Instituto Tecnológico Autónomo de México

UN MODELO PROBIT BAYESIANO NO LINEAL

Tesis

que para obtener el título de

licenciado en matemáticas aplicadas

presenta

GIANPAOLO LUCIANO RIVERA

Asesor: JUAN CARLOS MARTÍNEZ-OVANDO

Ciudad de México 2018

"Con fundamento en los artículos 21 y 27 de la Ley Federal del Derecho de Autor y como titular de los derechos moral y patrimonial de la obra titulada "**UN MODELO PROBIT BAYESIANO NO LINEAL**", otorgo de manera gratuita y permanente al Instituto Tecnológico Autónomo de México y a la Biblioteca Raúl Bailléres Jr., la autorización para que fijen la obra en cualquier medio, incluido el electrónico, y la divulguen entre sus usuarios, profesores, estudiantes o terceras personas, sin que pueda percibir por tal divulgación una contraprestación".

Gianpaolo Luciano Rivera

Fecha

Firma

Resumen

En respuesta al cambiante mundo del *Aprendizaje de Máquina*, se desarrolla, desde sus cimientos, un modelo aplicable a esta categoría. El modelo, no es más que un modelo lineal generalizado, particularmente un probit, que busca la predicción de variables binarias. A este, se le añadió un proyector aditivo no lineal para transformar a las covariables y lograr captar patrones más complejos. La transformación esta basada en polinomios por partes de continuidad arbitraria los cuales se explican en detalle. Bajo un paradigma de aprendizaje bayesiano, se desarrolla un algoritmo eficiente para su entrenamiento. Posteriormente, este algoritmo se implementa en un paquete para el software estadístico `R`. Usando este paquete, finalmente, se prueba y valida el modelo, presentan una variedad de resultados, que exponen de forma clara las capacidades de el modelo. **Palabras clave: modelos lineales generalizados, probit, modelos aditivos, bayesiano, no lineal, machine learning, splines, polinomios por partes.**

A mi madre, Irma, que no sólo le debo la vida, sino todo lo que soy y todo lo que tengo; que está tesis también es suya.

A mi padre, Antonio, que aunque no esté, siempre lo he sentido presente.

A mi otro padre, Fernando, por su guía, apoyo incondicional y sobre todo el amor que siempre me ha dado.

A mi abuelo, Carlos, por enseñarme el amor a la sabiduría y el valor del trabajo.

A Paulina, que es lo mejor que me ha pasado en la vida.

A Iñigo por su amistad sin medida, por su apoyo, compañía e inteligencia fuera de este mundo.

A los DonChis: Pamela, Rodrigo, Brenda, Hector y George. Por ser los mejores amigos que alguien podría pedir, las personas más exitosas que conozco y por siempre inspirarme a crecer y seguir adelante.

A Luis, Jime, Eli, Charly, Mercy, Santiago y Tulio por acompañarme en la carrera y rompernos más de una vez la cabeza en demostraciones oscuras.

A Toño, Chris, Edu, Mau y Luis V. por ser los mejores actuarios que conozco.

A Jorge, por enseñarme de perseverancia y resiliencia y por ser uno de mis más viejos amigos.

A Pau C. y a Fernanda, por ser algunas de las mejores amigas que podría pedir.

A todos mis alumnos, porque más que un negocio, fueron un medio para consolidar mis conocimientos y siempre aprender más.

A mi asesor, Juan Carlos Martínez-Ovando que, aunque no siempre fácil, sin su guía y consejo jamás habría logrado avanzar de la primera página.

A los grandes profesores que tuve en la carrera, que no sólo me enseñaron, sino que me inculcaron el amor por las matemáticas. En particular al profesor E. Barrios, M. Gregorio, J. Alfaro, R. Espinoza, G. Gravinsky y Z. Parada

A mis profesores de matemáticas y física del Green Hills que plantaron en mi la curiosidad por sus temas y siempre creyeron en mi, impuslandome a seguir mis sueños.

A lo grandes estadistas que han dedicados sus vidas a los datos. En particular a T.Hastie, R. Tibshirani y J. Friedman que sin sus contribuciones, hubiera estado perdido.

Al Café Parabien por ser un espacio de trabajo y un hogar para mi los meses de arduo trabajo.

Índice general

Índice de figuras

Índice de tablas

Notación y abreviaciones

Datos y variables

$y_i \in \{0, 1\} \quad \forall i = 1 \ldots, n$: variables de respuesta binarias. Usualmente representadas por el vector $\mathbf{y} = (y_1, \ldots, y_n)^t$

$\mathbf{x}_i \in \mathcal{X}^d \subseteq \mathbb{R}^d \quad \forall\ i = 1 \ldots, n$: covariables o regresores. Si se usa por si sola x o $\mathbf{x}$ (vector), esta representa una variable arbitraria. Si se habla de toda la matriz de datos, se denota por $\mathbf{X} \in \mathcal{X}^{n \times d} \subseteq \mathbb{R}^{n \times d}$. Juntos con las y_i, se tienen los datos para el modelo: $\{(y_i, \mathbf{x}_i)\}_{i=1}^n$

$n \in \mathbb{N}$: número de observaciones en la muestra.

$d \in \mathbb{N}$: número de covariables, dimensionalidad de los regresores.

$\mathcal{X}^d$: subconjunto de $\mathbb{R}^d$, espacio de covariables. Formado por el producto punto de los rangos de cada variable: $\mathcal{X}^d = [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_d, b_d]$, donde

$[a_j, b_j] \subset \mathbb{R}$ es un intervalo cerrado estándar en los reales.

Especificos del modelo

$z_i \sim N(\cdot) \forall i = 1 \ldots, n$: variables latentes del modelo cuya distribución es normal . Usualmente se acomodan en un vector $\mathbf{z} = (z_1, \ldots, z_n)^t$

$f(\mathbf{x})$: función de proyección.

$f_j(x_j) \quad \forall j = 1, \ldots, d$: polinomio por partes anidada en la función de proyección. Sirve para hacer una transformación no lineal de la dimensión J. En ocasiones se acomodan en su forma vectorial $\mathbf{f}(\mathbf{x})$. Ver ecuación (2.7)

$\beta = (\beta_0, \beta_1, \ldots, \beta_j)^t$: vector de coeficientes para la regresión lineal.

$w_j = (w_{j,1}, \ldots, w_{j,N^*})^t \quad \forall j = 1, \ldots d$: vector de pesos para las funciones base de la transformación lineal en j. Si se habla de todos los pesos en conjunto, esos se acomodan en una matriz $\mathbf{w} = [w_1, \ldots w_d]^t \in \mathbb{R}^{N^* \times d}$

$\Psi_j(\cdot) = (\Psi_{j,1}(\cdot), \ldots, \Psi_{j,N^*}(\cdot))^t$: vector de funciones base para los polinomios por parte flexibles. Ver ecuación (2.16) y (2.17)

N^*: número total de funciones base. Ver ecuación (2.14) para su expansión final.

M: número de bases para los polinomios por partes. M-1 indica el grado de

los polinomios.

J: número de sub-intervalos en los que se parte cada $[a_j, b_j]$.

K: número de restricciones de continuidad impuestas.

$\mathcal{P}_j = \{\tau_1, \ldots, \tau_{J-1}\} \quad \forall j = 1, \ldots, d$: partición del espacio de la dimensión j.

τ: nodos, se omiten los índices para evitar confusión, pero se tienen un total de $d * (J-1)$ nodos acomodados en una matriz de igual tamaño.

Contadores e índices

i: contador, usado para denotar un conjunto de observaciones *hacia abajo*, i.e. $i = 1, \ldots, n$. En la sección 2.3.1 se usa para contar sobre el grado del polinomio i.e. $i = 1, \ldots, M$. (Ver ecuación 2.16)

j: contador, usado para denotar el conjunto de variables *a lo largo*, i.e. $j = 1, \ldots, d$. Usualmente se hace referencia a la dimensión arbitraria j. En la sección2.3.1 se usa para contar sobre los nodos entre intervalos i.e. $j = 1, \ldots, J-1$. (Ver ecuación 2.16)

k: contador, usado para denotar el número de iteración en el algoritmo, i.e. $k = 1, 2, 3, \ldots$

l: contador adicional, asociado al número de funciones base N^* constante para

cada dimensión j.

Probabilidad

$F(\cdot)$: Distribución arbitraria de la familia exponencial.

$N(\cdot|\mu, \sigma^2)$: distribución normal con su correspondiente parametrización de media y varianza. Se utiliza la misma notación para su forma vectorial añadiendo un subíndice indicando su dimensionalidad: $N_\cdot(\cdot|z, \sigma)$ con su correspondiente vector de medias μ y vector de varianza covariazna Σ.

$\Phi(\cdot) : \mathbb{R} \to (0, 1)$: la función de distribución acumulada de una distribución normal estándar $N(\cdot|1, 0)$, con su correspondiente inversa Φ^{-1}.

$\text{Be}(\cdot|p)$: distribución bernoulli con probabilidad de éxito p.

$p \in [0, 1]$: probabilidad arbitraria.

$g(\cdot)$: función liga. Ver diagrama 2.2

ϵ: errores aleatorios, usualmente distribuidos $N(\epsilon|\mu, \sigma^2)$.

$P(\cdot)$, $\mathbb{E}[\cdot]$, $\mathbb{V}[\cdot]$: medida de probabilidad, operadores de esperanza y varianza respectivamente.

$\theta \in \Theta$ parámetros canónicos de distribuciones exponenciales, con Θ su corres-

pondiente espacio.

$\pi(\cdot)$: función de densidad.

α: símbolo de proporcionalidad.

$S(\cdot|\cdot)$: función de suavizamiento.

ρ: correlación.

Algoritmo

N_{sim}: número de simulaciones realizadas en el algoritmo.

k^*: número de observaciones por descartar, periodo de *burn-in*.

k_{thin}: parámetro de adelgazamiento.

r_{j^*}: residuales parciales para alguna j^* en particular.

Otros

$h(\cdot)$: función arbitraria.

$h^{(k)}$: (k)-ésima derivada de h.

$s : \mathbb{R} \to (0, 1)$: familia de funciones sigmoidales.

I: función indicadora.

$(\cdot)_+$: función parte positiva.

$\mathbf{1}$: vector de números uno.

ll: función *log-loss*.

El símbolo $\hat{\cdot}$ se usa para indicar que se trata de una variable estimada, i.e. $\hat{y}$ es la estimación de las variables correspondientes y.

Abreviaciones

GAM : *Generalized aditive model*, Modelo aditivo generalizado.

GLM : *Generalized linear model*, Modelo lineal generalizado

MCMC : *Markov Chain Monte Carlo*.

ML : *Machine Learning*, Aprendizaje de máquina.

RSS : *Residual sum of squares*, Suma de residuales cuadrados.

Capítulo 1

Introducción

En luz de las nuevas y populares tendencias en el mundo de la estadística computacional, llamada en ocasiones aprendizaje estadístico u aprendizaje de máquina,[1] este trabajo, busca desarrollar y entender desde sus cimientos, un modelo aplicable a esta categoría. Este modelo, buscará hacer inferencia sobre una base de datos y *aprender* sobre los patrones subyacentes que estos puedan contener. Se busca revisar todos los aspectos de su construcción: desde consideraciones teóricas hasta diferentes paradigmas de aprendizaje, así como su implementación y validación.

Este tipo de modelos, han resultado ser de enorme efectividad en ámbitos que van desde la medicina hasta las finanzas. En ocasiones sin embargo, por su complejidad

1. *machine learning (ML)*

dad, los métodos de ML son tratadas como *cajas negras* computacionales; se tienen datos que se alimentan a un modelo complejo y este arroja resultados. Sin dudarlo útiles, el tratamiento de los datos y el modelo en si no se debe dejar de un lado, pues, existen consideraciones técnicas y supuestos que se deben cumplir. Asimismo, la interpretación, validación y análisis de los resultados, deben ser realizados por alguien que conozca, al menos de manera general, lo que está haciendo el algoritmo empleado por la computadora.

