École polytechnique fédérale de Lausanne

Master thesis fall 2023

Master in Mathematics

Systematic Effects and Nuisance Parameters in Particle Physics Data Analyses

Author:
Dalil Koheeallee

Supervisor:
Prof. Anthony Davison

Contents

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Professor Anthony Davison, for suggesting this topic to me. I am thankful for the valuable time he spent discussing it with me, his invaluable guidance on conducting and writing mathematics, and for sharing his extensive knowledge. I consider myself fortunate to have been one of his students.

I am also deeply grateful to my family for their endless support and encouragement during my academic journey.

Lastly, I extend my gratitude to my friend François for his help on the subtleties of LaTeX.

Introduction

High-energy physics explores the subatomic realm, where particles interact at incredibly small scales and energies. High-energy physicists aim to gain insights into the behavior of fundamental particles and their properties. One of the most remarkable achievements in this field was the discovery of the Higgs boson, a particle that imparts mass to other particles. This discovery, made at the Large Hadron Collider at CERN, exemplifies the meticulous nature of this discipline, which involves the systematic examination of datasets produced by particle collisions. Researchers employ a diverse array of statistical, computational, and mathematical techniques to extract insights from these data. Through rigorous analysis, they endeavor to distinguish signal events, such as the production of the Higgs boson, from various sources of background noise. Such analyses can lead to significant advancements in our understanding, while also highlighting the inherent challenges of the domain.

This thesis centers on an analysis task inspired by the examination of $W\pm$ boson production rates at the Collider Detector at Fermilab (CDF). This analysis task employs a unique approach that uses two critical feature variables to discriminate between signal and background events. In this context, signal events refer to the desired $W\pm$ boson production, while background events encompass other particle interactions. The two pivotal feature variables in question are the missing transverse energy (MET) and the amount of energy contained within a cone around a lepton candidate (ISO), commonly referred to as "MET vs ISO". The primary objective of this paper is to construct 68% confidence intervals, accounting for the impact of unknown nuisance parameters, to precisely determine the signal rate, thereby enabling the effective discrimination of signals from background contributions.

While contemporary techniques for selecting $W\pm$ events at hadron colliders and estimating signal and background contributions have evolved considerably, this particular method offers the distinct advantage of consolidating all pertinent information within a single event sample, using only two feature variables. The distributions of these feature variables and their correlations are artificially generated for the purpose of this task, enabling rigorous evaluation and testing.

Chapter 1 of this document contains four sections. The first section serves as an introduction to the data under examination. Here, we present a comprehensive description of the dataset that will be central to our investigations. This includes a description of the datasets that will be used, the essential features they encompass,

some basic analyses, visualizations and transformations applied. The second section delves into the Poisson process, introducing key results that will be fundamental for our subsequent analyses. In the third section, we sketch generalized additive model (GAM) theory, elucidating essential principles and outcomes pivotal to our analytical framework. Topics covered include the formulation of GAMs and their utility in modeling intricate relationships between variables. It also includes the application of the theory using R. In the last part of this chapter, we shift our attention to the EM algorithm, which is a crucial idea in our study. We'll take a closer look at its basic theory to understand the core concepts that make it important in statistics and machine learning. To make things clearer, we'll also show how it works in a real-world applications.

Chapter 2 is dedicated to applying these tools to the transformed data. This segment primarily revolves around the development of the EM algorithm, aimed at distinguishing the signal from the background in unlabeled data. Additionally, it includes an evaluation of the suitability of fitting GAM to our data and assesses the effectiveness of the EM algorithm on labeled data, thereby testing the algorithm's overall performance.

Chapter 3 presents a thorough discussion of the results obtained, including an exploration of the subtle nuances revealed through the analysis. This chapter also speculates on potential improvements to the main findings, explores avenues for possible generalizations of the methods employed, and contemplates enhancements that could have been made with additional time.

Two appendices are included for supplementary analyses. Appendix A presents an in-depth exploration of the analyses closely mirroring those conducted in the core chapters. It is intended for readers interested in delving deeper into these analyses. Meanwhile, Appendix B provides a detailed discussion of confidence intervals, with a primary emphasis on their computation using Louis' Method.

Chapter 1

Data Description and Analytical Framework

1.1 Introduction to the Dataset

The dataset at our disposal comprises simulated individual collision events, with each event having measured values of (x_1, x_2). Two processes contribute to the collision data: a "signal" process representing $W\pm$ boson production and a "background" process that encompasses other particle interactions. In our analysis, we assume that the distributions of the feature variables for both the signal and background processes exhibit independence in the x_1 and x_2 variables. This assumption is expressed as

$$p_s(x_1, x_2) \approx f_s(x_1)g_s(x_2)$$

for the signal process and

$$p_b(x_1, x_2) \approx f_b(x_1)g_b(x_2)$$

for the background process. The principal challenge lies in the determination of the background production rate, which is unknown beforehand and must be estimated from the data. This rate is intentionally set to vary across different datasets.

The data are Monte Carlo samples categorized into two main types: background Monte Carlo and signal Monte Carlo. These samples simulate diverse probability distribution shapes for both the signal and background processes and are divided into two portions: the training set and the testing set. For our main analyses, we will specifically use the training set, which consists of a selection of these Monte Carlo simulations. The test portion of the document will explicitly detail the simulated datasets used for testing, which are composed of mixed samples from both signal and background collision simulations. The fraction of signal in the data varies across samples, ranging from no signal to a detectable signal. The simulated data are to be considered statistically and systematically independent of one another to avoid using any property of one simulated data sample to help interpret any other simulated data sample. Refer to Figure 1.1 for visual representations of the distributions in the dataset used for training.

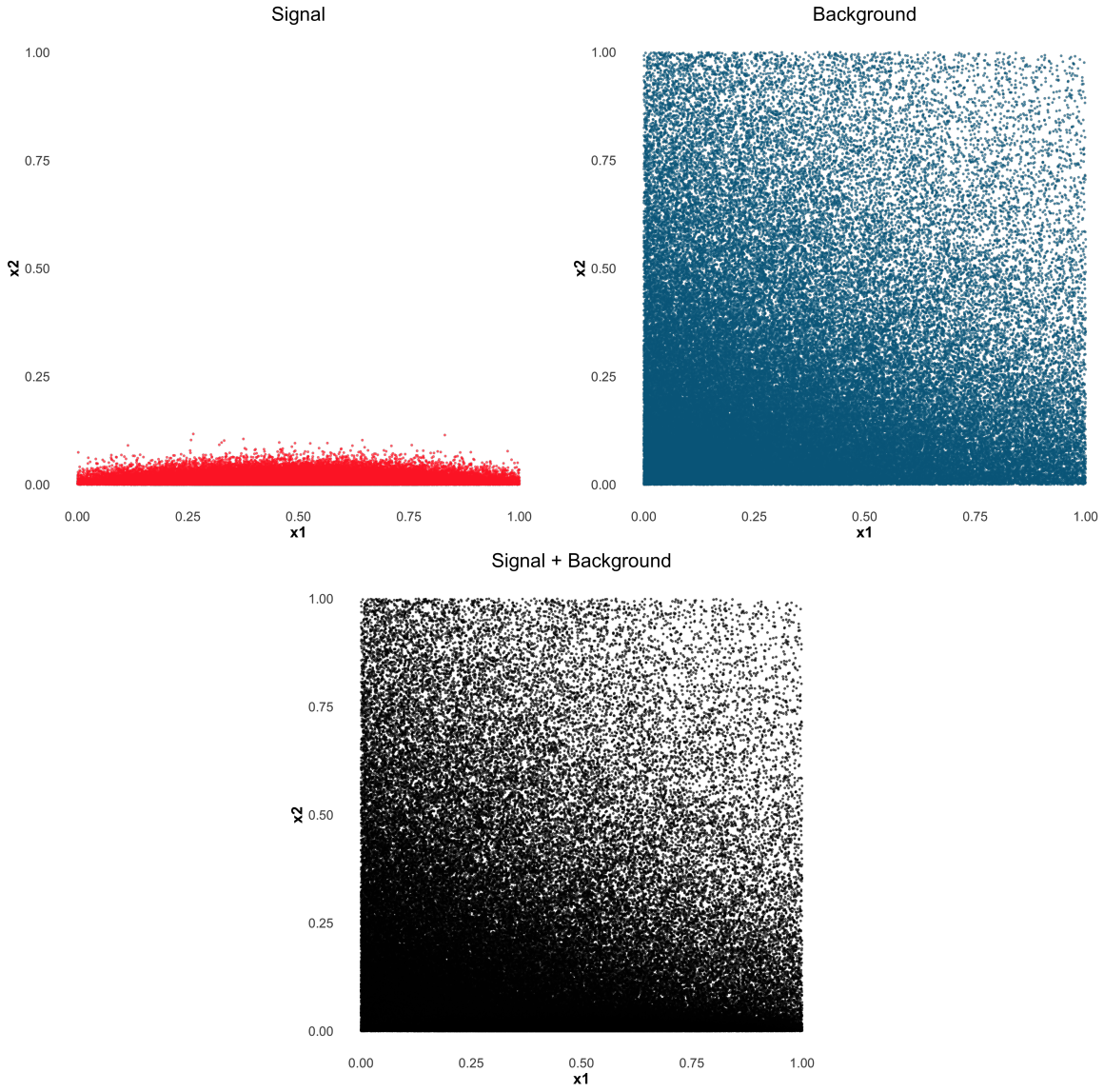

Figure 1.1: Distribution of signal, background and its mixture in the (x_1, x_2) feature plane for the training set.

The dataset consists of values for both the x_1 and x_2 coordinates ranging from 0 to 1. We treat the dataset by spatial discretization, dividing the data space into 10000 equally-sized squares. The discretization involves assigning data points to these squares based on their coordinates. The coordinates of the center of each square, denoted as (x_1, x_2), serve as reference points for grouping data points. By associating each data point with its corresponding square, we create a new data set with the coordinate of the center of the square and a count corresponding to the number of occurrences in that square. We examine the distribution of occurrences within individual squares in the training set in Table 1.1 and Table 1.2. The tables show distinct patterns in the occurrence distribution between the signal and background datasets and aid us in identifying critical features and understanding the underlying characteristics of the data, which are essential for our subsequent analysis. For the signal, we observe a high

number of squares with nearly no events but a few squares with a lot of events, so the distribution is concentrated on a smaller range of values for x and y. The distribution for the background is more even and more spread on the entire range of values. It is less concentrated, and there are almost no squares with a large concentration of events.

Number of events	Frequency
0	707
1 - 10	6544
11 - 20	1428
21 - 30	560
31 - 40	299
41 - 50	177
51 - 60	104
61 - 70	60
71 - 80	52
81 - 90	35
91 - 100	18
101 - 110	10
111 - 120	4
121 - 130	2

Table 1.1: Number of background events per square with their frequencies in the training set.

Number of events	Frequency
0	9290
1 - 100	507
101 - 200	75
201 - 300	22
301 - 400	21
401 - 500	29
501 - 600	8
601 - 700	2
701 - 800	8
801 - 900	5
901 - 1000	5
1001 - 1100	5
1101 - 1200	8
1201 - 1300	13
1301 - 1400	2

Table 1.2: Number of signal events per square with their frequencies in the training set.

1.2 Poisson Process

This section introduces the general theory of the Poisson process, which is a fundamental mathematical model. It helps us understand and model random events occurring over time or space.

1.2.1 Point process

A point process is a stochastic model representing a collection of points, denoted as P, distributed over a state space $\mathcal{E}$, where these points are identified as $\{x_1, x_2, \ldots\}$. While we often visualize P in a two-dimensional Euclidean space, it can actually be more complex. For instance, if $\mathcal{E}$ is defined as $\mathbb{R}_{>0} \times C$, where C is a suitable space of functions, each point takes the form $x = (u, f)$, with $u > 0$ and $f \in C$. Despite this, we refer to the elements of P as "points" in our general discussion. Irrespective of the specific choice for $\mathcal{E}$, it should be possible to determine how many points from P lie

within certain test sets A. In other words, we must be able to compute the number of points within A using the expression

$$N(A) = N(P \cap A) = \sum_x I(x \in P \cap A), \quad A \subset \mathcal{E}, \tag{1.1}$$

where $I(\cdot)$ is an indicator function. To ensure that this expression is unambiguous, it is necessary for two points in P not to coincide exactly; in such cases, P is referred to as "simple". To distinguish between its points, we assume that $\mathcal{E}$ has a topology that comprises a system of neighborhoods and open sets satisfying the Hausdorff property. In simpler terms, any two distinct points should have disjoint neighborhoods. Additionally, it is advantageous if $\mathcal{E}$ is locally compact, meaning that any point has a compact neighborhood and has a countable basis, indicating the existence of open sets $\{U_j : j \in \mathbb{N}\}$ such that any open set can be represented as a finite or countable union of the U_j.

To facilitate mathematical developments, we assume that $\mathcal{E}$ is a measurable space, implying that certain subsets of $\mathcal{E}$ are measurable, and these sets together form a σ-algebra. This σ-algebra has the following three properties:

1. The empty set $\varnothing$ is measurable.

2. Complements of measurable sets are measurable.

3. Any union of countably many measurable sets is measurable.

A measure μ is a non-negative function defined on measurable sets in $\mathcal{E}$ such that $\mu(\varnothing) = 0$, and if $\{A_j\}_{j=1}^{\infty}$ are disjoint subsets of $\mathcal{E}$, then the measure of their union is equal to the sum of their individual measures, that is

$$\mu\left(\bigcup_{j=1}^{\infty} A_j\right) = \sum_{j=1}^{\infty} \mu(A_j).$$

From now on, we will assume that subsets of $\mathcal{E}$ and functions defined on them are measurable unless explicitly stated otherwise.

The function $N(\cdot)$ defined in (1.1) is essentially a counting measure, as it enumerates the number of elements of P within any set A. It is referred to as a Radon measure if $N(A) < \infty$ for any compact set A, and it becomes a random measure if the points of P occur stochastically, making $N(A)$ a random variable.

Another noteworthy property of point processes is the Laplace functional $\mathcal{L}_P(\cdot)$, which can be likened to the moment-generating function of a random variable. For a function $\zeta > 0$ that is zero outside a bounded subset of $\mathcal{E}$, we define

$$\mathcal{L}_P(\zeta) = \mathbb{E}\left[\exp\left(-\int \zeta \, \mathrm{d}P\right)\right],$$

where $\mathbb{E}(\cdot)$ denotes the expectation. Here, the integral term is given by

$$\int \zeta \, \mathrm{d}P = \int \zeta(x) P(\mathrm{d}x) = \sum_j \zeta(x_j).$$

If we set $\zeta(x) = \sum_j t_j I(x \in A_j)$ for $t_j \geq 0$, then $\mathcal{L}_p(\zeta)$ is the joint moment-generating function of $N(A_1), \ldots, N(A_k)$ for any positive integer k and sets $A_1, \ldots, A_k$ and therefore summarizes the joint distribution

$$\Pr\{N(A_1) = n_1, \ldots, N(A_k) = n_k\}, \quad n_1, \ldots, n_k \in \{0, 1, 2, \ldots\}.$$

1.2.2 Poisson process and applications

We now focus on the Poisson point process. This is characterized by the Poisson distribution, whose probability mass function (PMF) is defined as

$$\Pr(X = k) = \frac{e^{-\lambda}\lambda^k}{k!}, \quad k \in \{0, 1, 2, \ldots\}, \quad \lambda > 0.$$

A Poisson process on $\mathcal{E}$ is a random countable subset P of $\mathcal{E}$ such that:

1. The random variables $N(A_1), \ldots, N(A_k)$ corresponding to any collection of disjoint subsets $A_1, \ldots, A_k$ of $\mathcal{E}$ are independent.

2. For any $A \subset \mathcal{E}$, $N(A)$ has the Poisson distribution with mean $\mu(A)$, where $0 \leq \mu(A) \leq \infty$, and $\mu(A)$ is finite for any compact A.

If A is a union of disjoint sets $A_1, A_2, \ldots$, then, since $N(A) = \sum_j N(A_j)$, the function $\mu(\cdot)$ defines a measure often called the mean measure of the Poisson process. This measure must be diffuse, i.e., $\mu(\{x\}) = 0$ for every $x \in \mathcal{E}$. For if not, then for some x, $m = \mu(\{x\}) > 0$, so the event $N(\{x\}) > 2$ has positive probability $1 - e^{-m} - me^{-m}$, and points of P could coincide, making $N(A)$ not well-defined. Figure 1.2 shows a Poisson process in $\mathbb{R}^2$ to highlight the properties of this process.

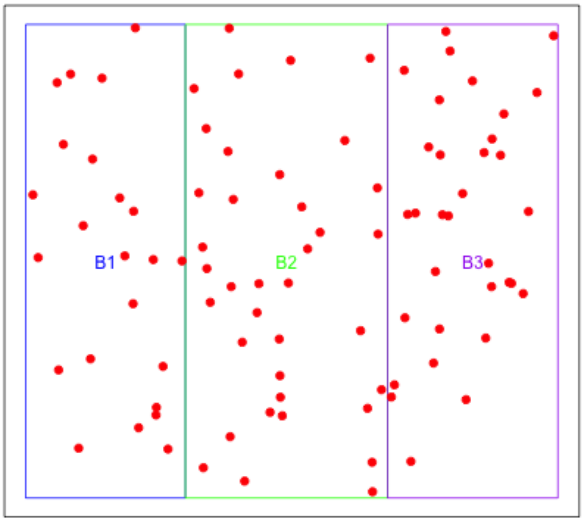

Figure 1.2: Poisson process on $\mathbb{R}^2$. Each box is disjoint with $N(B_k)$ independent for $k \in \{1, 2, 3\}$.

In many cases, $\mathcal{E} \subset \mathbb{R}^D$, and $A = [a_1, b_1] \times \ldots \times [a_D, b_D]$, and then if

$$\dot{\mu}(b_1, \ldots, b_D) = \frac{\partial^D \mu(A)}{\partial b_1 \partial b_2 \cdots \partial b_D}$$

exists and is finite, it is called the intensity function of P, necessarily, $\dot{\mu}(\cdot) \geq 0$. The Poisson process is said to be homogeneous if $\dot{\mu}(\cdot) = \dot{\mu}$, and otherwise, it is said to be *inhomogeneous*. The two main results of this section are the following propositions.

Proposition 1.2.1 (Conditioning)**.** *Let P be a Poisson process with mean measure μ, and suppose that $A \subset \mathcal{E}$ is such that $0 < \mu(A) < \infty$. Conditional on the event $N(A) = n$, the n points of P lying in A have the distribution of n points generated independently at random in A according to the measure $\mu_A(B) = \mu(B)/\mu(A)$, for $B \subset A$.*

Proof. To have the general intuition, we focus on the situation where $\mathcal{E} = \mathbb{R}_{>0}$. The aim is to compute the density corresponding to observing n events at times $x_1, \ldots, x_n$ conditional on the event $N(A) = n$. We first compute the density functions of the first arrival X_1. Let $N(x)$ be the number of arrivals up to time x included. We have

$$\Pr(X_1 > x) = \Pr\{N(x) = 0\} = \exp\{-\mu(x)\}, \quad \mu(x) = \int_0^x \dot{\mu}(u)\mathrm{d}u, \quad x > 0.$$

Hence, the probability density function of the first arrival X_1 is simply

$$f_{X_1}(x) = -\frac{d}{dr} \Pr(X_1 > x) = \dot{\mu}(x) \exp\{-\mu(x)\}, \quad x > 0.$$

Now suppose that we have $n > 0$ occurrences $x_1, \ldots, x_n$ until time t. The events are independent as they occur in disjoint intervals $(x_k, x_{k+1}]$ and there are no arrivals in $(x_n, t]$. The joint density is

$$\dot{\mu}(x_1) \exp\left\{-\int_0^{x_1} \dot{\mu}(u)\ \mathrm{d}u\right\} \times \prod_{j=2}^{n} \dot{\mu}(x_j) \exp\left\{-\int_{x_{j-1}}^{x_j} \dot{\mu}(u)\ \mathrm{d}u\right\} \times \exp\left\{-\int_{x_n}^{t} \dot{\mu}(u)\ \mathrm{d}u\right\}.$$

The joint distribution is

$$\exp\{-\mu(A)\} \times \prod_{j=2}^{n} \dot{\mu}(x_j).$$

Using the fact that $N(A)$ is Poisson with mean $\mu(A)$, the probability of n events is $\mu(A)^n \exp\{-\mu(A)\}/n!$. Using Bayes' theorem, we find that the density corresponding to observing events at $x_1, \ldots, x_n$ conditional of the event $N(A) = n$ is

$$n! \prod_{j=1}^{n} \frac{\dot{\mu}(x_j)}{\mu(A)}, \quad x_1, \ldots, x_n \in A.$$

This equals the joint density for n independent random variables each with probability density function $\dot{\mu}(x)/\mu(A)$ for $x \in A$, without regard to their labelling. This concludes the proof.

