s$ requires more sampling times to ensure all the data points can be sampled. Based on my preliminary tests, the following setting works well: $N_{sample} \geq 100$, $\frac{m}{n} \leq \frac{2}{3}$, and $\frac{q}{p} \leq \frac{2}{3}$.

By applying RLAR to the example given in Equation 3.9, the result was the desired ranking of $x_1$ higher than $x_2$.

[7] As shown later in Algorithm 3, RLAR can be applied to the problem with non-continuous response variables by utilizing $L_1-$regularization path algorithm. For simplicity, this dissertation still calls it as randomized LAR or RLAR.

[8] $n$ is the number of observations and $p$ is the number of predictors

**Algorithm 3** Randomized LAR

```
**Input**: data set $D = \{x, y\}$
**for** s from 1 to $N_{sample}$ **do**
    sample $D^s$ from $D$
    depending on the type of the response variable, apply LAR or $L_1-$regularization path algorithm on $D^s$ to get ranked features $rank^s = \langle feature_{s1}, \ldots, feature_{sp} \rangle$; where $feature_{si}$ is added in the $i^{th}$ step of LAR or $L_1-$regularization path algorithm
**end for**
**for** i from 1 to p **do**
    **for** s from 1 to $N_{sample}$ **do**
        **if** $feature_i \in rank^s$ **then**
            $vote_i = vote_i + r_{si}$, where $rank^s[r_{si}] = feature_i$
            $sample.time_i = sample.time_i + 1$
        **end if**
    **end for**
    $vote_i = \frac{vote_i}{sample.time_i}$
**end for**
**Output**: ranked features: the smaller $vote_i$ is, the more important $feature_i$ is.
```

### 3.3.3 Evaluation of RLAR

This subsection evaluates the performance of RLAR and compares it with LAR using both simulation data and real data.

*Evaluation with Simulation Data*

First, consider the following simulation problem:

$$y = 1000(x_1 - 0.5)^2 + 500(x_2 - 0.5)^2 + \sum_{i=3}^{50} 0 \cdot x_i + \epsilon \tag{3.10}$$

where $y$ is the response variable; $x_i$ ($i \in \{1, \cdots, 50\}$) are predictors; $x_i \sim U(0,1)$; $\epsilon \sim N(0,1)$ is random noise; $U(0,1)$ denotes the uniform distribution ranging from 0 to 1; and $N(0,1)$ denotes the Gaussian distribution with the mean of 0 and the variance of 1.

As we can see from Equation 3.10, the response variable $y$ depends only on the two predictors $x_1$ and $x_2$. Let $D = \{y, x_1, x_2, \cdots, x_{50}\}$ be the data set and apply both LAR and RLAR on this data set to rank predictors. A good feature selection method should rank $x_1$ and $x_2$ higher than all the other predictors. Figure 3.3 shows the ranks of $x_1$ and $x_2$ from both LAR and RLAR methods based on 100 simulations and each simulation had the data set with 100 observations. The left two box plots show the ranks from LAR and the right two box plots show the ranks from RLAR. Both $x_1$ and $x_2$ had higher ranks from RLAR than the corresponding ranks from LAR. Especially, RLAR ranked $x_1$ with very high priorities, the median of which was 2. On the contrary, the median of the ranks of $x_1$ from LAR was 22. To see whether the differences were significant, the one-sided paired Wilcoxon tests were performed. The p-values were $2.67 \times 10^{-16}$ and 0.002928 for $x_1$ and $x_2$ respectively. At the significance level of 0.05, both differences were significant.

Next, consider a more complex problem:

$$y = \sum_{i=1}^{4} a_i (x_i - b_i)^i + \sum_{i=5}^{50} 0 \cdot x_i + \epsilon \tag{3.11}$$

where $y$ is the response variable; $x_i$ ($i \in \{1, \cdots, 50\}$) are predictors; $x_i \sim N(0,1)$; $\epsilon \sim N(0,1)$ is random noise; $a_i, b_i \sim U(0,1)$; $U(0,1)$ denotes the uniform distribution ranging from 0 to 1; and $N(0,1)$ denotes the Gaussian distribution with the mean of

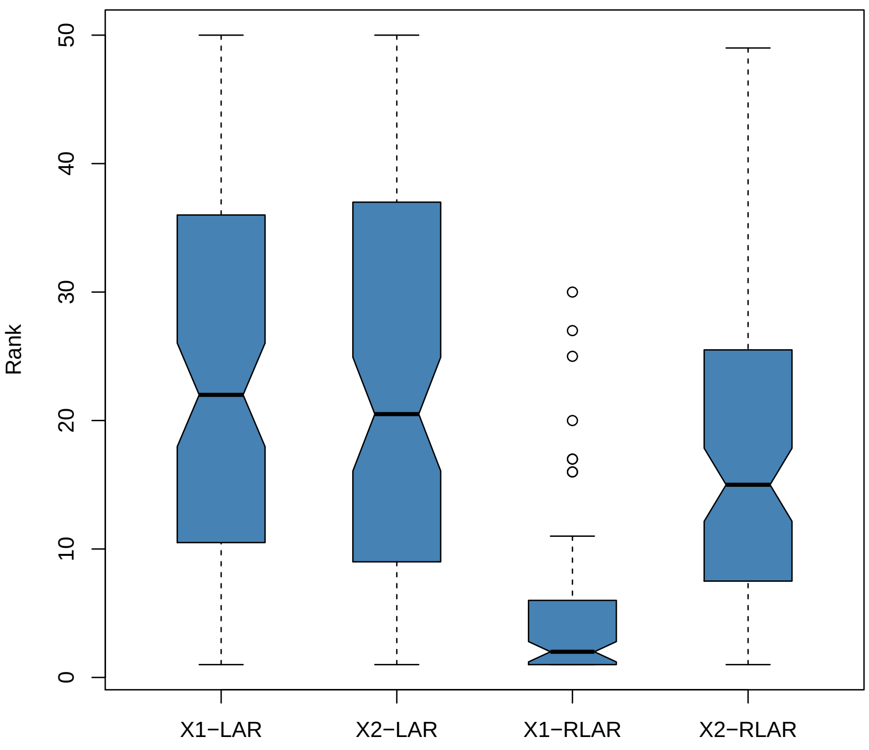


Figure 3.3: Evaluation of RLAR on the Two-Variable Selection Problem

0 and the variance of 1.

As defined in Equation 3.11, the response variable $y$ is a polynomial function of the predictors $x_1$, $x_2$, $x_3$ and $x_4$. Similar to the previous simulation test, both LAR and RLAR were applied to the data set $D = \{y, x_1, x_2, \cdots, x_{50}\}$ to rank predictors with 100 simulations and each simulation with 100 observations. Instead of comparing individual ranks[9], the following two metrics were used to compare the performance:

$$\mu(rank) = \frac{1}{4}\sum_{i=1}^{4} rank(x_i) \tag{3.12}$$

$$\max(rank) = \max\{rank(x_1), rank(x_2), rank(x_3), rank(x_4)\} \tag{3.13}$$

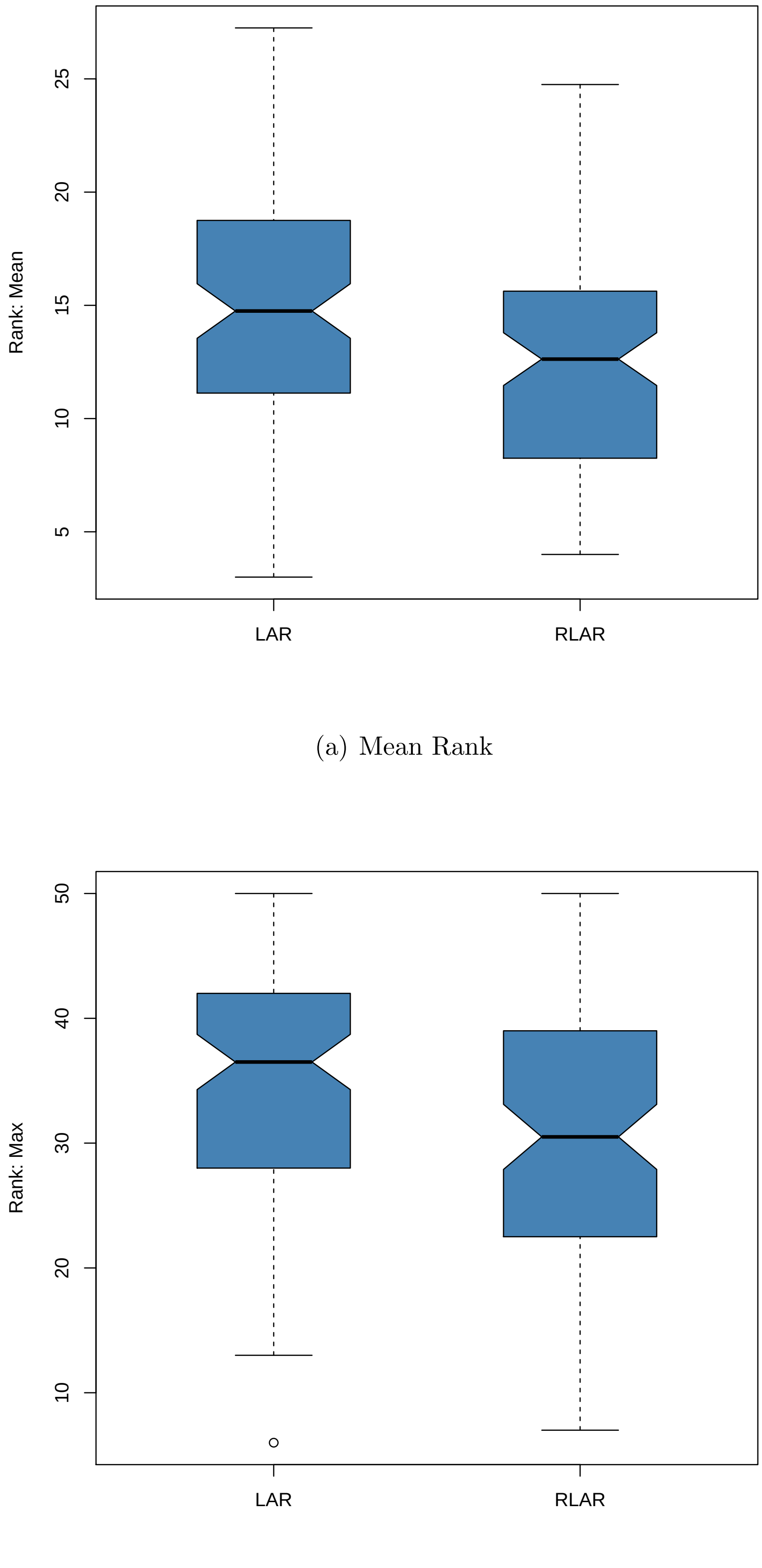
The first metric measures the mean of the ranks. The second one measures the maximum of the ranks. The second metric also measures the minimum number of features to be selected so that all the truly related features can be included. A good method should have low values for both metrics. Figure 3.4(a) shows the results on the mean metric and Figure 3.4(b) shows the results on the max metric. As we can see, RLAR performed better than LAR on either metric. Two one-sided paired Wilcoxon tests had the values of 0.0003245 and 0.003951. Therefore, the differences were significant.

*Evaluation with Real Data*

The task to predict violent crimes as described in Section 3.3.1 can be used to compare RLAR and LAR. As before, two-thirds of the data were used for training and the rest were used as test set to compute RMSE.

In the comparison, RLAR and LAR were applied to the training set to rank features. Next, the linear and additive models with different numbers of features were built and the RMSE of each model was calculated using the test set. Figure 3.5 shows the result. The x-axis is the number of top-ranked predictors used to predict the violent crimes. The y-axis is the average predicted RMSE. The black line is from the linear models using the features selected by LAR; the blue line is from the additive models using the features selected by LAR; and the red line is from the additive

[9] This is because we do not care about the ranks among $x_1, x_2, x_3$ and $x_4$. Instead we are interested in the overall ranks of them.



(a) Mean Rank

(b) Max Rank

Figure 3.4: Evaluation of RLAR on the Four-Variable Selection Problem

models using the features selected by RLAR. As shown in the plot, the additive models performed better than the linear model. Given any number of predictors, RLAR performed the same or better than LAR. RLAR achieved the lowest RMSE with 12 predictors whereas LAR achieved the lowest RMSE with 17 predictors.

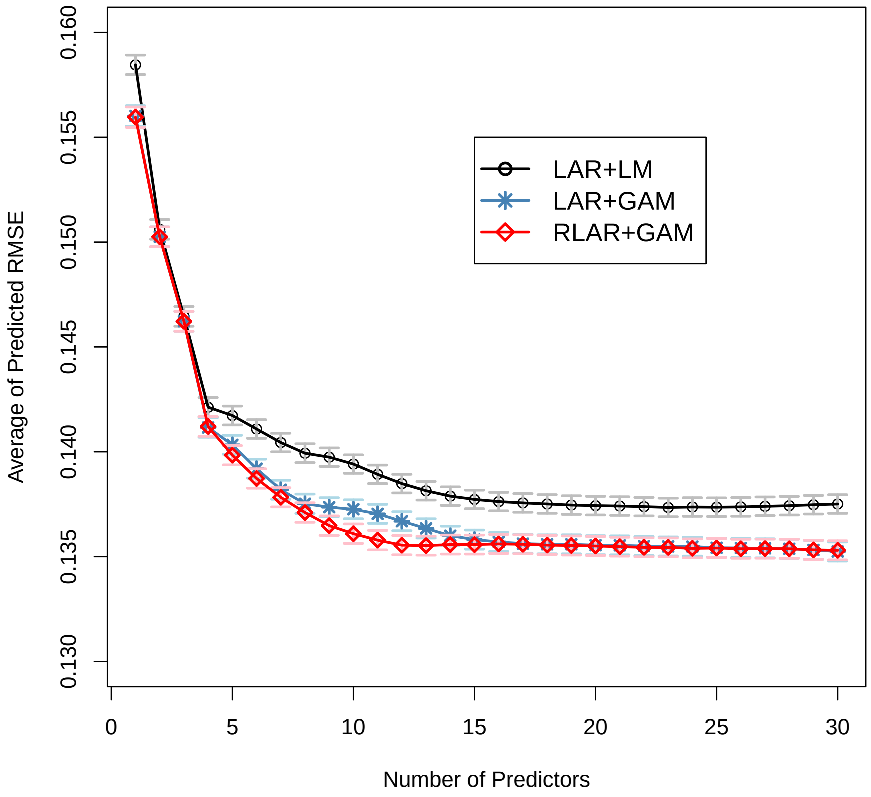


Figure 3.5: Comparison of Models with Different Number of Features

Based on the above evaluations, we can conclude that RLAR performed better than LAR.

## 3.4 Model Estimation

### 3.4.1 Estimation of STGAM

A good property of STGAM is that it has the form of a regular GAM. Therefore, it can be estimated as a standard GAM. GAM has been studied extensively in many different research areas. There are well developed methods and algorithms available to estimate GAM efficiently. Standard statistical softwares, such as R, S-plus and SAS, also have implements to estimate GAM. This section reviews the method to estimate GAM based on the work by Wood [109]. Details about this method and other estimation methods can be found in [50, 109].

*Review of Estimation of GAM*

The fundamental question for the GAM estimation is what the smooth functions $f.$ are. Intuitively, we need to find smooth curves which describe relationships between predictors and the response. This is usually achieved by fitting high-order polynomial functions locally. To estimate the GAM model in Equation 3.4, the smooth function $f.(x)$ is first represented by a sum of basis functions:

$$f.(x) = \sum_{i=1}^{B} \beta_i \cdot b_i(x) \tag{3.14}$$

where $b_i(x)$ is the $i^{th}$ basis function; and $\beta$ are the unknown parameters to be estimated.

A popular choice of basis functions is the cubic regression spline[10]. The basis

[10] A smooth function with more than one predictor, such as $f_0(s_g)$ with two dimensional predictors in Equation 3.4, can also be represented by a sum basis functions. A popular choice is the thin plate regression splines [107].

functions for this spline include:

$$b_1(x) = 1 \tag{3.15}$$

$$b_2(x) = x \tag{3.16}$$

$$b_{i+2}(x) = R(x, x_i^*) \tag{3.17}$$

where $\{x_i^* | i \in \{1, \cdots, B-2\}\}$ are knots of the spline; and $R(x,z)$ is defined as follows [45, 109]:

$$\begin{aligned} R(x,z) =& \frac{1}{4}\left[\left(z-\frac{1}{2}\right)^2 - \frac{1}{12}\right]\left[\left(x-\frac{1}{2}\right)^2 - \frac{1}{12}\right] \\ &- \frac{1}{24}\left[\left(|x-z|-\frac{1}{2}\right)^4 - \frac{1}{2}\left(|x-z|-\frac{1}{2}\right)^2 + \frac{7}{240}\right] \end{aligned} \tag{3.18}$$

By representing all smooth functions $f_{\cdot}(x)$ with basis functions, Equation 3.4 becomes a GLM:

$$\mathcal{G}\left(E[y_{g,t}]\right) = \beta_0 + \sum_{n=0}^{N}\sum_{i=1}^{B_n} \beta_{n,i} \cdot b_{n,i}(x_{[g,t,n]}) + \sum_{i=1}^{B_T} \beta_{T,i} b_{T,i}\left(f^{feature}(Y_{NN(g),t-1:t-K})\right) \tag{3.19}$$

The above GLM can be estimated efficiently by the penalized iteratively reweighted least squares method (P-IRLS) [109]. As discussed in [108], P-IRLS might not be able to converge in some cases. In recent work [111], Wood proposed a method to approximate the restricted maximum likelihood (REML) by a Laplace approximation and used Newton-Raphson iteration to fit GLM. This method is more computationally stable than previous estimation methods and has better convergence property. It has a computational cost on the order of $O(Mnq^2)$, where $M$ is the number of smoothing parameters, $n$ is the number of observations, and $q$ is the number of basis coefficients. The implementation of this method is available in the "mgcv" package in R [112].

*Subsampling for Classification STGAM*

The training data set for STGAM is usually huge, because spaces and time intervals are represented by grids. For example, in our modeling of criminal incidents in Charlottesville, Virginia[11] with the grid size of 32m × 32m and the time interval of one month, there were 1,062,094 records for the 46 month time period [100]. It is time consuming to estimate parameters using all the records. For the regression problem, we can use a random sample to estimate parameters. For the classification problem, the response variable is usually a sparse vector. For example, in the above example of criminal incidents modeling, there were about only 1,700 records with the response of 1. It is very likely to have a random sample with too few incidents. For the STGAM for classification, the following subsampling method is suggested.

To generate a sample from the records, all the records with the response of 1 and a random sample from the records with the response of 0 are included. This is a biased sampling method. Based on the analysis in [57], the effect from this biased sampling can be approximately corrected by adding an offset term $O$:

$$O = \log\left(\frac{\text{sample size}}{\text{total number of records}}\right) \tag{3.20}$$

in the estimation process. Thus, the subsampling technique can reduce the size of training set and save estimation time.

However, the subsampling method introduces stochastic effects to parameter estimates. If possible, using all of training data to estimate parameters is preferred.

[11] The total area of the city is approximately 26.6 km$^2$

### 3.4.2 Estimation of LSTGAM

There are two components of LSTGAM estimation. The first component is to estimate the STGAM or Equation 3.6 in each region $\mathbb{S}_r$. This can be done by the above method of GAM estimation. The second component is to define or estimate regions $\{\mathbb{S}_r\}$.

$\{\mathbb{S}_r\}$ can be defined by domain knowledge. For example, if law enforcement agencies believe that criminal patterns are different in different cities, each $\mathbb{S}_r$ can be a different city. When no such knowledge is available, we can estimate $\{\mathbb{S}_r\}$ with features $\{X.\}$. Any unsupervised learning method discussed in Section 2.1.2 can be applied. For example, for the spatio-temporal classification problem, the region with high risk usually has different underlying patterns from the low risk region. Algorithm 4 shows how to generate $\{\mathbb{S}_r\}$ based on the observed incident density.

**Algorithm 4** Region Generation for LSTGAM

```
Estimate the incident density over the entire area $\mathbb{S}$:
    $\{d_g | d_g \in [0,1], s_g \in \mathbb{S}\}$;
Pick threshold points:
    $\{d_1^*, \cdots, d_{R-1}^* | 0 = d_0^* < d_1^* < d_2^* < \cdots < d_{R-1}^* < d_R^* = 1\}$;
$\{\mathbb{S}_r\}$ based on the incident density are:
    $\mathbb{S}_r = \{s_g | d_{r-1}^* \leq d_g < d_r^*, s_g \in \mathbb{S}\}$.
```

## 3.5 Model Evaluation

For spatio-temporal regression problems, models can be evaluated with widely accepted metrics like predicted mean square errors. However, there is no unique metric available for the spatio-temporal classification problems. This section describes a new method to evaluate the prediction performance of the spatio-temporal classification models. This method can also help users to pick thresholds for the decision function such as the one defined in Equation 1.2.