En particular, el modelo presentado a continuación, realiza la estimación de variables binarias en un contexto de regresión bayesiana a través de un proyector aditivo con una transformación no lineal de los datos. Esta maraña de términos técnicos, se irá esclareciendo poco a poco conforme se construye el modelo. En el fondo, el modelo busca encontrar patrones de segmentación. Esto lo logra, clasificando cada una de las n observaciones como *éxito o fracaso*, *positivo o negativo*, *hombre o mujer* o cualquier otra posible respuesta binaria y_i, dependiendo de información adicional $\mathbf{x}_i$ conocida como covariable donde $i = 1, \ldots, n$. El problema radica en que la información adicional es compleja y puede contener patrones difícil de identificar, lo cual, hace que distinguir entre los posibles resultados de y_i sea difícil. En la Figura 1.1 se tiene un ejemplo gráfico de este tipo de clasificadores. Con algunos de los modelos tradicionales, por construcción, llevar a cabo esta clasificación sería imposible.

Para llevar a cabo esta construcción, se comienza con una extensa discusión teórica y matemática. No obstante, se hace énfasis en el desarrollo del algoritmo, esto, pues

la implementación del modelo es fundamental para su aplicación practica.

> *[...], it is more common in machine learning to view the model as core, and how this is implemented is secondary. From this perspective, understanding how to translate a mathematical model into a piece of computer code is central.*[2]

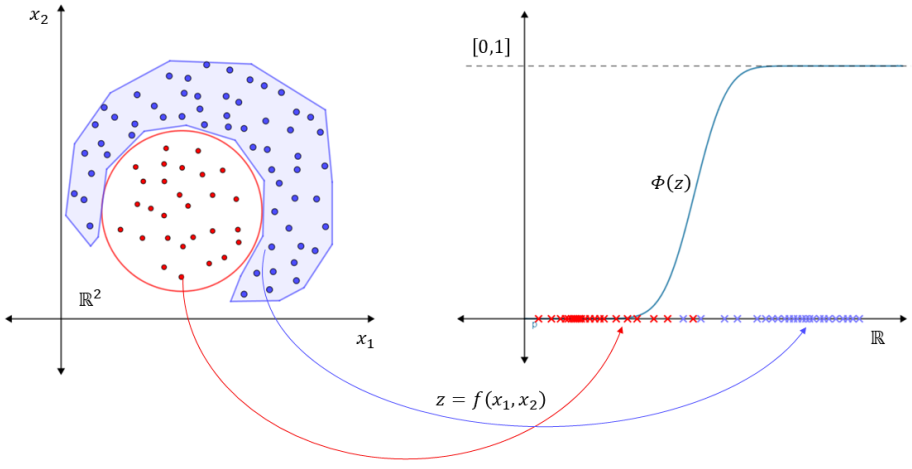


Figura 1.1: Diagrama explicativo de un modelo de clasificación no lineal

Se tienen observaciones del grupo azul y del grupo rojo con una clara separación no lineal en las covariables x_1 y x_2. El modelo busca *entrenar* una función f que logre separar lo mejor posible este espacio. Posteriormente, esta separación, induce una clasificación (0 y 1 correspondiendo a rojo y azul respectivamente) a través de la función de distribución normal Φ.

Es fundamental entender a fondo cada pedazo del modelo, por ello, en el Capítulo 2

2. Barber (2012)

se hace una exploración de su forma matemática más rigurosa. Dada su estructura, el modelo se puede estudiar de arriba hacia abajo, es decir, de la parte más general a la parte más profunda. Por lo tanto, primero se estudian los Modelos Lineales Generalizados (GLM), específicamente los modelos probit asociados a la distribución normal. Los GLM dan el salto de una regresión donde la respuesta y_i es real, a regresiones donde la respuesta, puede ser discreta o restringida a cierto dominio (MacCullagh y Nelder 1989). Los GLM, como su nombre lo indica, siguen siendo lineales, pero este proyector lineal se puede flexibilizar un poco más usando las ideas de los Modelo Aditivo Generalizado (GAM) presentadas en Hastie y Tibshirani (1986). Los GAM, buscan transformar a las covariables $\mathbf{x}_i$, previamente a la regresión usando métodos no paramétricos. Este trabajo, toma esas ideas y las combina con las de Denison, Mallick y Smith (1998), en el que se llevan un paso más allá la transformación para hacerla *tan flexible como sea posible*. Esta transformación, corresponde a una serie de polinomios por partes de continuidad y grado arbitrarios, sujetos a ciertos nodos, lo cual representa la parte más profunda del modelo. La expansión que se presenta, resulta que conectan muchas disciplinas y ramas de las matemáticas que han sido de mucha utilidad no sólo en el campo de la estadística. Al final del Capítulo 2, se verá que con estos principios se abre un mundo de posibilidades en cuanto a modelos y datos sobre los que se pueden aplicar.

Posteriormente en el Capitulo 3, se hace una breve introducción a la estadistica bayesiana, en particular al apredizaje bayesiano en el contexto de regresión lineal. Esto se debe a que, usando las ideas de Albert y Chib (1993), el algoritmo asociado al modelo recae en una técnica fundamental de esta diciplina: el *Gibbs sampler*.

Con esta poderosa herramienta, se presenta los detalles y lógica detrás de la implementación. Además, se explica a grandes razgos como se hizo el desarrollo y de un paquete computacional de código abierto en el software `R` para el uso del modelo. El desarrollo de un paquete se detalla en el Apéndice B, y corresponde a que, no sólo se simplifica el proceso de apredizaje del modelo, sino que se fomentando su fácil uso y su validación por terceras personas que se pudieran interesaran en el. Se hace notar, que al paquete se le añadió funcionalidad adicional para la visualización de ciertas partes del modelo bajo algunos supuestos facilitando su interpretación.[3]

Una vez que el modelo fue funcional y fácil de implementar, se probó y se validó contra una serie de bases de datos, tanto simulados como reales para probar su efectividad. En el Capítulo 4, se puede estudiar el modelo en una forma más pragmática, pues el uso del paquete lo facilita mucho. En particular, las bases de datos simuladas ejemplifican muy bien las matemáticas detrás del modelo y muestran la flexibilidad de los polinomios por partes, pues, logran encontrar fronteras de clasificación complejas y evidentemente no lineales. Uno de los ejemplos replica de forma fiel, la Figura 1.1.

Finalmente, en el Capítulo 5, se verán las consideraciones finales y limitaciones del modelo. Sin embargo, se abre una discusión a posibles extensiones para mejorarlo. Posteriormente, se da un vistazo a modelos más modernos los cuales han sido capaces de proezas computacionales que se creían imposibles hace algunas décadas. Se verá, sin embargo, que muchos de los modelos más avanzados y usados hoy en

3. El paquete se puede descargar libremente de: `https://github.com/PaoloLuciano/bpwpm`

día, son generalizaciones de los modelos tradicionales presentados en este trabajo. Si estos modelos, se comienzan a anidar unos dentro de los otros se logra extender el *aprendizaje* más allá de datos binarios y lograr clasificaciones de imágenes, sonidos y datos poco ortodoxos para la estadística, pero comunes en la actualidad.

Capítulo 2

Modelo en su forma matemática

Como base fundamental de este trabajo, a continuación, se expondrá a detalle la formulación matemática. A grandes rasgos, se tiene un modelo de clasificación supervisada. Por lo tanto, se irá construyendo construir un clasificador binario flexible con buena capacidad predictiva. La notación se irá explicando conforme aparece pero existe un compendio en el preámbulo del trabajo. En general, se trata de respetar la notación que usan en los libros de Hastie, Tibshirani y Friedman (2008) y James y col. (2013).

Se supone la siguiente estructura en los datos:

$\{(y_i, \mathbf{x}_i)\}_{i=1}^{n}$ con n el tamaño de la muestra.

$y_i \in \{0, 1\} \quad \forall i = 1 \ldots, n$ variables de respuesta binarias o *output*.

$\mathbf{x}_i \in \mathcal{X}^d \subseteq \mathbb{R}^d \quad \forall\, i = 1 \ldots, n$ covariables, regresores o *input*.

$d \in \mathbb{N}$ dimensionalidad de las covariables.

El modelo, se presenta a continuación de forma general para cualquier pareja de datos $(y, \mathbf{x})$:

$$y \,|\, z \;\sim \mathrm{Be}(y \,|\Phi(z)) \tag{2.1}$$

$$z \,|\, x \;\sim \mathrm{N}(z \,|f(\mathbf{x}), 1) \tag{2.2}$$

$$f(\mathbf{x}) \approx \sum_{j=0}^{d} \beta_j f_j(x_j) = \beta^t \mathbf{f}(\mathbf{x}) \tag{2.3}$$

$$f_j(x_j) \approx \sum_{l=1}^{N^*} w_{j,l} \Psi_{j,l}(x_j, \mathcal{P}_j) \quad \forall j = 0, 1, \ldots, d \tag{2.4}$$

En las expresiones (2.1) y (2.3), se tiene una versión ligeramente modificada de un GLM (Sec. 2.1). Esto, pues la variable de respuesta binaria y es perfectamente modelada a través de una distribución Bernoulli. En (2.3), f es una función de proyección lineal como las que se usan el los modelos de regresión lineales tradicionales. En el contexto de un modelo de clasificación, esta función f busca separar el espacio d-dimensional de covariables $\mathcal{X}^d$ en regiones identificables donde se tengan *éxitos o fracasos*. Lo logra, colapsando $\mathcal{X}^d$ en una sola dimensión $\mathbb{R}$ donde se puede hacer la clasificación con la ayuda de la variable latente z. Esta variable latente vista en la

ecuación (2.2), es la liga entre y y $\mathbf{x}$. Cabe mencionar, que es meramente estructural pues es modelada a través de una distribución normal, lo cual deriva en un modelo probit.

La función de proyección f, asume que la dependencia entre covariables se puede modelar como la suma ponderada de los componentes f_j (Sec. 2.2). En las ecuaciones (2.4) se definen las funciones f_j las cuales son transformación no lineales para cada dimensión j. Su trabajo, es el de tratar de encontrar las tendencias individuales de cada una de las covariables. Lo logran, haciendo un suavizamiento por medio de polinomios por partes que dependen de 3 componentes: una partición del intervalo $\mathcal{P}_j$, un vector de pesos w_j y parámetros que captura la N^* especificando la forma y grado de los polinomios. La forma funcional de Ψ es compleja y relativamente arbitraria dependiendo de la selección de la base, por lo tanto, no se especifican aún y se posterga para la Sección 2.3. Se hace notar que el componente bayesiano se explora hasta el Capítulo 3 pues va estrechamente ligado con su implementación. En la Figura 2.1 se hace una representación visual del modelo para su mejor comprensión.

Antes de continuar, vale la pena mencionar que al trabajar bajo la perspectiva de modelación:

> *All models are wrong but some are useful*[1]

Escoger un modelo que explique perfectamente los datos o que logre predecir todo sería una tarea inútil. Sin embargo, no significa que no se pueda intentar discernir

1. Box (1979)

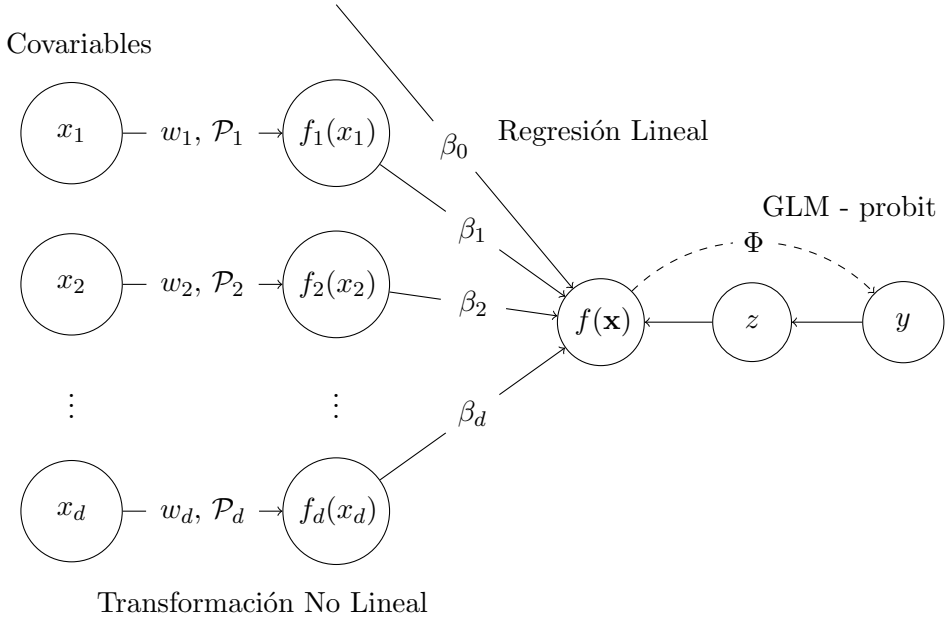


Figura 2.1: Diagrama del modelo

Se hace una transformación no lineal de cada una de las covariables x_j a través de los parámetros w_j, P_j y N^* por medio de las funciones f_j. Con los datos transformados, se lleva a cabo un modelo probit con función liga Ψ para lograr la clasificación binaria en y.

un patrón y esto, es justamente lo que se busca con la construcción de este modelo. Además de entender a profundidad la base de modelos más avanzados que se están usando en el mundo de la inteligencia artificial. En particular, este modelo tiene la ventaja que es flexible y, al menos en teoría, debería de servir para representar una gran cantidad de datos.