□

Proposition 1.2.2 (Superposition)**.** *If P_1 and P_2 are independent Poisson processes on $\mathbb{R}^D$ with mean measures μ_1 and μ_2, respectively, then their union $P_1 \cup P_2$ is Poisson with mean measure $\mu_1 + \mu_2$.*

Proof. For this result, we will use the Laplace functional for Poisson process. Take ζ a non-negative function positive on a bounded set A with $\mu(A) < \infty$. Conditional on $N(A) = n$, $\int \zeta(x)P(\mathrm{d}x) = \sum_{j=1}^{n} \zeta(X_j)$, where $\{X_1, \ldots, X_n\} \subset A$ are independent with density $\dot{\mu}(x)/\mu(A)$. The conditional independence of $X_1, \ldots, X_n$ given $N(A)$ shows that

$$\begin{aligned}\mathbb{E}\left[\exp\left\{-\int \zeta(x)P(\mathrm{d}x)\right\}\middle| N(A) = n\right] &= \mathbb{E}\left[\exp\left\{-\sum_{j=1}^{n}\zeta(X_j)\right\}\middle| N(A) = n\right] \\ &= \left\{\int_A \exp(-\zeta(x))\mu(\mathrm{d}x)/\mu(A)\right\}^n.\end{aligned}$$

It follows that

$$\begin{aligned}\mathbb{E}\left[\exp\left\{-\int \zeta(x)P(\mathrm{d}x)\right\}\right] &= \sum_{n=0}^{\infty}\left\{\int_A \exp(-\zeta(x))\mu(\mathrm{d}x)/\mu(A)\right\}^n \frac{\mu(A)^n}{n!}\exp(-\mu(A)) \\ &= \exp\left[\int_A \exp(-\zeta(x))\mu(\mathrm{d}x) - \mu(A)\right] \\ &= \exp\left[-\int_A \{1-\exp(-\zeta(x))\}\,\mu(\mathrm{d}x)\right] \\ &= \exp\left[-\int_{\mathcal{E}} \{1-\exp(-\zeta(x))\}\,\mu(\mathrm{d}x)\right],\end{aligned}$$

since $\zeta(x)$ is 0 outside A. We hence found that the Laplace functional of a Poisson process P on $\mathcal{E}$ with mean measure μ is

$$\mathcal{L}_P(\zeta) = \exp\left[-\int_{\mathcal{E}} \{1-\exp(-\zeta(x))\}\,\mu(\mathrm{d}x)\right].$$

Now that the Laplace functional is established, we need to state that the two processes are disjoint. We will use the result without providing a detailed proof in this context. For those interested in a comprehensive proof, we recommend referring to page 15 of Kingman (1993). We can now prove the theorem. The two processes are independent and disjoint, hence

$$\int \zeta(x)(P_1 \cup P_2)(\mathrm{d}x) = \int \zeta(x)P_1(\mathrm{d}x) + \int \zeta(x)P_2(\mathrm{d}x).$$

The Laplace functional for $P_1 \cup P_2$ is

$$\begin{aligned}
\mathcal{L}_{P_1 \cup P_2}(\zeta) &= \mathbb{E}\left[\exp\left\{-\int \zeta(x)(P_1 \cup P_2)(\mathrm{d}x)\right\}\right] \\
&= \mathbb{E}\left[\exp\left\{-\int \zeta(\mathrm{d}P_1)\right\}\right] \times \mathbb{E}\left[\exp\left\{-\int \zeta(\mathrm{d}P_2)\right\}\right] \\
&= \exp\left[-\int_{\mathcal{E}} \{1-\exp(-\zeta)\}\,\mathrm{d}\mu_1\right] \times \exp\left[-\int_{\mathcal{E}} \{1-\exp(-\zeta)\}\,\mathrm{d}\mu_1\right] \\
&= \exp\left[-\int_{\mathcal{E}} \{1-\exp(-\zeta)\}\,\mathrm{d}(\mu_1+\mu_2)\right],
\end{aligned}$$

which corresponds to a Poisson process with mean measure $\mu_1 + \mu_2$. □

1.3 Generalized Additive Models

In this section, we will introduce the generalized additive model (GAM). We will lay the foundational concepts leading to GAM without delving too deeply into the details. This section is based on Wood (2017).

Now, let's explore one of the essential building blocks upon which GAMs are constructed: the generalized linear model (GLM).

1.3.1 Generalized linear models (GLM)

In this section, we will only cover the basic framework for GLMs. We will explain more completely the next section about GAM models. Generalized linear models (Nelder and Wedderburn, 1972) extend the scope of linear models by accommodating response distributions beyond the normal distribution and enabling a degree of non-linearity in the model's structure. Given n observations (x_i, y_i) with y_i an observation on random variable Y_i, the basic structure is

$$g(\mu_i) = \alpha + x_i^\top \beta,$$

where $\mu_i \equiv \mathbb{E}(Y_i)$, g is a smooth monotonic "link function" and α and β are the unknown parameters. A central notion for GLM is the exponential dispersion family because a GLM in general makes the assumptions that the Y_i are independent and follow some exponential dispersion distribution. As a reminder, the probability density function given a parameter θ is in the form

$$f_\theta(y) = \exp[\{y\theta - b(\theta)\}/a(\varphi) + c(y, \varphi)],$$

where b, a and c are specified functions determined by the distribution and φ is an arbitrary "scale" parameter. If φ is known, this is an exponential family with canonical parameter θ.

The link function relates the linear predictor to the expected value. In basic linear models, the linear predictor and the expected value are identical. The identity link is

also used when the expected value and the predictor can be any value. However, when we are dealing with counts and the distribution is Poisson, we must have $\mu > 0$, so the identity link is less attractive because μ may then be negative. For Poisson, the log link is generally used.

Many of the general ideas and concepts of linear modelling generalize to this new model with little modification. Model fitting necessitates an iterative approach, and the distributional outcomes used for making inferences are now approximations. These approximations are justified based on large-sample limiting results, instead of exact solutions.

The maximum likelihood estimation of the parameters α and β in the linear predictor can be obtained by using iterative weighted least squares (IWLS). In this process, the dependent variable is $g(y)$ and the weights are function of the fitted values. The process is iterative because both the adjusted dependent variable $g(y)$ and the weight on the fitted values for which only current estimates are available. See McCullagh and Nelder (1983, p. 31) for more details on the procedure.

While GLMs are powerful tools for statistical analysis, they come with several limitations and challenges that may necessitate alternative approaches:

1. **Large Sample Requirement:** GLMs often require large sample sizes for accurate parameter estimation. Maximum likelihood estimation (MLE), while asymptotically efficient, can be unreliable with smaller datasets, leading to increased variance in the parameter estimates.

2. **Data Separation in Logistic Regression:** A specific challenge in logistic regression, a variant of GLM, is data separation. This issue arises when outcomes can be perfectly predicted by the predictors, causing the MLE to become ineffective and leading to infinite parameter estimates.

3. **Complex Interpretation:** The non-linear link functions in GLMs complicate the interpretation of model coefficients, especially when compared to the straightforward relationships in linear models. This non-linearity can obscure the understanding of the relationships between variables.

4. **Assumptions and Model Fit:** The performance of GLMs hinges on the assumption that the response variable follows a specific distribution from the exponential dispersion family. Deviations from this assumption, or an inappropriate choice of link function, can lead to poor model fit and inaccurate results.

5. **Computational Intensity:** The iterative algorithms used for parameter estimation in GLMs add significant computational cost. This can be particularly challenging for complex models or large datasets.

Given these considerations, particularly the need for a more flexible approach in modeling non-linear relationships and handling complex data structures, it becomes

evident that an extension of the GLM framework could be advantageous. This leads us to the study of generalized additive models, which build upon the foundation of GLMs while offering greater flexibility and capability in handling the intricacies of real-world data.

1.3.2 GAM theory

A generalized additive model (GAM) is an extension of a GLM. In a GAM, the linear predictor is a sum of smooth functions of covariates. If we take $\mu_i \equiv \mathbb{E}(Y_i)$ and each response variable Y_i follows an exponential dispersion family distribution with mean μ_i, θ the parameter vector, φ the scale parameter, the model has the structure

$$g(\mu_i) = A_i\theta + f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{1i}, x_{2i}) + \dots,$$

where A_i is a row of the model matrix for the parametric model components and f_j are smooth functions of the covariates. We'll now show how to represent a GAM using basis expansions for each smooth component. Each smooth component has its own penalty, which controls how "smooth" it should be. We estimate these models by using penalized regression methods, and we figure out the right level of smoothness for each component by either doing cross-validation or maximizing the marginal likelihood. Let's consider at first a model containing one function of one covariate,

$$y_i = f(x_i) + \epsilon_i, \tag{1.2}$$

where $y_i \overset{\text{ind}}{\sim} N(f(x_i), \sigma^2)$ is a response variable, x_i a covariate, f a smooth function and ϵ_i are independent $N(0, \sigma^2)$ random variables. We can now try to estimate the function f with basis expansions in order for Equation (1.2) to be expressed as a linear model. For this, we need to choose a basis defining the space of functions which will approximate f. If we take $b_j(x)$ to be the j^{th} basis functions and β_j to be the corresponding unknown parameter, we have the representation

$$f(x) = \sum_{j=1}^{k} b_j(x)\beta_j,$$

which clearly makes (1.2) linear.

A well known basis is the polynomial basis. Taylor's theorem suggests that polynomial bases are beneficial when you want to analyze the behavior of a function near a particular point. However, when the entire range of the function is involved, polynomial basis may be unsuitable. We can compare it to another basis called the smoothing spline basis. Instead of interpolating the data (x_i, y_i), we try to smooth them. For g representing the estimate of f, rather than setting $g(x_i) = y_i$, we treat $g(x_i)$ as n free parameters of the cubic spline and we aim to minimize

$$\sum_{i=1}^{n} \{y_i - g(x_i)\}^2 + \lambda \int g''(x)^2 \mathrm{d}x, \tag{1.3}$$

where λ is used to control the relative weight to be given to the conflicting goals of matching the data and producing a smooth g. The estimate g is a smoothing spline. The function g is also the unique function continuous on $[x_1, x_n]$ with absolutely continuous first derivatives minimizing (1.3).

Here is an example to illustrate the problems with polynomial basis in comparison to a simple basis for a cubic regression spline defined by $b_1(x) = 1, b_2(x) = x, b_{j+2}(x) = |x - x_j^*|^3$ for $j = 1, \ldots, q-2$ where q is the basis dimension and x_j^* are knot locations. We use this basis to fit a rank 11 cubic regression spline and we also fit 5^{th} order polynomials to the simulated data. It is obvious from Figure 1.3 that these basic polynomial regression models are inadequate for accurately capturing the underlying patterns in the data. In contrast, the bottom graph, which depicts the cubic regression spline, provides a much better fit.

Cubic spline basis naturally arises from the smoothing objective, providing a basis-independent way to define model fit and smoothness. Smoothing splines are great at creating smooth curves. The main challenge is that they have as many free parameters as there are data points, which can be wasteful since we often prefer smoother results. In practice, the penalty in the model often suppresses many degrees of freedom, making the large number of parameters less problematic for univariate smoothing with cubic splines. However, when dealing with multiple covariates, it can lead to significant computational complexity and expense. Using penalized regression splines represents a balanced approach that maintains the beneficial attributes of splines while also ensuring computational efficiency.

Penalized regression splines involve constructing a spline basis and associated penalties. This construction is typically done using a smaller dataset that adequately covers the distribution of covariate values in the original dataset. The choice of how many basis functions to use in this process depends on the specific problem under consideration, influenced by the balance between model complexity and the amount of data available. Typically, we need to allow the basis dimension to scale with the sample size, denoted as n, as n goes to infinity.

In this report, we use thin plate splines as formulated by Duchon (1977). The objective is to estimate a smooth function $g(x)$ from n data points (x_i, y_i), where

$$y_i = g(x_i) + \epsilon_i,$$

with ϵ_i random error and x being a d-dimensional vector. The thin plate spline approach is to estimate g by optimizing the function $\hat{f}$, which minimizes

$$\|y - f\|^2 + \lambda J_{md}(f),$$

where y denotes the data vector y_i and $f = [f(x_1), f(x_2), \ldots, f(x_n)]^T$. The term $J_{md}(f)$ is a penalization functional quantifying the function's "wiggliness", with λ being a smoothing parameter balancing data fit and smoothness of f to prevent overfitting.

Figure 1.3: The first graph shows a regression polynomial of degree 5, the second graph represents a regression polynomial of degree 10 and the bottom graph corresponds to cubic spline regression. The simulated data are a set of random data, (x, y) with 40 data points taken from a rescaled uniform distribution.

The wiggliness penalty is defined as

$$J_{md} = \int_{\mathbb{R}^d} \sum_{\nu_1+\cdots+\nu_d=m} \frac{m!}{\nu_1!\cdots\nu_d!} \left(\frac{\partial^m f}{\partial x_1^{\nu_1} \cdots \partial x_d^{\nu_d}} \right)^2 \mathrm{d}x_1 \cdots \mathrm{d}x_d.$$

Typically, m is chosen such that $2m > d$, and for visually smooth results, $2m > d+1$ is preferred. When $2m > d$, the minimization function takes the form

$$f(x) = \sum_{i=1}^{n} \delta_i \, \eta_{md}(\|x - x_i\|) + \sum_{j=1}^{M} \alpha_j \varphi_j(x), \tag{1.4}$$

with

$$M = \begin{pmatrix} m+d-1 \\ d \end{pmatrix},$$

and δ and α as coefficient vectors to be estimated, subject to $T^T\delta = 0$, where $T_{ij} = \varphi_j(x_i)$. The M functions φ_i are linearly independent polynomials spanning the space of polynomials in $\mathbb{R}^d$ of degree less than m and represent the "null space" of J_{md}. For example, in the case of $m = d = 2$, these functions are $\varphi_1(x) = 1$, $\varphi_2(x) = x_1$, and $\varphi_3(x) = x_2$. The basis functions in Equation (1.4) are then

$$\eta_{md}(r) = \begin{cases} \frac{(-1)^{m+1+d/2}}{2^{2m-1}\pi^{d/2}(m-1)!(m-d/2)!} r^{2m-d} \log r, & d \text{ even}, \\ \frac{\Gamma(d/2-m)}{2^{2m}\pi^{d/2}(m-1)!} r^{2m-d}, & d \text{ odd}. \end{cases}$$

Defining matrix $\mathbf{E}$ as $\mathbf{E}_{ij} = \eta_{md}(\|x_i - x_j\|)$, the thin plate spline fitting problem is then to

$$\text{minimize } \|y - \mathbf{E}\delta - \mathbf{T}\alpha\|^2 + \lambda^\top \mathbf{E}\delta, \quad \text{subject to } \mathbf{T}^\top\delta = 0,$$

for δ and α. This method precisely defines smoothness, balances the goals of data matching and smoothness in f, and identifies the optimal function for the smoothing objective. It eliminates the need to specify knot positions or select basis functions, as these are determined by the mathematical framework of the smoothing problem. Thin plate splines are versatile, allowing for multiple predictor variables and selection of derivative order to quantify wiggliness. The main challenge with thin plate splines is their computational demand, as the number of parameters equals the number of data points, leading to cubic scaling of computational costs with parameter quantity. Given that effective degrees of freedom for a model term are usually much lower than n, using many parameters might seem inefficient, raising the question of whether a low rank approximation could maintain smoothness without excessive computational burden. Thin plate regression splines (TPRS) address this by reducing the space of wiggly components of the thin plate spline, while keeping the "zero wiggliness" components intact. Let $\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{U}^\top$ be the eigen-decomposition of $\mathbf{E}$, where $\mathbf{D}$ is a diagonal matrix of eigenvalues and $\mathbf{U}$ the corresponding eigenvectors. Restricting to the column space of $\mathbf{U}_k$, with $\delta = \mathbf{U}_k\delta_k$, the rank k approximation problem becomes

$$\text{minimize } \|y - \mathbf{U}_k\mathbf{D}_k\delta_k - \mathbf{T}\alpha\|^2 + \lambda^\top \mathbf{D}_k\delta_k, \text{ subject to } \mathbf{T}^\top\mathbf{U}_k\delta_k = 0$$

for δ_k and α. An orthogonal column basis $\mathbf{Z}_k$ is found such that $\mathbf{T}^\top \mathbf{U}_k \mathbf{Z}_k = 0$, and by restricting δ_k to this space ($\delta_k = \mathbf{Z}_k \delta'$), we solve the unconstrained problem

$$\text{minimize } \|y - \mathbf{U}_k \mathbf{D}_k \mathbf{Z}_k \delta' - \mathbf{T}\alpha\|^2 + \lambda^\top \mathbf{D}_k \mathbf{Z}_k \delta'$$

for δ' and α, with a computational cost of $O(k^3)$. Post-fitting, evaluating the spline at any point is straightforward using (1.4). In Figure 1.4, we compare thin plate spline regression based on rank on our background data, using only the covariate x_2. Visually, rank nine seems to fit our data best, with further analysis confirming nine as the optimal rank for this specific regression.

The main computational problem lies in efficiently determining $\mathbf{U}_k$ and $\mathbf{D}_k$. Conducting a complete eigen-decomposition of $\mathbf{E}$ demands $O(n^3)$ operations, potentially restricting the practicality of the TPRS method. However, this issue can be mitigated by using Lanczos iteration, which allows for the calculation of $\mathbf{U}_k$ and $\mathbf{D}_k$ at a significantly reduced computational expense of $O(n^2 k)$ operations.

Lanczos iteration, as described in Demmel (1997), can be employed to achieve a rank k truncated eigen-decomposition of a symmetric matrix $\mathbf{E}$. This method requires $O(kn^2)$ operations and operates by iteratively constructing a tri-diagonal matrix, the eigenvalues of which converge to the needed values.

In each iteration, the algorithm generates a symmetric tri-diagonal matrix of dimensions $i \times i$. The eigenvalues of this matrix progressively approximate the i eigenvalues of the largest magnitude from the original matrix, with convergence typically occurring first for the largest ones. Additionally, the eigenvectors of the original matrix are obtained from those of the iterative matrix $\mathbf{K}_i$.

A complete version of the algorithm taken from McLachlan and Krishnan (2007, p. 426), suitable for finding the truncated decomposition of $\mathbf{E}$, is as follows:

1. Let b be an arbitrary non-zero n vector.
2. Set $q_1 = b/\|b\|$.
3. Repeat steps (4) to (12) for $j = 1, 2, \ldots$ until enough eigenvectors have converged.
4. Form $c = \mathbf{E} q_j$.
5. Calculate $\gamma_j = q_j^\top c$.
6. Reorthogonalize c to ensure numerical stability, by performing the following step twice:
$$c \leftarrow c - \sum_{i=1}^{j-1} (c^\top q_i) q_i.$$
7. Set $\xi_j \leftarrow \|c\|$.
8. Set $q_{j+1} \leftarrow c/\xi_j$.