As discussed in Section 1.2, we want to minimize the loss function $L(\cdot)$ defined in Equation 1.3 for the classification problem. The first part of the function $\sum_{g\in\mathbb{G}^*,t\in\mathbb{T}^*} weight_0 \cdot I(\delta_{[g,t]} = 0|x_{[g,t,P]} = 1)$ is the weighted sum of the times of incorrect predictions for the spatial grids and time when incidents actually happen. The second part of the function $\sum_{g\in\mathbb{G}^*,t\in\mathbb{T}^*} weight_1 \cdot I(\delta_{[g,t]} = 1|x_{[g,t,P]} = 0)$ is the weighted sum of times of incorrect predictions for the spatial grids and time when no incident happens. To minimize the first part, the probability model should predict high probabilities for the spatio-temporal grids where incidents actually happen. To minimize the second part, the total size of the spatio-temporal grids with high probabilities should be small within a given time period because of the sparseness of incidents over the entire space $S$.

Both criteria are important. The first criterion means the model should not miss a high risk area so that users can know all future locations and times of incidents. The second criterion is important, because users usually have limited resources and only a part of an area can be focused on. For example, law enforcement agencies need to predict high risk areas. They have limited watch units. At a given time, only a small area can be patrolled. With a good prediciton model, they can better allocate limited resources to help prevent crimes. Based on these criteria, the high risk percentage (HRP) versus true incident percentage (TIP) method is developed to evaluate the performance of spatio-temporal predictions within a given time period.

To measure the performance of a model at location $\mathbb{S}^* = \{s_g\}$ and time period $\mathbb{T}^* = \{t\}$, the method first computes:

$$\text{HRP}_\delta = \frac{\left\|\{(s_g,t)|Pr(x_{[g,t,P]} = 1) > \delta\}\right\|}{\|\mathbb{S}^* \times \mathbb{T}^*\|} \tag{3.21}$$

$$\text{TIP}_\delta = \frac{\left\|\{x_{[g,t,P]} = 1|(s_g,t) \subset \{(s_g,t)|Pr(x_{[g,t,P]} = 1) > \delta\}\}\right\|}{\left\|\{x_{[g,t,P]} = 1\}\right\|} \tag{3.22}$$

where $\{Pr(x_{[g,t,P]} = 1)|s_g \in \mathbb{S}^*, t_i \in \mathbb{T}^*\}$ are predictions from the model; $(s_g, t)$ refers to the spatio-temporal grid at location $s_g$ and time $t$; $\| \cdot \|$ is the size of a set; and $\delta$ is a threshold ($\delta \in [0, 1]$). HRP$_\delta$ represents the percentage of high risk area predicted by the model given $\delta$. TIP$_\delta$ represents the percentage of incidents (from test set) that happened within the high risk area given $\delta$.

Two vectors of HRP and TIP can be computed with different thresholds $\{\delta_i|\delta_i \in [0, 1]\}$. Then, TIP is plotted against HRP. The resulting plot looks like the receiver operating characteristic (ROC) curve [33]. Ideally, we hope as many as incidents happen within the high risk area predicted from the model with a given size. Therefore, the curve from a good model should be close to the upper left corner. An example of this plot is shown in Figure 3.6. This HRP-TIP plot is also helpful for users to pick threshold values for the decision functions. With a HRP-TIP plot, users can pick a HRP value and know how many real incidents can be covered by focusing on the high risk areas. After having the HRP value, the threshold is the corresponding $\delta$.

Similar to ROC analysis, we can use the area under the curve (AUC) to compare the performance of different models by a single score. Because a good model has the curve close to the upper left corner, AUC of a good model should be close to 1. It is easy to see that a random guess model has the curve along the diagonal. Thus, it has AUC$= 0.5$. As a result, AUC of a bad model should be close to or less than 0.5.

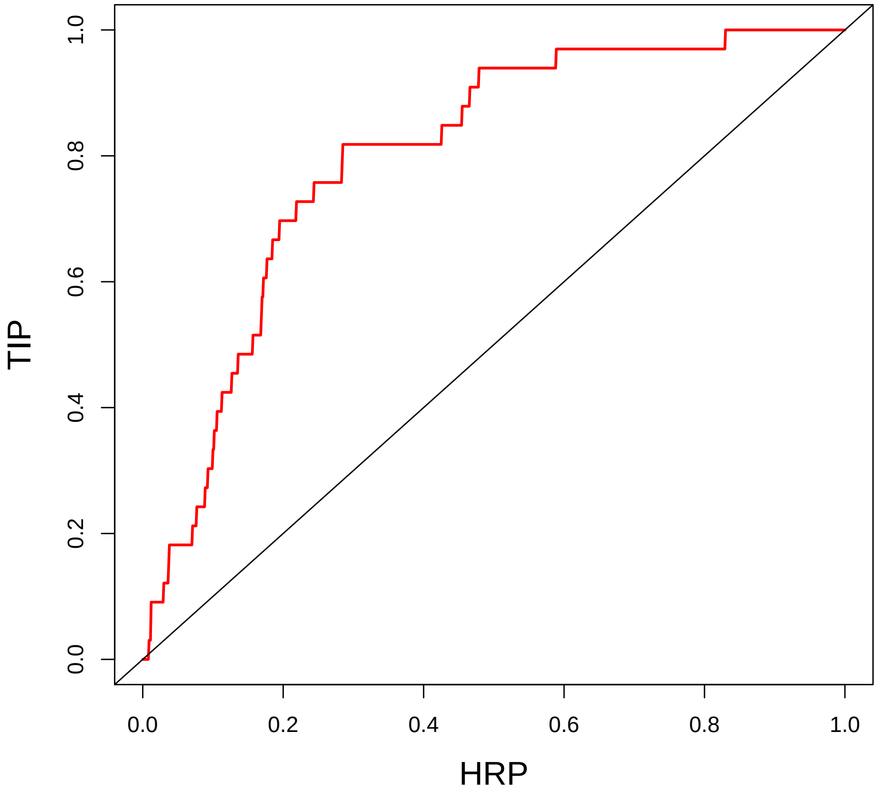


Figure 3.6: An Example of HRP-TIP Plot

# CHAPTER 4

# THE SEMANTIC ROLE LABELING-BASED LATENT DIRICHLET ALLOCATION MODEL

This chapter describes a new text mining model, the semantic role labeling-based latent Dirichlet allocation model (SRL-LDA), to extract information from unstructured text. This method improves a well-performed language model, latent Dirichlet allocation model, by utilizing semantic analysis from semantic role labeling systems. SRL-LDA can structure high dimensional text features into numerical vectors with a few dimensions. Structured textual features can then be incorporated into LSTGAM to model the spatio-temporal data. SRL-LDA can also be applied independently for other types of text mining tasks.

## 4.1 Incorporating Unstructured Textual Features into LSTGAM

LSTGAM developed in Chapter 3 can incorporate various features as long as they can be represented numerically as a vector or matrix. For example, suppose we have the textual feature $D_{g,t} \in \mathbb{X}_{txt}$ associated with location $s_g$ and time $t$. If we can represent $D_{g,t}$ using a numerical vector $X_{g,t}^{txt} = (x_{[g,t,1]}^{txt}, \cdots, x_{[g,t,m]}^{txt})$, we can add $x_{[g,t,1]}^{txt}, \cdots, x_{[g,t,m]}^{txt}$ into Equation 3.6. Doing so adds the textual information to the overall LSTGAM model.

Extracting information from text and structuring textual data as numerical vectors are fundamental text mining tasks. As discussed in Section 2.3.2, the most widely applied method to structure text is the vector space model, where documents[1] are represented by term-document matrices. Weighting schemes such as term frequency-inverse document frequency (TF-IDF) are often used in conjunction with vector space models [85]. Applying this method directly to structure text for LSTGAM has the following problems:

First, this type of model represents a collection of documents[2] within a high-dimensional feature space. For example, a corpus usually has more than 5000 different words. The standard TF-IDF method requires a vector with more than 5000 elements to represent a single document. This means we need to include more than 5000 numerical features in the LSTGAM. In most cases, it is extremely hard to estimate parameters with so many features.

Second, because of polysemy, using a single term[3] as a feature might not be able to represent the meaning of documents correctly. Polysemy means the same word might have different meaning in different sentences (e.g. "bank" in Example 2.25 and 2.26). For example, if we use the TF-IDF method to structure text, the two sentences *"I went to the bank to cash the check."* and *"I walked along the river bank. "* in Example 2.25 and 2.26 will be considered similar because both of them include the word "*bank*".

Third, synonymy might cause the multicollinearity problem for regression models if a single term is used as a feature. Synonymy means different words might have the similar meaning (e.g. "car" and "vehicle"). It may cause the multicollinearity problem for a regression model. For example, both words "gun" and "weapon" are likely to appear together in the same criminal reports. In this case, the two vectors

[1] This section uses "document" to represent a piece of text. It can be a sentence, a paragraph, or an article.

[2] In text mining, a collection of documents is usually called a corpus.

[3] A term can be a word or a phrase.

which represent these two words will have high linear correlation. This high linear correlation will result in unstable estimation of the regression models.

Recent text mining research has developed probabilistic topic models such as latent Dirichlet allocation (LDA) [10], which can partly solve the above problems. LDA performs well for different tasks like document tagging, summarization, and image labeling [7, 95]. LDA can be used to structure text by representing each document as a vector of $K$ probabilities, which describe the likelihoods of the document belonging to the $K$ different topics. This dissertation develops a new text model to extract textual information based on LDA. The following section describes this new model.

## 4.2 The Semantic Role Labeling-Based Latent Dirichlet Allocation Model (SRL-LDA)

As indicated by the name, the new semantic role labeling-based latent Dirichlet allocation model (SRL-LDA) is based on two separate models, the latent Dirichlet allocation (LDA) model and the semantic role labeling (SRL) model. This section first reviews the two previous models and discusses the reason why those two models should be combined. Then, SRL-LDA is described in details.

### 4.2.1 Review of the Latent Dirichlet Allocation Model (LDA)

*Model Description*

The standard LDA model can be represented graphically as shown in Figure 4.1. It is a hierarchical Bayesian model that extracts latent variables from a collection of documents.

LDA assumes a document $d$ is a bag of words, which means relationships between words are not considered. Each document can be represented by a set $d = \{w_{d,1}, \cdots, w_{d,n}\}$, where $w_{d,n}$ is the $n^{\text{th}}$ word in the document $d$. In total, there are $N_d$ different words in the document $d$. In addition, LDA assumes there are $T$ different

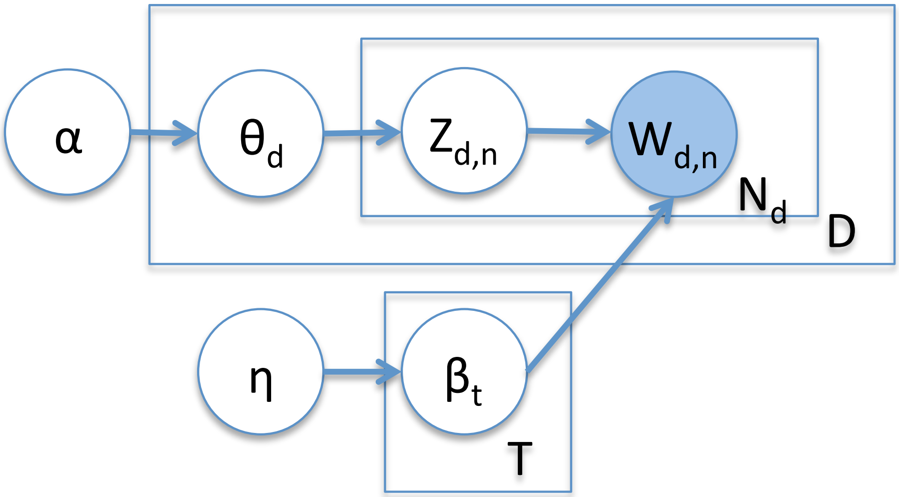


Figure 4.1: LDA Model

topics $\{\beta_t\}$ in the corpus $D = \{d\}$. A topic $\beta_t$ is a distribution over $V$ words, where $V$ is the total number of different words in the corpus. Each word $w_{d,n}$ is about the $Z_{d,n}^{\text{th}}$ topic. $Z_{d,n}$ is determined by the multinominal distribution $\theta_d$. In LDA, both $\theta$ and $\beta$ are assumed to have Dirichlet priors. LDA can be described by the following generative process:

1. Draw $T$ topics from a Dirichlet distribution $\beta_t \sim Dir_V(\eta)$;
2. For each document $d$ in the corpus $D$,
   (a) Draw topic proportions from another Dirichlet distribution: $\theta_d \sim Dir_T(\alpha)$;
   (b) For each word $w_{d,n}$ in the document $d$,
      i. Draw a topic $z_{d,n}|\theta_d \sim Multinomial(\theta_d)$;
      ii. Draw a word $w_{d,n}|z_{d,n}, \beta_{1:T} \sim Multinomial(\beta_{z_{d,n}})$.

Mathematically, the above generative process defines the probability of the corpus

$D$ being observed:

$$Pr(D) = Pr(\mathbf{w}) = \prod_{w_i \in \mathbf{w}} \sum_{t=1}^{T} Pr(w_i|z_i = t) \cdot Pr(z_i = t) \tag{4.1}$$

where $\mathbf{w} = \{w_{d,\cdot}|d \in D\}$ is a sequence including all the words in the corpus $D$; $Pr(w_i|z_i = t) = \beta_t^{w_i}$ means the probability of the word $w_i$ under the $t^{\text{th}}$ topic; and $Pr(z_i = t) = \theta_d^t(w_i \in d)$ is the probability that the document $d$ is about the $t^{\text{th}}$ topic.

In the above model, $\alpha$, $\eta$, and $T$ are hyper-parameters. The topic number $T$ depends on the specific problems. It is can also be estimated by the hierarchal Dirichlet process model [96]. It is usually chosen to be from 10 to 100. $\alpha$ and $\eta$ decide the sparseness of topic distributions. As suggested in [94], a good choice of these two parameters is $\alpha = \frac{50}{T}$ and $\eta = 0.01$. In practice, we can try different values of those hyper-parameters and select the best values by cross-validation.

In addition to the hyper-parameters, we need to estimate $\beta$ and $\theta$. Both of these two parameters are interesting to us. $\beta$ is the topic distribution describing what each document is about. It can be used to represent the document numerically. $\theta$ is the word distribution describing what words each topic is related to. It can be used to explain the meaning of the numerical representation.

*Estimation of LDA*

To estimate the parameters of interest $\beta$ and $\theta$, we can use the maximum likelihood estimation:

$$(\hat{\beta}, \hat{\theta}) = \arg\max Pr(D|\beta, \theta) \tag{4.2}$$

However, the above optimization is intractable [10, 43]. The following two methods have been developed and applied widely to estimate $\beta$ and $\theta$ approximately.

The first method is the variational expectation maximization (EM) algorithm [10].

Instead of directly maximizing the likelihood function, the variational EM algorithm attempts to maximize a lower bound of the log likelihood function. The E-step estimates the parameters using the current estimation, and the M-step updates the model parameters by maximizing a lower bound of the log likelihood. The above two steps are repeated until convergence. It is well known that the EM algorithm can converge [26]. Therefore, the above estimation algorithm can also converge. However, a problem with the variational EM algorithm is that it is likely to find local optima and lead to inaccurate estimates [69].

The second method is the Gibbs sampling (GS) algorithm [43]. This method is widely applied to estimate LDA, because it is straightforward and converges fast in practice. Instead of estimating $\beta$ and $\theta$ directly, this method estimates the posterior distribution of the assignments of words to topics, $Pr(\mathbf{z}|\mathbf{w})$, by GS. To use GS, we need to know the conditional distribution $Pr(z_i|\mathbf{z}_{-i}, \mathbf{w})$, which can be calculated by the following equation [43, 94]:

$$Pr(z_i = j|z_{-i}, w_i, d_i, \cdot) \propto \frac{C^{WT}_{w,j} + \eta}{\sum_{w=1}^{W} C^{WT}_{w,j} + W\eta} \cdot \frac{C^{DT}_{d,j} + \alpha}{\sum_{t=1}^{T} C^{DT}_{d,t} + T\alpha} \tag{4.3}$$

where $C^{WT}$ and $C^{DT}$ are matrices of counts; $C^{WT}_{wj}$ is the number of times word $w$ is assigned to topic $j$ (not including the current topic $i$); $C^{DT}_{dj}$ is the number of times topic $j$ is assigned to some word in document $d$ (not including the current topic $i$); and $W = \sum_{d \in D} N_d$ is the total number of words in the corpus.

After estimating $Pr(\mathbf{z}|\mathbf{w})$, $\beta$ and $\theta$ can be estimated by integrating across the entire corpus:

$$\hat{\beta}_j^w = \frac{C_{w,j}^{WT} + \eta}{\sum_{w=1}^{W} C_{w,j}^{WT} + W\eta} \tag{4.4}$$

$$\hat{\theta}_d^j = \frac{C_{d,j}^{DT} + \alpha}{\sum_{t=1}^{T} C_{d,t}^{DT} + T\alpha} \tag{4.5}$$

where $\hat{\beta}_j^w$ is the estimated probability of word $w$ under the $j^{\text{th}}$ topic; and $\hat{\theta}_d^j$ is the estimated probability that document $d$ is about the $j^{\text{th}}$ topic. As we can see, $\hat{\beta}_j^w$ is the first part of the right hand side of Equation 4.3, while $\hat{\theta}_d^j$ is the second part. $\hat{\beta}_j^w$ is related to how likely word $w$ is for topic $j$. $\hat{\theta}_d^j$ is related to how dominant topic $j$ is in document $d$.

As studied in [20], the Gibbs sampling can generate a Markov chain of random variables that converges to the distribution of interest after some large number of iterations. The empirical study showed GS converged relatively fast for LDA estimation, within 500 iterations of sampling [43]. In each GS iteration, we need to generate a topic assignment for each word from a conditional distribution over $T$ topics. Therefore, it requires $O(WT)$ computations for a single iteration, where $W$ is the total number of words in the corpus; and $T$ is the number of topics. The overall computation complexity of the GS algorithm with $K_{gs}$ iterations is $O(K_{gs}WT)$. In practice, $K_{gs}$ is usually fixed to a small number (compared to $W$). So the complexity of GS for LDA estimation is $O(WT)$.

Different methods have been developed to reduce the computation time required by the GS algorithm, such as FastLDA [78], approximate distributed LDA algorithm [71], and parallel LDA [105]. There are also many implements to estimate LDA models, such as GibbsLDA++ [114] and “topicmodels” [44] and “lda” [21] packages in R[4].

### 4.2.2 Review of Semantic Role Labeling (SRL)

Because humans utilize semantic roles of words in sentences in the understanding of languages, the text models like LDA which treat words as independent tokens are inadequate to fully extract information from text. NLP researchers have been developing automatic methods to analyze text semantically. Semantic role labeling (SRL) is such a type of NLP techniques. SRL automatically extracts the verbal and nominal predicates (usually called events) mentioned a sentence, the entities associated with the events, and the roles of the entities with respect to the events [39, 79]. Mathematically, for a sentence $sentence = \{w_1, \cdots, w_n\}$, SRL associates a label with each word: $SRL(sentence) = \{(w_1, sl_1), \cdots, (w_i, sl_i), \cdots, (w_n, sl_n)\}$, where $sl_i \in SL$ is either an event or a semantic role for word $w_i$; and $SL = \{sl.\}$ is a set of events and all possible semantic roles. An example of the output from SRL is shown in Example 2.30.