2.1. Modelos lineales generalizados (GLM)

Los modelos lineales generalizados, surgen como una generalización del modelo lineal ordinario

$$y = \beta^t x + \epsilon$$

donde $y \in \mathbb{R}$, β es un vector de parámetros y ϵ es error estadistico (Sundberg 2016), (MacCullagh y Nelder 1989). En esta generalización, se busca darle rangos diferentes a y pues, se tienen casos donde los datos están restringidos a un subconjunto de los reales como lo es el caso binario. Sin embargo, este cambio vuelve el modelo más complejo y deriva en diversas técnicas para la estimación de β. Asimismo, el cambio hace que se pierda algo de la interpretabiliad del modelo. Dependiendo de la especificación, su interpretación puede ser complicada.[2] Sin embargo, estos modelos han resultado ser realmente útiles.

Los GLM se especifican (de manera muy general) de la siguiente manera:

$$\begin{aligned} y &\sim F(\theta(x)) \\ z &= \beta^t x \\ \theta &= g^{-1}(z) \end{aligned} \tag{2.5}$$

con los siguientes tres elementos:

2. Por ejemplo, cuando se tiene un modelo logit, se logra expresar el logaritmo de la proporción de probabilidades (*Log-Odds-Ratio*) como una combinación lineal de las covariables. $\ln(\pi_i/\pi_0) = \beta^t x$.

F: tipo de distribución de la familia exponencial que describe el dominio de las respuestas y. Por ejemplo: Bernoulli si y es binaria, Poisson si $y \in \mathbb{Z}^+$ o una distribución Gamma si $y \in \mathbb{R}^+$

z: proyector lineal que explique (linealmente) la variabilidad sistemática de tus datos.[3]

g: función liga que una la media (o los parámetros canónicos) θ de mi distribución con el proyector lineal. Es decir: $\theta(x) = \mathbb{E}[y|x] = g^{-1}(\beta^t x)$.

g puede ser cualquier función monotónica diferenciable. Como ejemplos clásicos se tiene la función $\text{logit}(p) = \ln(p/(1-p))$ o la $\text{probit(p)} = \Phi^{-1}(p)$, donde $p = \mathbb{E}[y|x]$ y $\Phi(\cdot)$ es la función de acumulación de una distribución normal estándar.

En la Figura 2.2 se presenta una representación gráfica de g para su mejor comprensión.

Para este trabajo, se busca construir un clasificador binario donde $y \in \{0, 1\}$, por lo cual, es natural modelar y como una distribución Bernoulli. Notese que si $Y \sim \text{Be}(y|p)$ se tienen las siguientes propiedades:

$$\mathbb{E}[Y] = p = P(y = 1)$$
$$\mathbb{V}[Y] = p(1-p)$$

3. Como restricción adicional, en el modelo clásico se pide que $\dim(\beta) = d < n$.

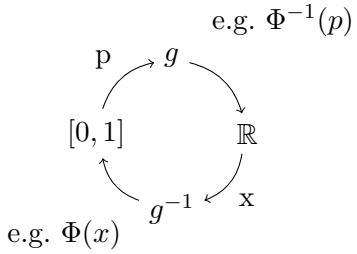


Figura 2.2: Esquema de función liga g

Por lo que con un solo parámetro p, la varianza queda determinada automáticamente. Además, esta especificación deja como opción para las función liga, a las inversas de las funciones *sigmoidales* $s(x)$. Las funciones sigmoidales, son funciones $s : \mathbb{R} \to (0, 1)$, estrictamente monótonas y por ende, biyectivas. Algunos ejemplos son las ya mencionadas logit, probit y la curva de Gompertz[4]. Estas funciones cumplen un papel de activación, es decir, una vez que se rebase cierto umbral, crecen rápidamente y toman valores más cercanos a uno, *activando* asi la probabilidad de que y sea un éxito.[5] Las propiedades de las funciones liga, las hacen perfectas herramientas para ligar el proyector lineal $z \in \mathbb{R}$ con probabilidades $p \in (0, 1)$.

4. Para no caer en redundancia de notación para este trabajo se tiene a partir de ahora: $s(x) = g^{-1}(x) = \Phi(x)$

5. En un contexto de redes neuronales, lo que se activa es la neurona y recientemente, se usa la función *ReLu* $(x) := \text{máx}\{0, x\}$

2.1.1. Uso de la variable latente

Para entender el papel de z, se necesita entender que es posible estructurar estos modelos como *modelos de variable latente* (Albert y Chib 1993). Bajo esta formulación, se asume que la relación entre y y x no es directa, sin embargo, existe una variable no observada z estructural que ayuda a discernir un vínculo entre ellas. En la Figura 2.3 se tiene una representación gráfica de esto.

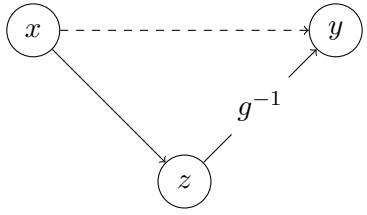


Figura 2.3: Modelo de variable latente

Tradicionalmente, la normalidad en z es derivada de suponer normalidad en los errores; es decir, dada la regresión lineal $z = \beta^t x + e$ se supone (y se debe verificar) que $e \sim N(0, \sigma^2)$. Lo cual lleva a $z \sim N(\beta^t x, \sigma^2)$. Además este supuesto facilita la estructura de los modelos y el algoritmo de ajuste. Bajo un paradigma frecuentista, la estimación de los parámetros β se reduce a encontrar los estimadores de mínimos cuadrados. Sin embargo, bajo el paradigma bayesiano, dentro de un modelo probit como el de este trabajo, se adopta la normalidad en z pues en Albert y Chib (1993), se sugiere un algoritmo *Gibbs sampler* con distribuciones truncadas de la normal para encontrar β.

La función liga probit se escoge como consecuencia de la normalidad en z, o viceversa,

dependiendo de como se quiera ver. Esta función $\Phi^{-1}(p)$ es la inversa de la función de acumulación normal estándar:

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}\, dt$$

la cual no tiene forma analítica cerrada, sin embargo, es claramente sigmoide. Esta función, cumple el propósito de modelar y cuantificar la incertidumbre pues está transformando la variable real z en una probabilidad p con su característica forma de "s". Se hace notar, que se podría haber usado una función más flexible o que, incluso, se podría dejar la función como una parte del modelo a estimar. Sin embargo, al adoptar el algoritmo antes citado, se requiere esta especificación.[6] La parte flexible de este modelo se encuentra en el proyector lineal.

Habiendo definido la función liga, distinguir entre si $y = 1$ ó 0, éxito o fracaso respectivamente, se reduce a distinguir en que área del espacio de covariables $\mathcal{X}^d$ se encuentra el dato. Esto se debe a que $y = 1$ cuando $\Phi(z) > 1/2$ que sucede *si y sólo si* $z > 0$ lo cual, depende en gran media de su media; en este caso la función de proyección $f(\mathbf{x})$. Si esta función es muy positiva en alguna región, implicará que el modelo tiene mucha evidencia para confiar que, al menos en esa área, la respuesta y es un éxito. El razonamiento, funciona de forma análoga para los casos donde $y = 0$, claramente, para esas regiones, se busca que $f(\mathbf{x})$ sea negativa. Por lo tanto, es fundamental para el modelo que se realice una correcta estimación de

6. En modelos multinomiales bayesianos, tomar esta decisión estructural lleva a que se puede asumir una estructura de interdependencia en los errores aleatorios. $\epsilon_k \sim N_k(0, \Sigma)$ con Σ una matriz de correlaciones

los parámetros de la función de proyección. Nótese además, que z le agrega cierta *estocasticidad* al modelo. Bajo la suposición que existe una pareja $(y_i, \mathbf{x}_i)$ tal que $f(\mathbf{x}_i) = 0$; alrededor de una vecindad de este punto, no se tendría evidencia para clasificar a y_i como un éxito o como un fracaso; sería mejor dejar la clasificación a la suerte.

Otro factor importante a considerar, es que el modelo supone que la varianza de z es constante, específicamente $\sigma^2 = 1$. Dado que la escala de z es completamente arbitraria pues es una variable auxiliar, se puede *restringir* z al rango que se desee. El método de simulación para z usando una distribución normal truncada se simplifica ligeramente usando esta varianza unitaria. Se verá en los resultados, sin embargo, que dada la naturaleza global de los polinomios que se usan, la escala de z, o al menos la estimación de su media $\hat{f}(\mathbf{x})$, puede variar mucho dependiendo de los datos, mas esto no representa un problema. Pues, en la practica, al usar el algoritmo de Albert y Chibb z sirve mucho más, para hacer la ligadura de y hacia f y no viceversa. En z se codifica, mediante una distribución normal truncada, los casos de éxito y de fracasos de y; posteriormente, se estima el vector β para la función f. Por ello, en la Figura 2.1, se representan las flechas de y a f por medio de z y solidas, en contraposición con la flecha punteada que va, directamente de f a y y pasa por la función Φ. Los detalles y su justificación probabilista, se tocan en detalle en el Capítulo 3.

Es importante mencionar, que el *corte* que se hace en $z = 0$ para la clasificación, es resultado de hacer una clasificación binaria. En modelos multinomiales también se debe tomar en cuenta los intervalos en $\mathbb{R}$ para los que la observación se clasificaría

en alguna de las posibles clases y por ende, estimar los umbrales o usar una función diferente a las sigmoides. Este hecho, lleva a la realización de que z y su media f son *ajenas más no independientes*. Esto quiere decir que la parametrización de z como una normal $N(\mu, 1)$ es equivalente a la parametrización $N(0, 1)$. Este hecho se hará más claro cuando se hable del papel de β_0 en la función de proyección.

Finalmente, si se quisiera ver la relación de x en y directamente, se puede lograr usando el teorema de la probabilidad total. Se puede calcular (al menos de forma teórica), la distribución marginal de y dado $\mathbf{x}$ sumando sobre z:

$$\begin{aligned} P(y|x) &= \int_{-\infty}^{\infty} P(y|z)P(z|x)\,dz \\ &= \int_{-\infty}^{\infty} p(y;\Phi(z))p(z;f(\mathbf{x}),\sigma^2)\,dz \\ &= \int_{-\infty}^{\infty} \Phi(z)^y(1-\Phi(z))^{1-y}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{1}{2\sigma^2}(z-f(\mathbf{x}))^2}\,dz \end{aligned}$$

Sin embargo, está claro que esta derivación no lleva a ningún resultado analítico cerrado pues la relación es bastante más compleja como para resultar en una distribución tradicional; si lo hiciera, el propósito de la z se perdería.

Recapitulando, mediante la función liga Φ se une la media p, la probabilidad de éxito o fracaso de la respuesta y con los datos $\mathbf{x}$. Esto se logra, a través de una variable

auxiliar z cuya media $f(\mathbf{x})$ es una función de proyección lineal.

$$P(y = 1) = p(\mathbf{x}) = \mathbb{E}[y|\mathbf{x}] = g^{-1}(f(\mathbf{x})) = \Phi(f(\mathbf{x})) \tag{2.6}$$

2.2. Función de proyección f

En la sección anteriór, se vió que tradicionalmente, se asumía z como una combinación lineal de los parámetros β y las covariables x.[7] Pero, como se explica en la página 6 de James y col. (2013), conforme avanzaron los métodos y el poder computacional disponible se fueron desarrollando técnicas cada vez más poderosas que permitieron romper la linealidad. En 1986, Hastie y Tibshirani introducen los modelos aditivos generalizados (GAM), una clase de modelos donde se rompe la linealidad en las covariables, flexibilizando aún más el modelo.