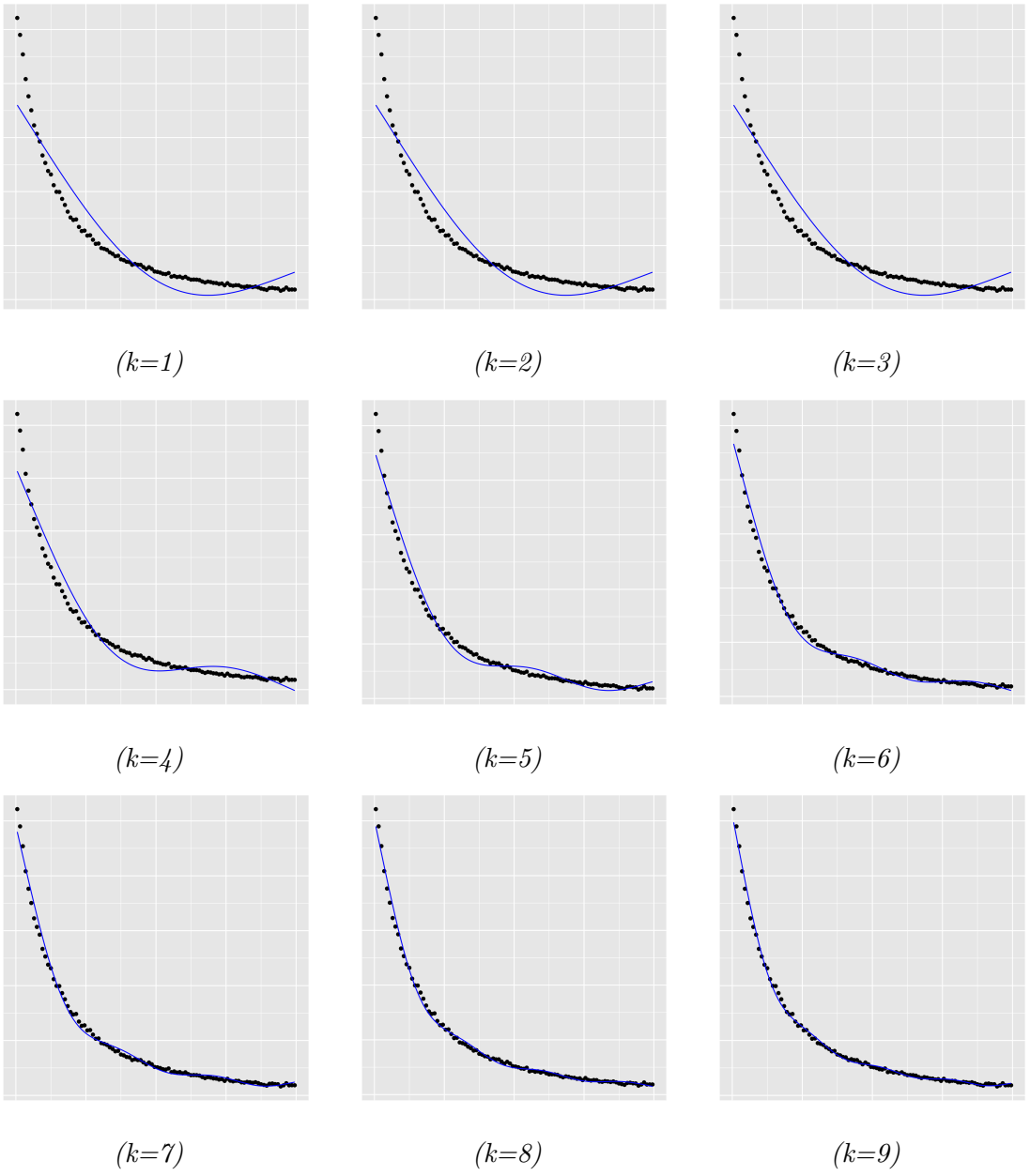

Figure 1.4: Illustration of rank one to rank nine thin plate regression spline basis for representing a smooth function of one variable. The black dots are the background data points, the blue curve is the smooth curve.

9. Let $\mathbf{K_j}$ be the $(j \times j)$ tri-diagonal matrix with $\gamma_1, \ldots, \gamma_j$ on the leading diagonal, and $\xi_1, \ldots, \xi_{j-1}$ on the leading sub- and super-diagonals.

10. If iteration has proceeded far enough to make it worthwhile, find the eigen-decomposition (spectral decomposition) $\mathbf{K_j} = \mathbf{V}\Lambda\mathbf{V}^\top$, where the columns of $\mathbf{V}$ are eigenvectors of $\mathbf{K_j}$ and Λ is diagonal with eigenvalues on leading diagonal.

11. Compute "error bounds" for each $\Delta\lambda_i : |\xi_j \mathbf{V}_{j,i}|$.

12. Use the error bounds to test for convergence of the k largest magnitude eigenvalues. Terminate the loop if all are converged.

13. The i-th eigenvalue of $\mathbf{E}$ is λ_i. The i-th eigenvector of $\mathbf{E}$ is $\mathbf{Q}v_i$, where $\mathbf{Q}$ is the matrix whose columns are the q_i (for all j calculated) and v_i is the i-th column of $\mathbf{V}$ (again calculated at the final iteration). Hence D_k and U_k can easily be formed.

The major computational burden arises from the $O(n^2)$ operation: $c \leftarrow \mathbf{E}q_j$, implying that the advantages of using a selective approach are expected to be minimal.

We will now briefly discuss smoothness selection. The estimation of the smoothing parameter λ constitutes the most challenging part of model estimation. We will generally use two methods, the known scale parameter method with the unbiased risk estimator (UBRE), and generalized cross validation (GCV). The Akaike's information criterion (AIC) will be used for model selection.

Consider estimating smoothing parameters in a basic additive model for data with a consistent and known variance σ^2. A desirable strategy is to aim for the estimated mean, denoted by $\hat{\mu}$, to closely approximate the true mean, μ. One way to measure this closeness is using the expected mean square error (with the Euclidean norm $\|\cdot\|$)

$$M = \mathbb{E}\left(\frac{\|\mu - X\hat{\beta}\|^2}{n}\right) = \frac{\mathbb{E}\left(\|y - Ay\|^2\right)}{n} - \sigma^2 + \frac{2\mathrm{tr}(A)\sigma^2}{n}.$$

It seems reasonable to select smoothing parameters in order to minimize an estimate of M,

$$V_u(\lambda) = \frac{\|y - Ay\|^2}{n} - \sigma^2 + \frac{2\mathrm{tr}(A)\sigma^2}{n},$$

which is the unbiased risk estimator UBRE. UBRE is essentially a generalization of the concept of M. The generalized cross validation score,

$$V_g = \frac{n\|y - \hat{\mu}\|^2}{(n - \mathrm{tr}(A))^2},$$

is a modification of the ordinary cross validation (OCV) but invariant to any orthogonal rotation of $y - X\beta$.

Akaike's Information Criterion (AIC) is a number used for model selection, where models are chosen to minimize an estimate of the expected Kullback Leibler divergence

between the model that has been fitted and the actual, underlying true model. The AIC serves as a criterion for this selection process. The criterion is

$$\text{AIC} = -2l + 2p,$$

with l the maximized log likelihood of the model and p the number of model parameters estimated. We select the model with the lowest AIC.

1.3.3 GAMs in Practice

This section focuses on the application of generalized additive modeling functions, with a particular emphasis on the *R* package `mgcv`. Although the package offers several modeling functions suitable for specific scenarios, such as handling large datasets, our main focus will be on the `gam` functions. The inclusion of this section is crucial, as various expressions and terms from the `mgcv` package will be used throughout the document, making familiarity with these concepts essential for understanding the subsequent content. The package provides three types of smooth functions, which can be employed individually or in combination:

1. `s()` is used for single variable smooths and for random effects.

2. `ti()` is used to specify interactions between marginal smooths using tensor products.

3. `te()` is used to specify tensor product smooths constructed from any single penalized marginal smooths usable with `s()`.

The first argument for all of these functions is the covariate for the smooth. Additional arguments are used to control specific aspects of the smoothing process, with the following being the most crucial:

1. **bs**: A character that specifies the type of basis. The default type of basis is the thin plate spline.

2. **k**: The basis dimension or marginal basis dimension (in the tensor case). **k** can also be a vector in the tensor case to specify a dimension for each marginal component.

3. **family** : Specifies the family distribution with a possible link. In this document, we will use the Poisson family with the log link.

4. **sp**: This parameter specifies the smoothing parameter, which controls the trade-off between smoothness and fit to the data. A higher value of **sp** leads to a smoother function, while a lower value allows for more flexibility in fitting the data.

In the context of `mgcv` estimation functions, a smooth, whether a regular smooth or a random effect, can be understood as a combination of model matrix columns and one or more associated penalties. This is exploited in the design of the code to ensure that the implementation of smooths is highly modular. Within the code, smooths are implemented through a smooth constructor method function. Additionally, each smooth is equipped with a prediction matrix method function, which generates a matrix facilitating the transformation of smooth coefficients into predictions for new covariate values. In GAMs, each predictor's effect is modeled using a smooth function, which is represented by a basis function, and the degree of smoothness is controlled by the smoothness parameter. The choice of the basis function can affect the model, with thin plate regression splines being the default choice in mgcv.

The package automatically estimates the degree of smoothness for the smooth functions using criteria like GCV. It reports the estimated degrees of freedom for each smooth term, which reflects the flexibility of the model in capturing the data patterns. For more details, refer to Wood (2017).

1.4 EM algorithm

This section on the Expectation-Maximization (EM) algorithm is based on work of McLachlan and Krishnan (2007). The EM algorithm is a versatile and widely employed method for iteratively computing maximum likelihood (ML) estimates. It is particularly useful in dealing with various incomplete-data problems, where other optimization methods, such as the Newton–Raphson method, may become computationally complex. The EM algorithm consists of two essential steps in each iteration: the "Expectation" step (E-step) and the "Maximization" step (M-step). The situations where the EM algorithm can be applied to include incomplete-data situations, where there are missing data, truncated distributions, or grouped observations, but also in a whole variety of situations where the incompleteness of the data is not obvious.

1.4.1 Maximum likelihood estimation

Maximum likelihood estimation (MLE) plays a central role in this algorithm. As a reminder, we let X be a p-dimensional random vector with probability density function $g(x;\theta)$ on $\mathbb{R}^p$ where $\theta = (\theta_1, \ldots, \theta_d)^\top$ is the vector of unknown parameters in the space Ω. Let x be an observed random sample from a sample of size n. The aim is to estimate the vector θ using maximum likelihood, with the likelihood function for θ formed from the observed data x is given by

$$L(\theta) = g(x;\theta).$$

In many cases, but not all cases, an estimate $\hat{\theta}$ of θ can be obtained as a solution of

$$\partial L(\theta)/\partial\theta = 0,$$

which is equivalent to

$$\partial \log L(\theta)/\partial\theta = 0. \tag{1.5}$$

The aim of the estimation is to establish a consistent and asymptotically efficient sequence of roots for the likelihood equation (1.5). Such a sequence exists under some mild regularity conditions. These roots usually correspond to a local maximum in the interior of the parameter space. In general, there is usually a global maximum in the interior of the parameter space. We will henceforth refer to the estimate $\hat{\theta}$ as the MLE. It is important to note that for each sample size n, the MLE typically identifies a root that corresponds to a local maximum of the likelihood function. However, this does not guarantee that the MLE always represents the global maximum of the likelihood function. In practical applications, maximizing the log-likelihood function analytically is often not possible. In such situations, it may be possible to iteratively compute the MLE of a parameter using a Newton–Raphson procedure or a variant. This approach is viable when the total number of parameters d in the model is not excessively large. An alternative method is to use the EM algorithm.

1.4.2 EM Algorithm

We define Y as the random vector associated with the observed data y, characterized by the probability density function (p.d.f.) $g(y;\theta)$. As stated before, the EM algorithm provides an alternative method for computing the MLE in situations where data are missing. We will now consider y as being the incomplete data, and we will refer to x as the complete data. We can establish the relation $y = y(x)$ where $y \in \mathcal{Y}$ and $x \in \mathcal{X}$. Incomplete data includes not only situations with missing data, but also situations where the complete data are never observable. We now let $g_c(x;\theta)$ denote the probability density function corresponding to the random vector X for the complete data vector x and let $\log L_c(\theta) = \log g_c(x;\theta)$ denote the complete-data log likelihood function. We can write

$$g(y;\theta) = \int_{\mathcal{X}(y)} g_c(x;\theta)\,\mathrm{d}x,$$

where $X(y)$ is the subset of $\mathcal{X}$ determined by the equation $y = y(x)$. The aim of the EM algorithm is to solve the equation (1.5) indirectly. It proceeds by alternating between the E-step (Expectation step) and the M-step (Maximization step). If we let $\theta^{(0)}$ be the initial value for θ, on the first iteration, the E-step computes

$$Q(\theta;\theta^{(0)}) = \mathbb{E}_{\theta^{(0)}}[\log L_c(\theta) \mid y].$$

The M-step's goal is to maximize this quantity. In other words, we seek $\theta^{(1)}$ such that for all $\theta \in \Omega$,

$$Q(\theta^{(1)};\theta^{(0)}) \geq Q(\theta;\theta^{(0)}).$$

We then continue to apply these steps, each time replacing $\theta^{(k)}$ by $\theta^{(k+1)}$. To summarize, for the E-step of the $(k+1)$th iteration, we compute $Q(\theta;\theta^{(k)}) = \mathbb{E}_{\theta^{(k)}}[\log L_c(\theta) \mid y]$

and for the M-step, we choose $\theta^{(k+1)}$, which maximizes $Q(\theta;\theta^{(k)})$ for all $\theta \in \Omega$. The steps are iterated until the difference between $L(\theta^{(k+1)})$ and $L(\theta^{(k)})$ is arbitrarily small. The key to the EM algorithm lies in the specification of the complete-data vector x and the conditional density of X given the observed data vector y, which is required for the E-step. The choice of this conditional density allows us to perform the E-steps and M-steps conveniently. While there is flexibility in selecting the complete-data vector x, it is chosen with consideration for optimizing the convergence of the EM algorithm. It has often been pointed out that the use of the term algorithm might not be appropriate, as it does not specify the exact sequence of steps for a single E or M-step. The EM algorithm is essentially a framework that can be adapted to various incomplete-data problems.

Proposition 1.4.1. *The likelihood function $L(\theta)$ is not decreased after an EM iteration. In other words, for $k = 0, 1, 2, 3, \ldots$ we have*

$$L(\theta^{(k+1)}) \geq L(\theta^{(k)}).$$

This proposition implies that a bounded sequence of likelihood values converges monotonically. The self-consistency of the EM algorithm is also a consequence. If the MLE $\hat{\theta}$ globally maximizes $L(\theta)$, it must satisfy

$$Q(\hat{\theta};\hat{\theta}) \geq Q(\theta;\hat{\theta})$$

for all θ. Otherwise, $Q(\hat{\theta};\hat{\theta}) \leq Q(\theta_a;\hat{\theta})$ for some θ_a which implies $L(\theta_a) > L(\hat{\theta})$. But it would then contradict that $\hat{\theta}$ is the global maximum of $L(\theta)$.

The EM algorithm can be generalized by slightly modifying the M-step. The generalized algorithm aims to satisfy a lighter condition, namely that the updated parameter estimate $\theta^{(k+1)}$ must satisfy the inequality

$$Q(\theta^{(k+1)};\theta^{(k)}) \geq Q(\theta^{(k)};\theta^{(k)}).$$

Hence, we choose $\theta^{(k+1)}$ to ensure an increase in $Q(\theta;\theta^{(k)})$ instead of being a maximizer of the Q-function with respect to θ. This condition is sufficient to ensure the non-decrease of the likelihood. This approach is taken when the solution of the M-step does not exist in closed form, leading to what is called the generalized EM (GEM) algorithm.

We now present a simple example of EM algorithm. Suppose that the probability density function of a random vector W has 2-component mixture form

$$f(w;\theta) = \frac{1}{2}\sum_{i=1}^{2} f_i(x),$$

where $\theta = (\lambda_1, \lambda_2)$ is the vector containing the unknown parameters corresponding to the parameters of the two exponential random variable

$$f_i(x) = \lambda_i \exp(-\lambda_i x), \quad i = 1, 2.$$

The λ_i are the unknowns parameters we want to estimate for $i = 1, 2$. This corresponds to the situation where the population is modeled by two distinct groups with the same proportion. We let

$$y = (w_1, \ldots, w_n)^\top$$

denote the observed random sample obtained from the mixture density. The log likelihood function for θ formed from the observed data y is

$$\begin{aligned}\log L(\theta) &= \sum_{j=1}^{n} \log f(w_j; \theta) \\ &= \sum_{j=1}^{n} \log \left\{ \sum_{i=1}^{2} \frac{1}{2} f_i(w_j) \right\}.\end{aligned}$$

For this example, we simulated n points for each distribution. In order to pose this problem in the EM context, we denote the missing data by

$$z = (z_1^\top, \ldots, z_n^\top)^\top,$$

where z_j is a two-dimensional vector where $z_{ij} = (z_j)_i$ is one or zero according to whether w_j come from the ith component of the mixture. As stated before, the EM algorithm handles the addition of the unobserved data to the problem by working with the current conditional expectation of the complete-data log likelihood given the observed data. The complete data vector is denoted by

$$x = (y^\top, z^\top)^\top,$$

and the complete-data log likelihood for θ has the form

$$\log L_c(\theta) = \sum_{i=1}^{2} \sum_{j=1}^{n} z_{ij} \log(1/2) + \sum_{i=1}^{2} \sum_{j=1}^{n} z_{ij} \log f_i(w_j; \theta_i). \tag{1.6}$$

This expression is linear in the unobserved data z_{ij} so the E-step requires only the calculation of the current conditional expectation of Z_{ij} given the observed data y, where Z_{ij} is the random variable corresponding to z_{ij}. Now, on the $(k+1)$th iteration,

$$\mathbb{E}_{\theta^{(k)}}(Z_{ij} \mid y) = P_{\theta^{(k)}} \{Z_{ij} = 1 \mid y\} = z_{ij}^{(k)},$$

where by Bayes theorem, we have

$$z_{ij}^{(k)} = \frac{f_i(w_j; \theta^{(k)})}{2 f(w_j; \theta^{(k)})}.$$

Replacing the corresponding probability density function, we have

$$z_{ij}^{(k)} = \frac{\lambda_i^{(k)} \exp\left(-\lambda_i^{(k)} w_j\right)}{\lambda_1^{(k)} \exp\left(-\lambda_1^{(k)} w_j\right) + \lambda_2^{(k)} \exp\left(-\lambda_2^{(k)} w_j\right)}.$$

Computing the right term of (1.6), we have

$$\sum_{i=1}^{2}\sum_{j=1}^{n} z_{ij} \log f_i(w_j) = \sum_{i=1}^{2}\sum_{j=1}^{n} z_{ij}(\log \lambda_i - \lambda_i w_j).$$

The M-step requires the computation of the values of $\lambda_1^{(k+1)}$ and $\lambda_2^{(k+1)}$ that maximizes $Q(\theta, \theta^{(k)})$. The maximum likelihood estimator of λ_i would be

$$\frac{n}{2\sum_{j=1}^{n} z_{ij} w_j}$$

if the z_{ij} were observable. As $\log L_c(\theta)$ is linear in the z_{ij}, we can replace z_{ij} by $z_{ij}^{(k)}$, their current conditional expectations. This yields

$$\lambda_i^{(k+1)} = \frac{n}{2\sum_{j=1}^{n} z_{ij}^{(k)} w_j}.$$

For this example, we used simulated points using the true parameters $\lambda_1 = 0.5$ for the first distribution and $\lambda_2 = 5$ for the second one. We can observe in Figure 1.5 the variation of performance of the algorithm depending on the number of simulated data points n. The number of iterations stays reasonably the same for every number of simulated data points. The algorithm yields a much more precise estimate with a greater number of simulated data points, which could be expected. The boxplots for parameter λ_1 show a relatively tight interquartile range, suggesting that the bulk of the estimation errors are concentrated around the median. The median of the errors is consistently around zero, which implies that the estimator is unbiased. Outliers are present but not excessive, indicating occasional large errors but nothing systematic as sample size changes. In contrast, the boxplots for parameter λ_2 exhibit a much wider spread in errors. The larger spread suggests that individual estimates can be quite far from the actual value, even if on average the estimator is unbiased.