Modern SRL systems use statistical learning models to label words. The first step of these systems is to analyze the syntactic structures of sentences. The syntactic structures are usually represented by syntactic parse trees. Figure 4.2 shows the syntactic parse tree of the sentence "Casey throws the ball."[5]. In the figure, blue square nodes are phrase types. 'S' stands for "sentence", the top-level of the sentence structure. "NP" stands for "noun phrase"; and "VP" stands for "verb phrase", which serves as the predicate in this example. A popular method to generate syntactic parse trees from sentences is by probabilistic context free grammars (PCFG) [64]. A PCFG $G$ defines a set of nodes (e.g. words and phrase types), a set of rules (e.g. VP → V NP,

[4] Additional implements can be found at `http://www.cs.princeton.edu/~blei/topicmodeling.html`

[5] This example is from Penn Treebank [6].

which means a verb plus a noun phrase is a verb phrase.), and a corresponding set of probabilities (e.g. $Pr(\text{VP} \rightarrow \text{V NP}) = 0.7$). Given $G$, a sentence is parsed such that the probability of the corresponding parse tree being observed is maximum. This solution can be generated by a popular dynamic programming algorithm, Cocke-Younger-Kasami (CYK) algorithm. Its running time is $O(n^3)$, where $n$ is the length of the sentence.

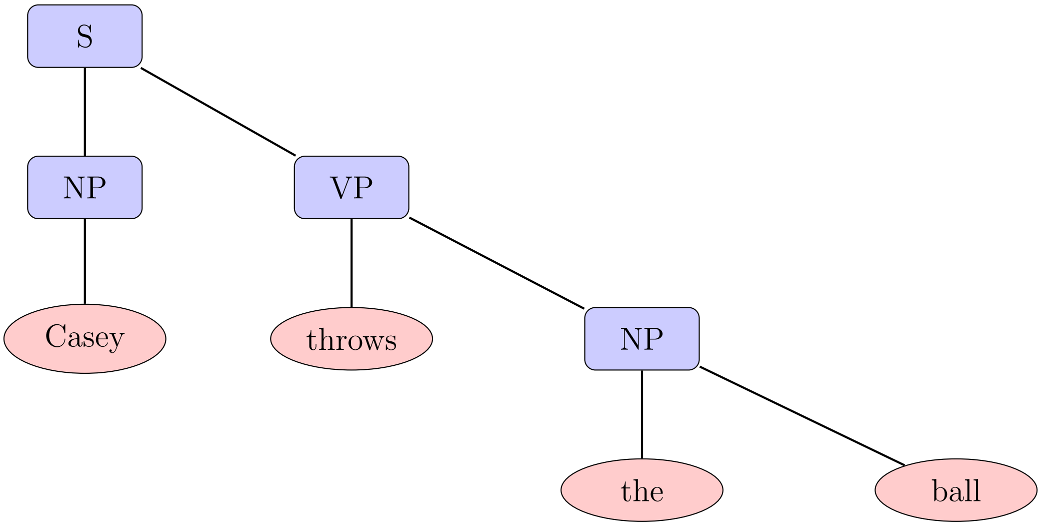


Figure 4.2: An Example of Syntactic Parse Tree

After getting syntactic representation of sentences, SRL classifies nodes in the parse trees into labels in $SL$. This is a classic supervised learning problem as discussed in Section 2.1.1. For example, a recent SRL system developed by Gerber [39] used a feature-based GLM to classify the nodes. The features used in his model included positions of nodes, head words, paths between two words, syntactic categories and so on. As we can see, this method well considered the orders between words and other syntactic rules. To estimate the parameters of the classification models, manually annotated corpora like Penn Treebank [65] and NomBank [68] can be used.

### 4.2.3 The Semantic Role Labeling-Based Latent Dirichlet Allocation Model (SRL-LDA)

The generative LDA model regards each document as a bag of words that contains no information beyond the words themselves. The SRL method provides such semantic information, but does not provide a directly numerical representation of unstructured textual data. This dissertation proposes to use SRL to incorporate additional semantic information into the LDA model. In this way, SRL can extract useful words, such as events, mentioned in a document. LDA then can extract numerical latent variables about only the important words in the documents. Such a semantic role labeling-based latent Dirichlet allocation (SRL-LDA) model can be represented graphically as shown in Figure 4.3.

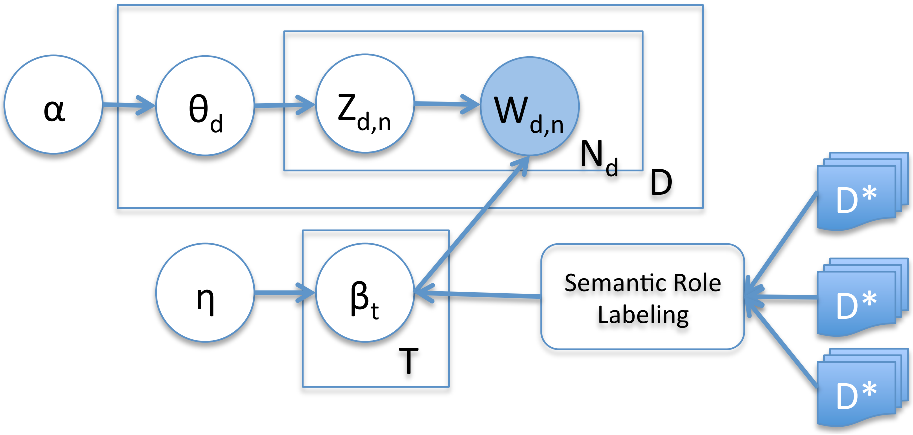


Figure 4.3: SRL-LDA Model

In Figure 4.3, the LDA part is defined the same as in Section 4.2.1, except for $\beta_t$. In the SRL-LDA model, the topic distribution $\beta_t$ is defined in terms of both the words in the corpus $D$ and the labels of the words from SRL. Only the words with certain labels in $SL^* \subset SL$[6] are assumed to be related to $\beta_t$. In other words, $\beta$ is no

[6] For example, we know the predicates have the most important information about sentences. In this case, $SL^*$ includes both nominal and verbal predicates.

longer defined on the entire set of words $\mathbf{w}$[7] in the corpus. It is defined on a subset $\mathbf{w}^* = \{w_{i,d} | sl_{i,d} \in SL^*, w_{i,d} \in \mathbf{w}\}$, where $sl_{i,d}$ is the label assigned to word $w_{i,d}$ by SRL. The block "Semantic Role Labeling" means any available SRL systems[8] . $D^*$'s are corpora used to build SRL systems. $D^*$ is usually different from $D$ and has larger size than $D$.

Mathematically, SRL-LDA can be formulated as follows:

$$Pr(D) = Pr(\mathbf{w}) = \prod_{w_i \in \mathbf{w}^*} \sum_{t=1}^{T} Pr(w_i | z_i = t) \cdot Pr(z_i = t) \cdot \prod_{w_i \in \mathbf{w} - \mathbf{w}^*} Pr(w_i | SRL) \quad (4.6)$$

where $\mathbf{w}^*$ is a sequence including all the important words in the corpus $D$ determined by SRL; $Pr(w_i|z_i = t) = \beta_t^{w_i}$ means the probability of the important word $w_i$ under the $t^{\text{th}}$ topic; $Pr(z_i = t) = \theta_d^t(w_i \in d)$ is the probability that document $d$ is about the $t^{\text{th}}$ topic; $\mathbf{w} - \mathbf{w}^*$ is a set of words not included in $\mathbf{w}^*$; and $Pr(w_i|SRL)$ means these words not in the set $\mathbf{w}^*$ are decided by syntactic rules and not related to topics. Similar to LDA, we are interested in the estimates of $\beta$ and $\theta$.

SRL-LDA assumes a corpus $D$ is generated as follows:

1. Draw $T$ topics from a Dirichlet distribution $\beta_t \sim Dir_V(\eta)$; topics are only related to a small number of important words;

2. For each document $d$ in the corpus $D$,

   (a) Draw topic proportions from another Dirichlet distribution: $\theta_d \sim Dir_T(\alpha)$;

   (b) Draw topic-related words $\{w_{d,i}^*\}$:

      i. Draw a topic $z_{d,i} | \theta_d \sim Multinomial(\theta_d)$;

      ii. Draw a word $w_{d,n}^* | z_{d,n}, \beta_{1:T} \sim Multinomial(\beta_{z_{d,n}})$.

[7] $\mathbf{w}$ is defined as a sequence including all the words in the corpus $D$.

[8] This dissertation uses the SRL systems developed by Punyakanok et al. [79] and Gerber and Chai [41].

(c) Given $\{w^*_{d,i}\}$, fill other words in the document according to syntactic rules.

SRL-LDA has the following desired properties to be applied to structure textual data for the LSTGAM model:

1. Compared to vector space models, SRL-LDA can represent unstructured textual data with meaningful vectors with lower dimensions. Fundamentally, SRL-LDA is an LDA model. It has all the good properties of LDA. SRL-LDA extracts topic distributions as the numerical representation of the documents. The number of topics is usually much less than the number of words. Therefore, SRL-LDA requires much less dimensions than vector space models.

2. SRL-LDA can partly solve the synonymy and polysemy problems by representing documents with topic distributions. This is possible because of the following two properties of LDA: synonyms which frequently appear together can be assigned to the same topic by LDA; and because a topic is determined by more than one words, a word can appear in different topics.

3. By incorporating SRL, SRL-LDA can utilize semantic information of words in documents. Because SRL well considers the orders between words and syntax, SRL-LDA can extract information from text with the consideration of semantics and syntax.

4. Compared to LDA, SRL-LDA provides more meaningful topics. LDA learns topics from a corpus using all the words. As we know, not all of the words can be assigned to topics. For example, the article words, like “the”, “an”, “a” and so on, can appear in a document with any topic. Even content words, like “car”, “ball” and so on, might not be important for topic discovery in some cases. By focusing on only important words labeled from SRL, SRL-LDA can provide more interpretable topics.

5. SRL-LDA is more suitable to be applied to the problems with a limited number of documents. LDA learns the topics with only the documents in the corpus $D$. When the number of documents in $D$ is small, LDA can hardly discover meaningful topics. This is because LDA learns topics by computing the occurrences of a word in a specific topic and the co-occurrences of words in a document[9]. When the number of documents is small, occurrences and co-occurrences are more likely to be random. The performance of LDA is limited by the existence of unimportant words. On the other hand, SRL-LDA uses the knowledge learned from the large corpora $D^*$'s to reduce the number of words to be considered. This can improve the performance of LDA.

6. As will be discussed in the later of this section, SRL-LDA can be estimated easily and efficiently with current algorithms. With the development of SRL and LDA, the performance of SRL-LDA can also be improved.

## 4.3 Estimation of SRL-LDA

For SRL-LDA, we are interested in two parameters $\beta$ and $\theta$. $\theta_d$ is the numerical representation of the document $d$, describing what topics $d$ is about. $\beta$ are useful to interpret the meaning of topics.

To estimate these two parameters, we first need to decide $\mathbf{w}^*$. Because we have the knowledge of what types of semantic labels are important, this step is straightforward: apply SRL to the corpus, define important label set $SL^*$ by experts, and $\mathbf{w}^*$ is then decided.

With $\mathbf{w}^*$, SRL-LDA can be easily estimated by available LDA algorithms. Similar

[9] This can be seen by examining Equation 4.3.

to LDA, we use maximum likelihood estimation by solving:

$$(\hat{\theta}, \hat{\beta}) = \arg\max Pr(\mathbf{w}|\theta, \beta) \tag{4.7}$$

As shown in Equation 4.6, $\theta$ and $\beta$ are only related to the first part of the right hand side. So, the problem in the above Equation 4.7 is equivalent to:

$$(\hat{\theta}, \hat{\beta}) = \arg\max Pr(\mathbf{w}^*|\theta, \beta) \tag{4.8}$$

This is the LDA problem defined on $\mathbf{w}^*$. Therefore, we can use Algorithm 5 to estimate SRL-LDA by using available SRL systems and LDA estimation methods.

**Algorithm 5** Estimation of SRL-LDA

```
1: Apply SRL to corpus $D$;
2: Keep only the words with labels $sl \in SL^*$; call the filtered corpus $D_{SRL}$;
3: Apply the standard LDA estimation algorithm (e.g. Gibbs sampling algorithm)
   to $D_{SRL}$ to estimate $\theta$ and $\beta$.
```

The complexity of the above algorithm depends on the choice of SRL and LDA methods. This dissertation uses the SRL systems by Punyakanok et al. [79] and Gerber and Chai [41] as well as the Gibbs sampling method for LDA estimation. The complexity of SRL is $O(n_s^2 W)$, where $n_s$ is the average of sentence length; and $W$ is the total number of words in the corpus. The complexity of LDA estimation is $O(WT)$, where $T$ is the number of topics. Therefore, the complexity of Algorithm 5 is $O(n_s^2 W + WT)$.

# CHAPTER 5

# MODEL EVALUATION AND APPLICATIONS

This chapter shows the applications of the models developed in Chapter 3 and 4 to four real problems. The first application is the modeling of the spatio-temporal patterns of breaking and entering incidents with numerical features. The second application is the prediction of hit and run incidents with information extracted from Twitter posts. The third application is the modeling of the spatio-temporal patterns of criminal incidents with both numerical and unstructured textual features. The last application is the prediction of equipment damages of train accidents using both spatio-temporal and textual features. In addition, each application evaluates the performance of the models developed in this dissertation and compares them with several previous models.

## 5.1 The Spatio-Temporal Modeling for Breaking and Entering Incidents[1]

### 5.1.1 Introduction

As discussed in Chapter 1, law enforcement agencies have the need to study the spatio-temporal patterns of criminal incidents. A good mathematical model of criminal incidents can help them in at least two ways. First, they can find out the factors which affect crimes and study the causality of crimes with those factors. Second, they can predict the locations and time of future criminal activity. If the model can pre-

[1] Most of the material in this section has been published in previous papers [100, 101].

dict future crimes accurately, law enforcement can deploy limited resources, such as walking and driving patrols, surveillance systems, and neighborhood watch programs, to improve security and reduce threats.

Many types of data are available to assist building such models. Law enforcement agencies in the United States usually monitor criminal incidents as they occur. For example, they have locations and times of criminal incidents, as well as victim and perpetrator information. In addition to criminal incident data, most agencies can also acquire spatial information from geographic information systems (GIS) and demographic and economic data from the census.

Several techniques and models have been developed to meet the need for predictive policing with available data. One of the most popular methods is spatial hot spot models. In hot spot models [29], current criminal incident data are collected and clustered over spaces. The locations of such clusters are so-called hot spots. The models assume the current crime clusters to persist over the forecast horizon. Future criminal incidents are predicted to occur in these same areas. Methods to generate hot spots include spatial histograms, clustering, mixture models, scan statistics, and density estimation. Hot spot models only utilize criminal incident data, such as types of crimes, locations and time of criminal incidents. They only show the current patterns of crimes without the insight into the relationship between crimes and environment over time. As the local environment changes, hot spot models cannot indicate changes of crime patterns.

To address this problem, more sophisticated statistical models using both criminal data and environmental data have been built by researchers. Liu and Brown [60] applied a point pattern density model to criminal incidents. The spatial density of criminal incidents was assumed to be conditioned on features associated with locations. These features included geographic features, such as distances to the nearest interstate highways, demographic features and consumer expenditure features. Xue and Brown [115] developed a spatial choice model. They assumed criminals made choices to pick places that could be modeled by random utility maximization. This utility maximization was over all alternatives, where the utility was defined by the gain from crimes and the risk of being caught. Brown and his colleagues [15,92] then discussed a method that used generalized linear models (GLM) to compute the risk over a territory. They first partitioned the space into grids. Each grid was associated with a response indicating whether incidents happened and features about the grid. Then, a spatial GLM was built with all grids. They applied the spatial GLM to predict terrorist events. Results showed the spatial GLM had better prediction performance than the density models. Rodrigues and Diggle [84] combined point process models and GAM to build a semiparametric point source model. In their model, features affected the risk nonlinearly. They applied the model to study the effect of installed security cameras on crimes.

None of the above models directly incorporate the temporal information of criminal incidents. For instance, Liu and Brown used a Bayesian approach to model building that can include a variety of time series but no specifics is recommended or tested. Other models usually estimated different parameter sets based upon coarse divisions of time. For example, these models used criminal incidents that happened within the most recent year to generate hot spots for this year. Another intuitive method discussed by Ivaha, Al-Madfai, Higgs, and Ware [54] first modeled the temporal behaviors of crimes with time series models and then modeled the spatial behaviors given the predicted number of incidents at a certain time. However, this approach did not model interactions between space and time.

In this application, we applied the feature-based spatio-temporal model developed in Chapter 3 to utilize a variety of data types, such as spatial, temporal, geographic, and demographic data, to model criminal incidents.

### 5.1.2 Data Description

We used three data sets for this study. The first data set included breaking and entering incidents in Charlottesville, Virginia from April 2001 to February 2005. In total, there were 1,795 incidents[2]. Each incident in this study had coordinates of the incident and the time of when it happened. The second data set was the geographic information of the city in the form of GIS layers, such as locations of roads, interstate highways, small businesses and schools. The third data set had demographic data of Charlottesville measured in census block groups, including population, median values of all houses, races, marriages and so on. Figure 5.1 shows a small number of geographic features of Charlottesville with all the breaking and entering incidents.

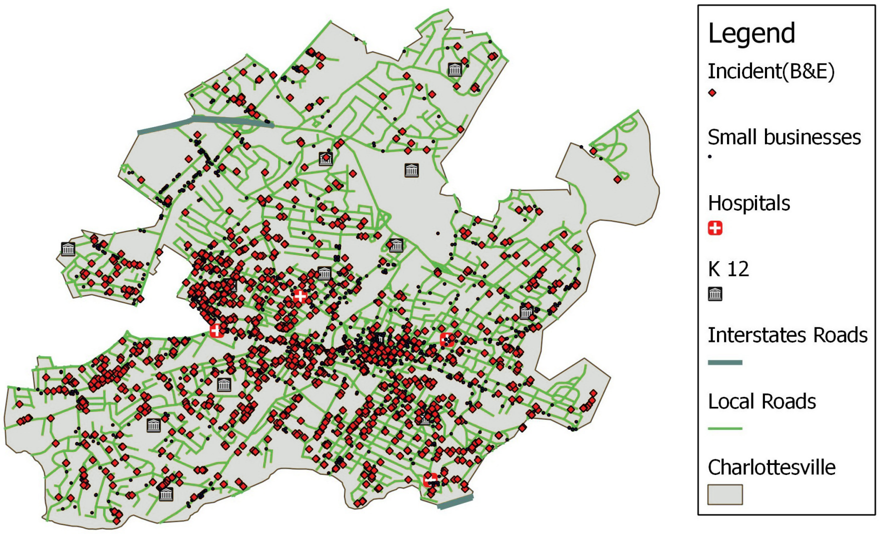
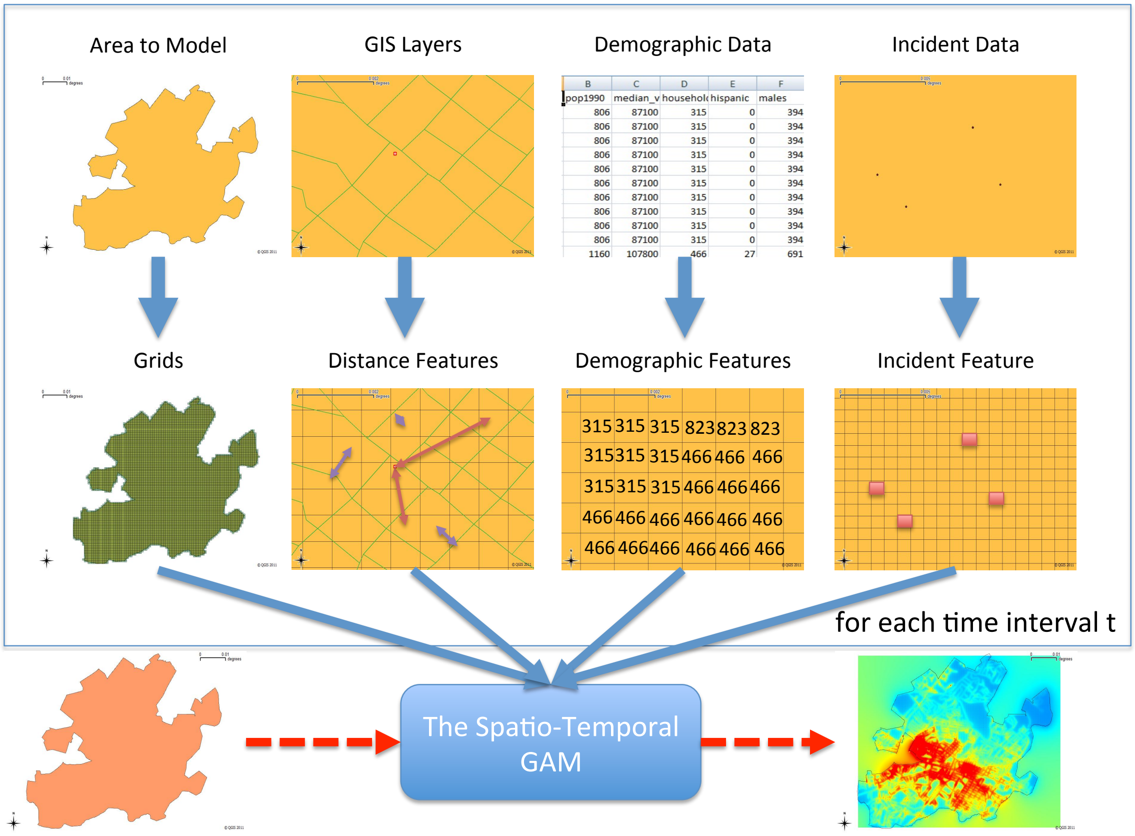


Figure 5.1: Criminal Incidents in Charlottesville, Virginia

[2] 58 incidents without exact coordinates were excluded.