Se hace notar que esta generalización es sutil pues el modelo aún conserva una parte lineal *a lo largo*.[8] Se ve en la ecuación (2.3) que el modelo aún es lineal en las β_j y en las f_j. Donde se pierde la linealidad es *hacia abajo*,[9] pues cada f_j en realidad es la transformación no lineal de la covariables x_j. Además, las dimensiones j's se asumen independientes entre si pues, se busca que corresponden a diferentes *características*

7. Segunda ecuación de (2.5)

8. Con esta frase se hace alusión a que, en una tabla de datos de tamaño $n \times d$, siendo cada fila una observación y cada columna una variable, el *largo* se piensa como la segunda dimensión de tamaño d. Por lo tanto, al usar esta expresión se busca considerar todas las variables. En este trabajo y en su implementación, se usa el subíndice j para denotar esta idea.

9. De forma análoga, esta idea, hace alusión a las diferentes observaciones. Denotado por el subíndice i, el cual no se ha usado para simplificar la notación.

o variables con las que se planea modelar la variable de respuesta.

En este modelo, el proyector f de la ecuación (2.3), no hace otra cosa más que ir colapsando dimensiones, en particular: $\mathcal{X}^d \to \mathbb{R}$ que posteriormente se colapsa por medio de Φ en $[0, 1]$. Es por esto que se le llama función de proyección, pues *proyecta* el espacio $\mathcal{X}^d$ en $\mathbb{R}$. Sin embargo, la forma en la que lo haga, debe de ser lo más precisa posible pues el modelo recae en que este colapso detecte los patrones correctos en las covariables que llevan a la correcta identificación de y. Por lo tanto, f como función de proyección es el corazón del modelo, por lo que su correcto entrenamiento es fundamental. La idea, recapitulando, es que f separe el espacio de covariables para que sea positiva en las regiones donde se tengan éxitos y negativa donde se tengan fracasos; para ello, es fundamental entender los GAM.

2.2.1. Modelos aditivos generalizados (GAM)

Un GAM como se introduce en Hastie y Tibshirani (1986) remplaza la forma lineal $\sum_1^d \beta_j x_j = \beta^t x$ con una suma de funciones *suaves* $\sum_j^d f_j(x_j)$. Estas funciones no tienen una forma analiítica cerrada y son no especificadas, es decir, no hay tienen una forma funcional concreta y representable algebraicamente. Donde recae la fuerza de estos modelos, es que se estiman usando técnicas no paramétricas de suavizamiento[10] como lo sería un suavizamiento loess. En estos modelos, se supone que por más

10. Las técnicas no paraméticas están fuera del alcance de este trabajo. Sin embargo, vale la pena una mención especial por su funcionalidad, practicidad y forma intuitiva, además del sinfín de aplicaciones que tienen. Una guía comprensiva de estas se encuentra en el libro Wasserman (2007).

grande que sea $\mathcal{X}^d$, la relación que existe entre cada una de las dimensiones j, se puede explicar de manera aditiva, es por ello que cada función f_j tiene como argumento exclusivamente de x_j. Esta especificación fue revolucionaria pues no sólo regresa interpretabilidad al modelo, sino que simplifica la estimación usando técnicas prácticamente automáticas con el algoritmo de *backfitting*. La idea fundamental de este algoritmo será de vital importancia para el ajuste de el modelo. Los principal ventaja de los GAM es que logran descubrir efectos no lineales en las covariables, justamente lo que se busca. La función f de este trabajo, es una versión versión modificada de un GAM con tres cambios fundamentales.

La primera modificación es que se está ponderando cada f_j por un parámetro β_j, esto, para suavizar aún más cada dimensión y captar el patrón general y no tanto los componentes individuales de cada x_j. Al entender que cada f_j es una transformación no-lineal de x_j (como lo sería una transformación logarítmica o una transformación Box-Cox) se le regresa cierta interpretabilidad al parámetro β_j como el efecto que tiene la dimensión i en particular para el modelo. Asimismo, se reincorpora un término independiente β_0 pues este ayuda a ajustar la escala en la estimación de los parámetros. La inclusión de este parámetro es fundamental para la correcta especificación del modelo pues ayuda a dar un *sesgo o nivel* base contra el cual comparar la suma y escalar la f para que sea compatible con el umbral de corte en 0 haciendo equivalente la parametrización de z con una distribución normal estándar. Por convención $f_0(\cdot) = 1$ por lo que se puede re-expresar la ecuación (2.3) como:

$$f(\mathbf{x}) \approx \beta_0 + \sum_{i=1}^{d} \beta_i f_i(x_i) = \beta^t \mathbf{f}(\mathbf{x}),$$

donde usando notación vectorial $\beta \in \mathbb{R}^{d+1}$ y $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^{d+1}$:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_0 \\ f_1(x_1) \\ f_2(x_2) \\ \vdots \\ f_d(x_d) \end{bmatrix} = \begin{bmatrix} 1 \\ w_1^t \Psi_1(x_1, \mathcal{P}_1) \\ w_2^t \Psi_2(x_2, \mathcal{P}_2) \\ \vdots \\ w_d^t \Psi_d(x_d, \mathcal{P}_d) \end{bmatrix} \tag{2.7}$$

La segunda modificación, se ve en la expresión anterior (2.7). Aunque es muy práctico manejar las f_j's como indeterminadas y estimarlas con procedimientos de suavizamiento no paramétricos, también se puede optar por la vía en la que se especifica su forma funcional, en este caso $w_j^t \Psi_j(\cdot)$. No por ello, se le quita flexibilidad al procedimiento; esto se verá con todo detalle en la sección 2.3, donde se trata de adapar el procedimiento de Denison, Mallick y Smith (1998). Esta modificación, obedece a que para ciertas aplicaciones, sirve hacer el modelo paramétrico en donde cada f_j se modela en su expansión de bases, vease Capitulo 9.1 y Ejemplo 5.2.2 de Hastie, Tibshirani y Friedman (2008).

La tercera modificación, es que en la practica, se tiene una aproximación en vez de una igualdad para f. El simple hecho de asumir que existe una f que puede separar el espacio en regiones positivas y negativas es uno de los supuestos más fuertes del modelo, esto responde a que el *error aleatorio*, sistemático de los datos, está siendo capturado por una aproximación a la f real, dentro de cada una de las f_j's. Bajo el paradigma frecuentista, se desarrolla toda una amplia teoría de convergencia y se explica a detalle el porqué de esta modificación usando técnicas de análisis más

avanzadas (Bergstrom 1985), (Stone 1985).

En la peculiaridad de que $d = 2$, se podrá visualizar $f(\mathbf{x})$ en $\mathbb{R}^3$ como una serie de picos y valles donde será positiva en caso de ser éxito y negativa en caso contrario.

Modelos en bases de funciones lineales

Finalmente, se aclara que este modelo podría ser confundido con un modelo en bases de funciones lineales[11] como los presentados en Capitulo 3 de Bishop (2006):

$$f(\mathbf{x}) = \beta_0 + \sum_{j=1}^{d} \beta_j f_j(\mathbf{x}), \tag{2.8}$$

claramente con una forma funcional simular. La diferencia radica en que cada f_j es función de todas las covariables en lugar de sólo la que le corresponde en el índice. A estas funciones se les conocen como funciones base, sobre las que se hará una exposición más a detalle en la siguiente sección. La f una vez más es lineales en β, no-lineales en $\mathbf{x}$, pero la diferencia radica en que f_j es de naturaleza global. Algunos ejemplos son:

bases gaussianas,

$$f_j(\mathbf{x}) = \exp\left\{-\frac{(\mathbf{x} - \mu_j)^2}{2s^2}\right\}$$

11. *Linear basis function models* por falta de una mejor tradición.

funciones sigmoidales,

$$f_i(\mathbf{x}) = \sigma\left(\frac{\mathbf{x} - \mu_j}{s}\right).$$

Estos modelos son muy útiles en la estimación de mezclas de distribuciones y en la estimación de curvas. Sin embargo, estos son un grupo de modelos diferentes y su aplicación no es tanto inferencial como lo que se busca. En realidad, estos modelos son más una extensión al modelo que se construye en este trabajo.

2.3. Funciones f_j - polinomios por partes

Finalmente se trata la parte más profunda del modelo, las funciones f_j que, como se mencionó anteriormente, son transformaciones no lineales de cada componente x_j. Lo que buscan es suavizar la nube de datos, para posteriormente sumarlas entre si y dar una medida f que resuma toda la información en un número real. Como se menciona en la introducción de Härdle y col. (2004), el suavizamiento de los datos es central en la estadística inferencial. La idea es extraer la señal entre el ruido y para ello, se intenta estimar y modelar la estructura subyacente. Este suavizamiento, se llevará a cabo usando una expansión en bases funcionales, particularmente en polinomios por partes. Toda la siguiente sección se concentra en darle formas funcionales a Ψ y a explicar el papel de los pesos w. Se usa como referencia en la exposición, las primeras dos secciones de el Capitulo 5 de Hastie, Tibshirani y Friedman (2008).

Expansión en bases funcionales

Saliendo por un momento del domino de la estadística, se definen las expansiones en bases de funciones. Sin entrar mucho en los detalles técnicos, dado un espacio funcional, se puede representar cualquiera de sus elementos, en este caso una funciones arbitrarias h, como la combinación lineal de los elementos de la base Ψ (también funciones) y constantes w. En particular (y dados los objetivos del trabajo) se considera el espacio funcional que mapea $\mathbb{R}^d$ a $\mathbb{R}$, quedando entonces la expansión:

$$h(\mathbf{x}) = \sum_{l=1}^{N^*} w_l \Psi_l(\mathbf{x}) = w^t \Psi(\mathbf{x}) \tag{2.9}$$

con el vector de funciones base $\Psi(\mathbf{x})^t = (\Psi_1(\mathbf{x}), \ldots, \Psi_{N^*}(\mathbf{x}))^t$, donde cada elemento Ψ_l es también una función con el mismo mapeado que h, $w^t = (w_1, \ldots, w_{N^*})^t$ un vector de coeficientes constantes y N^* un entero mayor o igual a la dimensión del espacio funcional que se maneja[12].

Regresando a las regresiones en el mundo de la estadística. Se busca representar la media condicional de la respuesta y por una función que depende de los datos: $h(\mathbf{x}) = \mathbb{E}[y\,|\mathbf{x}]$. Se puede pensar, que esta h también puede ser expresada como su expansión en bases funcionales.[13] La idea, es que se remplace (o se aumente) la cantidad de covariables $\mathbf{x}$ con transformaciones de estas, capturadas en el vector $\Psi(\mathbf{x})$. Por ejemplo:

12. Dependiendo de el espacio funcional, en ocasiones $N^* = \infty$
13. Un supuesto fuerte pero necesario en ocasiones.

$\Psi_l(\mathbf{x}) = x_j \quad \forall\, l = 1, \ldots, d$, donde se recupera un GLM tradicional.

$\Psi_l(\mathbf{x}) = \ln x_j$ ó $x_j^{1/2}$ donde se tienen transformaciones no lineales en cada una (o algunas) de las covariables.

$\Psi_l(\mathbf{x}) = ||\mathbf{x}||$ una transformación no lineal de todas las covariables.[14]

$\Psi_l(\mathbf{x}) = x_j^l \quad \forall l = 0, \ldots, N^*$ donde se tiene una expansión en bases polinómicas.

$\Psi_l(\mathbf{x}) = x_j x_k \quad \forall l,$ p.a. j, k donde se incluyen términos de interacción.

Esta representación engloba muchos de los modelos y transformaciones posibles en el mundo de las regresiones, uniendo temas de análisis funcional con estadística aplicada. Además de que en general han resultado ser de gran utilidad en la práctica. Se hace notar que el último ejemplo rompe con la aditividad inherente de las combinaciones lineales, demostrando que esta generalización no está restringida a ser completamente aditiva.

Dependiendo del tipo de datos y el propósito del modelo, puede ser conveniente usar algún tipo de funciones base sobre las otras. Sin embargo, sobre todo cuando se tiene poca o ninguna experiencia con los datos, se busca una representación más flexible (por no decirla ingenua) de éstos. El método más común es tomar una familia grande de funciones que logre representar una gran variedad de patrones.