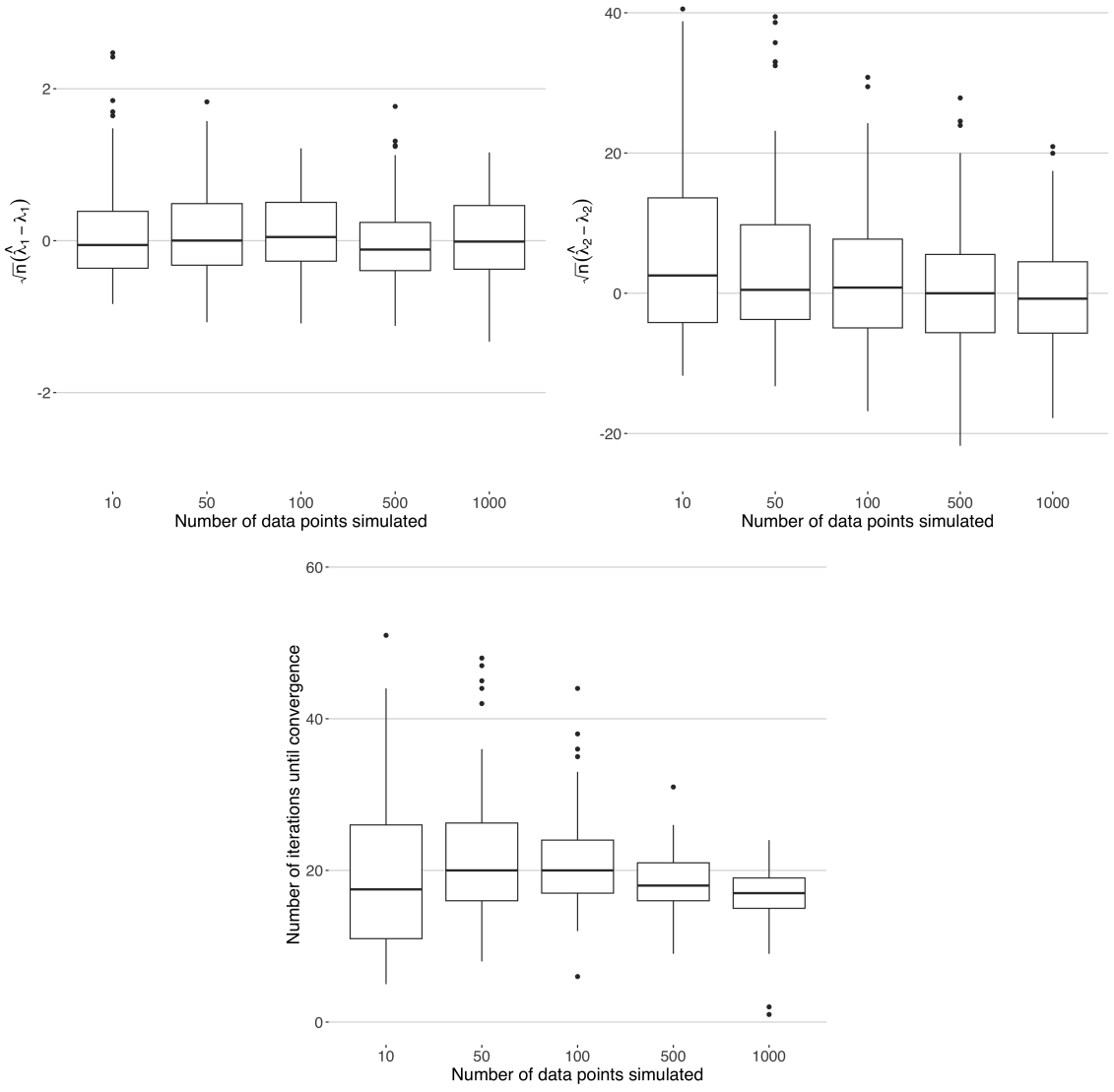

Figure 1.5: The top graphs are boxplots evaluating the standardized difference between the estimated parameter and the true parameter. The bottom graph corresponds to the number of iterations necessary for the algorithm to converge.

Chapter 2

Applications

2.1 GAM fitting

In the initial phase of our research, we begin by fitting a GAM to the training set detailed in section 1.1. To reiterate, our study focuses on a dataset divided into 10000 grid squares, each represented by its center point coordinates and the recorded collision frequency. Our goal is to assess if a GAM with a Poisson distribution is suited for our signal and background data and to find the best model structure accordingly. We also compare this approach to a GLM to determine if the GAM provides a better fit. In this section, x_1 corresponds to the missing transverse energy and x_2 corresponds to the amount of energy contained within a cone around a lepton candidate.

2.1.1 Background data fitting

We fitted five GAMs for the background data and compared their respective AIC values in Table 2.1. For the notation used, see subsection 1.3.3 for detailed explanations.

Model	Degrees of Freedom	AIC
$s(x_1) + s(x_2)$	9.89	44425.41
$s(x_1) + s(x_2) + ti(x_1, x_2)$	11.43	44426.25
$s(x_1, x_2)$	24.19	44507.90
$s(x_1, x_2) + ti(x_1, x_2)$	37.62	44483.45
$te(x_1, x_2)$	9.82	44479.69

Table 2.1: Models with Degrees of Freedom and AIC. We set log *as the link function and Poisson distribution.*

The additive model, characterized by the lowest AIC value among the models considered, appears to provide a better fit to the data. In particular, it explains 92.1%

of the deviance and achieves full convergence after eight iterations. The estimated intercept is highly significant at 1.67, indicating a robust association with the response variable. The model incorporates non-linear relationships through x_1 and x_2 with effective degrees of freedom of approximately 1.48 and 8.71, respectively. The chi-square values for these smooth terms, 26619 for x_1 and 84195 for x_2, are highly significant, reinforcing the presence of non-linear patterns in the data. The adjusted R-squared of 0.95 highlights a strong fit, explaining about 95% of the variability in the response variable. Diagnostic metrics, including a UBRE of 0.06 and a scale estimate of 1, indicate a satisfactory model with low bias and appropriate dispersion estimation. The detailed results and metrics described can be found in the "gam.summary" output. To deepen the analysis of the model fit, we can utilize the "gam.check" method. It generates residual plots and provides additional insights into the success of the fitting process.

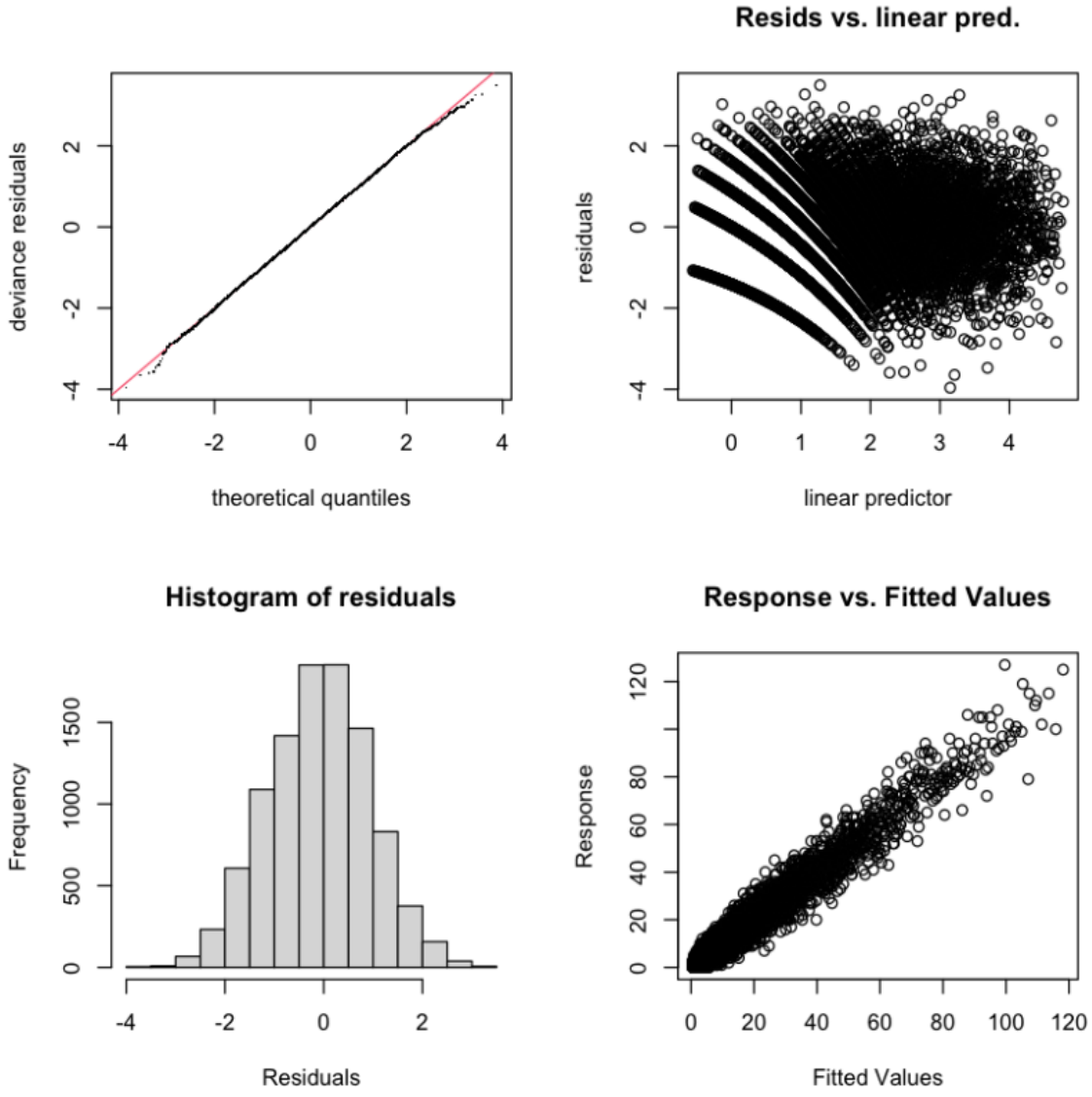

Figure 2.1: Residual plot from gam.check for background data with additive structure.

In the "gam.check" plots in Figure 2.1, the QQ plot in the upper left graph exhibits a remarkable alignment with the red line, indicating a strong adherence to the expected pattern. Unlike a standard QQ plot, which compares residuals to a normal distribution, this plot uses theoretical quantiles that match the model's specified error distribution, which, in our scenario, is the Poisson distribution. The bottom left graph shows a histogram resembling a normal distribution, suggesting the data's goodness of fit with a slight tendency to underestimate a few background points. The bottom right graph displays a smooth scatter like an identity function, further corroborating the model's accuracy. The upper right graph, characterized by a considerable scatter around the y-axis's origin without discernible patterns, signifies nothing problematic. These observations collectively affirm the data's good fitting to the GAM model.

Alternatively, we consider a model based on a framework of generalized linear model with the use of Poisson distribution

$$\log(\mu) = \beta_0 + \beta_1 x_1 + \beta_2 x_2,$$

where μ represents the expected count or mean of the response variable "count", β_0 is the intercept term, β_1 and β_2 are the coefficients associated with the predictor variables x_1 and x_2. Compared to the GAM models, we notice that the AIC for GLM is larger in Table 2.2. It implies that a GAM structure is more adapted to our data.

Model	Degrees of Freedom	AIC
$s(x_1) + s(x_2)$	9.90	44425.41
GLM	3.00	48131.37

Table 2.2: Models with Degrees of Freedom and AIC comparing GAM and GLM model for background data.

The series of plots in Figure 2.2 demonstrate the model's fit across various values of x_2. The y-axis scale is not uniform across the graphs, which may influence the visual interpretation of the fit. Nonetheless, despite the varying scales, the model's fit is uniformly consistent for all values of x_2. This uniformity is reflected by the model's trajectory, which reliably follows the trend of the data points throughout the entire range of x_2, irrespective of the frequency of event counts.

2.1.2 Signal data fitting

We also fitted five GAMs for the signal data and compared their respective AIC values in Table 2.3.

The most suitable model indicated by the lowest AIC 4501.15 demonstrates an excellent match with the data. The deviance explained is an exceptionally high 99.9%, and the model achieves full convergence in nine iterations. It effectively captures non-linear relationships using variables x_1 and x_2, with their respective degrees of freedom

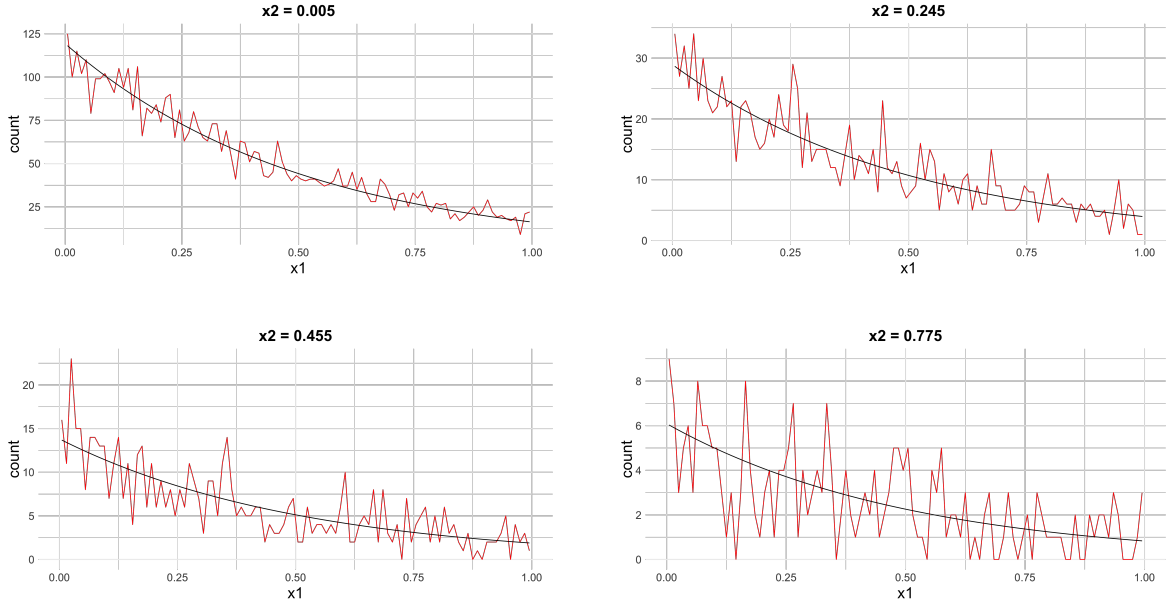

Figure 2.2: The red curves correspond to the actual values whereas the black curves correspond to the prediction with the GAM model for different values of x_2.

Model	Degrees of Freedom	AIC
$s(x_1) + s(x_2)$	10.61	4501.15
$s(x_1) + s(x_2) + ti(x_1, x_2)$	13.62	4505.52
$s(x_1, x_2)$	13.45	4512.11
$s(x_1, x_2) + ti(x_1, x_2)$	12.32	4506.85
$te(x_1, x_2)$	8.18	4541.22

Table 2.3: Models with Degrees of Freedom and AIC. We set log *as the link function and Poisson distribution.*

around 8.61 and 1.00. The chi-square statistics for these variables are significant, with values of 33600 for x_1 and 89802 for x_2. The model's adjusted R-squared value at 0.99 indicates a robust fit. Other diagnostic measures, such as a UBRE of -0.90 and a scale estimate of 1, also support the model's effectiveness. The deviance explained is much greater for the signal data, with 99.9% explained compared to 92.1% for the background data. We can also deepen the analysis the same way with "gam.check".

In Figure 2.3, the QQ plot in the upper left shows the deviance residuals from the GAM aligned with the theoretical distribution, indicating a good model fit. This QQ plot is constructed in the same manner as the one previously described. The bottom right also displays a good result, with a scatter following the identity function. The bottom left graph showcases a histogram with a heavy representation for the 0 residuals. It does not closely resemble a normal distribution, but the outcome is not a cause for concern. This graph is the result of having 9290 squares with no signal events inside on a total of 10000 squares. The residual values are relatively

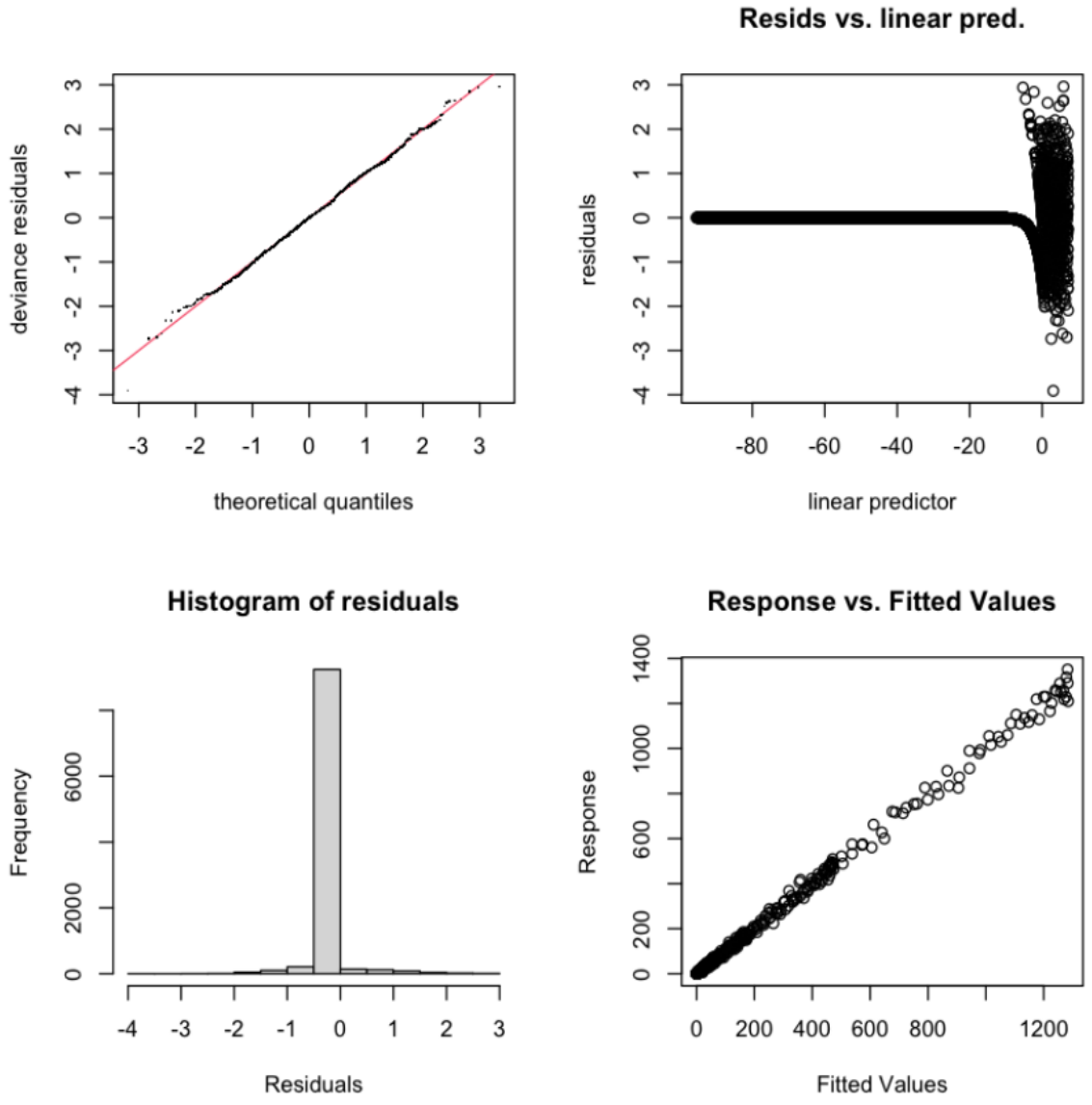

Figure 2.3: Residuals plot from gam.check for signal data with additive structure.

low, especially considering the large values we are estimating. Finally, the upper right graph predominantly shows a significant concentration of residuals around 0, with a scattering of values towards the far right of the graph. This concentration is primarily attributed to a substantial portion of the signal data having count values of 0 across many coordinates. The model often predicts values of 0 in these cases. Despite the results being notably different from those of the background data, these observations still affirm the data's great fitting to the GAM model. In comparison to a GLM model, the GAM structure exhibits a significantly lower AIC, as seen in Table 2.4.