### 5.1.3 Model Construction and Estimation

The overall modeling process is shown in Figure 5.2. In the figure, blue solid lines represent the steps to build the model, while red dotted lines represent the steps to predict when the model was built.



| B | C | D | E | F |
|---|---|---|---|---|
| pop1990 | median_v | househol | hispanic | males |
| 806 | 87100 | 315 | 0 | 394 |
| 806 | 87100 | 315 | 0 | 394 |
| 806 | 87100 | 315 | 0 | 394 |
| 806 | 87100 | 315 | 0 | 394 |
| 806 | 87100 | 315 | 0 | 394 |
| 806 | 87100 | 315 | 0 | 394 |
| 806 | 87100 | 315 | 0 | 394 |
| 806 | 87100 | 315 | 0 | 394 |
| 806 | 87100 | 315 | 0 | 394 |
| 806 | 87100 | 315 | 0 | 394 |
| 1160 | 107800 | 466 | 27 | 691 |

| | | | | | |
|---|---|---|---|---|---|
| 315 | 315 | 315 | 823 | 823 | 823 |
| 315 | 315 | 315 | 466 | 466 | 466 |
| 315 | 315 | 315 | 466 | 466 | 466 |
| 466 | 466 | 466 | 466 | 466 | 466 |
| 466 | 466 | 466 | 466 | 466 | 466 |

Figure 5.2: Overall Process of the Spatio-Temporal Modeling of Criminal Incidents in Charlottesville, Virginia

To model criminal incidents in Charlottesville, we first partitioned the city into spatial grids with the size of 32m × 32m. The total number of grids covering the area was 23,089. We used the time interval with the length of one month and there were 46 months in the data set. Therefore, we had 1,062,094 ($= 23,089 \times 46$) records. Each record had a response variable indicating whether at least one incident happened within the grid and the time period. There were also two types of features associated with each record as explanatory variables. The first type was the distance feature. We calculated the shortest distance between the centroid of a grid and a certain geographic landmark, such as the distance to the nearest road. This calculation was done by a toolkit programmed in Visual C#[3] and PostGIS[4]. The second type was the demographic feature, such as population, marriage status, house values. Because we used the demographic data measured in census block groups, the demographic features of a grid actually measured the properties of the neighborhood where the grid was located. There were 14 distance features and 20 demographic features. For this study, we only kept the most important 11 features out of 34 features. These 11 features were selected by the stepwise selection of GLM. Table 5.1 shows the description of the features for modeling.

To test our models, we kept the incident data that happened in the last 12 months as the test data. Thus, the training data set included incidents between April 2001 and February 2004. The test data set included incidents between March 2004 and February 2005.

We first built the following STGAM:

$$\text{logit}\left[Pr(inci_{g,t} = 1)\right] = \sum_{n=1}^{N} f_n(x_{[g,t,n]}) + \kappa_{g,t} \tag{5.1}$$

and LSTGAM:

$$Pr(inci_{g,t} = 1) = \sum_{r=1}^{2} I(s_g \in \mathbb{S}_r) \cdot Pr_r(inci_{g,t} = 1) \tag{5.2}$$

$$\text{logit}[Pr_r(inci_{g,t} = 1)] = \sum_{n=1}^{N} f_{r,n}(x_{[g,t,n]}) + \kappa_{[r,g,t]}, \text{for all } r \in \{1, 2\} \tag{5.3}$$

where $\text{logit}(p) = \log(\frac{p}{1-p})$ is a logit function; $inci_{g,t} = 1$ means at least one

[3] `http://www.microsoft.com/visualstudio/en-us`

[4] `http://www.postgis.org`

Table 5.1: Features Used for the Spatio-Temporal Modeling of Breaking and Entering Incidents

| Feature | Type | Description | Significance* |
|---|---|---|---|
| $\kappa$ | temporal | the dummy variable indicating the time of the previous incident | 2,3,4 |
| college_univ_dist | distance | the distance to the nearest college or university | 1,2,3 |
| k_12_dist | distance | the distance to the nearest K-12 shool | 1,2,4 |
| roads_all_dist | distance | the distance to the nearest road | 1,2,3,4 |
| roads_interstates_dist | distance | the distance to the nearest interstate highway | 2,3 |
| small_businesses_dist | distance | the distance to the nearest small business | 1,2,3,4 |
| median_val | demographic | median value of all housing unites | 1,2,3 |
| males | demographic | number of males | 1,2,3 |
| widowed | demographic | number of people whose spouse died | |
| divorced | demographic | number of people who are divorced | 1,2,3,4 |
| owner_occ | demographic | count of owner-occupied households | 1,2,4 |
| medianrent | demographic | median rent charged for all housing units that are rented | 1 |

* The significance column describes whether a feature is significant (at the level of 95%) in the following models: 1-Spatial GLM, 2-STGAM, 3-LSTGAM (in region $\mathbb{S}_1$), and 4-LSTGAM (in region $\mathbb{S}_2$).

incident happens at grid $s_g$ and time $t$; $N$ is the total number of features; $x_{[g,t,n]}$ is the $n^{\text{th}}$ feature associated with location $s_g$ and time $t$; $f_n$ is the smooth function of the $n^{\text{th}}$ feature to be estimated from data; $\kappa_{g,t} = \{1, \cdots, K\}$ is the dummy variable indicating the length of the continuous zeros (no accident happens) that precede the current observation at location $s_g$ and time $t$; $I$ is an indicator function; $Pr_r$ models the incident probability in the $r^{\text{th}}$ region; and $f_{r,n}$ and $\kappa_{[r,g,t]}$ are the same as defined above, but different for the different regions.

To build the above STGAM and LSTGAM, we chose the parameter $K = 13$, which means incidents happened before one year would not be considered. To build the LSTGAM, we defined two regions, $\mathbb{S}_1$ and $\mathbb{S}_2$, using Algorithm 4 discussed in Section 3.4.2. The high risk region $\mathbb{S}_2$ included 10% of the area with the highest incident density. The low risk region $\mathbb{S}_1$ included the other 90% of the area. We used the package “mgcv” in R [112] to estimate the smooth functions and parameters in STGAM and LSTGAM. To avoid stochastic effects from subsampling, we used all the training data to estimate models.

To compare the STGAM and LSTGAM models with the previous work, we also built a spatial GLM and a hot spot model. The spatial GLM used the same features in Table 5.1 and parameters were estimated with all the training data. The hot spot model estimated the density with all the incidents in the training data set using Gaussian kernels. Both models were estimated by the software R [81].

### 5.1.4 Results

*Prediction Performance*

We applied STGAM, LSTGAM, the spatial GLM and the hot spot model to predict the probability of criminal incidents in Charlottesville from March 2004 to February 2005 using the test data set. Then, we compared those four models with the metrics described in Section 3.5.

Figure 5.3 shows the AUC of the twelve-month predictions using the four models. The larger the AUC value is, the better the model predicted. STGAM and LSTGAM performed better than the previous work, the spatial GLM and the hot spot model. The performance of LSTGAM was a little better than the performance of STGAM. To test whether the difference between any two curves in Figure 5.3 was significant, we performed paired Wilcoxon significance tests on the groups of AUC values. Table 5.2 shows the test results. Small p-values mean the differences are significant ($p < 0.05$). Because all of the p-values were less than 0.05, the difference between any two curves was statistically significant at the level of 0.05. For example, the p-value of the test for the difference between LSTGAM and STGAM was 0.02686. Although the two curves for LSTGAM and STGAM are close in Figure 5.3, LSTGAM was still significantly better than STGAM in this evaluation. In addition, we can see both LSTGAM and STGAM were significantly better than the spatial GLM and the hot spot model.

Table 5.2: Wilcoxon Significance Test Results for AUC Comparisons

| Models | Hot Spot Model | Spatial GLM | STGAM | LSTGAM |
|---|---|---|---|---|
| Hot Spot Model | - | 0.0009766 | 0.0004883 | 0.0004883 |
| Spatial GLM | 0.0009766 | - | 0.0004883 | 0.0004883 |
| STGAM | 0.0004883 | 0.0004883 | - | 0.02686 |
| LSTGAM | 0.0004883 | 0.0004883 | 0.02686 | - |

Figure 5.4 shows the HRP-TIP plots for the predictions in March 2004, July 2004, November 2004 and February 2005. In the plots, HRP and TIP are the percentage of high risk area and the percentage of incidents happened within the high risk area respectively, as defined in Section 3.5. From these plots, we can confirm that STGAM and LSTGAM had better prediction performance in these four months. Especially, STGAM and LSTGAM can capture about half of the real incidents that happened in a very small high risk area in each case. For example, about 50% of real incidents

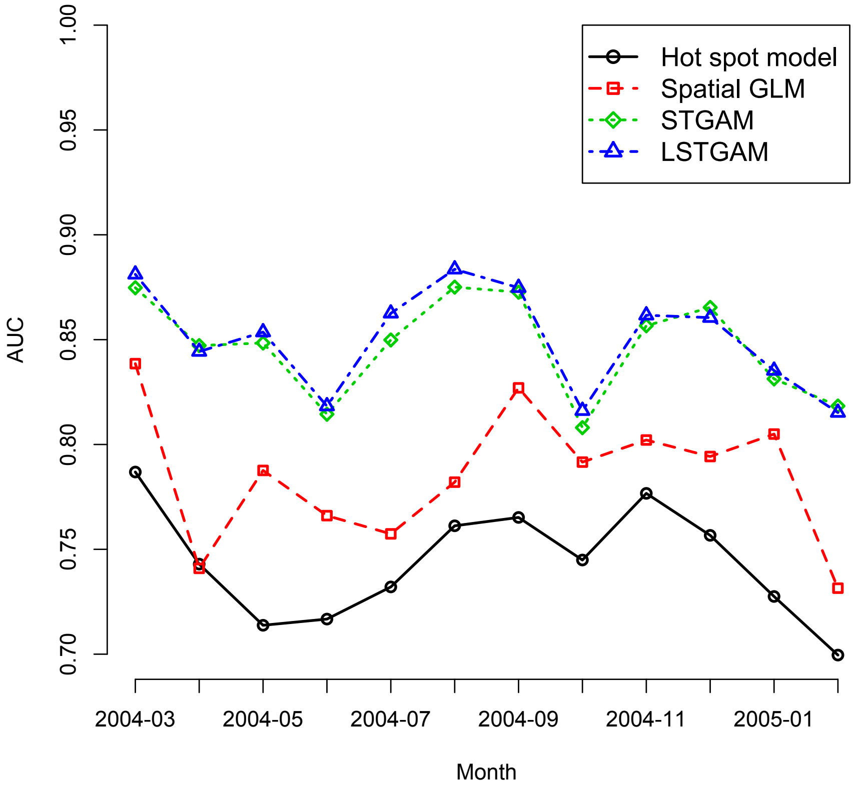
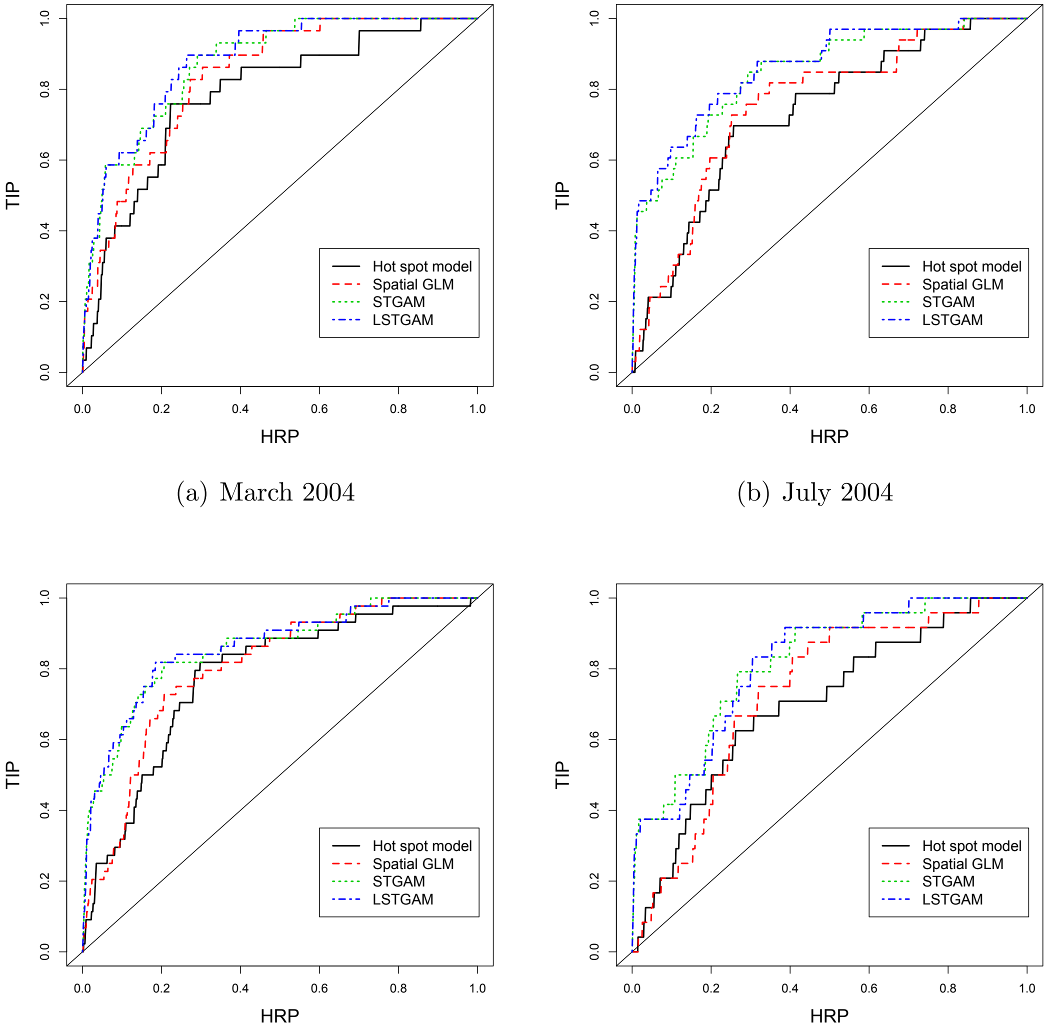


Figure 5.3: AUC from Different Models

happened within the top 2% area with the highest risk predicted from LSTGAM in July 2004.The police department can use this prediction to patrol more efficiently.

Predictions from STGAM and LSTGAM are probabilities on spatio-temporal grids. These type of data can be visualized easily with available GIS softwares. Figure 5.5 shows the heat map of the prediction from STGAM in February 2005 generated by Quantum GIS [80]. We used a kernel density with 3 standard deviation to smooth the prediction. On this map, red color means high predicted probabilities while the light blue color means low predicted probabilities. The red stars are the real criminal incidents happened in February 2005. As we can see from this map, most of the real



(a) March 2004

(b) July 2004

(c) November 2004

(d) February 2005

Figure 5.4: HRP-TIP Plots to Compare Hot Spot Model, Spatial GLM, STGAM and LSTGAM

criminal incidents are located within the predicted high probability area.

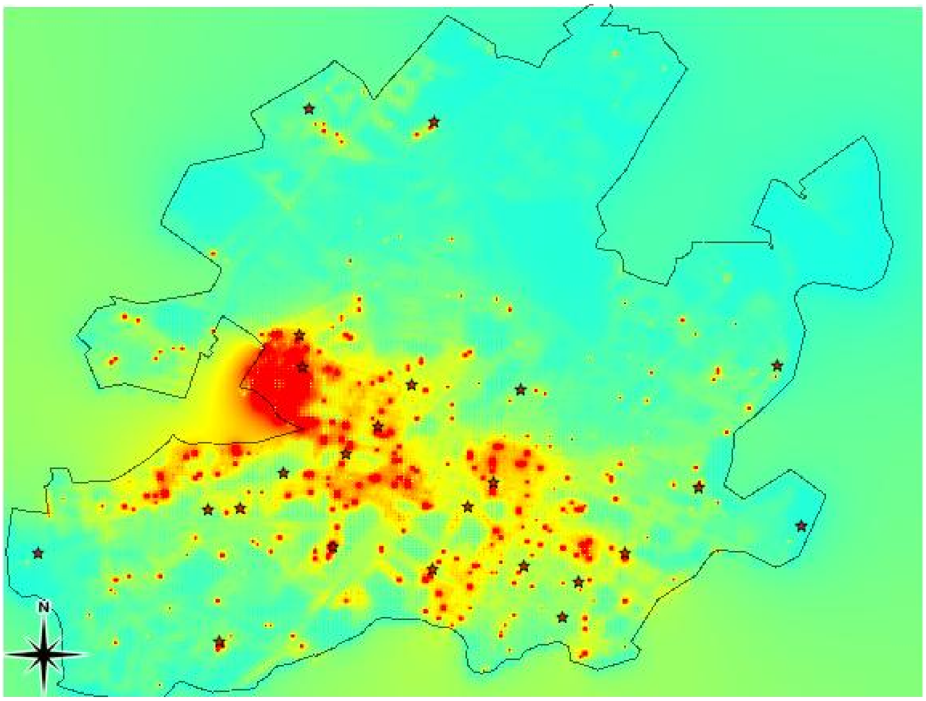


Figure 5.5: Heat Map of the Prediction of Criminal Incidents in 2005-02 by STGAM

*Model Interpretation*

Table 5.1 shows feature significance in different models. As we can see, the temporal dummy variable $\kappa$ was significant in both STGAM and LSTGAM. The significance means if the variable was removed from the models, the performance of these two models would be worse. Therefore, it was helpful to predict the criminal incident probability. All selected features were significant in at least one model, except for the feature *widowed*. Features *roads_all_dist*, *small_businesses*, and *divorced* were significant in all the models. Comparing the features in LSTGAM in $\mathbb{S}_1$ and $\mathbb{S}_2$, we can see the different regions had different sets of significant features. For example, *median_val* was important to predict the probability in the low risk area $\mathbb{S}_1$, but not in the high risk area $\mathbb{S}_2$.

Figure 5.6, 5.7 and 5.8 show the estimated parameters and smooth functions of STGAM, LSTGAM in region $S_1$ and LSTGAM in region $S_2$ respectively. Only significant features were plotted. In the figures, solid lines represent the estimated smooth functions while the dotted lines are 95% confidence intervals. Clearly, we can see the nonlinear effects of features on the crime probability.

Based on Figure 5.6, locations with no incident happened in the previous year were less likely to have a new incident. Out of the locations where incidents happened in the previous year, the locations with incidents just happened in the past half year were more likely to have a new incident. Incidents were more likely to happen at locations closer to schools, roads, and small businesses. The neighborhoods with the least and the most expensive median house values were less likely to have breaking and entering. The neighborhood with the median house value of about $60,000 was the most likely to have such incidents. The number of males in the neighborhood also impacted crimes. For the neighborhoods that had less than 350 males, it was more likely to have incidents in the neighborhoods with less males. For the neighborhoods that had more than 350 males, there was no such effect. In addition, breaking and entering incidents were less likely to happen in the neighborhoods with less divorced people and more owner occupied houses.