14. Como se vio en la ecuación (2.8)

Una de estas familias, es la de los polinomios por partes usadas en este modelo. Una desventaja de estos métodos, sin embargo, es que al contar con una cantidad muy grande de funciones base y por ende parámetros, se requiere controlar la complejidad del modelo para evitar el *sobre-ajuste*. Algunos de los métodos más comunes para lograrlo son los siguientes:

Métodos de restricción: donde se selecciona un conjunto finito de funciones base y su tipo, limitando así las posibles expansiones. Los modelos aditivos como los usados en este trabajo, son un ejemplo perfecto de este tipo.

Métodos de selección de variables: como lo son los modelos CART y MARS, donde se explora de forma iterativa las funciones base y se incluyen aquellas que contribuyan a la regresión de forma significativa.

Métodos de regularización: donde se busca controlar la magnitud los coeficientes, buscando que la mayoría de ellos sean cero, como los son los modelos *Ridge* y *LASSO*.

Consideraciones para la expansión en bases de este trabajo

Para simplificar un poco la exposición y reducir la notación, se supone por lo pronto que $d = 1$, por lo tanto $\mathbf{x} = x$ y se puede pensar únicamente en funciones que mapean reales a reales. Esto permite librar el subíndice j para indicar componente del vector $\mathbf{x}$ y usarlo para otros fines.

Para el modelo de este trabajo, se aplicaron las ideas de Denison, Mallick y Smith a los modelos GLM presentados con anterioridad. Los autores presentan un método revolucionario, automático y bayesiano, que permite estimar con un alto grado de precisión relaciones funcionales entre la variable de respuesta y y sus covariable $x \in \mathbb{R}$. En el trabajo original, se buscaba ajustar una curva tal que $y = h(x)$. El modelo en su forma estadística se plantea para un conjunto de datos $\{(x_i, y_i)\}_{i=1}^{n}$:

$$y_i = h(x_i) + e_i \quad i = 1, \ldots, n \tag{2.10}$$

donde las e_i son variables aleatorios con media cero. Este método, combina los procedimientos paramétricos y no paramétricos desarrollados con anterioridad para hacer más robusto el algoritmo de Hastie y Tibshirani (1986). La idea, es ajustar un *polinomio por partes* muy flexible. Estos polinomios, se componen de partes de menor orden entre *nodos* adyacentes. Una de las muchas genialidad de su trabajo es que estos nodos, tradicionalmente fijos, se vuelven parámetros a estimar, usando un paradigma bayesiano. Y no sólo eso, sino que permiten *aumentar o disminuir el número de nodos* desarrollando un algoritmo Gibbs sampler trans-dimensional. Esta generalización, logra estimaciones tan robustas, que logran aproximar funciones continuas *casi en todas partes*, como lo son la función Doppler, funciones por bloques y funciones con picos pronunciados (Denison, Mallick y Smith 1998).

2.3.1. *Splines*

Antes de exponer estos polinomios tan flexibles, se busca entender que son los polinomios por partes simplificando (bastante) el trabajo de Wahba (1990). Sea $x \in [a, b] \subseteq \mathbb{R}$, se busca separar $[a, b]$ en J intervalos. Por lo tanto, se construye una partición correspondiente $\mathcal{P} = \{\tau_1, \tau_2, \ldots, \tau_{J-1}\}$ tal que $a \leq \tau_1 < \ldots < \tau_{J-1} \leq b$. Estas τ's son llamadas *nodos*. Se hace notar, que se puede incluir o no la frontera dependiendo de la especificación.[15] Con estos nodos seleccionados, se puede hacer una representación de h en su expansión de bases como en la ecuación (2.9), donde cada Ψ_j será una función que depende, tanto de la partición como de la variable x. Por ejemplo, se puede pensar en un caso sencillo donde se tiene que $J = 3$ y a cada subintervalo les corresponde una función $\Psi_j \quad j = 1, \ldots, 3$. Simplificando aún más, se hacen que estas Ψ_j's sean funciones constantes en cada intervalo. Por lo tanto, las funciones base son:

$$\Psi_1(x, \mathcal{P}) = I(x < \tau_1)$$
$$\Psi_2(x, \mathcal{P}) = I(\tau_1 \leq x < \tau_2)$$
$$\Psi_3(x, \mathcal{P}) = I(\tau_2 \leq x),$$

con $I(\cdot)$ la función indicadora que vale 1 si x se encuentra en la región y 0 en otro

15. Dependiendo de si se busca hacer inferencia o no fuera del los intervalos.

caso. Por lo tanto, la expansión es:

$$\begin{aligned} h(x) &= \sum_{j=1}^{J} w_j \Psi_j(x) \\ &= w_1 I(x < \tau_1) + w_2 I(\tau_1 \leq x < \tau_2) + w_3 I(\tau_2 \leq x). \end{aligned}$$

Resultando en una función *escalonada*, en el sentido de que para cada región de x se tiene un nivel w_j.[16] Esta aproximación, podría servir para datos que estén agrupados por niveles, sin embargo, rara vez será este el caso.

Con este ejemplo sencillo, se ilustra a grandes rasgos como funcionan los polinomios por partes. Sin embargo, en cada intervalo se puede ajustar un polinomio de grado arbitrario, aumentando así, el número de funciones base. Adicionalmente, se pueden añadir restricciones de continuidad en los nodos, y no sólo continuidad entre los polinomios, sino continuidad en las derivadas, lo cual logra una estimación más robusta. Esta es la magia de los polinomios por partes, que se les puede pedir cuanta *suavidad* (o no) se requiera, entendido como la continuidad de la (K)-ésima derivada. Tradicionalmente, se construyen polinomios cúbicos con segunda derivada continua en los nodos. Esto, pues resultan en curvas suaves al ojo humano, además de que logran aproximar una gran cantidad de funciones.

16. Sin entrar en el detalle, usando una función de perdida cuadrática, es fácil demostrar que cada $\hat{w}_j = \bar{x}_j$ es decir, para cada región, el mejor estimador constante, es el promedio de los puntos de esa región.

Orígenes y justificación de su uso

La palabra *spline* usualmente se usa para designar a un grupo particular de polinomios por parte. Sin embargo, no hay consenso en la literatura de su definición exacta. Dependiendo de las particularidades se pueden denotar funciones diferentes. Para este trabajo se usa la definición de Wasserman (2007) y Hastie, Tibshirani y Friedman (2008). Un *spline de grado* M es un polinomio por partes de grado $M-1$ y continuidad hasta la $(M-2)$-derivada. Se hace notar, que existen muchos tipos de *splines*, además de que pueden ser, puede ser más flexibles o más rápidos en su implementación computacional como los B-Splines. En Boor (1978) y más recientemente Wahba (1990) se hacen tratados extensivos sobre ellos.

Como breviario historico, los splines originales, los desarrolla el matemático I. J. Schoenberg como la solución al problema de encontrar la función h en el espacio de Sobolev W_M de funciones con $M-1$ derivadas continuas y M-ésima derivada integrable al cuadrado que minimice:

$$\int_a^b (h^{(M)}(x))^2\,dx,$$

sujeta a que interpole los puntos $h(x_i) = h_i \quad i = 1, 2, \ldots, n$ (Schoenberg 1964). Posteriormente, la teoría sobre los splines se fue expandiendo y fueron adoptados por ramas de la matemática tan diversas como los gráficos por computadora y, como es el caso, la estadística computacional. Bajo este contexto, los splines también surgen de forma orgánica pues, la ecuación (2.10) se puede plantear como encontrar la función

h que minimice:

$$\sum_{i=1}^{n}(y_i - h(x_i))^2 + \lambda \int_a^b (h^{(M)}(x))^2\, dx, \tag{2.11}$$

para alguna $\lambda > 0$. Donde, la solución se demuestra que son *splines cúbicos naturales* ($M = 4$). Cabe mencionar, que esta formulación del problema engloba muchas de técnicas estadísticas interesantes además de conceptos de optimización. El lector reconocerá que el primer término claramente es la *suma de residuales cuadrados (RSS)* y el segundo término del sumando es un caso particular de los métodos de regularización mencionados anteriormente. No es el enfoque del trabajo entrar en estos detalles pues, cambios menores en la formulación y diferentes elecciones de λ llevan a modelos que cada uno merece una tesis por si mismo. Sin embargo, es importante mencionar que la regularización y modelos de este tipo, son algunos de los más usados y útiles en ML, pues logran captar patrones muy complejos al incluir muchos términos de orden superior e interacciones sin sobreajustar en los datos. Como ejemplo, se puede encontrar fronteras de clasificación circulares usando un modelo logístico normal en $\mathbb{R}^2$ al incluir todos los términos polinomiales y las interacciones hasta orden 6. Por lo pronto, lo esencial en la expresión (2.11) es que al tratar de minimizar el RSS se puede caer en problemas de sobre-ajuste en donde los parámetros no estén capturando efectos y patrones subyacentes, sino sólo se trata de seguir los datos. Para compensar la complejidad, se penaliza la función a minimizar con segundo termino que controla el número de parámetros y la suavidad deseada mediante λ. A este segundo término, se le conoce como *penalización* y crece a medida

que h se vuelve más complicada.[17]

Posterior a estas formulaciones, los splines vuelven a ser relevantes con el modelo aditivo de Hastie y Tibshirani. Ellos extienden la formulación de un espacio de covariables en una sola dimensión, a muchas. La formulación del problema es prácticamente la misma que (2.11) pero ahora se busca estimar d funciones h, dando lugar a tener más parámetros λ:

$$y = \sum_{j=0}^{d} h_j(x_j) + \epsilon$$

$$\text{RSS}(h_0, h_1, \ldots, h_d) = \sum_{i=1}^{n} [y_i - \sum_{j=0}^{d} h_j(x_{ij})]^2 \; + \; \sum_{j=1}^{d} \lambda_j \int h_j^{''}(t_j)\, dt_j$$

con la convención de que h_0 es una constante. Ellos muestran que $h_j \quad j = 1, \ldots, d$ son splines cúbicos. Sin embargo, sin restricciones adicionales, el modelo no sería *identificable*, es decir, la h_0 podría ser cualquier cosa. Para asegurar la unicidad de la solución se añade la condición de que las funciones estimadas, promedien cero sobre los datos:

$$\sum_{i=1}^{n} h_j(x_{ij}) = 0 \quad \forall j \tag{2.12}$$

Esto lleva a la conclusión natural de que h_0 sea la media de las variables de respuesta, es decir: $h_0 = \bar{y}$. Por lo que si se ve cada dimensión j, se tiene que su función

17. Si el lector tiene una intuición de análisis, notará que integrar la función al cuadrado, corresponde con el producto interno de las funciones pertenecientes al espacio de Hilbert $\mathcal{L}_2([a, b])$.

correspondiente h_j está centrada alrededor de la media $\bar{y}$. Esta idea es fundamental para el modelo de este trabajo. En el, se permite que h_j sean *arbitrarias* para toda j, por lo que sólo se necesita que tenga la magnitud necesaria para ajustar los datos. Es decir, dada h_0, la estimación y entrenamiento de los parámetros que definen por completo a h_{j^*} (con j^* alguna $j = 1, \ldots, d$) deben ser tales para que esta ajuste los *residuales parciales*:

$$\hat{h}_{j^*} = y - h_0 - \sum_{\substack{j=1 \\ j \neq j^*}}^{d} h_j \tag{2.13}$$

y se vaya captando en esta h_{j^*} la información aún no captada por el modelo. Esta lógica, además de brillante, es la que le da fuerza a los GAM, pero sólo se puede entender de forma completa hasta que se estudie el algoritmo de *backfitting* en la Sección 3.3.1.

Formalización matemática de *splines*

Retomando la discusión de la página 28, se está buscando definir un polinomio de grado $M - 1$ por partes en J intervalos. Tomando una expansión de bases para cada intervalo, como en el primer ejemplo que se dio, el número de funciones base aumenta en J por cada grado que se agregue, dando un total de $J \times M$ bases funcionales, y en consecuencia, el mismo número de parámetros por estimar. Esto ocurre porque se necesita definir una base de tamaño M para cada subintervalo

$j = 1, \ldots, J$. Es decir: $\mathcal{B}_j = \left\{1, x, x^2, \ldots, x^{M-1}\right\}$, para $j = 1, \ldots, J$. Sin embargo, esto lleva a polinomios que se comportan de forma independiente en cada intervalo y no se conecten. Naturalmente, la primera condición en la que se piensa es imponer continuidad en los nodos, lo cual devuelve $J - 1$ parámetros que corresponden a los $J - 1$ nodos. De la misma forma, cada grado de continuidad nodal en las derivadas que se le pida al polinomio, lo restringe y por ende, devuelve el mismo número de funciones bases. Sea K este número, es decir, se construye un polinomio por partes con continuidad hasta la K-ésima derivada, que tiene un total de:

$$N^*(M, J, K) = M \times J - K(J - 1) \tag{2.14}$$

bases funcionales. Por ende, este polinomio tiene el mismo número de parámetros por estimar w.[18] Es claro que N^* es la *dimensión mínima* necesaria para construir polinomios por partes con estas características. Pues, el número de funciones N^* está, a su vez, en función de M definiendo el grado, el número de intervalos J (por ende el número de nodos) y el número de restricciones K.