The plots in Figure 2.4 illustrate a good overall estimate of the signal with a generalized additive model. The y-axis scales are not consistent across all graphs, which is crucial to notice when comparing the relative heights and spreads of the estimated signals. Despite the discrepancies in scale, the estimations preserve a bell-shaped distribution, indicative of a robust signal detection by the model. Specifically,

Model	Degrees of Freedom	AIC
$s(x_1) + s(x_2)$	10.613	4501.152
GLM	3	54254.218

Table 2.4: Models with Degrees of Freedom and AIC comparing GAM and GLM for signal data.

for $x_2 = 0.135$ and generally when $x_2 > 0.1$, the plots reveal estimates approaching zero. This suggests that even when the signal counts are zeros, the model still retains its ability to capture the distribution of the signal, maintaining the characteristic bell shape but predicting values that are very small, almost 0.

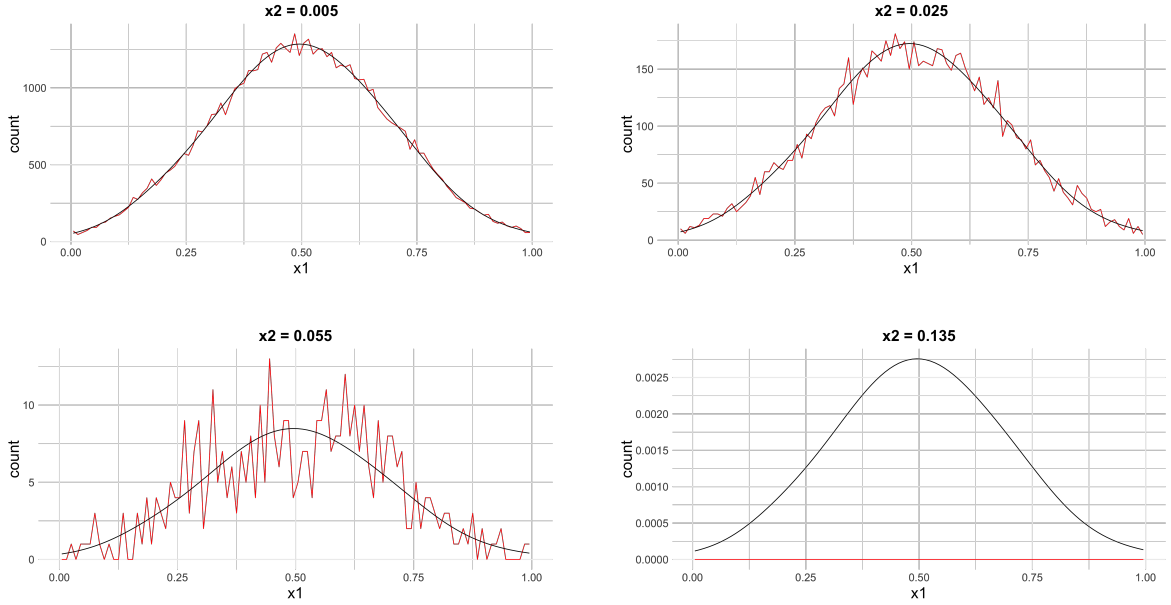

Figure 2.4: The red curves correspond to the actual values whereas the black curves correspond to the prediction with the GAM model for different values of x_2.

The primary objective of our study is to determine the overall Poisson rate for the signal estimate. In our observations of 100000 signal events, the GAM accurately forecasts a rate of 100000. The first heatmap in Figure 2.5, which illustrates error, shows minimal color variation, suggesting a consistent error across the predictor's range. This indicates that the model is neither overfitting nor underfitting in any systematic way, although some outlier data points are still present. The second heatmap indicates an absence of consistent trend within the deviance residuals, which remain within a tolerable range, and the absence of outliers among these residuals suggests that the model provides an adequate fit. The analysis is confined to $x_2 \leq 0.2$ due to the lack of signal events outside this segment.

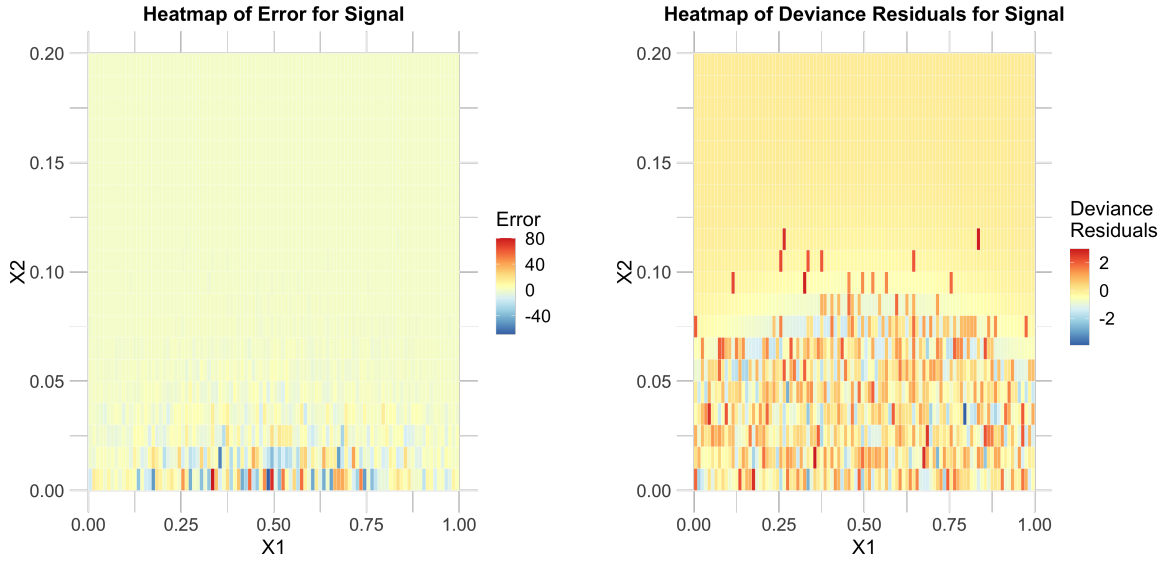

Figure 2.5: Heatmaps of observed error and deviance residuals for GAM on signal of training set.

2.2 EM algorithm application

2.2.1 EM algorithm for GAM

We now present the EM algorithm for our data. We consider a model where the observed data w_i are the sums of two latent variables Z_{i1} (the signal) and Z_{i2} (the background), with each latent variable following a Poisson distribution. The expected values of these distributions are functions of two variables x_{1i} and x_{2i} following a GAM structure

$$\begin{aligned} \lambda_{i1} &= \exp(\alpha_0 + s_1(x_{1i}) + s_2(x_{2i})), \\ \lambda_{i2} &= \exp(\alpha_0 + s_3(x_{1i}) + s_4(x_{2i})), \end{aligned} \tag{2.1}$$

where

$$s_i(x) = \sum_{j=2}^{k} \beta_{ij}\varphi_{ij}(x), \quad i = 1, \ldots, 4$$

are the smooth functions with $\varphi_{ij}(x)$ the thin plate spline basis functions. We let k denote the degrees of freedom for the smooth. We let θ denote the vector containing the unknown parameters β_{ij}, and let

$$y = (w_1, \ldots, w_n)^\top$$

denote the observed random sample obtained from the sum of the two variables. The log-likelihood for a single observation from a Poisson distribution with mean λ is

$$\log L(Z; \lambda) = Z \log \lambda - \lambda - \log Z!, \quad \lambda > 0.$$

For the complete data, comprising both Z_{i1} and Z_{i2} for each observation i, the log-likelihood is the sum of the log-likelihoods of these two independent Poisson distributions

$$\log L_c(Z_{i1}, Z_{i2}; \theta) = Z_{i1} \log \lambda_{i1} - \lambda_{i1} - \log Z_{i1}! + Z_{i2} \log \lambda_{i2} - \lambda_{i2} - \log Z_{i2}!, \quad \lambda_1, \lambda_2 > 0.$$

As stated before, the EM algorithm handles the addition of the unobserved data to the problem by working with the current conditional expectation of the complete-data log likelihood given the observed data. This expression is linear in the unobserved data Z_{ij} so the E-step requires only the calculation of the current conditional expectation of Z_{ij} given the observed data y. Now, on the $(k+1)$th iteration, we have

$$\mathbb{E}_{\theta^{(k)}}[Z_{ij} \mid y] = w_i \times \frac{\lambda_{ij}^{(k)}}{\lambda_{i1}^{(k)} + \lambda_{i2}^{(k)}}.$$

Substituting these expected values into the conditional expectation of log-likelihood function, we get

$$Q(\theta, \theta^{(k)}) = \sum_{i=1}^{n} \left(\mathbb{E}_{\theta^{(k)}}[Z_{i1} \mid y] \log \lambda_{i1} - \lambda_{i1} + \mathbb{E}_{\theta^{(k)}}[Z_{i2} \mid y] \log \lambda_{i2} - \lambda_{i2} \right).$$

Note that in the expression for $Q(\theta, \theta^{(k)})$, the terms involving $\log Z!$ are omitted, because these terms do not contain the parameters and thus do not contribute to estimation in the M-step of the algorithm. Their omission simplifies the computation without affecting the optimization of the parameters.

The M-step requires the computation of the values of $\lambda_1^{(k+1)}$ and $\lambda_2^{(k+1)}$ that maximize $Q(\theta, \theta^{(k)})$. To find the optimal parameters, we take derivatives with respect to λ_{i1} and λ_{i2}, and set them to zero. The derivative with respect to λ_{ij} is

$$\frac{\partial Q}{\partial \lambda_{ij}} = \frac{\mathbb{E}_{\theta^{(k)}}[Z_{ij} \mid y]}{\lambda_{ij}} - 1.$$

Setting this derivative to zero, we get

$$\frac{\mathbb{E}_{\theta^{(k)}}[Z_{ij} \mid y]}{\lambda_{ij}^{(k+1)}} - 1 = 0,$$

and simplifying, we find

$$\lambda_{ij}^{(k+1)} = \mathbb{E}_{\theta^{(k)}}[Z_{ij} \mid y].$$

From this and using the formulas in (2.1), we have to find the parameters of the GAM structure for each j where $j = 1, 2$. We achieve this by fitting a separate GAM per value of j using $\lambda_{ij}^{(k+1)}$ for $i = 1, \ldots, 10000$ as the new fitted values. The process is repeated until the parameters reach a sufficient level of stability. This algorithm corresponds to a generalized EM algorithm because of the lack of closed form formula for the parameters. It is necessary to set the initial values of $\lambda_{ij}^{(0)}$ for the initialization process.

2.2.2 Analysis

In this algorithm, the hyperparameters we have to set are the degrees of freedom for each smooth and the smoothing parameter. The latter is chosen to avoid under- and over-fitting, so the resulting function is as close as possible to the true function. This parameter is chosen with cross-validation. But the principal difficulty to make the algorithm perform correctly is the setting of the initial parameters. This initial choice has a huge impact on the EM algorithm and changes completely its results. To estimate the best choice of initial parameters, we have six other generated datasets to test the performance of the EM algorithm. Each of the six datasets contains 100000 signal events and 100000 background events. Figure 2.6 shows the overall distributions for signal and background. Dataset 5 exhibits a significant variance in the concentration of background events compared to the others, with higher concentration of background events.

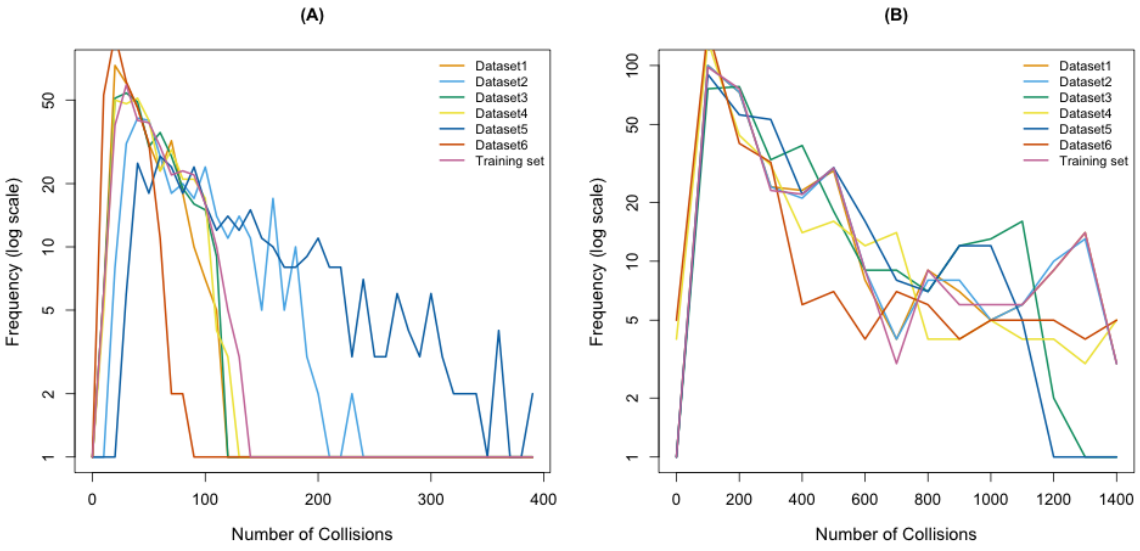

Figure 2.6: The x axis denotes the number of collisions, while the y axis indicates the number of times each count was observed. The left plot (A) corresponds to the background distribution and the right plot (B) corresponds to the signal distribution.

Table 2.5 presents a comparative analysis of event counts for small values of x_2. A notable observation is the unusually high number of background events in dataset 5, which is atypical for such values of x_2. Additionally, a significant majority of the signal events in dataset 6 fall within small values of x_2. This variation in distribution patterns may present challenges for the EM algorithm in adapting to these diverse distributions. Before fitting the EM algorithm, we fitted a GAM with an additive structure to each dataset signal and background separately. Following the same procedure as used for the training set, the models demonstrate a suitable fit for the data. The GAMs fitted also indicate that the degrees of freedom for the smooth of the variable x_2 for the signal estimation is usually smaller than three. The remaining degrees of freedom

Dataset	Signal	Background
1	98185	15463
2	98171	29164
3	98163	17879
4	98148	18056
5	93223	44376
6	99978	9431

Table 2.5: Comparison of number of collisions recorded for signal events and background events for $x_2 < 0.035$.

for the other smooths and other variables are usually smaller than ten. For our first choice of initialization values in the EM algorithm, the initial values are chosen to be the parameters from the GAM that was fitted to the training set. In other words, we start the algorithm using the rate obtained from this dataset. The maximum degrees of freedom are selected based on previous analyses: for the smooth term using covariate x_1, we set the degrees of freedom to ten, and for covariate x_2, we set it to three. To avoid any overfit because of the choice of initial parameters, we set the smoothing parameter to 0.5. The second choice of initial values is the set of parameters from the GAM that was fitted to the dataset 5, which exhibits a greater concentration of background events for small values of x_1 and x_2. The third choice of initial values is the set of parameters of the GAM fitted to the dataset 3, which exhibits a smaller concentration of background events for small values of x_1 and x_2. The parameters obtained from this algorithm are presented in Table 2.6, alongside the number of iterations required. It is evident that the algorithm demonstrates high accuracy for all datasets, except for the fifth. The parameter estimates are notably similar to those of the GAMs, which were trained using the actual values of the signal and background. The choice of different initialization seems to have minimal influence on the algorithm's performance, particularly for dataset 1, where the results remain unchanged. The most significant variations in parameter values are observed in dataset 5. However, the primary focus of our study is to determine the overall signal rate for each dataset, which, in our context, corresponds to the summation of the predicted rates across all individual small squares.

Table 2.7 shows that, excluding the dataset 5, which exhibits a poor fit even when initializing the EM algorithm with its true parameters, we have relatively small error between the true rate and the estimated rate. The model might not be appropriate for dataset 5 which exhibits a significant error.

From now on, we will only use the first initialization for our analyses. As a reminder, this initialization uses the parameters obtained from the training set and only affects the proportion of signal and background for the first step of the EM algorithm. In this study, we adopted an approach to validate our model by simulating 100

Initial	Dataset	Intercept	$s(x_1).1$	$s(x_1).2$	$s(x_1).3$	$s(x_1).4$	$s(x_1).5$	$s(x_1).6$	$s(x_1).7$	$s(x_1).8$	$s(x_1).9$	$s(x_2).1$	$s(x_2).2$	Iterations
True	1	−43.63	−0.47	−1.68	−0.48	−1.17	−0.50	1.15	−0.42	4.66	0.03	0.00	−28.99	2
Init1	1	−43.78	−0.44	−1.49	−0.43	−1.02	−0.47	1.02	−0.39	4.34	0.02	0.01	−29.09	45
Init2	1	−43.78	−0.44	−1.49	−0.43	−1.02	−0.47	1.02	−0.39	4.34	0.02	0.01	−29.09	60
Init3	1	−43.78	−0.44	−1.49	−0.43	−1.02	−0.47	1.02	−0.39	4.34	0.02	0.01	−29.09	35
True	2	−43.60	−0.25	−1.86	−0.37	−1.25	−0.37	1.09	−0.36	4.88	0.23	0.00	−28.97	2
Init1	2	−44.26	−0.25	−2.05	−0.40	−1.42	−0.42	1.23	−0.41	5.17	0.22	0.01	−29.37	73
Init2	2	−44.40	−0.33	−2.19	−0.45	−1.51	−0.45	1.30	−0.43	5.40	0.31	0.01	−29.45	43
Init3	2	−44.26	−0.25	−2.05	−0.40	−1.42	−0.42	1.23	−0.41	5.17	0.22	0.01	−29.37	77
True	3	−43.39	−0.19	−1.31	−0.24	−0.93	−0.30	0.83	−0.31	3.34	−0.02	0.00	−28.98	2
Init1	3	−43.76	−0.16	−1.36	−0.23	−0.96	−0.29	0.85	−0.30	3.44	−0.09	0.01	−29.18	44
Init2	3	−43.83	−0.17	−1.36	−0.23	−0.95	−0.29	0.85	−0.30	3.44	−0.08	0.01	−29.22	52
Init3	3	−43.56	−0.17	−1.36	−0.23	−0.96	−0.29	0.85	−0.30	3.44	−0.08	0.01	−29.07	39
True	4	−44.10	−0.48	−2.63	−0.62	−1.80	−0.55	1.72	−0.50	7.70	−0.12	0.00	−28.98	2
Init1	4	−44.51	−0.65	−2.91	−0.71	−1.91	−0.59	1.77	−0.52	8.14	0.09	0.01	−29.19	45
Init2	4	−44.56	−0.71	−3.01	−0.74	−1.97	−0.62	1.83	−0.54	8.31	0.16	0.01	−29.21	33
Init3	4	−44.51	−0.65	−2.91	−0.71	−1.91	−0.59	1.77	−0.52	8.14	0.09	0.01	−29.19	48
True	5	−27.30	−0.52	−2.17	−0.54	−1.49	−0.57	1.33	−0.49	5.37	0.39	0.00	−19.33	2
Init1	5	−27.76	0.13	−0.81	−0.17	−0.69	−0.28	0.64	−0.24	3.10	−0.44	0.03	−19.73	35
Init2	5	−28.42	0.00	−1.12	−0.25	−0.88	−0.35	0.80	−0.30	3.62	−0.28	0.03	−20.09	90
Init3	5	−27.58	0.16	−0.73	−0.15	−0.65	−0.26	0.60	−0.22	2.97	−0.48	0.03	−19.63	71
True	6	−93.06	−0.52	−2.17	−0.54	−1.49	−0.57	1.33	−0.49	5.37	0.39	0.00	−58.01	2
Init1	6	−93.83	−0.55	−2.17	−0.56	−1.48	−0.58	1.33	−0.50	5.37	0.44	0.00	−58.46	15
Init2	6	−93.83	−0.55	−2.17	−0.56	−1.48	−0.58	1.33	−0.50	5.37	0.44	0.00	−58.46	26
Init3	6	−93.74	−0.54	−2.15	−0.56	−1.47	−0.57	1.32	−0.49	5.34	0.43	0.00	−58.41	20

Table 2.6: Parameters for signal found for GAM model with complete data (Initial = True) and for incomplete data with different initializations.