Figure 5.7 and 5.8 shows the different patterns in the different regions. In the low risk region $\mathbb{S}_1$, the features had similar effects on crimes as in the STGAM, but the number of significant features was less. In the high risk region $\mathbb{S}_2$, locations with no incident happened in the previous year were still less likely to have a new incident. However, out of the locations where incidents happened in the previous two months, the locations with incidents just happened in the past month were less likely to have a new incident in the following month. In addition, the effects of the number of divorced people and owner occupied houses in the high risk region were different from the effects of the same features in the low risk region. In the high risk region, incidents were more likely to happen in the neighborhoods with less divorced

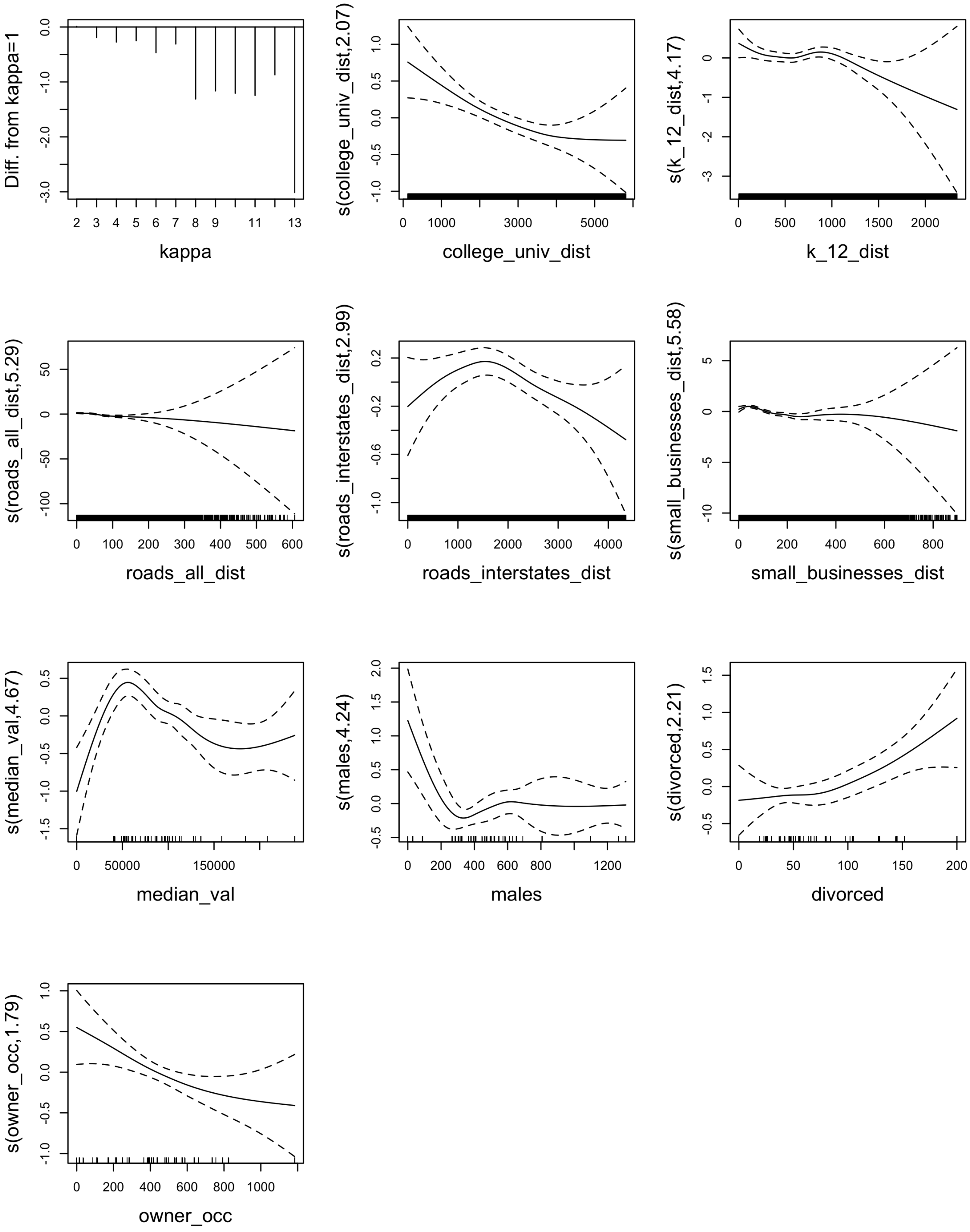


Figure 5.6: Estimation of STGAM

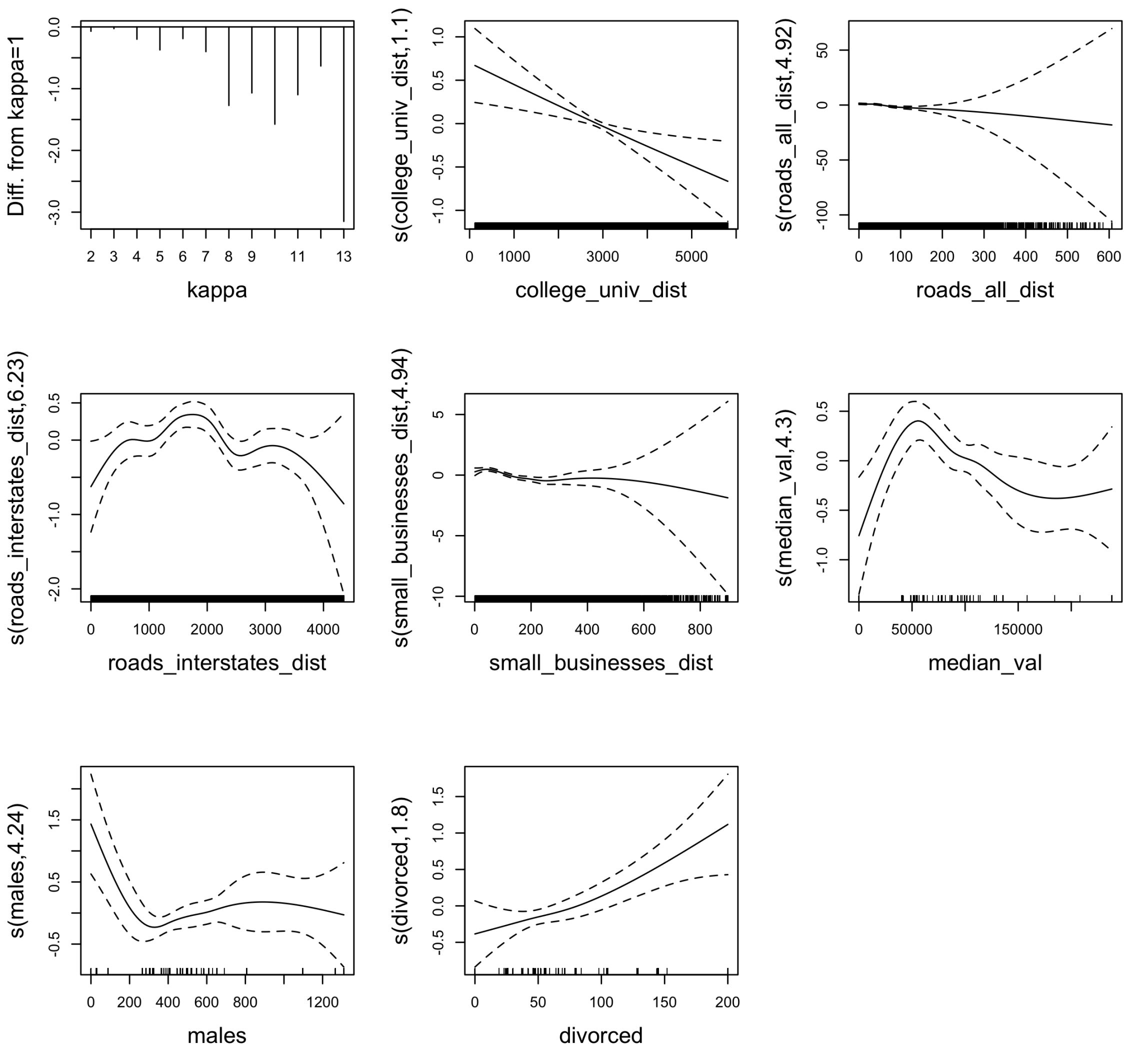


Figure 5.7: Estimation of LSTGAM in $\mathbb{S}_1$

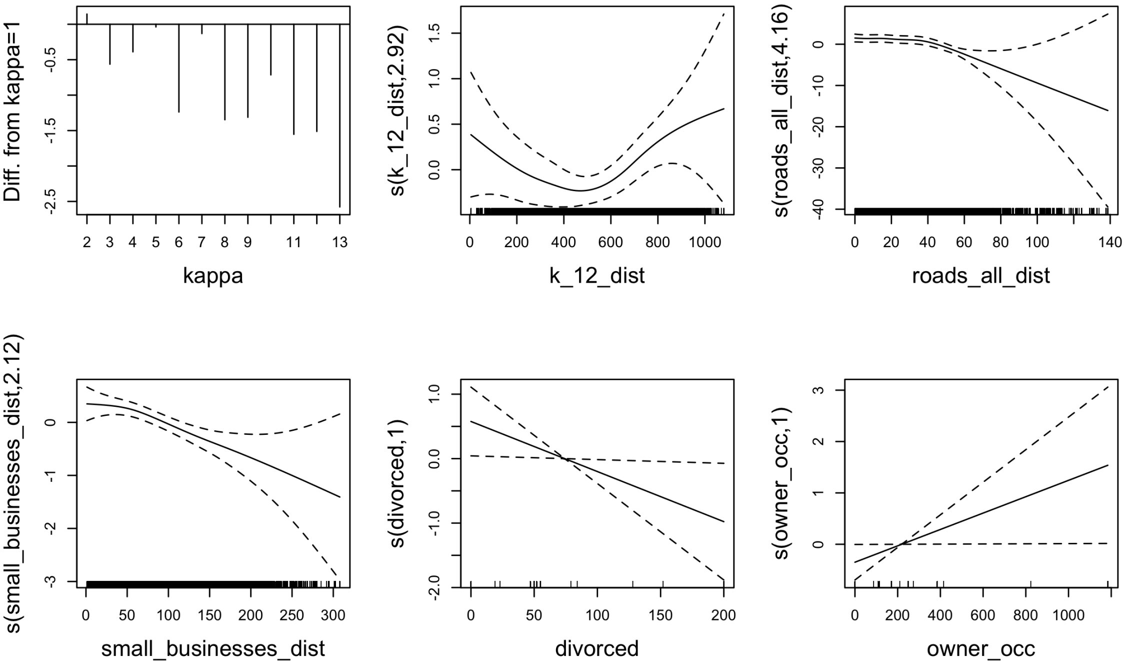


Figure 5.8: Estimation of LSTGAM in $\mathbb{S}_2$

people; and incidents were less likely to happen in the neighborhoods with more owner occupied houses .

### 5.1.5 Conclusion

Based on our assessments with the real criminal incident data in Charlottesville, Virginia, both STGAM and LSTGAM models can predict future incidents accurately. Results showed that the two models outperformed the previous spatial GLM and the hot spot model. Compared with STGAM, LSTGAM had better performance in prediction. Law enforcement agencies can use STGAM and LSTGAM to model criminal incidents, predict future incidents and prevent crimes. In addition, those two models can be applied to other areas with the need to study the spatio-temporal patterns and predict future incidents. For example, we can use STGAM and LSTGAM to predict terrorist events and car accidents.

## 5.2 Automatic Crime Prediction using Events Extracted from Twitter Posts[5]

### 5.2.1 Introduction

Traditional crime prediction models (e.g., the ones described in Section 5.1) make extensive use of numerical features, like historical incident patterns as well as layers of information provided by geographic information systems (GISs) and demographic information repositories. Although crucial, these information sources do not account for the rich and rapidly expanding social media context that surrounds incidents of interest. Without utilizing those context, prediction of criminal incidents might not be possible.

For example, in the attempt to predict the daily numbers of hit-and-run criminal incidents in Charlottesville, Virginia, we tried to build a classic autoregressive integrated moving average model (ARIMA) of the time series data. After plotting the autocorrelation (ACF) plot and partial autocorrelation (PACF) plot as shown in Figure 5.9, we found no interesting significant autocorrelation. The prediction based on the historical data would be the average of daily incidents or 1.2 incidents. We studied the incidents further and found that the likelihood of incidents might be able to be predicted by surrounding events. For example, there were 5 hit-and-hun incidents on August 16th, 2011. The local news showed that the area was hit by severe storms on the night of August 14th and many roads were closed in the following week.

The question is how to collect all the events in the area efficiently and automati-

[5] Most of the material in this section has been published in the previous paper [104].

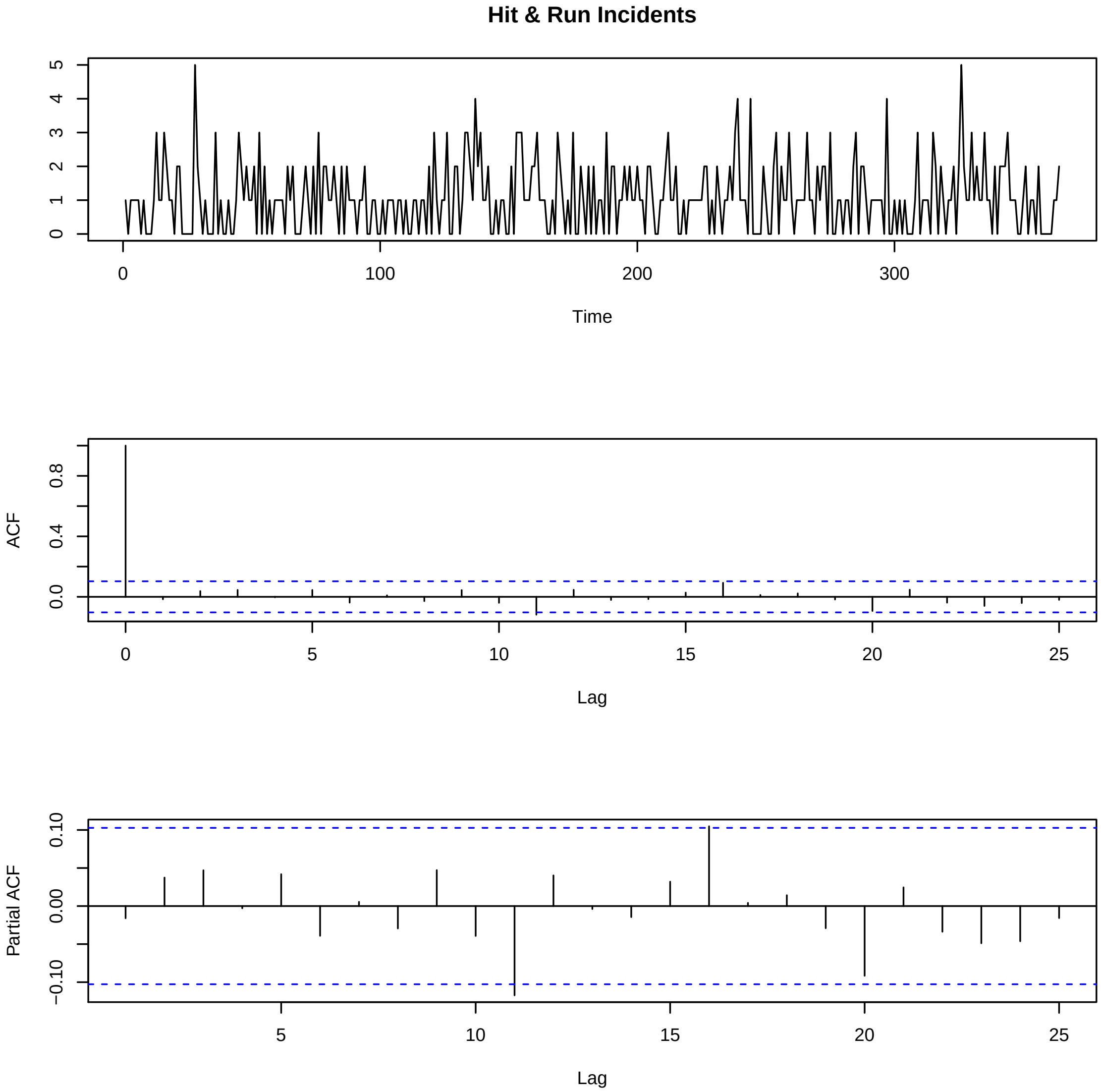


Figure 5.9: Time Series Plots of Hit-and-Run Incidents in Charlottesville, Virginia

cally. The expanding social media, such as Facebook[6] and Twitter[7], provide us with a possible solution. These social media services allow users to instantly create, disseminate, and consume information from any location with access to the Internet. For example, we found the following Twitter posts[8] from CBS19 on August 15th, 2011:

(5.4) JessicaJaglois reports 2 major roads still closed due to storm debris: Park St. and McIntire Rd. at 250 Bypass.

(5.5) Traffic Alerts: McIntire Road now open. Park Street partially open, but drivers advised not to use it as a way out of town.

(5.6) Bus Routes Adjusted Due to Storm Damage.

All these tweets provide evidence of an increased hazard level along roadways, which, in turn, might lead to an increased number of accidents or hit-and-run crimes.

There is a surge of interest in using the Twitter data for various predictive purposes. For example, Twitter posts have been used in models that predict weekend box office results [3], election results [5], and stock market trends [12]. Popular techniques in these studies include keyword volume analysis (e.g., the frequency of a movie title) and sentiment analysis (e.g., whether tweets about a movie are favorable). These methods have proven useful for the tasks described above; however, these methods lacked a deep semantic understanding of tweets as our study to predict discrete criminal incidents.

This study presented a preliminary investigation of the predictive power of social media information, in particular information produced by the Twitter service. We hypothesized that information extracted from the Twitter service would provide indicators about the likelihood of future incidents. We focused on the use of tweets pulled

[6] `http://www.facebook.com`

[7] `http://www.twitter.com`

[8] Twitter posts are also called tweets.

from the Twitter feed of a news agency covering the city of Charlottesville, Virginia and its surrounding areas. The goal of our investigation was to build a predictive model of criminal incidents that leverages the type of evidence shown in Example 5.4 to 5.6. The SRL-LDA model described in Chapter 4 was used to extract information from the tweets. Then a GLM, a simplified version of LSTGAM, was build to predict future occurrences of criminal incidents.

### 5.2.2 Data Collection and Modeling

Figure 5.10 shows the overall operation of our Twitter-based predictive model. We first collected a corpus of tweets from Twitter. We then extracted events from the main textual content of each tweet using SRL. Next, we applied latent Dirichlet allocation (LDA) to identify salient topics within the extracted events. A predictive model was then built upon these latent topics. These steps are described in details as follows.

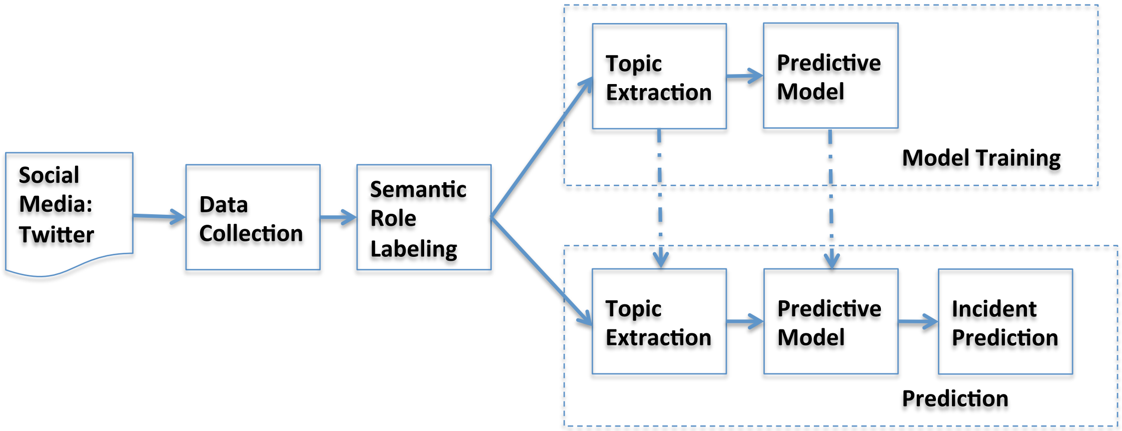


Figure 5.10: Overall Process of Criminal Incident Prediction Using Tweets

*Data Collection*

The user base of Twitter comprises a vast community of news agencies, journalists, and casual users who post tweets from their Internet-connected devices. Each tweet is restricted to 140 characters and can be observed by those who subscribe to the poster's Twitter feed. As of March 11, 2011, Twitter was processing approximately 140 million tweets per day, with approximately 460,000 new accounts being created daily[9]. Traditional news stations and newspapers actively use Twitter to publish breaking news in real-time. For example, CBS19[10] in Charlottesville, Virginia published 3,659 tweets during the period of February 22, 2011 through October 21, 2011 (approximately 15 per day). We collected these tweets using the public interface provided by Twitter.