Por lo pronto, y para continuar con una exposición constructiva, se centra la discusión cuando $K = M - 1$, devolviendo la definición de spline: polinomios de grado $M-1$ con continuidad hasta la $(M-2)$-derivada. Por ende, $N^* = M+J-1$. Ahora, se recuerda que el objetivo es darle forma funcional a Ψ. Para lograr esto, habiendo

18. En ocasiones es más fácil pensar en K como el número de restricciones que se imponen en los nodos. Así, $K = 0$ implica que los intervalos son independientes, $K = 1$, implica que los polinomios se conectan, $K = 2$ implica continuidad en la primera derivada y así sucesivamente. Naturalmente $K < M$

incorporado el número de bases, se define la función auxiliar *parte positiva*:

$$x_+ = \text{máx}\left\{0, x\right\}.$$

Esta función, ayuda a se pueda representar la expansión en bases de una forma relativamente sencilla. A esta expansión, se le conoce como *expansión en bases truncada*:

$$\begin{aligned} h(x) &= \sum_{i=1}^{M+J-1} w_i \Psi_i(x, \mathcal{P}) \\ &= \sum_{i=1}^{M} w_i \; x^{i-1} + \sum_{j=1}^{J-1} w_{M+i} \; (x - \tau_i)_+^{M-1}. \end{aligned} \tag{2.15}$$

El primer sumando de (2.15) representa el *polinomio base*[19] de grado $M-1$ que afecta a todo el rango. El segundo sumando, está compuesto únicamente de funciones parte positivas que se van activando a medida que x recorre el rango $[a, b]$ a la derecha y va pasando por los nodos. Estas funciones parte positiva, capturan el efecto de todos los intervalos anteriores que, al combinarlos con el primer sumando definen un polinomio de grado $M-1$ en todo el intervalo.[20] Esta derivación de las bases, surge cuando se integra un polinomio por partes constante $M-1$ veces. En cada iteración, las constantes se juntan y se integran por si solas, independientemente de los intervalos, lo cual deriva en este polinomio base. De forma explicita, se tiene que

19. *Baseline*, una vez más a falta de una mejor traducción
20. En realidad lo hace en todo $\mathbb{R}$

$\Psi(x, \mathcal{P})$ es:

$$\begin{aligned}
&\Psi_1(x, \mathcal{P}) = 1 \\
&\Psi_2(x, \mathcal{P}) = x \\
&\vdots \\
&\Psi_M(x, \mathcal{P}) = x^{M-1} \\
&\qquad \text{(el } \textit{polinomio base}\text{)} \\
&\Psi_{M+1}(x, \mathcal{P}) = (x - \tau_1)_+^{M-1} \\
&\vdots \\
&\Psi_{M+J-1}(x, \mathcal{P}) = (x - \tau_{J-1})_+^{M-1} \\
&\qquad \text{(la base truncada)},
\end{aligned}$$

las cuales forman un espacio lineal de funciones $(M + J - 1)$-dimensional. En la particularidad que $M = 4$, se les conoce como splines cúbicos y son los más usados cuando se buscan funciones suaves. En la práctica han resultado ser de gran utilidad pues el ojo humano no detecta la posición de los nodos.

2.3.2. Polinomios por parte flexibles

Independientemente de la elección de parametros en la construcción del polinomio, se tiene el problema de seleccionar la posición de los nodos. Existen procedimientos adaptativos, como los propuestos en Friedman (1991). No obstante, y como ya se

mencionó anteriormente, Denison, Mallick y Smith (1998), proponen un método bayesiano más atractivo, que aunque no se implemente en este trabajo, se implementa su expansión en bases aún más general. Esta es una ligera modificación a la ecuación (2.15) la cual la convierte, de un spline, a un polinomio por partes más general, con grado arbitrario de continuidad en las derivadas. Dejando atrás el supuesto que $K = M-1$ y devolviendole esa flexibilidad al modelo. Su expansión en bases resulta:

$$h(x) = \sum_{l=1}^{N^*} w_l \; \Psi_l(x, \mathcal{P}) = w^t \Psi(x, \mathcal{P}) \qquad \text{con } N^* = J \times M - K(J-1) \tag{2.16}$$

$$= \sum_{i=1}^{M} w_{i,0} \; x^{i-1} + \sum_{i=K}^{M-1} \sum_{j=1}^{J-1} w_{i,j} \; (x - \tau_j)_+^i \tag{2.17}$$

la cual es la expansión de bases implementada en el modelo final.

Dado que se tiene una doble suma, es necesario incluir un segundo índice, al menos temporalmente, a los pesos. El primer índice, denotado por i está asociado al grado de su función base; si $i = 2$ entonces, $w_{2,j}$ está asociado a una término de grado 1 cuando $j = 0$, pero a uno de grado 2 si $j > 0$.[21] El segundo índice $j = 1, \ldots, J-1$ denota el nodo al que está asociado el peso. Como convención, si $j = 0$, se hace referencia al polinomio base que siempre tiene efecto. En el segundo sumando de (2.17) la primera suma comienza en K. Recordando, K es el número de restricciones de continuidad que se imponen al polinomio en los nodos. Por ejemplo, $K = 0$ implicaría que cada polinomio es independiente; $K = 2$, se tiene continuidad en la función y en la primera derivada, etc. En el caso que $K = M - 1$ se regresa a

21. Esta desgraciada disparidad en la notación surge para ser consistente con la anterior, y no se puede indexar directamente en el primer sumando.

la ecuación (2.15) y se recuperan los splines que, por construcción, son suaves. La suavidad, aunque útil, no siempre es necesaria. Existen muchas funciones con primera y segunda derivada que varían rápidamente e incluso funciones discontinuas que no se podrían estimar usando splines, todo depende de los datos. Esta construcción, con su doble suma, permite tener $M - K$ términos por nodo, codificando así las continuidades arbitrarias en las derivadas.[22] La ecuación (2.16) es una vez más la expansión en bases arbitrarias, igual a (2.9) pero definiendo de forma completa a N^*. Además, si finalmente en esta ecuación se deja que $h(x)$ sea igual a $f_j(x_j)$ para toda $j = 1, \ldots, d$, se regresa a la ecuación canónica del modelo (2.4) presentada al principio de este trabajo. Este era el último componente que quedaba por definir, completando así la exposición matemática del modelo.

Para ayudar con la interpretación y lectura de la ecuación (2.17), la Tabla 2.1, de la página 39, hace un compendio de los polinomios por partes. Esto ayuda no sólo a esclarecer la notación, sino a formar una biyección entre w_l, $w_{i,j}$ y Ψ_l que posteriormente ayudará a expresar todo de forma matricial en su implementación en código.

Antes de cerrar la sección, se centra la atención en los nodos τ. A estos, se les ha dado poca importancia hasta el momento. Como ya se mencionó antes, en Denison, Mallick y Smith (1998) se desarrolla, además de la ecuación (2.17) un paradigma bayesiano en el que los nodos, son tratados como parámetros y por ende sus posiciones

22. Esta codificación es sutil pues, al hacer los cálculos de continuidad, hay que considerar los límites izquierdos y derechos, los cuales existen siempre. Sin embargo, los términos $(x - \tau)_+^K$ se desvanecen únicamente hasta la K-esima derivada. Para la $(K + 1)$-derivada, el coeficiente correspondiente se suma a la función y rompe la continuidad pues no corresponde con el límite izquierdo

w_l	$w_{i,js}$	$\Psi_l(x,\mathcal{P})$		
Subíndice l	Subíndices i,j	Función Base		
1	$1,0$	1	M elementos	
2	$2,0$	x		
$\vdots$	$\vdots$	$\vdots$		
M	$M,0$	x^{M-1}		
$M+1$	$K,1$	$(x-\tau_1)_+^K$	$M-K$	$J-1$ veces
$M+2$	$K+1,1$	$(x-\tau_1)_+^{K+1}$		
$\vdots$	$\vdots$	$\vdots$		
$M+(M-K)$	$M-1,1$	$(x-\tau_1)_+^{M-1}$		
$M+(M-K)+1$	$K,2$	$(x-\tau_1)_+^K$	$M-K$	
$M+(M-K)+2$	$K+1,2$	$(x-\tau_1)_+^{K+1}$		
$\vdots$	$\vdots$	$\vdots$		
$M+2(M-K)$	$M-1,2$	$(x-\tau_1)_+^{M-1}$		
$\vdots$	$\vdots$	$\vdots$		
$M+(J-2)(M-K)+1$	$K,J-1$	$(x-\tau_{J-1})_+^K$	$M-K$	
$M+(J-2)(M-K)+2$	$K+1,J-1$	$(x-\tau_{J-1})_+^{K+1}$		
$\vdots$	$\vdots$	$\vdots$		
$M+(J-1)(M-K)$	$M-1,J-1$	$(x-\tau_{J-1})_+^{K+1}$		

Tabla 2.1: Biyección entre w_l, $w_{i,j}$ y sus correspondientes funciones base Ψ_l.

Se termina con $N^* = M + (J-1)(M-K) = J \times M - K * (J-1)$ términos, ecuación (2.14). Por construcción, se es consistente con la definición (2.15) si $K = M-1$.

cambian. La ventaja de que estos estén indeterminados, es que se pueden concentrar en los lugares donde la función varia más. Y al contrario, si la función es relativamente suave para alguna sección, se usan pocos nodos. Aunque hubiera sido bueno implementar este proceso, el algoritmo que *mueve* los nodos va ligado directamente a un proceso de eliminación y nacimiento de estos, haciendo que la J sea variable. En el trabajo original, esto no era un problema pues sólo se hacían estimaciones para una dimension, $d = 1$. En el contexto de este modelo probit, implementar el algoritmo trans-dimensional que los autores proponen, hubiera implicado que N^* estrella, no fuera constante para toda variable, $j = 1, \ldots, d$. Sino que se tendría N_j^*, incorporado otra capa de complejidad innecesaria. Además, el algoritmo habría sido radicalmente diferente. En el Capítulo 3 se detalla como la simplificación de no incorporar los nodos como parámetros ayuda bastante a la velocidad del algoritmo. Posteriormente en el Capítulo 4, se ve que para fines prácticos, el modelo funciona de maravilla y finalmente en el Capítulo 5 se discute que habría cambiado de haberse implementado.

2.3.3. Consideraciones matemáticas adicionales

A pesar de la utilidad de los splines (y los polinomios por parte), todos sufren de problemas más allá del rango de entrenamiento $[a, b]$. Pues, su naturaleza global hace que fuera de la región con nodos, los polinomios crezcan o decrezcan rápidamente. Por lo tanto, extrapolar con polinomios o splines es peligroso y podría llevar a estimaciones erróneas. Para corregir esto, en ocasiones, se puede imponer una restricción

adicional para que el polinomio sea lineal en sus extremos. Se usa el adjetivo de *natural* para designarlos. Esta modificación, libera $2(M-2)$ funciones bases, pues quita todas las bases de orden mayor a 1 en los dos nodos frontera. Su expansión en bases, también se deriva de la ecuación (2.15). Es razonable que esta modificación mejore la fuerza predictiva fuera de el dominio de entrenamiento. Sin embargo, en general, en un contexto de regresión, se recomienda no hacer inferencia fuera de el espacio de covariables $\mathcal{X}$, pues en realidad, no se tiene evidencia para tomar conclusiones en esta región. Todo depende de los datos y el objetivo del modelo.

Se han usando los parámetros M, J y K para hablar del número de funciones base N^*, ecuación (2.14), pero se recuerda que también, dictan el número de *grados de libertad* del modelo. Es decir, el número de pesos o coeficientes w, los cuales son igual o más importantes que las bases, no sólo porque son parámetros a estimar, sino que son los que dictan el *ajuste* a los datos a diferencia de Ψ que únicamente los operan.