Dataset	Initialization 1	Initialization 2	Initialization 3
1	100210	100210	100210
2	99982	99624	99982
3	98648	98507	98992
4	99353	99255	99353
5	110074	106268	111105
6	99842	99842	99874

Table 2.7: Comparison of estimates across three initializations. The true rate is supposedly 100000*.*

datasets for each original dataset. This simulation serves multiple purposes. Firstly, it allows us to assess the stability and consistency of our models under varied conditions, ensuring that our conclusions are not hasty. Secondly, by replicating the analysis across these simulated datasets, we can rigorously test the robustness of our findings against variations in data, such as noise, outliers, or different underlying trends. This is particularly important in scenarios where the original data may have limitations in size or diversity, or where the underlying process is complex and subject to random fluctuations. To fully understand how the EM algorithm behaves, Figure 2.7 displays the average error between the expected value and the value estimated in each square. Excluding the problematic dataset 5, the heatmaps display a range of errors from negative (blue) to positive (red), with yellow representing a neutral or zero error. For small x_2 and as x_1 moves away from the midpoint of 0.5, most squares display minimal error. This suggests that the estimates from the EM algorithm are, generally, very accurate compared to the true value in these areas. Additionally, the generated data from dataset 6 have the least errors overall and the generated data from dataset 3 have the most negative error. The region of concern is primarily characterized by lower values of x_1 and x_2 for dataset 5. In this dataset, there is an exclusively positive error, with larger errors being more pronounced. This aligns with the observation of an overestimated overall rate of 110074 compared to the expected 100000, as detailed in Table 2.7. Specifically, the model's tendency to overestimate is more evident in the lower-left part of the heatmap for this dataset. This pattern of overestimation is not an isolated occurrence but rather a consistent outcome observed across the average of 100 generated datasets.

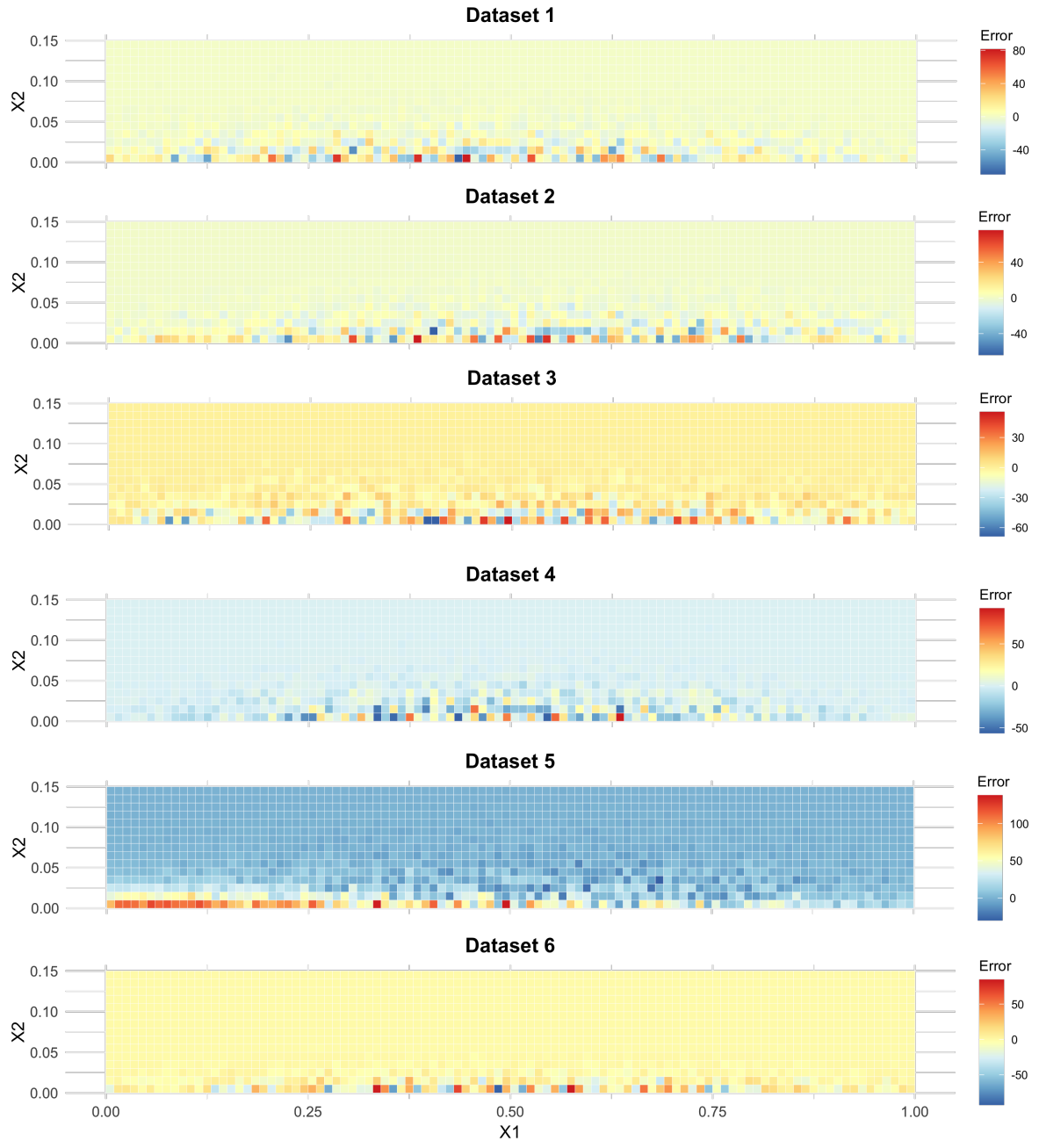

Figure 2.7: Average error heatmaps for signal collisions, generated from 100 simulated datasets for each original dataset. Values corresponding to $x_2 > 0.15$ are omitted as their average is consistently zero.

The graphs in Figure 2.8 highlight the contrast in the fitting effectiveness of the EM algorithm between a dataset with an adequate fit (dataset 3) and the performance on dataset 5. For dataset 5, there is a consistent overestimation in the values, the error is more pronounced for bigger values but the variance appears to be reasonably managed. On the other hand, dataset 3 shows a slight tendency to underestimate the values, though by less than with dataset 5.

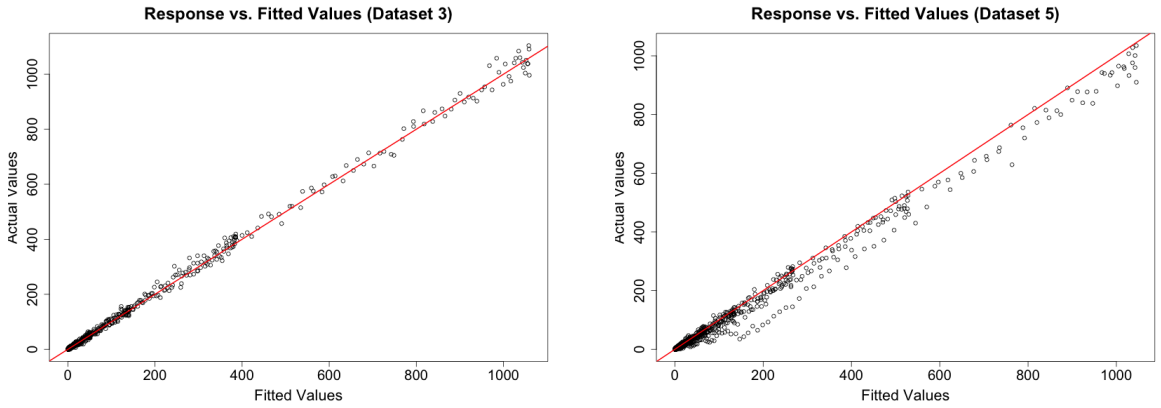

Figure 2.8: Plots of responses against fitted values for dataset 3 *and dataset* 5. *The red curves correspond to the perfect fit and the black dots correspond to the estimate of the EM.*

In Table 2.8, we compare the average sum of Mean Absolute Error (MAE) for 12 parameters obtained using the EM algorithm on simulated datasets against the parameters of the GAMs fitted with complete data. As anticipated, the data generated from dataset 5 exhibit a significantly larger MAE, indicating a higher overall prediction error. For the remaining datasets, the MAE values are comparatively lower. Dataset 3 displays the smallest error, which is somewhat unexpected given its tendency to underestimate the signal. This suggests that while dataset 3's predictions are generally more accurate, they may consistently fall short of the actual values, indicating a systematic bias. Despite the lower average MAE for dataset 3, the higher AIC in comparison to datasets 1, 2, and 6 suggests that the model might be overly complex. For dataset 5, the significantly higher AIC indicates that the model is not fitting the data as well as it does for the other datasets and aligns with the observed MAE of 8.86. While the MAE provides us with a broad measure of prediction error, a deeper dive into the model's performance is revealed when we examine the deviance residuals, offering a more nuanced view of the fit across individual observations.

Deviance residuals measure how well a model fits the data. They measure the discrepancy between the model and the observed data for each coordinate. Figure 2.9 displays the deviance residuals on the simulated datasets. Excluding the problematic dataset 5, we have a good result. The deviance range is between 3 and -3, which indicates that the model is a good fit for our data. This range suggests that there are no extreme outliers that the model is failing to account for. There is also no

Original Dataset	Average MAE	AIC
Dataset 1	1.74	4001
Dataset 2	2.04	3908
Dataset 3	1.26	4115
Dataset 4	2.60	3579
Dataset 5	8.86	5483
Dataset 6	1.99	2417

Table 2.8: MAE for the estimate of the parameters for the simulated dataset compared to the true parameters. AIC is given for the EM algorithm.

noticeable systematic pattern and the residuals are randomly scattered. On another hand, the heatmap indicates a noticeable pattern for dataset 5, with a concentrated area of very low residuals in the left bottom corner. The range of residuals from -9 to 1 is problematic in this case, with almost only negative values. The analysis indicates that the model does not adequately fit dataset 5, suggesting an issue with specific small ranges of the x_1 and x_2 values. Our next step will be to concentrate on how the algorithm performs in this particular region.

The analysis of plots in Figure 2.10 reveals distinct patterns in the distribution of data for $x_2 = 0.005$ across various datasets. While the signal plot maintains a consistent normal-like distribution centered at zero for all datasets, the background distribution shows notable variations, especially in dataset 5. In this dataset, the background distribution exhibits a significant spike, particularly at smaller x_1 values, diverging from the trends observed in other datasets. This deviation not only affects the background distribution, but also significantly influences the summed values. Consequently, the EM algorithm, which relies solely on these sums, is directly impacted. The EM algorithm's predictions are generally accurate for most datasets, except for dataset 5, where a substantial overestimation of the signal is observed. This could be attributed to the algorithm's inability to accurately capture the sharp increase in background levels in the lower-left corner of the plot. This anomaly, particularly in dataset 5, highlights a potential area for further investigation, especially since it suggests a critical limitation in the algorithm's performance under specific data conditions. In dataset 3, the EM algorithm's performance showcases a relatively minor estimation error compared to other datasets. Despite its overall accuracy, there's a tendency for slight overestimation in the background, indicating a nuanced deviation from the actual values. The analysis of the graph suggests that the algorithm demonstrates a lack of robustness in handling outliers within background events. This characteristic of the algorithm becomes evident when dealing with data points that deviate significantly from the general pattern. Similar observations can be made for every value of $x_2 < 0.035$.

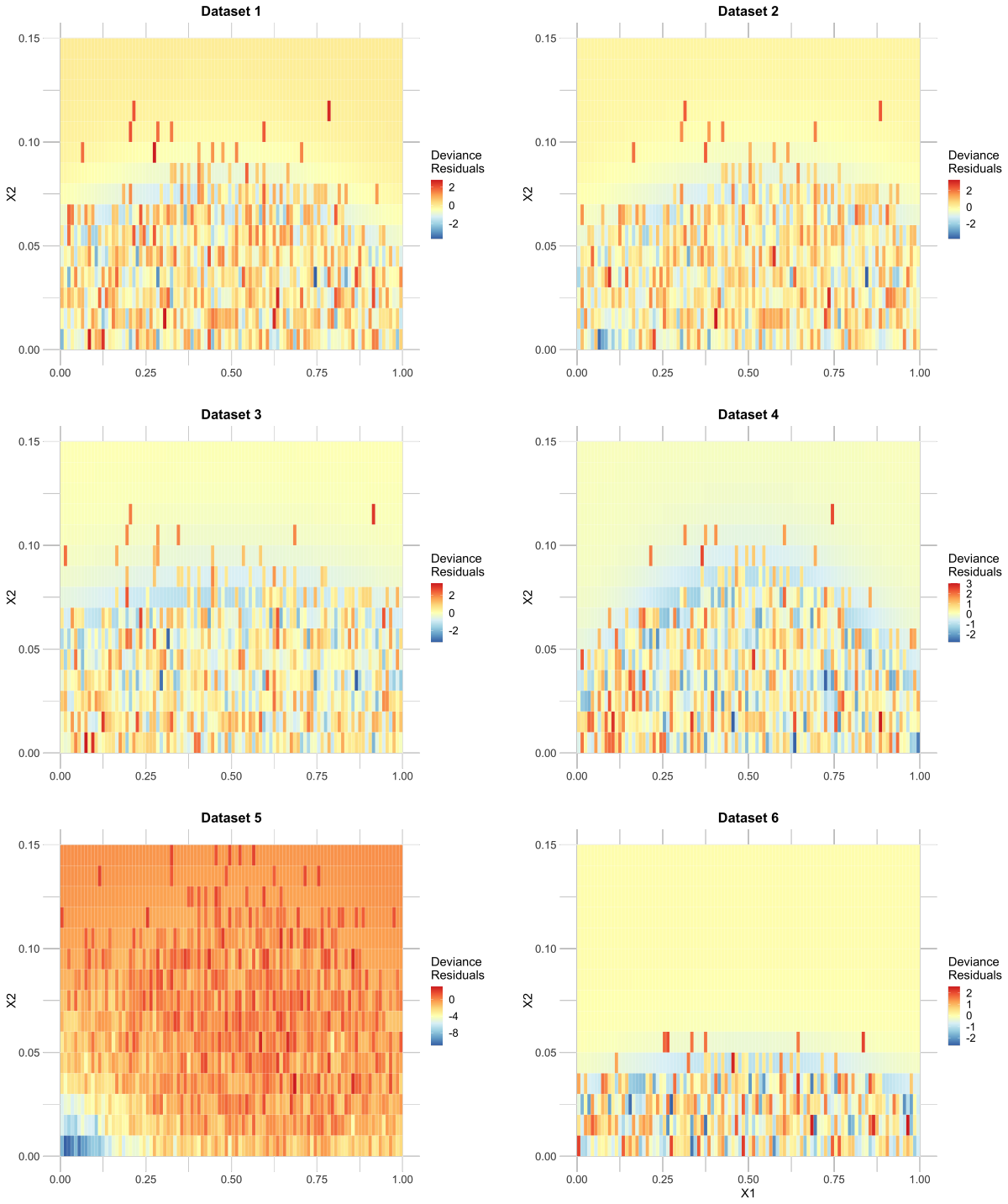

Figure 2.9: Heatmap of average deviance residuals from EM Algorithm. These heatmaps illustrate the average deviance residuals computed using the EM algorithm, aggregated on the 100 simulations for each original dataset. The residuals reflect the deviation of the estimated rates from the true rates. Values corresponding to $x_2 > 0.15$ *are omitted as their average is consistently zero.*

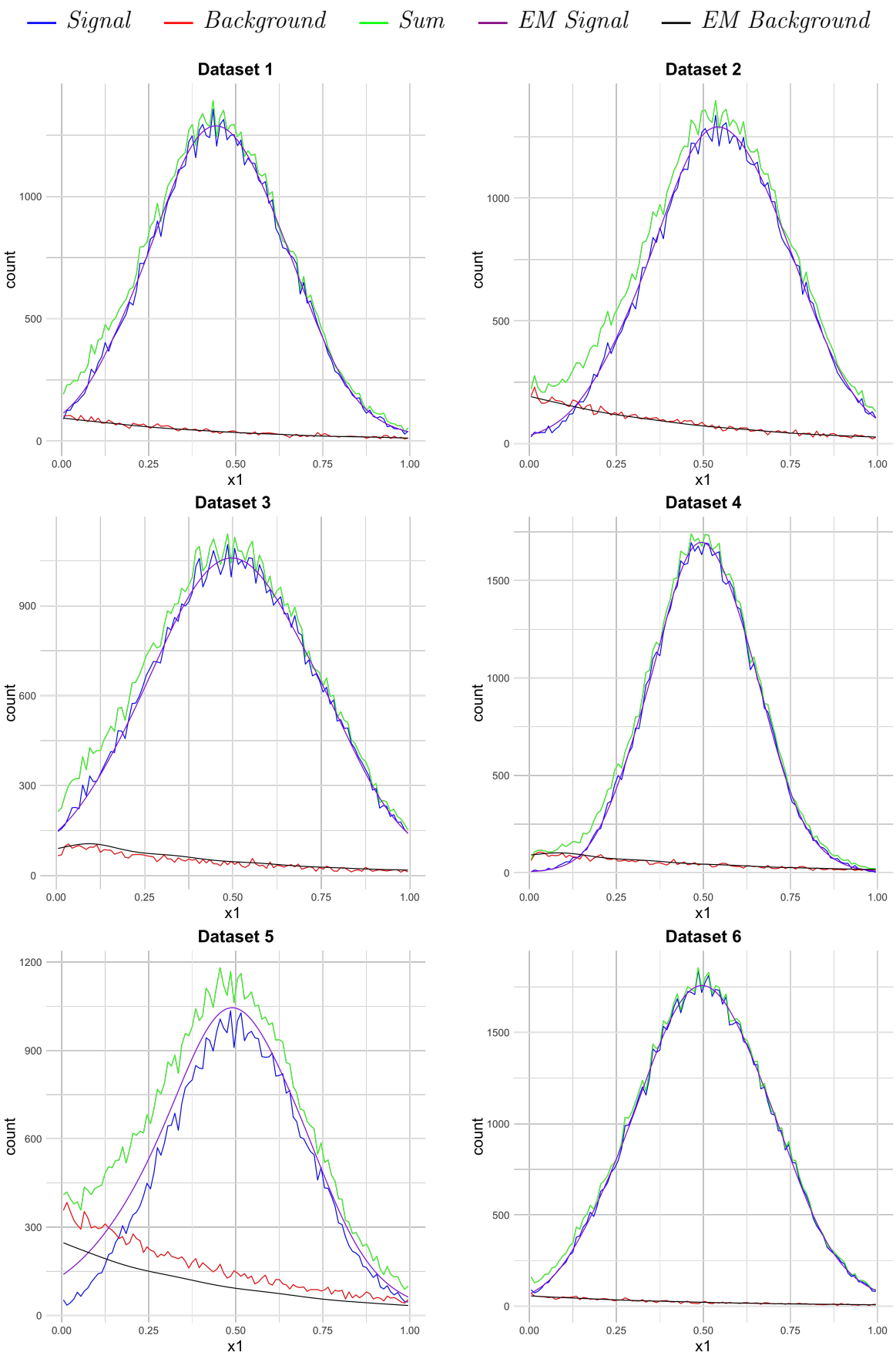

Figure 2.10: Series of plots for $x_2 = 0.005$ across the different datasets. Each plot illustrates the observed signal, background, their sum, and the respective estimates derived from the EM algorithm.