In addition to Twitter data, our investigation required ground-truth criminal incident data, which we used to estimate the parameters of our predictive model and evaluate its performance. We obtained these records from local law enforcement agencies[11], focusing on hit-and-run incidents during the same period covered by the Twitter data. In total, we collected records for 290 hit-and-run incidents (1.2 per day).

*Semantic Role Labeling (SRL)*

Our approach to Twitter-based crime prediction relied on a semantic understanding of tweets. Such an understanding can be derived from SRL, which extracts the events mentioned in tweets, the entities involved in the events, and the roles of the entities with respect to the events. An example of the SRL analysis of a tweet is

[9] `http://blog.twitter.com/2011/03/numbers.html` (accessed November 1, 2011)

[10] `http://www.newsplex.com`

[11] `http://www.charlottesville.org/index.aspx?page=257`

shown below:

(5.7) $[_{e1:warning}$ TRAFFIC$]$ $[_{e_1}$ ALERT$]$: $[_{e2:entity}$ Rt. 20$]$ $[_{e_2}$ closed$]$ $[_{e2:cause}$ due to a wreck$]$.

Two events were extracted from Example 5.7: (1) an *alert* event in which traffic is being brought to the reader's attention, and (2) a *close* event where a road is closed due to a wreck. In our study, we used the system created by Punyakanok et al. to analyze verb-based SRL structures [79] and the system created by Gerber and Chai to analyze noun-based SRL structures [41]. The SRL output from these systems formed the basis for event prediction, since it informed the model about current events, which might correlate with future criminal incidents.

*Event-based Topic Extraction via Latent Dirichlet Allocation (LDA)*

After processing the tweets with the SRL systems, we had multiple events $e_i$ associated with each day. In topic modeling terms, each day $t$ was associated with an abstract "document" $doc_t$ that contained "words" $\{e_1, e_2, \ldots, e_{n_t}\}$, where $n_t$ is the length of $doc_t$. These words described what happened on day $t$.

As with topic modeling of actual textual documents, we hypothesized that a day's events would be related in a particular (though hidden) way. Thus, instead of using $doc_t$ directly to predict future incidents, we further extracted topics $\{T_1, T_2, \ldots, T_k\}$[12] from $doc_t$ using latent Dirichlet allocation (LDA)[13]. As discussed in Section 4.2.1, LDA is a probabilistic language model that can be used to explain how a collection of documents is generated from a set of hidden (or latent) topics. LDA efficiently discovers word-based topics and reduces the dimensionality of documents to lie within

[12] In this section, we used $T_k$ to denote topic distributions, because $\beta$ is commonly used to denote the coefficient of a linear regression model, which was used in this study.

[13] We used GibbsLDA++ [114] in this experiment.

the $k$-dimensional space of topics. Given the number of topics $k$, LDA can estimate the topic-document distribution $\{T_{t,1}, T_{t,2}, \dots, T_{t,k}\}$, where $T_{t,i}$ is the probability that document $doc_t$ is related to topic $i$.

We applied LDA to derive $\{T_{t,1}, T_{t,2}, \dots, T_{t,k}\}$ for the events described in tweets on day $t$. Intuitively, this analysis can tell us about the relationship between the $k$ major (latent) events on day $t$ and the observable events $e_i$ that were reported by the news agencies. This reduced the dimensionality of $doc_t$ and provided meaningful structured data for our predictive model.

*Predictive Model*

$doc_t$ contains the events that occurred on day $t$. Our goal was to use $doc_t$ to make predictions about incidents in the future. Formally, we needed a function $y_{t+1} = f(doc_t)$, where $y_{t+1}$ is a binary random variable indicating whether an incident will occur on day $t+1$. We used the following generalized linear regression model (GLM) to meet this need[14]:

$$log\left(\frac{p[y_{t+1}=1]}{1-p[y_{t+1}=1]}\right) = \beta_0 + \beta_1 T_{t,1} + \cdots + \beta_k T_{t,k} \tag{5.8}$$

where each $T_{t,i}$ is derived via LDA. Parameters $\{\beta_0, \dots, \beta_k\}$ can be estimated using the set of prior criminal incidents.

With both the estimated LDA model and GLM model, we can make a prediction using new tweets. To make a prediction, we first processed tweets on day $t'$ using the SRL systems described above. Then, the LDA model was used to infer the event-based topic distribution $\{T_{t',1}, T_{t',2}, \dots, T_{t',k}\}$. Lastly, the predictive model (Equation 5.8) used this distribution to predict the likelihood of an incident occurring on day $t'+1$.

[14] This GLM is a STGAM built on a single grid: the entire Charlottesville city.

### 5.2.3 Evaluation and Results

We evaluated our predictive model using Twitter data and actual hit-and-run incidents that occurred in Charlottesville, Virginia. As described in Section 5.2.2, our data covered the period of February 22, 2011 through October 21, 2011. We studied the hit-and-run incidents per day using traditional time series methods, but discovered no trend, seasonality, or autocorrelation. Thus, without any additional information, a baseline system would assign a uniform probability of incidents to all future days.

We used the data before September 17, 2011 to train the LDA and predictive models, setting $k$ (the number of latent topics) to be 10. Table 5.3 presents the top 10 words for each topic. Some structures can be found in the topics. For example, topic 1 appears to be related to crashes, whereas topic 3 appears to be related to shootings and their associated criminal processes.

We trained the GLM on these topics as described in Equation 5.8, using stepwise selection to identify the most informative features. The resulting GLM is shown below:

$$\log\left(\frac{p[y_{t+1}=1]}{1-p[y_{t+1}=1]}\right) = 0.4 + 0.71T_{t,1} + 0.88T_{t,4} + 0.72T_{t,6} + 0.61T_{t,8} \tag{5.9}$$

In Equation 5.9, $p[y_{t+1}=1]$ denotes the probability of at least one hit-and-run incident occurring on day $t+1$. $T_{t,\cdot}$ is the topic distribution on day $t$. As shown, the topics 1, 4, 6, and 8 were positively related to the future hit-and-run incidents. This means if the events emphasized in those topics occurred, the likelihood of incidents increased.

We applied this model to predict hit-and-run incidents during the period of September 17, 2011 to October 21, 2011. Figure 5.11(a) shows the ROC curve of the prediction performance. Vertical bars are 95% confidence intervals derived with a bootstrap resampling procedure. The ideal ROC curve stretches toward the upper-

Table 5.3: Top 10 Most Likely Words for Each of the 10 Topics to Predict Hit-and-Run Incidents

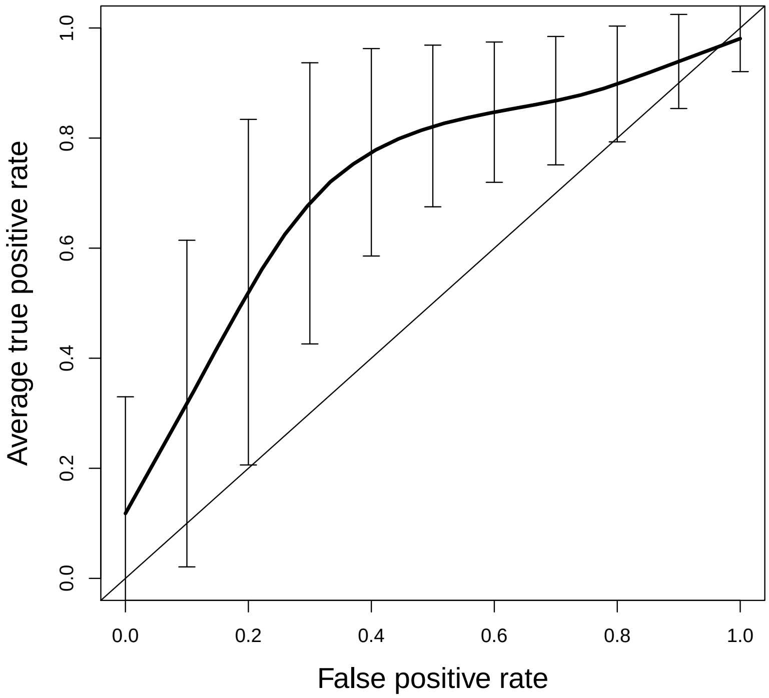
| Topic 1 | Topic 2 | Topic 3 | Topic 4 | Topic 5 |
|---|---|---|---|---|
| close | say | arrest | plan | report |
| fire | make | suspect | kill | say |
| crash | hanchettjim | death | use | student |
| look | search | murder | ask | vote |
| delay | confirm | shoot | plead | tell |
| come | run | rsb | life | hear |
| reopen | start | hear | sell | work |
| stay | move | protest | convict | speak |
| watch | begin | report | visit | head |
| driver | end | need | statement | call |
| Topic 6 | Topic 7 | Topic 8 | Topic 9 | Topic 10 |
| expect | report | say | trial | come |
| remain | close | found | make | cbs |
| rsb | open | die | break | help |
| cancel | confirm | crash | set | start |
| follow | block | fall | traffic | lead |
| close | wreck | want | begin | look |
| warn | follow | find | hope | check |
| make | accord | kill | bring | lawsuit |
| price | move | shut | hit | arrest |
| list | check | come | stop | left |

left corner. A curve along the diagonal indicates no predictive power. As shown by Figure 5.11(a), our GLM model using SRL-LDA was able to predict future hit-and-run incidents; although, due to the limited amount of testing data, we observed fairly wide confidence intervals.

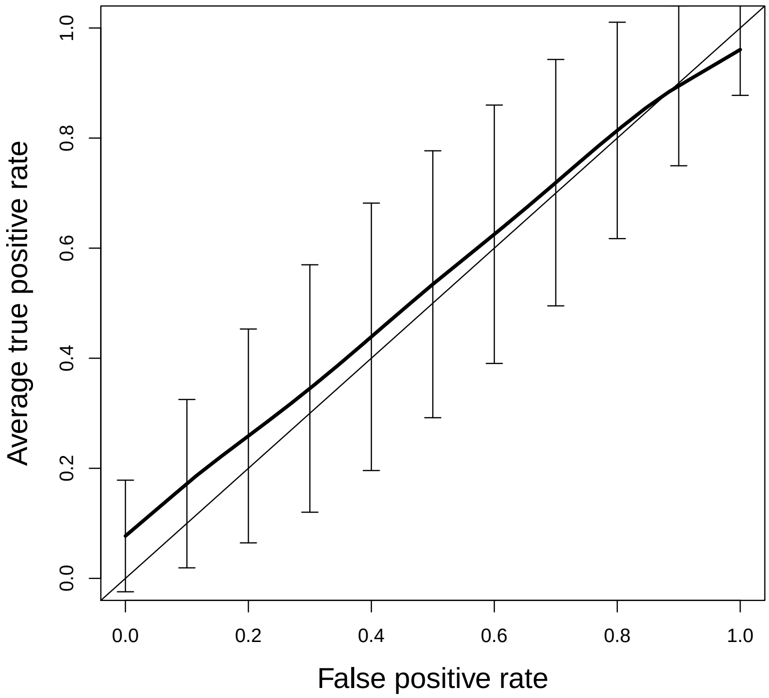
We also tested whether SRL-LDA can perform better than the standard LDA model. In order to compare the two models, we re-trained the predictive model with the topic distributions extracted by a standard LDA model with the same hyper-parameters as the SRL-LDA. The remaining experimental conditions were held constant, resulting in the ROC curve shown in Figure 5.11(b). As shown in the figure, the model using LDA had minimal predictive power. Based on this test, SRL-LDA outperformed LDA.

### 5.2.4 Conclusion

This application has presented a preliminary investigation into the use of unstructured textual data for criminal incident prediction. Our approach was based on the automatic semantic analysis and understanding of natural language tweets, combined with dimensionality reduction via latent Dirichlet allocation and prediction via linear modeling. Evaluation results demonstrated the model's ability to forecast hit-and-run crimes using only the information contained in the training set of tweets. This application also showed that SRL-LDA indeed performed better than the standard LDA model.



(a) Using Events Extracted from Tweets



(b) Using Words from Tweets

Figure 5.11: ROC Curves for Predicting Hit and Run Incidents

## 5.3 The Spatio-Temporal Modeling for Criminal Incidents using Textual Information[15]

### 5.3.1 Introduction

As reported by the Bureau of Justice and Statistics [67], there were 22,879,720 personal and property crimes with an estimated economic loss of 18 billion dollars in the United States in 2008. One approach to preventing crimes is to predict the location and time of future criminal activity. The previous two sections have shown that STGAM and SRL-LDA developed in this dissertation can predict the spatio-temporal patterns of criminal incidents separately. This section combines the two models and tests whether incorporating unstructured textual data can improve the predictability of STGAM. This section also illustrates how to build spatio-temporal models using different types of features, especially unstructured textual data.

### 5.3.2 Data Description

We used four datasets in this study. The first dataset contained breaking and entering crimes that occurred in Charlottesville, Virginia during the period of March 1st - October 31st, 2011. In total, there were 88 incidents. The dataset was obtained from local law enforcement agencies.[16] Each incident was associated with a street address and time. Each street address was mapped to geographic coordinates using the Mapquest API[17]. The second dataset was a collection of Twitter posts downloaded using Twitter's publicly accessible API. We used all tweets posted by the CBS19 news agency as used in Section 5.2. On average, there were 15 tweets per day. The last two datasets contained the geographic and demographic information used in

[15] Most of the material in this section is to be appeared in [102].

[16] http://www.charlottesville.org/index.aspx?page=257

[17] http://developer.mapquest.com

Section 5.1. The geographic dataset contained information layers such as locations of roads, small businesses, and schools. The demographic dataset measured features of Charlottesville in census block groups, including population and race.

To build spatio-temporal models, we used a grid size of 0.02 miles × 0.02 miles and a time interval of 24 hours. There were 23,089 grids within the area of Charlottesville and 5,656,805 spatio-temporal records. Each record had a response variable indicating whether at least one incident occurred within the grid and time interval. Each record also had three types of features describing characteristics of the space and time. The first feature type captured the minimum distances between the centroid of a grid and certain geographic landmarks. The second feature type captured demographic properties of grids' neighborhoods. The last feature type contained textual information describing the day's news in Charlottesville. All the records located on the same day were associated with the same textual feature. In total, there were 14 distance features, 20 demographic features, and 1 textual feature. A subset of the distance and demographic features is shown in Table 5.4. Features with names ending in "_dist" are distance features and the rest are demographic features. The textual feature contained Twitter posts from CBS19 grouped by date and analyzed using the SRL-LDA method described earlier. To test the model's prediction performance, we used the incident data between October 1st and October 31st. We used the remaining data to estimate the model's parameters.

### 5.3.3 Modeling and Results

*Feature Selection by RLAR*

We first applied the feature selection method developed in Section 3.3 to select distance and demographic features. Because of the large size of the training set, we used the sub-sampling technique described in Section 3.4 with RLAR. The top 20 features with ranks are shown in Table 5.4. To decide the number of features to use, we built STGAM for each different number of features and computed the restricted maximum likelihood (REML) score of each model. The smaller the score, the better the model is. Figure 5.12 shows the result. The x-axis indicates the number of features used in STGAM. For example, “5” means the STGAM used the five top-ranked features in Table 5.4. The y-axis is the REML score of the corresponding STGAM. The variances of REML scores were estimated by 100 replicates of the model estimation. The grey vertical bars show the 95% confidence intervals of REML scores based on the estimated variances. Based on this figure, we chose 18 features to model the criminal incidents within Charlottesville.

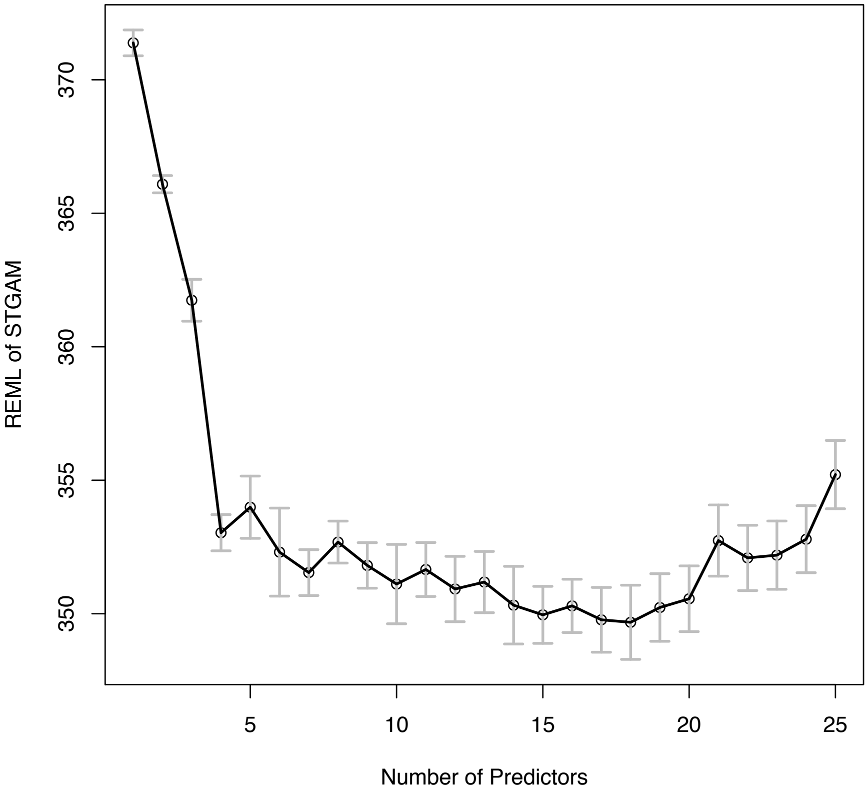


Figure 5.12: REML Scores of STGAM with Different Predictors

Table 5.4: Ranked Features by RLAR for the Spatio-Temporal Modeling of Criminal Incidents using Textual Information

| Rank | Feature | Description |
|---|---|---|
| 1 | small_business_dist | distance to the nearest small business |
| 2 | nursehomes_dist | distance to the nearest nurse home |
| 3 | nevermarry | number of people who are never married |
| 4 | roads_dist | distance to the nearest road |
| 5 | vacant | count of vacant houses |
| 6 | rivers_dist | distance to the nearest river |
| 7 | renter_occ | count of renter-occupied households |
| 8 | married | number of people who are married |
| 9 | it_hardware_dist | distance to the nearest IT hardware |
| 10 | roads_interstates_dist | distance to the nearest interstate highway |
| 11 | telecom_services_dist | distance to the nearest telecom services |
| 12 | medianrent | median rent charged for all housing units that are rented |
| 13 | separated | number of people who are separated |
| 14 | telecom_products_dist | distance to the nearest telecom products |
| 15 | females | number of females |
| 16 | black | number of African American persons |
| 17 | hispanic | number of hispanic persons |
| 18 | elect_trans_dist | distance to the nearest electricity transmission lines |
| 19 | it_services_dist | distance to the nearest IT services |
| 20 | owner_occ | count of owner-occupied households |

*Extracting Textual Features by SRL-LDA*

Next, we extracted textual information from the Twitter posts using SRL-LDA, which was described in Chapter 4. This process was similar to the process we used in Section 5.2 to predict the likelihood of daily criminal incidents by Twitter posts. We grouped Twitter posts by date. Thus, each day was associated with a "document" containing the day's news tweets. We processed all tweets with the verb-based SRL system of Punyakanok et al. [79] and the noun-based SRL system of Gerber and Chai [41]. We filtered out all twitter words that were not events, as indicated by the SRL systems. After the SRL analysis, we trained a 10-topic LDA model using the "topicmodels" package in R. The top 15 most likely words for each topic are shown in Table 5.5. Again, we can find interesting groups of words. For example, topic 1 was about the trail of violent crimes; topic 3 was related to road conditions; and topic 8 seemed to be related to the occurrence of violent crimes. We used the topic distributions as the numerical representation of the textual information.

*Prediction of Criminal Incidents by STGAM*

Using textual features as well as distance and demographic features, we built the STGAM to model the criminal incidents and predict on the test set. To see whether including text information can help to improve the prediction performance, we also built a STGAM without using textual features. In addition, we built a STGAM using distance and demographic features as well as textual features extracted by LDA without SRL. In all three models, we used $K = 8$ for the temporal dummy variable $\kappa$. To compare the models, we used the HRP-TIP plot described in Section 3.5.