Al estar trabajando en espacios funcionales, la elección de base es relativamente arbitraria y se podría cambiar como lo hace una transformación de coordenadas en un espacio euclidiano. Cada base tiene sus beneficios y desventajas. Para esta exposición, se escoge la expansión en bases truncadas pues es explicada fácilmente y tiene una forma funcional relativamente sencilla además, la interpretación de los coeficientes w es inmediata. Sin embargo, no es óptima computacionalmente cuando J es grande. En la practica, usualmente se implementan B-Splines[23] que se derivan

23. Vease el Capítulo 5.5 de Wasserman (2007) o el Apéndice del Capítulo 5 en Hastie, Tibshirani y Friedman (2008).

de lo vistos anteriormente. No obstante, para no complicar más la exposición (y el algoritmo en si) se implementó una versión optimizada de (2.17) con base en la Tabla (2.1) que funciona bastante rápido inclusive cuando J es grande.

En la practica, los parámetros M, J y K se calibran pues, como ya se mencionó anteriormente, hacer J variable y automático es muy complejo. Asimismo, la elección de M y K requeriría cierta exploración previa de los datos. No obstante, existen algoritmos que realizan esta tarea que, para los fines de este trabajo no aportaría mucho. Además de que los resultados que se obtuvieron por el método de calibración son bastante buenos.

Si se le da rigor al modelo, en realidad, hay dos expansiones en bases. La primera la primera *a lo largo* de la ecuación lineal (2.3) cuyos coeficientes son β y las funciones base $\mathbf{f}$. Posteriormente, se tiene la expansión de la ecuación de polinomios por partes de (2.4) cuyos coeficientes son w_j y las funciones base Ψ_j para toda j. Esto explica el salto conceptual y notacional que se da entre las representaciones de (2.8) y (2.9).

Al tener en mente que se tienen d covariables, y por ende d polinomios por partes, además de la estructura lineal de (2.16) se puede sustituir (2.4) dentro de (2.3) dando la siguiente estructura con doble suma:

$$\begin{aligned} f(\mathbf{x}) &\approx \sum_{j=0}^{d} \beta_j f_j(x_j) \\ &\approx \beta_0 + \sum_{j=1}^{d} \beta_j \left[\sum_{l=1}^{N^*} w_{j,l} \Psi_{j,l}(x_j, \mathcal{P}_j) \right]. \end{aligned}$$

Lo cual, es perfectamente lineal. Se tienen $1 + d \times N^*$ términos que se pueden acomodar en un solo vector. Sin embargo, se tiene un cruce de parámetros interesante: la multiplicación de $\beta_j \quad \forall j$ contra $w_{j,l} \quad \forall l$. Tradicionalmente, no se usan β y se deja que se capture ese efecto dentro de las f_j como en los GAM. Sin embargo, dado que el objetivo de este trabajo es la predicción, más que la estimación de funciones, se opta por dar una nueva capa de suavizamiento con β. No existe forma de garantizar ortogonalidad de β contra las todas las w, por lo tanto, se le da prioridad a la correcta estimación de w pues captura un mayor efecto además de que, por la construcción de los polinomios por partes, si está garantizada la ortogonalidad contra las funciones bases Ψ.

Capítulo 3

Paradigma bayesiano e implementación

Pasar de un modelo tan estructurado a su implementación computacional no resulto fácil. Sin embargo, se logró desarrollar un algoritmo que estima todos los parámetros del modelo de una forma eficiente y que funciona en la práctica. En el fondo, el algoritmo recae en el método de Gibbs sampling propuesto en Albert y Chib (1993), por lo que se hace una breve introducción a la escuela de inferencia bayesiana, y en el algortimo de backfitting descrito en Hastie y Tibshirani (1986). Al algoritmo se le titula: *bayesian piece wise polinomial model (bpwpm)*. Para facilitar la utilización del modelo en diversas bases de datos, así como su validación y visualización, a la par del algoritmo se desarrolló un paquete de código abierto (con el mismo nombre)

para el software estadístico `R`, más detalles en le Apéndice B.

Al darle un tratamiento bayesiano a los parámetros, más que estimarlos, se busca regresar una muestra de tamaño arbitrario de sus correspondientes distribuciones posteriores. La idea, es que estas distribuciones posteriores, se haya capturado toda la información de los datos de entrenamiento. Se considera, que una buena forma de entender el algoritmo es *visualizando* tanto los datos como los objetos que componen el modelo, por lo tanto se hace un paréntesis notacional. De las ecuaciones del modelo: (2.1) a (2.4), se tienen dos grupos de parámetros por estimar, $\beta \in \mathbb{R}^{d+1}$ y $w_j \in \mathbb{R}^{N^*} \quad \forall j = 1, \ldots, d$, donde:

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_d \end{bmatrix} \quad \text{y} \quad w_1 = \begin{bmatrix} w_{1,1} \\ \vdots \\ w_{1,N^*} \end{bmatrix} \quad \ldots \quad w_d = \begin{bmatrix} w_{d,1} \\ \vdots \\ w_{d,N^*} \end{bmatrix}$$

Se hace énfasis en que existen d vectores w_j, cada uno de tamaño N^*. Por lo tanto, se tienen un total de $1 + d + d \times N^*$ parámetros. Se usa el símbolo $\mathbf{w}$ para designar todos los vectores w_j, correspondiente a una matriz: $\mathbf{w} \in \mathbb{R}^{d \times N^*}$. Cuando se habla de datos: $\{(y_i, \mathbf{x}_i)\}_{i=1}^n$, estos se pueden representar en una tabla (o matriz):

$$\left[\begin{array}{c|ccc} y_1 & x_{1,1} & \ldots & x_{1,d} \\ \vdots & \vdots & & \vdots \\ y_n & x_{n,d} & \ldots & x_{n,d} \end{array}\right],$$

donde el vector de observaciones binarias $\mathbf{y} = (y_1, \ldots, y_n)^t$ es la primer columna de

la tabla, y la matriz de covariables $\mathbf{X}$ es el resto.

Bajo esta representación, se da contexto cuando se habla de que la estimación debe reflejar los patrones *hacia abajo* y *hacia lo largo*. Hacia abajo, se está captando la información existente entre las observaciones; cada f_j, mediante su parámetro w_j, representa una transformación no lineal de la variable (o dimensión) j. Hacia lo largo, la función de proyección f suma cada f_j a través de β, ponderando los efectos individuales de cada variable. Mantener el balance entre la estimación de β a lo largo y w_j hacia abajo, es fundamental para el algoritmo. Se concluye, que la estimación de ambos grupos de parámetros, se puede ver como una regresión separada para cada uno, y por ende, estos pueden ser estimados por el mismo algoritmo. Esto responde a la dualidad que se exploró en el capítulo pasado de que ambas expresiones son expansiones en bases funcionales. El puente que conecta, y controla el balance entre ambas, son los residuales parciales. Los siguientes capítulos, se concentran en explicar e implementar este curioso patrón.

3.1. Fundamentos de la estadística bayesiana

Dado el problema de describir fenómenos bajo incertidumbre, existen dos escuelas dominantes de la estadística: la frecuentista y la bayesiana. La primera, aunque increíblemente útil, está hasta cierto punto limitada y en ocasiones termina derivando en colecciones de algoritmos. La escuela bayesiana, por el contrario, nombrada así en honor a Thomas Bayes (1702 - 1761), enfatiza el componente *probabilista* del

proceso inferencial, dandole coherencia al proceso (Mendoza y Regueiro 2011), (Bernardo y Smith 2001). La estadistica bayesiana está axiomatizada bajo la *teoría de la decisión*. Esta teoría formaliza conceptos como la *coherencia entre preferencias* y *utilidad*, sobre los que desarrolla un marco metodológico para la toma de decisiones.

Esta metodología, además de proveer técnicas concretas para resolver problemas, también formaliza en una forma de pensar sobre la probabilidad como una *medida racional para cuantificar la incertidumbre* condicionando sobre el conocimiento existente. Este paradigma es el que más corresponde con el sentido que usualmente se le da a la palabra. La inferencia sobre creencias (o parámetros), se realiza mediante una *actualización* de estas en luz de nueva evidencia, modificando la medida de incertidumbre. El mecanismo que permite realizalo, es el aclamado teorema de Bayes. De manera informal se puede describir como: dado un evento E bajo condiciones C, la probabilidad *posterior* de ocurrencia del evento, será proporcional a la probabilidad *previa* que se tiene sobre este, ponderado por la probabilidad de ocurrencia de las condiciones presentes. En otras palabras, el teorema de Bayes, *actualiza* la probabilidad de ocurrencia de un evento, ponderandolo por la información condicional que se tiene sobre el. En su forma matemática:

$$P(E|C) \; \alpha \; P(C|E)P(E) \tag{3.1}$$

El término central $P(C|E)$ es una medida descriptiva de las condiciones (usualmente datos) llamada *verosimilitud*. En Se hace notar que para poder hacer cualquier intento de descripción, se debe especificar el *modelo probablistico* que se asume describe el estado por el que se dan las condiciones C.

En un contexto matemático más formal, la cuantificación de la incertidumbre se da a través de medidas de probabilidad $\pi(\cdot)$, que describan el fenómeno observado. Estas medidas de probabilidad, usualmente son funciones que dependen de cantidades desconocidas llamados parámetros θ. Aunque desconocidas, se tienen ciertas creencias u conocimiento previo, *a priori*, sobre ellos, descritos por su correspondiente medida de probabilidad $\pi(\theta)$. Además, se tienen datos $\mathbf{X}$, interpretados como *evidencia* (condiciones), a los cuales se les asigna un modelo de probabilidad dependiente de los parámetros, es decir, su verosimilitud: $\pi(\mathbf{X}|\theta)$. Usando la formula de Bayes, se puede actualizar el conocimiento que se tiene sobre los parámetros haciendo:

$$\pi(\theta|\mathbf{X}) \; \alpha \; \pi(\mathbf{X}|\theta)\pi(\theta) \tag{3.2}$$

La idea es que este proceso de actualización sea a la vez, un proceso de aprendizaje, en el cual los parámetros capturen la información contenida en los datos.

El paradigma frecuentista, adopta un enfoque diferente para el aprendizaje. Se asume que no hay incertidumbre en los parámetros dado los datos y, por lo tanto, estos son tomados como fijos. El mecanismo que permite su estimación, usualmente consiste en plantear una función objetivo y optimizarla. Por ejemplo, si se escoge la verosimilitud $\pi(\mathbf{X}|\theta)$, se busca dar un estimador que la maximice, pues equivaldría a encontrar los parámetros que hagan más *posibles* los datos, bajo el modelo planteado. Si por el contrario, es escoge una función como la RSS de los modelos ANOVA (primer sumando de (2.11)), se busca la θ que minimice estos errores, así, el modelo logra capturar toda la variabilidad que puede sobre los datos.

Independientemente del paradigma estadístico que se escoja, siempre es importante la validación del modelo y de sus supuestos. Además, tanto teoría bayesiana como frecuentista han resultado de infinita utilidad en la practica y el avance de la estadística y ciencia en general.

Una de las dificultades que surgen en la estadística bayesiana, es que la obtención de resultados analíticos cerrados es difícil o muy tedioso una vez que los modelos se empiezan a complicar. Por ejemplo, en las ecuaciones anteriores, se ha usado el argumento de proporcionalidad α. Esto pues, para que se de la igualdad, el lado derecho de la ecuación (3.2) se debe de dividir entre $\pi(\mathbf{X}) = \int \pi(X|\tilde{\theta})\pi(\tilde{\theta})\, d\tilde{\theta}$, el cual usualmente es difícil, sino imposible, de calcular. A este término se le conoce como *constante de proporcionalidad* y su función es la de reescalar la expresión del lado derecho para que en realidad se tenga una distribución en el izquierdo. Usualmente, para evitar estas complicaciones, se escogen *distribuciones conjugadas*, para que tanto la distribución a priori como la posterior pertenezcan a la misma familia. En el Apéndice A se detallan las distrubiciones conjugadas y se realiza más a fondo la derivación de los resultados de este trabajo. Sin embargo, con los avances en el poder computacional disponible y técnicas numéricas para resolver integrales (Robert y Casella 2004), se han desarrollado muchos métodos para aplicar el proceso de aprendizaje, independientemente de que tan complejo sea el modelo o las distribuciones, iniciales y resultantes. Muchos de estos métodos recaen en la teoría de las *cadenas de Markov*, como lo es, el Gibbs sampler presentado en la sección 3.2.