This tendency captured for dataset 5 can be explored by diminishing the anomalies in the lower-left corner. By decreasing background events to align more closely with the patterns seen in other graphs, we can reassess the model using the EM algorithm. With the modified dataset, we reduced the high concentration of background events for small values of x_2 and for small values of x_1. We removed nearly 8000 background events while leaving the signal intact. In Figure 2.11, we can visualize the effect of the modifications for small values of x_2. The new trained model appears to fit better the data. The error between the estimates and the true values is significantly reduced for the background and for the signal.

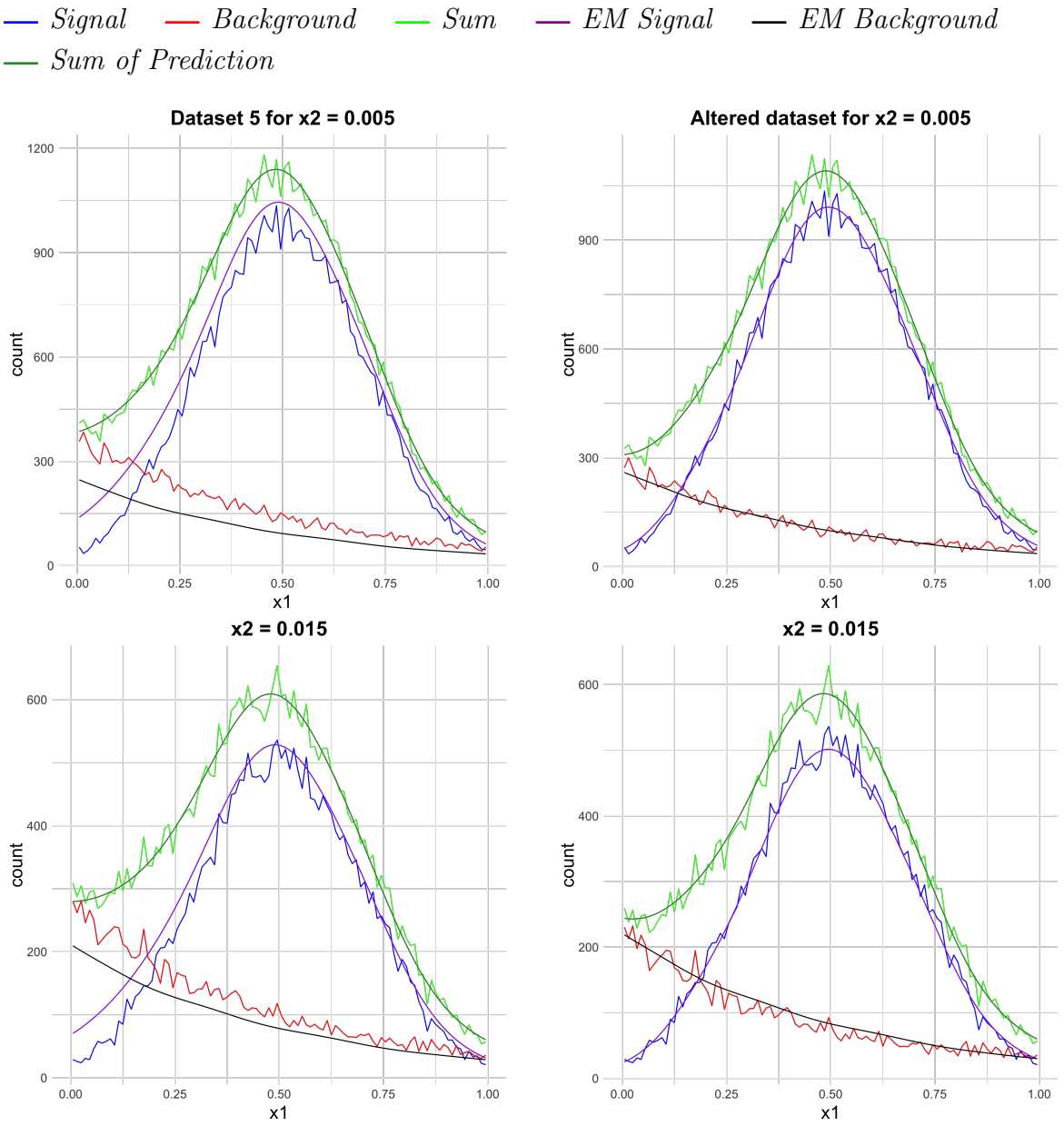

Figure 2.11: Series of plots for $x_2 = 0.005$ and $x_2 = 0.015$ for dataset 5. Left plots correspond to untouched dataset 5 while right plots correspond to modified dataset 5 with reduced background.

A closer examination of the revised outcomes reveals a Poisson rate that more accurately reflects the true value, demonstrating reduced error. In this case, we arrive at an estimated Poisson rate of 99509, drastically smaller than 110074 found with the raw data. Table 2.9 showcases the parameter estimates for the signal rate from the modified dataset, revealing a substantial enhancement in parameter estimation accuracy. The MAE now closely resembles that of the other datasets, which previously did not exhibit issues, as seen in Table 2.6.

Initial	Intercept	$s(x_1).1$	$s(x_1).2$	$s(x_1).3$	$s(x_1).4$	$s(x_1).5$	$s(x_1).6$	$s(x_1).7$	$s(x_1).8$	$s(x_1).9$	$s(x_2).1$	$s(x_2).2$	MAE
True	−27.30	−0.52	−2.17	−0.54	−1.49	−0.57	1.33	−0.49	5.37	0.39	0.00	−19.33	-
Init1	−27.76	0.13	−0.81	−0.17	−0.69	−0.28	0.64	−0.24	3.10	−0.44	0.03	−19.73	8.40
Mod	−27.99	−0.55	−1.90	−0.53	−1.36	−0.55	1.22	−0.46	4.94	0.49	0.03	−19.76	2.28

Table 2.9: Table of parameters and MAE for signal parameters found with GAM model for complete data, with EM for incomplete data and with EM for incomplete modified data.

The heatmaps in Figure 2.12 affirm a well-fitted model, with the spread of deviance residuals showing no apparent patterns, indicating a random distribution of errors. The error distribution balances out across positive and negative values, suggesting a more accurate estimation process, although a few outliers still stand out. The overall fit of the model is improved with the new modified dataset. While this process improved the accuracy of the model, it can only be done with complete data. This approach primarily revealed why the model was ineffective with this particular dataset. Excessively high or low levels of background elements can adversely affect the prediction outcome of the signal.

To conclude the analysis, we compute a 68% Poisson rate-based confidence interval for the EM algorithm's accuracy on simulated data. This interval is determined using the theoretical formula $\lambda \pm \sqrt{\lambda}$, where λ represents the Poisson rate. In this context, λ is calculated as the sum of the Poisson rates in each square of the dataset. It is important to note that this is the fundamental theoretical formula and does not account for the specific modeling approach we have employed. In our analysis, these confidence intervals are primarily used as a measure of the accuracy of the model. For a more detailed discussion on confidence intervals in the context of the EM algorithm used in our study, please refer to Appendix B.

In Table 2.10, we see that for datasets 1, 2, and 6, the EM algorithm demonstrates effectiveness in predicting this confidence interval, indicating good model performance. However, for the remaining datasets, the confidence interval is observed to be excessively narrow, leading to less accurate estimate by the algorithm. This outcome was somewhat anticipated for dataset 5, given its specific characteristics, but it presents an

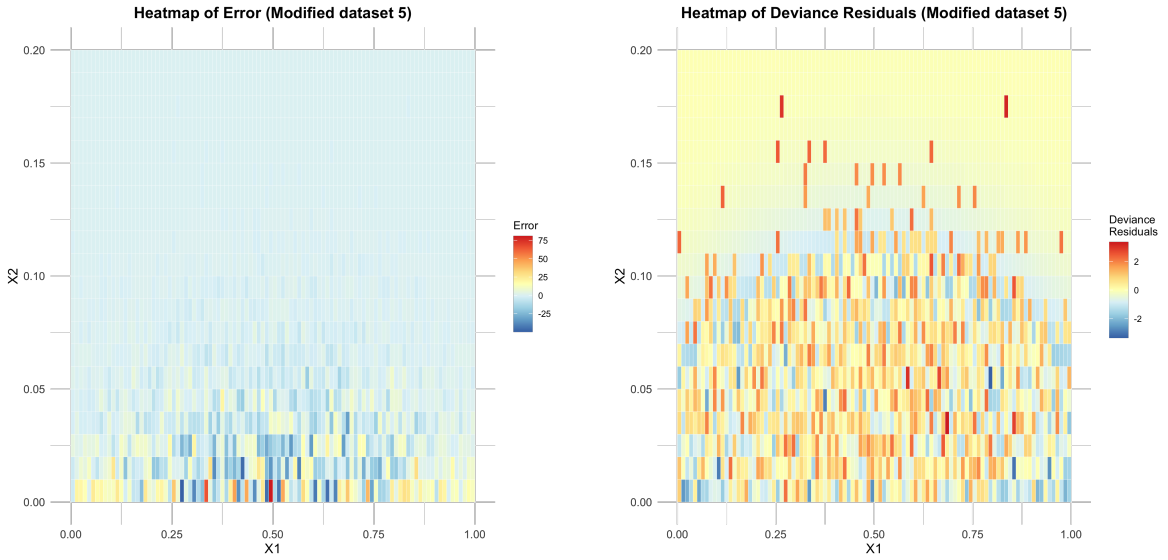

Figure 2.12: Heatmaps of observed errors and deviance residuals for EM algorithm on the modified dataset.

Original Dataset	Confidence successes
Dataset 1	59
Dataset 2	50
Dataset 3	2
Dataset 4	7
Dataset 5	0
Dataset 6	88

Table 2.10: Confidence interval successes for the overall rate over the simulated dataset. The length of each interval is approximately 600 *for each dataset. The desired confidence success rate is* 68*.*

unexpected result for datasets 3 and 4. While dataset 3 had previously shown issues related to model complexity, dataset 4 had not exhibited any prior signs that would predict this issue with the confidence interval.

2.2.3 EM algorithm variant

Two alternative approaches were explored to enhance the model's robustness. The first approach involved increasing the number of parameters in the GAM's smoothers to enable the model to capture a wider variety of smooth shapes. The model with 25 degrees of freedom per smooth showed largely similar performance to the one with fewer degrees, but exhibited fewer errors for dataset 5. The new model ended up more robust to irregularities in the background, but at the cost of reduced precision. As the analysis is similar to the main model, please refer to the Appendix A for more details on this new model.

The second approach entailed dividing each dataset into two subsets based on the value of x_2, specifically into subsets where $x_2 < 0.035$ and $x_2 \geq 0.035$. This division was informed by observations from Table 2.5, which highlighted notable distributional differences at this threshold. The EM algorithm was applied to these subdivided datasets. The segment with $x_2 \geq 0.035$ yielded satisfactory accuracy with standard initialization parameters. In contrast, the segment with $x_2 < 0.035$ showed significant variation in estimate depending on the initialization parameters of the algorithm. Table 2.11 presents data comparing the signal estimate across the different datasets, with a particular focus on the segment where $x_2 < 0.035$. In this segment, two methods are used, differentiated by their initialization parameters. The first method uses initialization parameters derived from the GAM fitted on dataset 5. The second method uses initialization parameters from the GAM modeled on the training set.

Dataset	$x_2 \geq 0.035$		$x_2 < 0.035$		
	Signal	Estimate	Signal	Estimate 1	Estimate 2
1	1821	1777	98185	84395	96071
2	1836	1709	98171	93817	104841
3	1847	1721	98163	83133	96060
4	1856	1737	98148	91930	99343
5	6783	7086	93223	93312	108656
6	28	30	99978	98634	98993
Training	1830	1758	98176	91986	98494

Table 2.11: Signal Estimate for Different Data Segments. For the segment where $x_2 < 0.035$, the first estimate uses initialization parameters derived from the GAM fitted on dataset 5. The second estimate uses initialization parameters from the GAM modelled on the training set.

For the segment $x_2 < 0.035$, there's a notable difference in the signal estimate between the two methods. The first estimate generally yields lower values compared to the second one. This difference can be attributed to the impact of initialization on the EM algorithm used. Initialization parameters play a crucial role in the EM algorithm, as they can significantly influence the convergence of the algorithm and the final estimate it produces.

In this context, the initialization parameters derived from dataset 5 and the training set lead to distinct starting points for the EM algorithm, resulting in underestimation of signal. Particularly, the use of parameters from dataset 5 seems to cause an overfitting effect on the proportion given at the beginning of the EM algorithm. Overfitting in this scenario means that the EM algorithm becomes too closely tailored to the specifics of dataset 5, reducing its generalizability and accuracy when applied to other datasets.

While a suitable modeling approach and initialization was found for dataset 5, this result does not extend to other datasets with different background shape. This

specificity indicates that the model, though effective for dataset 5, is not universally applicable.

Chapter 3

Discussion

In this study, we have demonstrated the feasibility of estimating the parameters of a generalized additive model for both signal and background by observing only their combined effect. This was achieved through the application of the Expectation-Maximization algorithm. A critical aspect of the EM algorithm's success is the selection of initial values. Our findings underscore the importance of this initialization phase, as it can significantly constrain the model's performance.

Two distinct iterations of the main model were developed: the first iteration was less robust to background anomalies but offered greater precision for a general array of datasets. Conversely, the second iteration was enhanced for robustness against background anomalies at the expense of precision. Given additional time, it is conceivable that a more refined model could be constructed, one that brings together the robustness of the second iteration with the precision of the first. This hybrid model holds the promise of offering a superior balance between resilience to background noise and accuracy.

While our model was initially tailored to distinguish between background noise and signals corresponding to the Higgs boson, its potential for generalization is significant. It could be adapted to identify signals that are dissimilar to the Higgs boson, demonstrating the model's versatility.

A limitation of our approach was the decision to confine the analysis to 10000 squares, equating to 10000 data points. A possible avenue for future research would be to increase the number of data points by reducing the size of each square. This refinement could potentially enhance the results, offering finer resolution and possibly more accurate parameter estimate.

In conclusion, the explorations and findings of this study contribute valuable insights into signal-background separation in high-energy physics and provide a strong foundation for subsequent research endeavors. The adaptability of the model and the prospect of improving its performance by adjusting the granularity of the data points offer exciting opportunities for further investigation and application in various domains where signal detection is crucial.

Appendix A

Other model

In the appendix, we replicate the analysis from the main model, applying the EM algorithm with increased degrees of freedom (25 per smooth) and using initialization parameters derived from the GAM for dataset 5. As seen in Figure A.1, the errors observed do not follow any discernible pattern, suggesting non-systematic behavior. The outcomes are broadly in line with those from the main model. However, there are noticeable differences for dataset 5. Although the errors are generally positive, they are less pronounced, and the more significant error values are not as densely clustered in the lower left portion of the graph. In this instance, these higher error values appear as outliers, rather than indicative of any consistent discrepancy.

Examining the deviance residuals in Figure A.2, we find similar observations to those in the main model. With the exception of dataset 5, no distinct patterns or anomalies are evident. The range of deviance is within reasonable bounds, spanning from -3 to 3. However, dataset 5 continues to show higher negative values in deviance residuals, predominantly in the bottom left area. This model seems to present an enhancement over the main model, particularly in how it handles dataset 5.

In Table A.1, we see that the number of successes of confidence interval is significantly diminished for dataset 2, and accuracy overall is reduced. Overall, the new algorithm does a better job for dataset 5 and dataset 6. This algorithm is more robust but loses in accuracy compared to the main model.

Original Dataset	**Confidence successes**
Dataset 1	50
Dataset 2	5
Dataset 3	0
Dataset 4	3
Dataset 5	7
Dataset 6	91

Table A.1: Confidence interval successes for the overall rate over the simulated dataset for EM algorithm with more parameters. The desired confidence success rate is 68%.

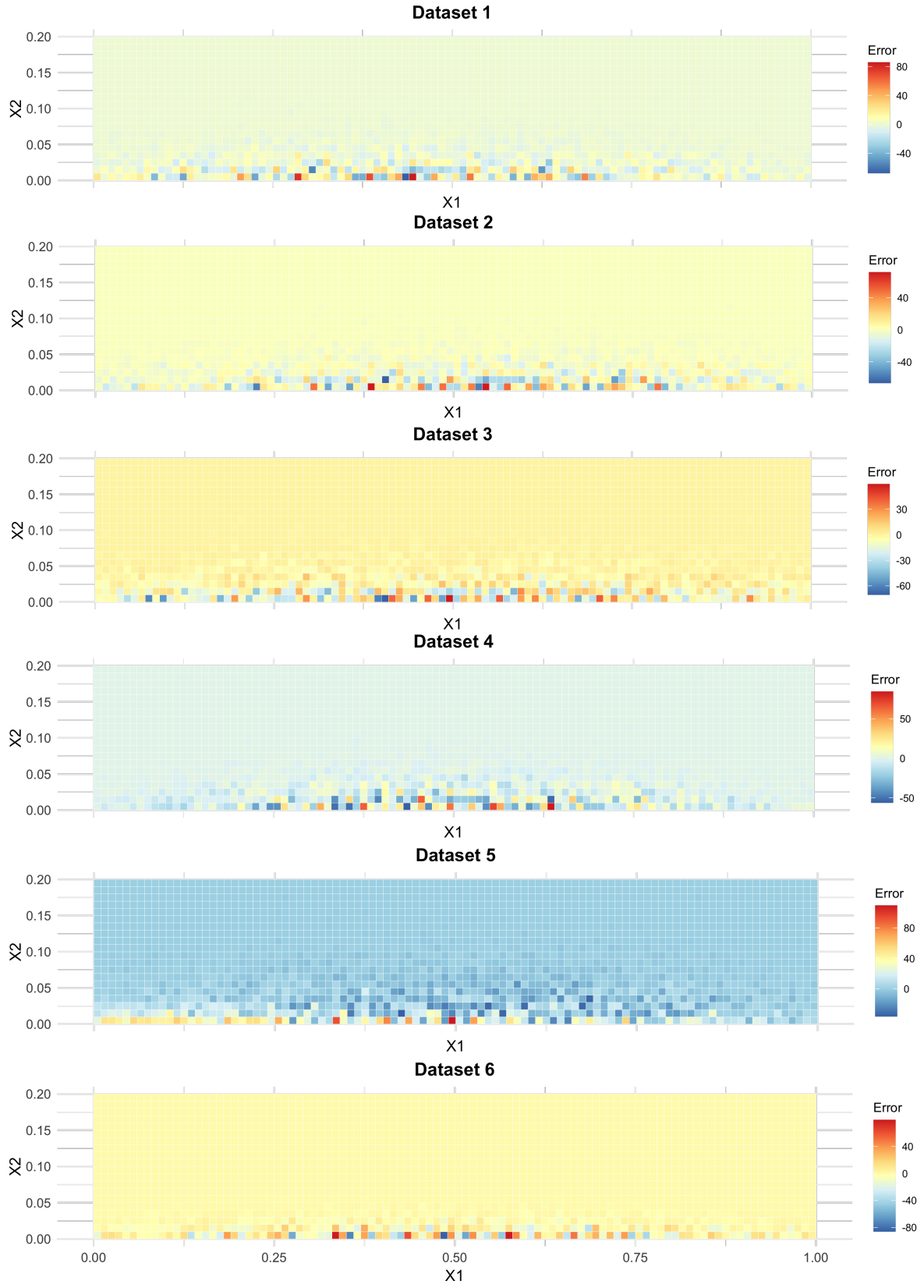

Figure A.1: Average error heatmaps for signal collisions, generated from 100 simulated datasets for each original dataset using the new EM algorithm. Values corresponding to $x_2 > 0.15$ are omitted as their average is consistently zero.