Figure 5.13 shows the prediction result. In the figure, HRP and TIP are the percentage of high-risk area and the percentage of incidents that occurred within the high-risk area, respectively. Overall, we can see that STGAM using textual

Table 5.5: Top 15 Most Likely Words for Each of the 10 Topics to Predict Breaking-and-Entering Incidents

| Topic 1 | Topic 2 | Topic 3 | Topic 4 | Topic 5 | Topic 6 | Topic 7 | Topic 8 | Topic 9 | Topic 10 |
|---|---|---|---|---|---|---|---|---|---|
| found | warn | update | close | announce | win | make | arrest | state | die |
| charge | cancel | close | report | say | report | arrest | suspect | region | crash |
| sentence | delay | crash | damage | confer | student | say | shoot | plan | say |
| murder | release | report | say | found | send | vote | death | hear | look |
| trial | expect | call | come | confirm | celebrate | watch | charge | leave | fall |
| come | flood | fire | follow | start | service | lead | kill | develop | investige |
| plead | consider | cause | open | kill | fire | want | search | need | come |
| say | effect | charge | remain | live | help | investigate | student | create | kill |
| ask | meet | approve | robbery | make | increase | debate | change | face | plan |
| enter | talk | injury | find | run | check | tell | connect | inauguration | injury |
| identify | appeal | vote | include | follow | honor | list | deny | look | join |
| hold | rain | expect | confirm | hold | hope | beat | invas | sell | watch |
| stop | assist | delay | work | speak | traffic | begin | include | address | move |
| continue | launch | driver | head | release | set | fire | attack | arrest | raise |
| seek | lawsuit | hit | rsb | sign | tell | break | award | expand | charge |

information from SRL-LDA performed better than the other two models. More than 60% of actual incidents occurred within the top 20% of the area predicted using STGAM with SRL-LDA. This is compared to the 50% of actual incidents captured by STGAM in the same area when not using textual information. The STGAM with LDA had better performance than the simple STGAM. It also performed better than the STGAM with SRL-LDA when $HRP > 0.5$. The AUC of the three models was 0.7616, 0.7235, and 0.7394 for STGAM with SRL-LDA, STGAM without textual information and STGAM with LDA, respectively. Based on this criterion, STGAM with SRL-LDA performed the best. In addition, we tested the significance of the difference between any two curves by computing the differences between points on the curves at 100 different $HRP$ values. The one-sided paired Wilcoxon significance test indicated a p-value of $3.806 \times 10^{-6}$ for the difference between STGAM with SRL-LDA and STGAM without textual information. The p-value was 0.008018 for the difference between STGAM with SRL-LDA and STGAM with LDA. At the significance level of $p < 0.05$, STGAM with SRL-LDA performed significantly better than the other two models.

Figure 5.14 shows several estimated smooth functions of the STGAM using SRL-LDA. The first two plots about $road\_dist$ and $females$ show similar crime patterns as presented in the study in Section 5.1: the houses closer to roads were more likely to be entered; and the crimes were more likely to happen in the neighborhoods where there were more females. The last plot is about topic 3 extracted from tweets. The more a day was related to this topic, the less likely the crimes happened on the day. As shown in Table 5.5, this topic was about road conditions, especially bad conditions.

### 5.3.4 Conclusion

This section presented the work that brought together STGAM and SRL-LDA models to criminal incident modeling. We evaluated the combined models using actual

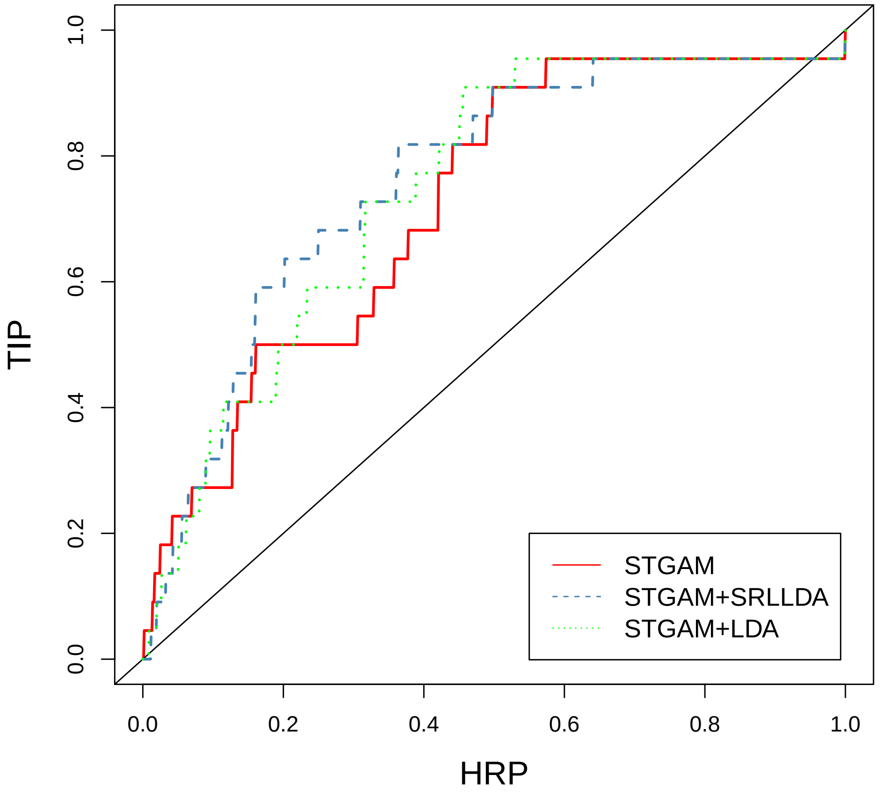


Figure 5.13: Prediction in October 2011

criminal incident data for Charlottesville, Virginia. Our results indicated that the STGAM using textual features extracted from SRL-LDA model exhibited improved prediction performance versus the standard STGAM. The hybrid STGAM+SRL-LDA model can be generalized to other application areas where unstructured textual information contains indicators relevant to the spatio-temporal properties of events.

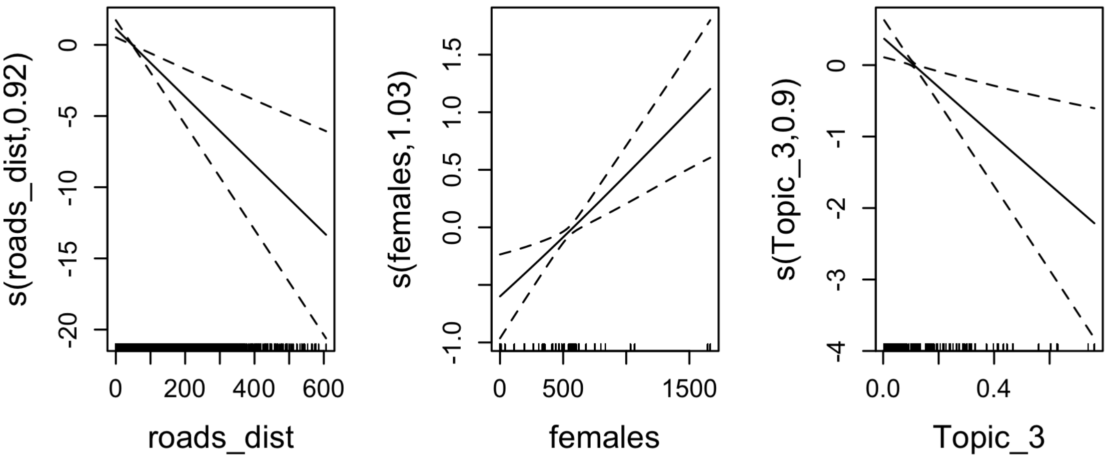


Figure 5.14: Selected Estimated Smooth Functions of STGAM using SRL-LDA

## 5.4 Prediction of Equipment Damage in Train Accidents using Spatio-temporal and Textual Features

### 5.4.1 Introduction

Federal Railroad Administration keeps track of train accidents in the United States [34] . For each train accident, they record the location and time of the train collision, accident damages, fatalities, attributes of trains, weather condition, and narratives about the accident. These data provide us the resources to study relationships between train damages and environmental, operational and other factors, which can support important decision making like how to prevent train accidents. For example, we studied whether positive train control technologies can reduce damages involved in train accidents [18].

These data are also helpful for building predictive models, which can quickly estimate final damages given the description of accidents. The objective of the study in this section was to build such a model using all types of features. Different from the previous three applications which built spatio-temporal classification models, this study built a spatio-temporal regression model. Especially, we're interested in whether using spatio-temporal and textual features can improve the prediction performance of the regression model.

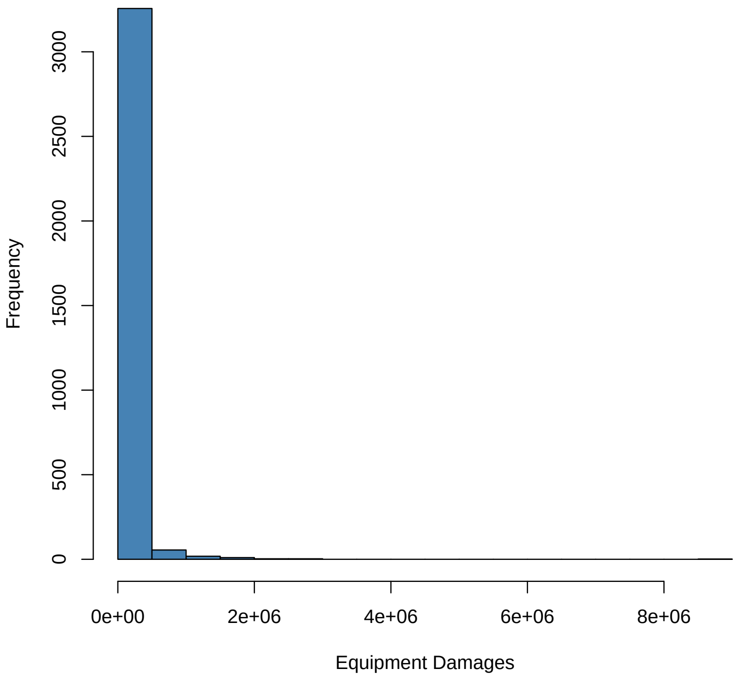
### 5.4.2 Data Description

The data sets used in this study were the train accident data from Federal Railroad Administration (FRA) in 2007 and 2008 [34]. We used 2008 data to build and test the model and used 2007 data to compute spatio-temporal features. In total, there were 3346 accident records[18]. In this study, we focused on the prediction of equipment damage. The histograms of this response variable are shown in Figure 5.15. Figure 5.15(a) shows the variable in its original scale. As we can see, most of accidents had very small cost. The distribution is highly skewed. Figure 5.15(b) shows the variable with a logarithm transformation[19]. The distribution is more symmetric. In the modeling, we used the log transformed damage as the response variable.

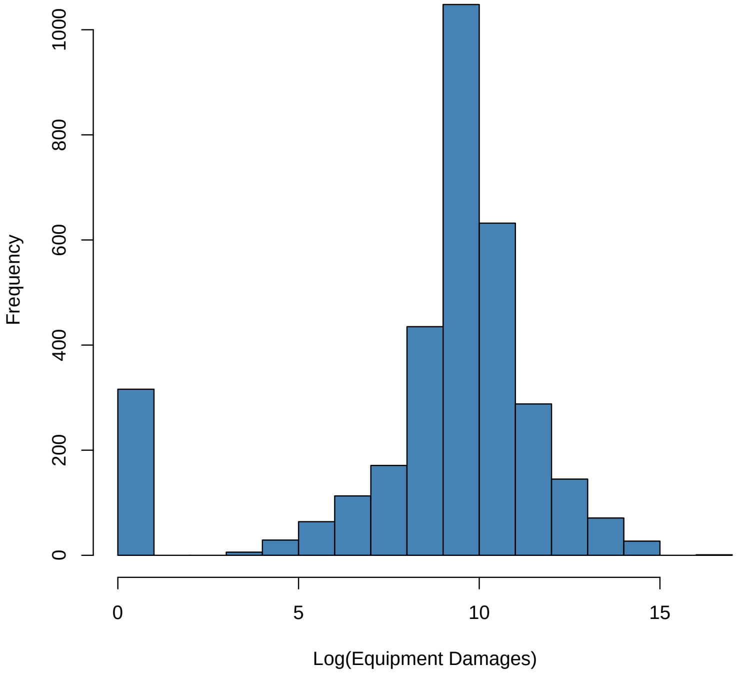
For each record, we had three types of features. The first type was ordinary features as shown in Table 5.6. These features were available in the data sets from FRA directly. The second type was locations of collisions. Because no exact geographic coordinate was available in the data sets, we extracted county names from the data sets and used coordinates of centroids of the counties where accidents happened as

[18] Some accidents had multiple accident records, each record for a different train involved.

[19] Because some damages were with the value of zero, we added 1 to the variable in the log transformation. In this section, all log transformations of equipment damages were performed in the same way.



(a) Histogram of Orignal Equipment Damages



(b) Histogram of Logarithm Transformed Equipment Damages

Figure 5.15: Histograms of Equipment Damages

locations[20]. The third type was narratives about accidents in the data sets[21].

Table 5.6: Ordinary Features for Equipment Damage Modeling

| Feature | Description |
|---|---|
| CARS | number of cars carrying hazardous materials and items |
| AMPM | am or pm when the accident happened |
| TYPE | type of accident |
| TEMP | temperature |
| WEATHER | weather conditions |
| TRNSPD | speed of train in miles per hour |
| TONS | gross tonnage (excluding power units) |
| CAUSE | primary cause of incident: signal failure, human errors, electronic failure, track failure, or miscellaneous failure |
| VISIBLTY | daylight period: dawn, day, dusk, or dark |
| TYPEQ | type of train |
| HIGHSPD | maximum speed reported for equipment involved |

### 5.4.3 Modeling and Results

A spatio-temporal regression model was built to predict equipment damages based on STGAM described by Equation 3.4. We specified the model as follows:

[20] Coordinates of counties were from [98].

[21] Misspelled words were corrected by the public interface from `http://www.spellcheck.net/`

$$E\left[log(y_{g,t})\right] = \sum_{p=1}^{P_O} f_p(x_{[g,t,p]}) + \sum_{q=1}^{P_{ST}} f_{ST,q}\left(f_q^{feature}(y_{g,t}, y_{\cdot,t-1})\right) + \sum_{k=1}^{T} f_{txt,k}(\beta_{[g,t,k]}) \tag{5.10}$$

$$f_q^{feature}(y_{g,t}, y_{\cdot,t-1}) = \|s_g^t - s_{HCA_q}^{t-1}\| \tag{5.11}$$

In the above model,

1. $y_{g,t}$ is the response variable equipment damage. In this study, we considered equipment damages grouped by years. Therefore, $t$ refers to year 2008 and $t-1$ refers to year 2007. $g$ means the $g^{\text{th}}$ accident record and the accident happened at the location $s_g$.

2. The smooth functions $f_p$, $f_{ST,q}$, and $f_{txt,k}$ are corresponding to ordinary features, spatio-temporal features and textual features respectively.

3. For ordinary features, there were $P_O = 11$ different features as shown in Table 5.6. For categorical variables, we used dummy variables for coding.

4. Equation 5.11 defines the $q^{\text{th}}$ spatio-temporal feature by the distance to the $q^{\text{th}}$ highest cost accident in the previous year. $\|\cdot\|$ means the distance between two locations. $s_g^t$ is the location of the accident described in the $g^{th}$ record in $t = 2008$. $s_{HCA_q}^{t-1}$ is the location of the $q^{\text{th}}$ highest cost accident in $t-1 = 2007$. To decide the maximum number of the highest cost accidents to be considered $P_{ST}$, we plotted damages in 2007 as shown in Figure 5.16. There was a cut-off at the damage of two million dollars, corresponding to the top ten highest cost accidents. Therefore, we chose $P_{ST} = 10$.

5. To structure narratives, we used a 10-topic LDA model. SRL-LDA was not applied in this application because the narratives about train accidents were

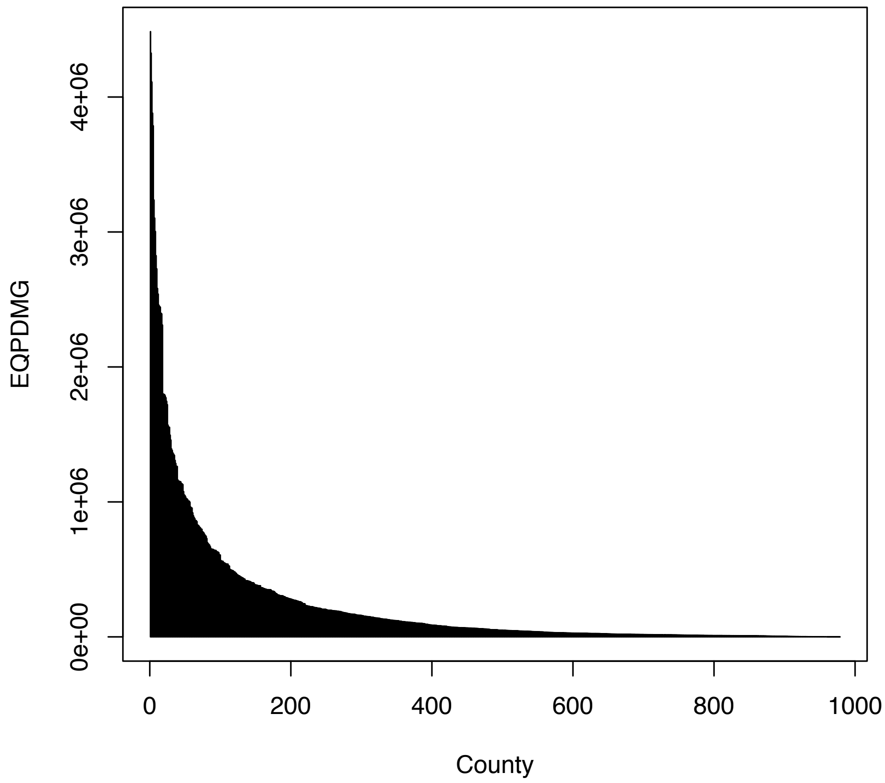


Figure 5.16: Equipment Damages of Train Accidents in 2007

not well-formed sentences. Instead, these narratives included individual words and phrases. We cleaned the narratives by removing stop words and correcting misspellings. The cleaned narrative of each accident report was considered as a document. The top words of each topic from the LDA model are shown in Table 5.7. These topics might not be interpreted easily. Generally, they described different situations about accidents. In the above equation, $T = 10$ is the total number of topics; and $\beta_{[g,t,k]}$ represents the probability that the narrative of the $g^{\text{th}}$ accident at time $t$ was about the $k^{\text{th}}$ topic.

We built the above model with 2008 accident data. To test whether including spatio-temporal and textual data can improve the predictability, we also built three

Table 5.7: Top 10 Most Likely Words for Each of the 10 Topics to Predict Equipment Damages Involved in Train Accidents

| Topic 1 | Topic 2 | Topic 3 | Topic 4 | Topic 5 |
|---|---|---|---|---|
| cars | track | switch | track | train |
| shoving | lead | conductor | cars | main |
| hazardous | end | engineer | out | went |
| two | yard | movement | cut | rear |
| derailed | west | pulled | rolled | emergency |
| one | east | lined | switching | found |
| leaking | south | move | standing | line |
| loaded | north | point | left | traveling |
| three | shoved | shove | back | mph |
| empty | job | stopped | kicked | approximately |
| Topic 6 | Topic 7 | Topic 8 | Topic 9 | Topic 10 |
| side | crew | struck | car | derailed |
| loads | not | damage | causing | rail |
| out | locomotive | crossing | bnsf | due |
| head | derailment | unit | derail | pulling |
| derailing | consist | engine | caused | broken |
| units | did | cwr | set | wheel |
| failed | moving | truck | released | under |
| pulling | cause | stop | wheels | account |
| resulting | power | equipment | other | curve |
| empties | speed | lead | between | gauge |

models using different sets of features: a model using only ordinary features, a model using ordinary features and spatio-temporal features, and a model using ordinary features and textual features. For all four models described above, we computed predicted root mean squared errors (RMSE) in the original scale by 10-fold cross-validations. Then the models with non-ordinary features were compared to the model using only ordinary features by one-sided Wilcoxon paired tests to test whether the differences between predicted RMSE were significant. Table 5.8 shows the results. Based on this test, we can see the model with all three types of features performed better than the other models. Including spatio-temporal features or textual features can improve the model with only ordinary features. At the significance level of $p < 0.05$, both of the models using spatio-temporal features were significant better than the model with only ordinary features.

Table 5.8: Predicted RMSE of Equipment Damages

| Features | Ordinary | Ordinary+Spatio-Temporal | Ordinary+Textual | All |
|---|---|---|---|---|
| RMSE | 183546.2 | 182261.5 | 180408.8 | 179346.5 |
| P-Values | - | 0.04199 | 0.05273 | 0.001953 |

### 5.4.4 Conclusion

In this study, we applied STGAM to a regression problem to predict equipment damages in train accidents. We showed a method to compute the spatio-temporal feature based on the distances to the highest cost sites in the previous year. Based on the evaluation with real data from FRA, we can see including spatio-temporal and textual features discussed in this dissertation can indeed improve the prediction performance.

# CHAPTER 6

# CONCLUSION AND FUTURE WORK

## 6.1 Conclusion

This dissertation described a spatio-temporal generalized additive model (STGAM) and its extension the local spatio-temporal generalized additive model (LSTGAM) to model spatio-temporal data. Both STGAM and LSTGAM can fully utilize many different types of data, such as spatial and temporal data, geographic data, demographic data, textual data, etc. Both models can be easily estimated by available algorithms and has good interpretability. Based on our assessments with real criminal incident data and train accident data, both models can predict accurately. Results showed that these two models outperformed several previous models, such as spatial GLM and hot spot models. These two models can be applied to other areas with the need to study spatio-temporal patterns and predict future incidents. For example, we can use STGAM and LSTGAM to predict terrorist events and car accidents.

In addition, this dissertation developed a new semantic role labeling-based latent Dirichlet allocation (SRL-LDA) model to extract key information from unstructured textual data. This model is based on the automatic semantic analysis and understanding of natural languages, combined with dimensionality reduction via latent Dirichlet allocation. The outputs from SRL-LDA are structured numerical vectors. These vectors are meaningful: they describe the probabilities of a document being related to different topics. It can also partly solve synonymy and polysemy problems which prevent many text mining methods from being used for prediction. The dissertation applied and tested SRL-LDA with two real problems about criminal incidents. In both problems, SRL-LDA were used to extract information from Twitter posts. Based on the tests, information extracted by SRL-LDA had the ability to predict criminal incidents. Compared to LDA, SRL-LDA performed better to predict hit-and-run incidents. These two applications also revealed interesting sources of data for criminal prediction: social media services.

This dissertation showed how to combine STGAM with SRL-LDA. The hybrid model was evaluated using actual criminal incident data for Charlottesville, Virginia. The results indicated that the hybrid model exhibited improved prediction performance versus the standard STGAM model. The hybrid model can be generalized to other application areas where unstructured textual information contains indicators relevant to the spatio-temporal properties of events.

In addition to the above models, this dissertation has described a new feature selection algorithm. Tests with simulated data and real data showed the algorithm performed better than a classic penalized linear regression model. This algorithm can be applied independently to choose features for nonlinear models.

## 6.2 Contribution

This dissertation has the following contributions to spatio-temporal modeling, supervised learning algorithms as well as text modeling:

1. It developed a methodology to model spatio-temporal data with numerous features. Particularly, it formalized an important class of problems related to spatio-temporal data. It built and analyzed an effective mathematical model, the local spatio-temporal generalized additive model(LSTGAM), incorporating spatio-temporal features, numerical features, categorical features, as well as textual features. It showed how to estimate the model with most recently developed algorithms.

2. It developed a new feature selection algorithm, randomized least angle regression (RLAR). Based on the test of both simulation data and real data, RLAR can select features efficiently and effectively for nonlinear regression models.

3. It designed a semantic role labeling-based latent Dirichlet allocation model (SRL-LDA) to combine a successful natural language processing method and a popular text mining method to extract key information from high dimensional unstructured textual data. Especially, SRL-LDA can consider both semantic information and syntax of the English language in the modeling process.

4. It showed the applicability of models to real problems, such as law enforcement, risk analysis, etc. In the law enforcement applications, it showed demographic features, distance features and textual features extracted from Twitter had the ability to predict spatio-temporal patterns of criminal incidents. The models also provided interpretable results to help law enforcement agencies to identify possible causal factors for criminal incidents. In the application of train damage prediction, it showed utilizing both structured data and unstructured textual data can quickly estimate final damages given the description of accidents. It demonstrated the validity of model estimates based on data sets from the real world. It also discussed how to extend the methods to a broader class of real problems.

5. It explored new sources of textual data, social media services like Twitter, for criminal incident prediction. It showed how to access those data and how to utilize them. With further investigation, these types of data are possible to be applied to enhance the predictability in many different areas.

## 6.3 Future Work

In the future, the research in this dissertation can be extended in the following ways:

1. For STGAM, more sophisticated penalized regression methods like group Lasso can be incorporated into STGAM estimation process to choose features automatically. This process requires fast algorithms to solve large-scale optimization problems. It is also possible to be realized with the development of computation powers.

2. For LSTGAM, only one method based on incident density was discussed in this dissertation. Better methods can be developed to generate optimal regions.

3. For RLAR, it was evaluated with a limited number of examples. Further tests and comparisons are required to fully conclude that it indeed performs better than any other feature selection methods. The theoretical study of why it works is also interesting.

4. For SRL-LDA, one could take advantage of the spatial and temporal extraction capabilities of the SRL systems. The current SRL-LDA model ignores textual information describing an event's spatial and temporal location. This information could be used to map tweets to particular spatio-temporal grid locations. This would improve the model's ability to identify textual information that correlates with spatio-temporal patterns.

5. There is another way to improve SRL-LDA model. Instead of manually deciding what types of words are important, one could use a model to select important words labeled by the SRL systems automatically.

6. The dissertation showed three applications to test and evaluate the newly developed models. All these applications were based on data from Charlottesville,

Virginia. A comprehensive comparison including additional models with more data sets in different geographical locations and for longer periods of time should be performed.

7. The textual data used in this dissertation were still limited. Instead of just using tweets from one user, additional textual data can be collected online. A large scale text analysis system can provide much information about our world. With these information, the ability of modeling and prediction can be greatly enhanced.

# BIBLIOGRAPHY

[1] A. Aizawa. An information-theoretic perspective of tf-idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

[2] L. Anselin, J. Cohen, D. Cook, W. Gorr, and G. Tita. Spatial analyses of crime. *Criminal justice*, 4:213–262, 2000.

[3] S. Asur and B. Huberman. Predicting the future with social media. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 492–499. IEEE, 2010.

[4] N. Beck, J. Katz, and R. Tucker. Taking time seriously: Time-series-cross-section analysis with a binary dependent variable. *American Journal of Political Science*, 42(4):1260–1288, 1998.

[5] A. Bermingham and A. Smeaton. On using twitter to monitor political sentiment and predict election results. In *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011)*, pages 2–10, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing.

[6] A. Bies, M. Ferguson, K. Katz, R. MacIntyre, V. Tredinnick, G. Kim, M. Marcinkiewicz, and B. Schasberger. Bracketing guidelines for treebank ii style penn treebank project. *University of Pennsylvania*, 1995.

[7] D. Blei, L. Carin, and D. Dunson. Probabilistic Topic Models. *Signal Processing Magazine, IEEE*, 27(6):55–65, 2010.

[8] D. Blei, T. Griffiths, and M. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 57(2):1–30, 2010.

[9] D. Blei and J. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[11] C. Block. STAC hot-spot areas: A statistical tool for law enforcement decisions. In *Crime analysis through computer mapping. Washington, DC: Police Executive Research Forum*, pages 15–32. Citeseer, 1995.

[12] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2011.

[13] L. Breiman. *Classification and regression trees.* Chapman & Hall/CRC, 1984.

[14] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[15] D. Brown, J. Dalton, and H. Hoyle. Spatial forecast methods for terrorist events in urban environments. *Intelligence and Security Informatics*, pages 426–435, 2004.

[16] D. Brown and S. Hagen. Data association methods with applications to law enforcement. *Decision Support Systems*, 34(4):369–378, 2003.

[17] D. E. Brown. Last of selection and discriminant analysis. Lecture 8 Note for SYS618: Data Mining, 2007.

[18] D. E. Brown and X. Wang. Postive train control. Laboratory 1 for SYS4021&6021: Linear Statistical Models, 2011.

[19] C. Burges, R. Ragno, and Q. Le. Learning to rank with nonsmooth cost functions. *Advances in Neural Information Processing Systems*, 19:193, 2007.

[20] G. Casella and E. I. George. Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174, Aug 1992.

[21] J. Chang. "lda" packge in R. `http://cran.r-project.org/web/packages/lda/`.

[22] F. Ciravegna, A. Lavelli, N. Mana, J. Matiasek, L. Gilardoni, S. Mazza, M. Ferraro, W. Black, F. Rinaldi, and D. Mowatt. Facile: Classifying texts integrating pattern matching and information extraction. In *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 16, pages 890–897. Citeseer, 1999.

[23] D. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 187–220, 1972.

[24] N. Cressie. *Statistics for spatial data*. John Wiley & Sons, New York, 1993.

[25] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[26] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):138, 1977.

[27] V. Denis, J. Lejeune, and J. Robin. Spatio-temporal analysis of commercial trawler data using General Additive models: patterns of Loliginid squid abundance in the north-east Atlantic. *ICES Journal of Marine Science*, 59(3):633, 2002.

[28] P. Diggle. *Statistical analysis of spatial point patterns.* Edward Arnold, second edition, 2003.

[29] J. Eck, S. Chainey, J. Cameron, M. Leitner, and R. Wilson. Mapping crime: Understanding hot spots. 2005.

[30] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of statistics*, 32(2):407–451, 2004.

[31] H. Erdogan, R. Sarikaya, S. Chen, Y. Gao, and M. Picheny. Using semantic analysis to improve speech recognition performance. *COMPUTER SPEECH AND LANGUAGE*, 19(3):321–343, JUL 2005.

[32] C. Fall, A. Törcsvári, K. Benzineb, and G. Karetka. Automated categorization in the international patent classification. In *ACM SIGIR Forum*, volume 37, page 25. ACM, 2003.

[33] T. Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[34] Federal Railroad Administration Office of Safety Analysis. Federal railroad administration (fra) accident/incident data). `http://safetydata.fra.dot.gov/officeofsafety/`, Last accessed: July 2010.

[35] Y. Freund and R. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of computer and system sciences*, 55:119139, 1997.

[36] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning.* Springer Series in Statistics, 2001.

[37] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Department of Statistics, Stanford University, Tech. Rep*, 2008.

[38] W. Fu. Penalized regressions: the bridge versus the lasso. *Journal of computational and graphical statistics*, pages 397–416, 1998.

[39] M. Gerber. *Semantic role labeling of implicit arguments for nominal predicates.* PhD thesis, Michigan State University, 2011.

[40] M. Gerber and J. Chai. Beyond NomBank: A study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1583–1592, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[41] M. Gerber, J. Chai, and A. Meyers. The role of implicit argumentation in nominal SRL. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 146–154, Boulder, Colorado, June 2009. Association for Computational Linguistics.

[42] Google. Google flu trends. `http://www.google.org/flutrends/`, March Last accessed: July 2010.

[43] T. L. Griffiths, , and M. Steyvers. Finding scientific topics. *PNAS*, 101:5228–5235, 2004.

[44] B. Grün and K. Hornik. “topicmodels” packge in R. `http://cran.r-project.org/web/packages/topicmodels/`.

[45] C. Gu. *Smoothing spline ANOVA models.* Springer Verlag, 2002.

[46] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[47] J. Hartigan and M. Wong. Algorithm AS 136: A K-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[48] T. Hastie. "gam" pakcage in R. `http://cran.r-project.org/web/packages/gam/index.html`, Last accessed: March, 2012.

[49] T. Hastie and R. Tibshirani. Generalized additive models. *Statistical Science*, 1(3):297–310, 1986.

[50] T. Hastie and R. Tibshirani. *Generalized Additive Models, volume 43 of Monographs on Statistics and Applied Probability.* Chapman and Hall, 1990.

[51] A. Hoang. Information retrieval with principal components. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, volume 1, pages 262–266. IEEE, 2004.

[52] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[53] A. Hotho, A. Maedche, and S. Staab. Ontology-based text document clustering. *KI*, 16(4):48–54, 2002.

[54] C. Ivaha, H. Al-Madfai, G. Higgs, and A. Ware. The Dynamic Spatial Disaggregation Approach: A Spatio-Temporal Modelling of Crime. In *Proceedings of the World Congress on Engineering*, volume 2. Citeseer, 2007.

[55] G. Kauermann, T. Krivobokova, and L. Fahrmeir. Some asymptotic results on generalized penalized spline smoothing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):487–503, 2009.

[56] S. Khudanpur and J. Wu. Maximum entropy techniques for exploiting syntactic, semantic and collocational dependencies in language modeling. *COMPUTER SPEECH AND LANGUAGE*, 14(4):355–372, OCT 2000.

[57] G. King and L. Zeng. Logistic regression in rare events data. *Political analysis*, 9(2):137, 2001.

[58] K. Koh, S. Kim, and S. Boyd. An Interior-Point Method for Large-Scale l 1-Regularized Logistic Regression. *The Journal of Machine Learning Research*, 8:1555, 2007.

[59] M. Konchady. *Text mining application programming*. Charles River Media, 2006.

[60] H. Liu and D. Brown. Criminal incident prediction using a point-pattern-based density model. *International Journal of Forecasting*, 19(4):603–622, 2003.

[61] P. Long and R. Servedio. Random classification noise defeats all convex potential boosters. In *Proceedings of the 25th international conference on Machine learning*, pages 608–615. ACM, 2008.

[62] I. Mani and M. Maybury. *Advances in automatic text summarization*. MIT Press, 1999.

[63] C. Manning, P. Raghavan, and H. Schutze. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, 2008.

[64] C. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT Press, 2000.

[65] M. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of English: the Penn TreeBank. *Computational Linguistics*, 19:313–330, 1993.

[66] B. Marx and P. Eilers. Direct generalized additive modeling with penalized likelihood. *Computational Statistics & Data Analysis*, 28(2):193–209, 1998.

[67] C. Maston and US Dept of Justice and Bureau of Justice Statistics. Criminal victimization in the united states, 2007–statistical tables. *Bureau of Justice Statistics (NCJ 227669), http://bjs.ojp.usdoj.gov/index.cfm?ty=pbdetail&iid=1743*, page Table 82, 2010.

[68] A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. The nombank project: An interim report. In *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31. Boston, Massachusetts, USA: Association for Computational Linguistics, 2004.

[69] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 352–359, 2002.

[70] M. Mittermayer and G. Knolmayer. Text mining systems for market response to news: A survey. *Institute of Information Systems University of Bern. http://www. ie. iwi. unibe. ch/publikationen/berichte/resource/WP-184. pdf*, 2006.

[71] D. Newman, A. Asuncion, P. Smyth, and M. Welling. Distributed algorithms for topic models. *The Journal of Machine Learning Research*, 10:1801–1828, 2009.

[72] T. Nguyen, K. Chang, and S. Hui. Supervised term weighting for sentiment analysis. In *Proceedings of the IEEE International Conference on Intelligence and Security Informatics: 9-12 July 2011; Beijing, China*. IEEE, 2011.

[73] P. Nougues and D. Brown. We know where you are going: tracking objects in terrain. *IMA Journal of Management Mathematics*, 8(1):39, 1997.

[74] C. Paciorek, J. Yanosky, R. Puett, F. Laden, and H. Suh. Practical large-scale spatio-temporal modeling of particulate matter concentrations. *Ann Appl Stat*, 3:369–396, 2009.

[75] J. Park and B. Seifert. Local additive estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(2):171–191, 2010.

[76] M. Park and T. Hastie. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):659–677, 2007.

[77] S. Perkins, K. Lacker, and J. Theiler. Grafting: Fast, incremental feature selection by gradient descent in function space. *The Journal of Machine Learning Research*, 3:1333–1356, 2003.

[78] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577. ACM, 2008.

[79] V. Punyakanok, D. Roth, and W.-t. Yih. The importance of syntactic parsing and inference in semantic role labeling. *Comput. Linguist.*, 34(2):257–287, 2008.

[80] Quantum GIS Development Team. *Quantum GIS Geographic Information System*. Open Source Geospatial Foundation, 2009.

[81] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[82] M. Redmond. Communities and crime data set, Last accessed: February 2012.

[83] C. Robertson, S. Geva, and R. Wolff. Can the content of public news be used to forecast abnormal stock market behaviour? In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 637–642. IEEE, 2007.

[84] A. Rodrigues, P. Diggle, and R. Assuncao. Semiparametric approach to point source modelling in epidemiology and criminology. *Journal of the Royal Statistical Society: Series C(Applied Statistics)*, 59(3):533–542, 2010.

[85] G. SALTON. Term Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[86] O. Schabenberger and C. Gotway. *Statistical methods for spatial data analysis.* CRC Press, 2005.

[87] R. Schumaker and H. Chen. Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Transactions on Information Systems (TOIS)*, 27(2):12, 2009.

[88] C. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):55, 2001.

[89] J. Shao. *Mathematical Statistics.* Springer, second edition, 2003.

[90] R. Shumway and D. Stoffer. *Time Series Analysis and Its Applications With R Examples* . Springer New York, second edition, 2006.

[91] A. Singhal. Modern information retrieval: A brief overview. *IEEE Data Engineering Bulletin*, 24(4):35–43, 2001.

[92] M. Smith and D. Brown. Discrete choice analysis of spatial attack sites. *Information Systems and E-Business Management*, 5(3):255–274, 2007.

[93] M. Steyvers and T. Griffiths. Matlab Topic Modeling Toolbox. `http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm`.

[94] M. Steyvers and T. Griffiths. Probabilistic topic models. *Latent Semantic Analysis: A Road to Meaning*, 2006.

[95] M. Steyvers and T. Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427, 2007.

[96] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[97] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[98] U.S. Census Bureau. Cartographic boundary files. `http://www.census.gov/geo/www/cob/co2000.html`, Last accessed: Jun 2010.

[99] V. Vapnik. *The nature of statistical learning theory.* Springer Verlag, 2000.

[100] X. Wang and D. Brown. The spatio-temporal generalized additive model for criminal incidents. In *Proceedings of the IEEE International Conference on Intelligence and Security Informatics: 9-12 July 2011; Beijing, China.* IEEE, 2011.

[101] X. Wang and D. Brown. The spatio-temporal modeling for criminal incidents. *Security Informatics*, 1, Feburary 2012.

[102] X. Wang, D. Brown, and M. Gerber. Spatio-temporal modeling of criminal incidents using geographic, demographic, and twitter-derived information. In *Proceedings of the IEEE International Conference on Intelligence and Security Informatics: 11-14 June 2012; Washington, D.C., USA*. IEEE, 2012.

[103] X. Wang, D. E. Brown, and J. H. Conklin. Crime incident association with consideration of narrative information. In *Systems and Information Engineering Design Symposium*, 2007.

[104] X. Wang, M. Gerber, and D. Brown. Automatic crime prediction using events extracted from Twitter posts. *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 231–238, 2012.

[105] Y. Wang, H. Bai, M. Stanton, W. Chen, and E. Chang. Plda: Parallel latent dirichlet allocation for large-scale applications. *Algorithmic Aspects in Information and Management*, pages 301–314, 2009.

[106] I. Witten, Z. Bray, M. Mahoui, and W. Teahan. Using language models for generic entity extraction. In *Proceedings of the ICML Workshop on Text Mining*. Citeseer, 1999.

[107] S. Wood. Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):95–114, 2003.

[108] S. Wood. Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association*, 99(467):673–686, 2004.

[109] S. Wood. *Generalized additive models: an introduction with R*, volume 66. CRC Press, 2006.

[110] S. Wood. Fast stable direct fitting and smoothness selection for generalized additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3):495–518, 2008.

[111] S. Wood. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2011.

[112] S. Wood. "mgcv" pakcage in R. `http://cran.r-project.org/web/packages/mgcv/index.html`, Last accessed: March, 2012.

[113] Q. Wu, C. Burges, K. Svore, and J. Gao. Adapting boosting for information retrieval measures. *Information Retrieval*, pages 1–17, 2009.

[114] Xuan-Hieu Phan and Cam-Tu Nguyen. Gibbslda++: A c/c++ implementation of latent dirichlet allocation (lda), 2007.

[115] Y. Xue and D. Brown. Spatial analysis with preference specification of latent decision makers for criminal event prediction. *Decision Support Systems*, 41(3):560–573, 2006.

[116] H. Yang, I. Spasic, J. Keane, and G. Nenadic. A Text Mining Approach to the Prediction of Disease Status from Clinical Discharge Summaries. *Journal of the American Medical Informatics Association*, 16(4):596–600, 2009.

[117] M. Yetisgen-Yildiz and W. Pratt. Using statistical and knowledge-based approaches for literature-based discovery. *Journal of Biomedical Informatics*, 39(6):600–611, 2006.

[118] J. Zhu, H. Huang, and P. Reyes. On selection of spatial linear models for lattice data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):389–402, 2010.