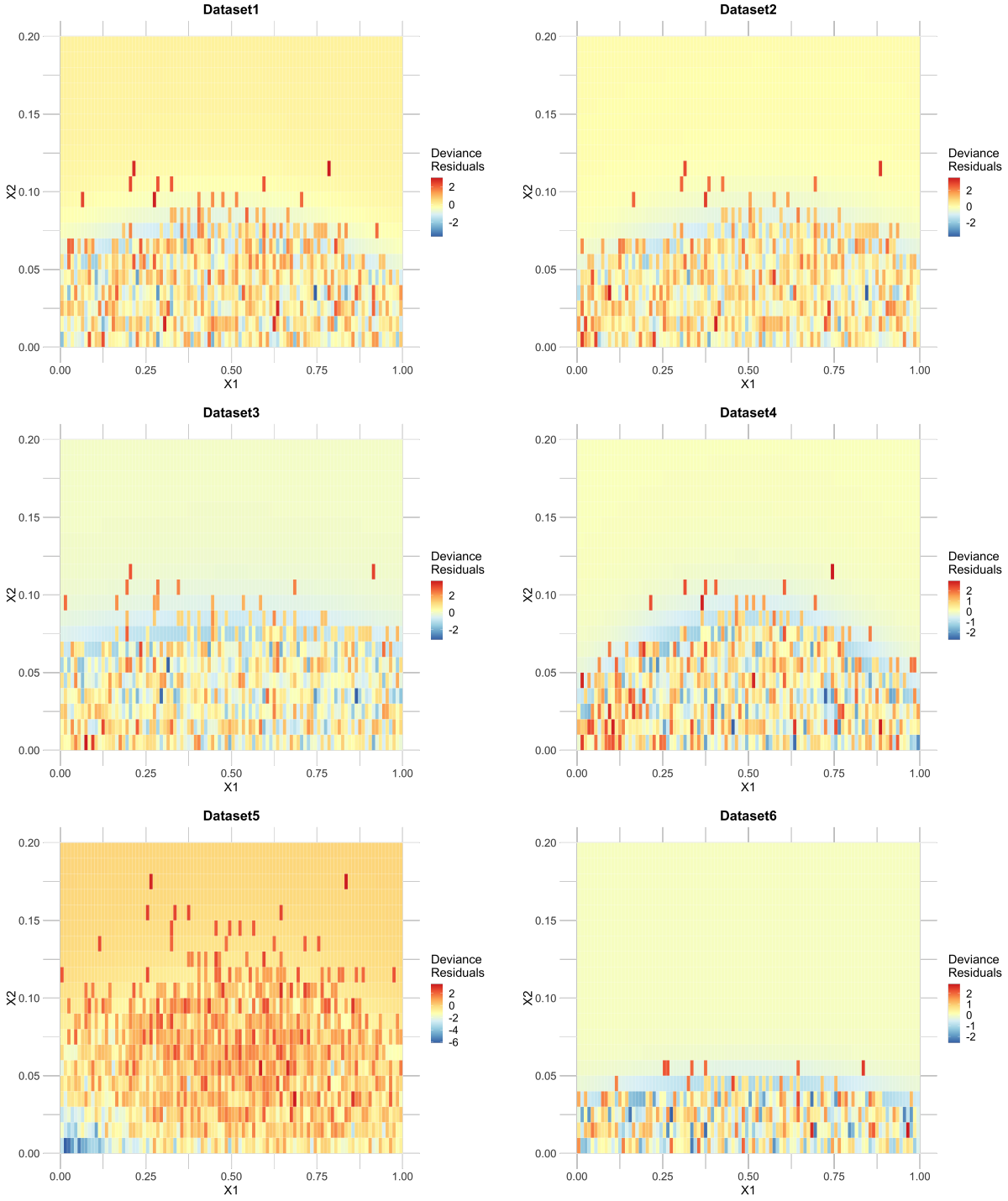

Figure A.2: Heatmap of average deviance residuals from EM Algorithm with more degrees of freedom. These heatmaps illustrate the average deviance residuals computed using the EM algorithm, aggregated the 100 simulations for each original dataset. Values corresponding to $x_2 > 0.15$ are omitted as their average is consistently zero.

Appendix B

Confidence Interval For EM Algorithm

In the main analysis, a 68% Poisson rate-based confidence interval was employed, following the formula $\lambda \pm \sqrt{\lambda}$, to evaluate the EM algorithm's accuracy on simulated data. This approach, while standard for preliminary analysis, was chosen for its simplicity and generality rather than its specificity to the modeling process at hand. It is important to clarify that this form of confidence interval is a broad tool used for initial assessment of model performance and not uniquely tailored to the nuances of our EM algorithm-based GAM model.

For a more tailored approach to confidence interval estimation that aligns closely with the intricacies of our modeling procedure, the Louis' method offers a promising alternative. This method involves calculating the observed information matrix, which can be quite complex for models fitted by the EM algorithm. The Louis' formula enables us to account for the uncertainty in both the E-step and M-step of the algorithm, leading to a more accurate representation of confidence intervals for the estimated parameters. The approach introduced by Louis (1982) simplifies the process of calculating the observed information matrix, which is a vital component in determining the standard error associated with maximum likelihood estimates (MLEs). For readers primarily interested in the final results, you may wish to start directly at the result (B.10).

We will discuss methods for calculating standard errors or the entire covariance matrix of MLEs within the framework of the EM algorithm. Unlike Newton-type algorithms, which automatically provide an estimate of the covariance matrix for the MLE, the EM algorithm does not inherently offer this. Our focus will be on a technique for deriving the covariance matrix of the MLE $\hat{\theta}$ obtained via the EM algorithm, using the observed information matrix $I(\hat{\theta}; y)$. We will explore approaches to compute $I(\hat{\theta}; y)$ within the EM algorithm. In situations where the observed information matrix is assumed to be independent, $I(\hat{\theta}; y)$ can be estimated conveniently, without requiring additional computational effort beyond the initial MLE calculation. Prior to delving into Louis' formula, it is crucial to understand key concepts such as the score statistic,

the missing information principle, and the observed information matrix. This detailed discussion and the methodologies described here are based on McLachlan and Krishnan (2007, p. 95).

Using the same notations as in the EM algorithm section, let Y be the random vector corresponding to the observed data y having probability density function $g(y;\theta)$. We denote y as the incomplete data, z the unobserved data and we will refer to x as the complete data. We can establish the relation $y = y(x)$ where $y \in \mathcal{Y}$ and $x \in \mathcal{X}$. We let $g_c(x;\theta)$ denote the probability density function corresponding to the random vector X for the complete data vector x and let $\log L_c(\theta) = \log g_c(x;\theta)$ denote the complete-data log likelihood function. We establish

$$S(y;\theta) = \frac{\partial}{\partial\theta} \log L(\theta)$$

as the derivative vector of the log likelihood function $L(y;\theta)$. This vector, referred to as the score statistic, is based on the observed data y, which might include incomplete entries. For the complete data scenario, the corresponding derivative vector of the log likelihood function is represented as

$$S_c(x;\theta) = \frac{\partial}{\partial\theta} \log L_c(\theta).$$

In the case of data with missing values, the score statistic, denoted $S(y;\theta)$, is computed as the expected value of the complete data score statistic, expressed as

$$S(y;\theta) = \mathbb{E}_\theta\{S_c(X;\theta) \mid y\}.$$

To see this, we note that

$$\begin{aligned} S(y;\theta) &= \frac{\partial}{\partial\theta} \log L(\theta) \\ &= \frac{\partial}{\partial\theta} \log g(y;\theta) \\ &= \frac{g'(y;\theta)}{g(y;\theta)} \\ &= \frac{1}{g(y;\theta)} \int_{\mathcal{X}(y)} g_c'(x;\theta)\mathrm{d}x, \end{aligned} \tag{B.1}$$

with the prime symbol indicating differentiation with respect to θ. By multiplying and dividing the integrand by $g_c(x;\theta)$, we get

$$\begin{aligned} S(y;\theta) &= \int_{\mathcal{X}(y)} \left\{ \frac{\partial}{\partial\theta} \log g_c(x;\theta) \right\} \frac{g_c(x;\theta)}{g(y;\theta)} \mathrm{d}x \\ &= \int_{\mathcal{X}(y)} \left\{ \frac{\partial}{\partial\theta} \log L_c(\theta) \right\} k(x \mid y;\theta)\mathrm{d}x \\ &= \mathbb{E}_\theta \left\{ \frac{\partial}{\partial\theta} \log L_c(\theta) \mid y \right\} \\ &= \mathbb{E}_\theta \left\{ S_c(X;\theta) \mid y \right\}, \end{aligned} \tag{B.2}$$

where $k(x \mid y;\theta) = g_c(x;\theta)/g(y;\theta)$. Having established the key results for the score statistic, we now turn our attention to the information matrix.

We define the matrix $I(\theta; y)$ as

$$I(\theta; y) = -\frac{\partial^2 \log L(\theta)}{\partial\theta\partial\theta^\top},$$

which represents the negative second-order partial derivative matrix of the log likelihood function for incomplete data with respect to the parameter θ. Given certain regularity conditions, the expected Fisher information matrix, denoted as $\mathcal{I}(\theta)$, can be computed as

$$\begin{aligned}\mathcal{I}(\theta) &= \mathbb{E}_\theta[S(Y;\theta)S^\top(Y;\theta)] \\ &= \mathbb{E}_\theta[I(\theta; Y)].\end{aligned}$$

With respect to the complete-data log likelihood, we define

$$I_c(\theta; x) = -\frac{\partial^2 \log L_c(\theta)}{\partial\theta\partial\theta^\top}.$$

The expected information matrix for the complete data is then given by

$$\mathcal{I}_c(\theta) = -\mathbb{E}_\theta\left[I_c(\theta; X)\right].$$

Using the previous notation $k(x \mid y;\theta) = g_c(x;\theta)/g(y;\theta)$, we have that

$$\log L(\theta) = \log L_c(\theta) - \log k(x \mid y;\theta).$$

By differentiating both sides of the previous equation twice with respect to θ and taking the negative, we obtain

$$I(\theta; y) = I_c(\theta; x) + \frac{\partial^2 \log k(x \mid y;\theta)}{\partial\theta\partial\theta^\top}.$$

Taking the expectation given y of both sides of the equation, we find

$$I(\theta; y) = \mathcal{I}_c(\theta; y) - \mathcal{I}_m(\theta; y), \tag{B.3}$$

where

$$\mathcal{I}_c(\theta; y) = \mathbb{E}_\theta\left[I_c(\theta; X) \mid y\right]$$

is the conditional expectation of the complete-data information matrix $I_c(\theta; X)$ given y, and

$$\mathcal{I}_m(\theta; y) = -\mathbb{E}_\theta\left[\frac{\partial^2 \log k(X \mid y;\theta)}{\partial\theta\partial\theta^\top} \mid y\right]$$

is the expected information matrix for θ based on x when conditioned on y.

In practical applications, it is a common to estimate the inverse of the covariance matrix for the MLE $\hat{\theta}$ using the observed information matrix $I(\hat{\theta}; y)$. One approach is to directly compute $I(\hat{\theta}; y)$ after obtaining the MLE $\hat{\theta}$. However, evaluating the

second-order derivatives of the incomplete-data log likelihood, $\log L(\theta)$, analytically can be challenging or at least time-consuming. Louis (1982) illustrates that the missing information matrix, denoted as $\mathcal{I}_m(\theta; y)$, can be expressed as

$$\begin{aligned}\mathcal{I}_m(\theta; y) &= \mathrm{cov}_\theta\{S_c(X;\theta) \mid y\} &\text{(B.4)}\\ &= \mathbb{E}_\theta\{S_c(X;\theta)S_c^\top(X;\theta) \mid y\} - S(y;\theta)S^\top(y;\theta), &\text{(B.5)}\end{aligned}$$

since

$$S(y;\theta) = \mathbb{E}_\theta\{S_c(X;\theta) \mid y\}.$$

By first replacing (B.4) and then (B.5) into (B.3), we have that

$$\begin{aligned}I(\theta; y) &= \mathcal{I}_c(\theta; y) - \mathcal{I}_m(\theta; y)\\ &= \mathcal{I}_c(\theta; y) - \mathrm{cov}_\theta\{S_c(X;\theta) \mid y\}\\ &= \mathcal{I}_c(\theta; y) - \mathbb{E}_\theta\{S_c(X;\theta)S_c^\top(X;\theta) \mid y\} + S(y;\theta)S^\top(y;\theta). &\text{(B.6)}\end{aligned}$$

The derivation of result (B.6) is achieved by showing that $\mathcal{I}_m(\theta; y)$ can be formulated as presented on the right-hand side of (B.4). Louis confirms (B.6) by manipulating the expression for the negative of the second derivative of $\log L(\theta)$. From (B.1), $I(\theta; y)$ can be calculated as

$$\begin{aligned}I(\theta; y) &= -\frac{\partial}{\partial\theta}S(y;\theta)\\ &= -\frac{\partial}{\partial\theta}\{\int_{\mathcal{X}(y)} \frac{g_c'(x;\theta)}{g(y;\theta)}\mathrm{d}x\}\\ &= -\frac{1}{g(y;\theta)}\int_{\mathcal{X}(y)} \frac{\partial^2 g_c(x;\theta)}{\partial\theta\partial\theta^\top}\mathrm{d}x\\ &\quad + \frac{1}{g_c(x;\theta)^2}\left\{\int_{\mathcal{X}(y)} g_c'(x;\theta)\mathrm{d}x\right\}\left\{\int_{\mathcal{X}(y)} g_c'(x;\theta)\mathrm{d}x\right\}^\top. &\text{(B.7)}\end{aligned}$$

It is assumed that the necessary regularity conditions are met to allow the interchange of differentiation and integration operations. Following the same approach as used in deriving (B.2), we find

$$I(\theta; y) = -\frac{1}{g(y;\theta)}\int_{\mathcal{X}(y)} \frac{\partial^2 g_c(x;\theta)}{\partial\theta\partial\theta^\top}\mathrm{d}x + S(y;\theta)S^\top(y;\theta), \qquad \text{(B.8)}$$

by using the result (B.1) for the last term on the right-hand side of (B.7). The first

term on the right-hand side of (B.8) can be expressed as

$$
\begin{aligned}
-\frac{1}{g(y;\theta)}\int_{\mathcal{X}(y)}\frac{\partial^2 g_c(x;\theta)}{\partial\theta\partial\theta^\top}\mathrm{d}x &= -\frac{1}{g(y;\theta)}\int_{\mathcal{X}(y)}\frac{\partial^2 \log g_c(x;\theta)}{\partial\theta\partial\theta^\top}g_c(x;\theta)\mathrm{d}x \\
&\quad -\frac{1}{g(y;\theta)}\int_{\mathcal{X}(y)}\{\frac{g_c'(x;\theta)}{g_c(x;\theta)}\}\{\frac{g_c'(x;\theta)}{g_c(x;\theta)}\}^\top g_c(x;\theta)\mathrm{d}x \\
&= \int_{\mathcal{X}(y)} I_c(\theta;x)k(x\mid y;\theta)\mathrm{d}x \\
&\quad -\int_{\mathcal{X}(y)} S_c(x;\theta)S_c^\top(x;\theta)k(x\mid y;\theta)\mathrm{d}x \\
&= \mathbb{E}_\theta\{I_c(\theta;X)\mid y\} - \mathbb{E}_\theta\{S_c(X;\theta)S_c^\top(X;\theta)\mid y\} \\
&= \mathcal{I}_c(\theta;y) - \mathbb{E}_\theta\{S_c(X;\theta)S_c^\top(X;\theta)\mid y\}. \qquad \text{(B.9)}
\end{aligned}
$$

Substitution of (B.9) into (B.8) gives the expression (B.6) for $I(\theta;y)$. From (B.6), the observed information matrix $I(\hat{\theta})$ can be computed as

$$
\begin{aligned}
I(\hat{\theta};y) &= \mathcal{I}_c(\hat{\theta};y) - \mathcal{I}_m(\hat{\theta};y) \\
&= \mathcal{I}_c(\hat{\theta};y) - [\mathrm{cov}_\theta\{S_c(X;\theta)\mid y\}]_{\theta=\hat{\theta}}. \qquad \text{(B.10)}
\end{aligned}
$$

Louis' method offers a valuable simplification for computing the observed information matrix by using the relationship between the missing information matrix $I_m(\theta;y)$ and the conditional moments of the score and curvature of the complete-data log likelihood function. This approach proves particularly advantageous within the EM algorithm framework, where the direct analytical evaluation of second-order derivatives of the incomplete-data log likelihood can often be challenging or time-consuming.

By employing the definition of the E-step, denoted as

$$
Q(\theta;\theta^{(0)}) = \mathbb{E}_\theta[\log L_c(\theta)\mid y],
$$

we can express $\mathcal{I}_c(\hat{\theta};y)$ as

$$
\mathcal{I}_c(\hat{\theta};y) = -\mathbb{E}_\theta\left[\frac{\partial^2 \log L_c(\theta)}{\partial\theta\partial\theta^\top}\mid y\right] = -\left[\frac{\partial^2 Q(\theta;\theta_o)}{\partial\theta\partial\theta^\top}\right]_{\theta_o=\theta}.
$$

With Louis' formula, we can compute the observed information matrix and invert it to obtain an estimate of the covariance matrix for the MLE $\hat{\theta}$. This estimation is crucial for computing standard errors and confidence intervals for the parameter estimates obtained through the EM algorithm.

These calculations, while providing a more accurate measure of uncertainty in the parameter estimates, require intensive computation that is specific to the model's structure and the data at hand.

The computation of confidence intervals using the Louis' method is theoretically more appropriate for our modeling framework. However, it requires a detailed, hand-crafted approach to GAM modeling. This process is intricate and time-consuming, involving iterative calculations that are beyond the scope of this current report. With

additional time and resources, this could be explored to potentially yield more precise and tailored confidence intervals for the challenge set.

Due to these constraints, the present analysis will not include these advanced confidence interval computations. Nonetheless, it is important to acknowledge their relevance and potential for future studies, where a more refined and model-specific approach to uncertainty quantification is warranted.

Bibliography

Davison, A. C. (2020) Poisson process. Lecture Notes.

Demmel, J. (1997) *Applied Numerical Linear Algebra*. SIAM.

Duchon, J. (1977) Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables*, eds W. Schempp and K. Zeller, pp. 85–100. Springer.

Kingman, J. F. C. (1993) *Poisson Processes*. Oxford: Clarendon Press.

Louis, T. A. (1982) Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society: Series B* **44**, 226–233.

McCullagh, P. and Nelder, J. A. (1983) *Generalized Linear Models*. Chapman and Hall.

McLachlan, G. J. and Krishnan, T. (2007) *The EM Algorithm and Extensions*. Second edition. Wiley.

Tufte, E. R. (1983) *The Visual Display of Quantitative Information*. Graphics Press.

Wood, S. N. (2017) *Generalized Additive Models*. Second edition. Chapman and Hall/CRC.