Estimadores Bayesianos

Una vez realizado el proceso de actualización, el estadista se enfrenta con un problema. Se tiene una distribución posterior de probabilidad para los parámetros de interés, usualmente dada por una muestra y no por una distribución analítica. Sin embargo, por practicidad y utilidad, en ocasiones se busca dar un *estimador puntual*. Por ejemplo, si se necesita dar un estimador $\hat{\theta}$ para usarlo en otros cálculos, o si $\pi(\theta|\mathbf{X})$ es multi-dimensional. Para superar este problema teórico, se ha adoptado por usar *funciones de perdida* $L(\hat{\theta}, \theta)$.[1] Estas, miden las *consecuencias* que se dan, al tomar $\hat{\theta}$ como el verdadero valor del parámetro θ, es decir, las funciones de perdida evalúan que tan bien se está representando el valor de θ con un estimador puntual. Por ello, vale la pena usar funciones que penalicen la distancia entre θ y $\hat{\theta}$. Sin entrar mucho en los detalles técnicos, se tiene que calcular:

$$\hat{\theta} = \mathbb{E}[L(\hat{\theta}, \theta)] = \int_{\Theta} L(\hat{\theta}, \theta)\pi(\theta)\, d\theta \tag{3.3}$$

con Θ el espacio de todas las posibles valores de θ. Sin embargo, se demuestra que para funciones de perdida sencillas, pero intuitivas, se tiene que el estimador puntual posterior es alguna medida de centralidad de la distribución posterior. Por ejemplo:

Función de pérdida cuadrática: $L(\hat{\theta}, \theta) = (\hat{\theta} - \theta)^2$, deriva en la media posterior, es decir: $\hat{\theta} = \mathbb{E}[\theta|\mathbf{X}]$

Función de perdida valor absoluto: $L(\hat{\theta}, \theta) = |\hat{\theta} - \theta|$, deriva en la mediana de

1. Formalmente se tiene un problema de decisión.

la distribución posterior.

Función de pérdida 0-1: $L(\hat{\theta}, \theta) = I[\hat{\theta} \neq \theta]$, deriva en la moda de la distribución posterior.

En la práctica, estas cantidades son fáciles de calcular cuando se tiene una muestra simulada de θ proveniente de la distribución posterior. En el paquete, se implementa una forma sencilla de obtener estimadores puntuales con cualquiera de las 3 funciones de pérdida. Sin embargo, dada la dorma de las distribuciones resultantes, los resultados no varían mucho al escoger entre las funciones de perdida.

3.1.1. Distribuciones condicionales completas

Retomando el modelo que concierne a este trabajo, se tienen dos grupos de parámetros, β y $\mathbf{w}$. Sin embargo, dados los supuestos del modelo, por el uso de la variable latente z, esta también se debe de incluir como parámetro pues es la liga entre la respuesta y y los datos $\mathbf{X}$, vista de forma bayesiana, también se debe de simular. Por lo tanto, los parámetros del modelo son: $\theta = (\mathbf{z}, \beta, \mathbf{w})$ con $\mathbf{z} = (z_1, \ldots, z_n)^t$. Esta sección, concierne en desglosar el proceso de aprendizaje sobre ellos; esta derivación es importante en si pues es la que induce el algoritmo. Usando la notación presentada al inicio de esta sección, los supuestos propuestos en las ecuaciones del modelo

(2.1) a (2.4) y sustituyendo en (3.2) se tiene:

$$\begin{aligned}
\pi(\mathbf{z},\beta,\mathbf{w}|\mathbf{y},\mathbf{X}) \quad &\alpha \quad \pi(\mathbf{y}|\mathbf{X},\mathbf{z},\beta,\mathbf{w})\ \pi(\mathbf{z},\beta,\mathbf{w}) \\
&\alpha \quad \pi(\mathbf{y}|\mathbf{z})\ \pi(\mathbf{z}|\mathbf{X},\beta,\mathbf{w})\ \pi(\beta,\mathbf{w}) \\
&\alpha \quad \pi(\mathbf{y}|\mathbf{z})\ \pi(\mathbf{z}|\mathbf{X},\beta,\mathbf{w})\ \pi(\beta)\ \pi(\mathbf{w}) \\
&\alpha \quad \prod_{i=1}^{n} \mathrm{Be}[y_i|\Phi(z_i))]\ \phi[z_i|f(\mathbf{x}_i),1] \times \pi(\beta)\ \pi(\mathbf{w}) \\
&\alpha \quad \prod_{i=1}^{n} \mathrm{Be}[y_i|\Phi(z_i))]\ \phi[z_i|\beta^t\mathbf{f}(\mathbf{x}_i),1] \times \pi(\beta)\ \pi(\mathbf{w}) \qquad (3.4)
\end{aligned}$$

donde $\phi(\cdot|\mu,\sigma^2)$ es la función de densidad de una variables aleatoria normal con media μ y varianza σ^2. Asimismo $\mathrm{Be}(\cdot|p)$ es función de densidad de una variable Bernoulli con probabilidad de éxito p. Esta factorización es válida dados los supuestos, donde se hace notar, la forma que conecta z_i a las dos partes del modelo a través de la función de proyección $f(\mathbf{x}_i) = \beta^t\mathbf{f}(\mathbf{x}_i)$ que contiene tanto a β como $\mathbf{w}$. Esta derivación es una forma extendida (aunque simplificada) de la verosimilitud para todas las observaciones $i = 1,\ldots,n$. Aunque aún no se han especificado las formas funcionales para las distribuciones a priori $\pi(\mathbf{w})$ y $\pi(\beta)$, estas se pueden separar ya que se suponen independientes. Esta propiedad, combinada con la forma funcional en expansiones de bases, lleva a que se piense en hacer una estimación *por bloques*, es decir, se estima primero β y posteriormente $\mathbf{w}$ en un bucle iterativo, esta es la idea de un Gibbs sampler.

3.2. Herramietnas de simulación

Una vez establecida el proceso de actualización, el estadista se ve en la necesidad de tener que desarrollar técnicas para simular de la distribución posterior $\pi(\theta|\mathbf{X})$ sin importar que complejo sea el modelo. Desde principios de los años noventa, se desarrollaron muchos algoritmos y paquetería estadística al aumentar el poder computacional. La gran mayoría de los algoritmos recae en los *métodos Monte Carlo de cadenas de Markov* (MCMC). Estos métodos, como su nombre lo indica, hacen alusión a principios de aleatoriedad, como se daría en un casino. Usando ideas intuitivas de probabilidad y números pseudoaleatorios, se pueden generar muestras prácticamente de cualquier distribución, incluso si su forma funcional es desconocida. La simulación, como tal es un tema que merece un estudio más profundo (Robert y Casella 2004). Estas poderosas simulaciones, permitieron que los estadistas y experimentadores pudieran hacer el menor número de supuestos posibles sobre los modelos, puesto que ya no se buscan resultados analíticos sino más bien, se buscaba reflejar la realidad y dejar que los cálculos los hiciera una computadora.

Breve introducción a cadenas de Markov

Al final, la gran mayoría de estos métodos recaen sobre las la teoría de *cadenas de Markov*. Una cadena de Markov, es una secuencia de variables aleatorias:

$X^{(1)}$, $X^{(2)}$, ... que cumplen la *propiedad Markoviana*:

$$P(X^{(t+1)}|X^{(t)} = x^{(t)}\,, X^{(t-1)} = x^{(t-1)}\,, \ldots, X^{(2)} = x^{(2)},\, X^{(1)} = x^{(1)}) = P(X^{(t+1)}|X^{(t)} = x^{(t)}) \quad \forall$$

con t interpretado como *tiempo*. Por lo tanto, la siguiente variable de la cadena, $X^{(t+1)}$, únicamente depende de *el estado* actual $X^{(t)}$ y no de los anteriores. Usualmente esta propiead es expresada como: el futuro, condicionando al presente, es independiente del pasado. El ejemplo canónico que se presenta es la *caminata aleatoria*: $X^{(t+1)} = X^{(t)} + e^{(t)}$, con $e^{(t)}$ error aleatorio generado de forma independiente. De esta idea se desarrolla toda una rica teoría revisada en cursos de procesos estocásicos, de donde surgen muchas propiedades aplicables a las cadenas(Ross 2009). Una de las ideas más relevantes para lo que concierne este trabajo, es la de *matrices de transición*. Dada una cadena con n posibles estados, es decir, $X^{(t)}$ únicamente puede tomar valores de un subconjunto de cardinalidad n. Se puede construir una matriz cuadrada $P \in \mathbb{R}^{n\times n}$ donde cada entrada $0 \leq p_{i,j} \leq 1$ representa la probabilidad de transicionar del estado i al estado j. Se demuestra, que si una cadena es *ergodica*[2], entonces existe una *distribución límite* que es igual a la *distribución estacionaria*: $\exists\, \pi$ tal que $\pi P = \pi$. Sin entrar en los detalles técnicos, la ergodicidad es la propiedad que asegura que eventualmente se alcanza la convergencia de la cadena sin importar el estado inicial tras repetidas aplicaciones de la matriz de transición P.[3] Esto es, dado un vector de estados inicial $X^{(0)} \in \mathbb{R}^n$ tal que $\mathbf{1}^t X^{(0)} = 1$, se puede

2. Aperiódica, irreducible y recurrente positiva. Para efectos de simplicidad en la exposición, la ergodicidad es tratada como una propiedad en si misma. Las definiciones formales, puede ser consultadas en cualquier texto de procesos estocásticos.

3. Esta convergencia es una convergencia estocástica aplicable al paradigma bayesiano. El paradigma frecuentista, presenta resultados de convergencia que recaen en el análisis funcional (Stone 1985)

encontrar la distribución estacionaria dejando:

$$\pi = \lim_{t\to\infty} X^{(t)} \quad \text{si} \quad X^{(t+1)} = P^t X^{(t)} \tag{3.5}$$

Esta idea se puede extender a casos más complejos donde se relajan o se cambian supuestos. Incluso, se extiende a casos donde el número de estados es no finito, pero la idea fundamental es la misma.

En el contexto de este trabajo la idea es poder simular *secuencialmente* cadenas de parámetros θ que estén ligados unos con los otros, que dependan únicamente de el presente y, sobre todo, que una vez simuladas un número arbitrario de estas, converjan a la distribución estacionaria. Precisamente lo que hace un Gibbs sampler.

Gibbs sampler

El Gibbs sampler como tal, es una técnica, para simular variables aleatorias de una *distribución conjunta* sin tener que calcularla directamente, análogo a lo anteriór(Gelfand y Smith 1990) y (Casella y George 1992). Usualmente, el muestreo de Gibbs se usa dentro de un contexto bayesiano, aunque también funciona para otras aplicaciones. A primera vista, parece misterioso, pero en realidad, se basa únicamente en las propiedades revisadas (relativamente sencillas) de las cadenas de Markov. Sin perdida de generalidad, se busca simular una muestra de parámetros $\theta = (\theta_1, \ldots, \theta_p)$ que provienen de la distribución conjunta $\pi(\theta|\cdot)$. Esta distribución usualmente no es conocida analíticamente, sin embargo el Gibbs sampler permite

dar, no la distribución como tal, pero si una muestra arbitrariamente grande con la que se puede aproximar empíricamente $\hat{\pi}(\theta) \approx \pi(\theta)$. En la practica usualmente más que aproximar la distribución, se busca alguna función de los parámetros como la media o la varianza de la distribución.

Para llevar a cabo el muestreo, se intercambia el difícil cálculo de la distribución conjunta al cálculo de las distribuciones condicionales que usualmente son más fáciles de derivar. Las distribuciones condicionales están dadas por:

$$\begin{aligned} \theta_1 &\sim \pi(\theta_1|\theta_2, \ldots, \theta_p) \\ \theta_2 &\sim \pi(\theta_2|\theta_1, \theta_3, \ldots, \theta_p) \\ &\vdots \\ \theta_p &\sim \pi(\theta_p|\theta_1, \ldots, \theta_{p-1}) \end{aligned} \tag{3.6}$$

Se comienza con una muestra inicial arbitraria $\theta^{(0)} = (\theta_1^{(0)}, \ldots, \theta_p^{(0)})^t$, donde el superíndice $^{(k)}$ corresponde a la iteración k. Se comienza a simular de las correspondientes distribuciones condicionales, las cuales quedan especificadas para los valores iniciales. En este caso, para $k = 1, 2, 3, \ldots$ se tiene:

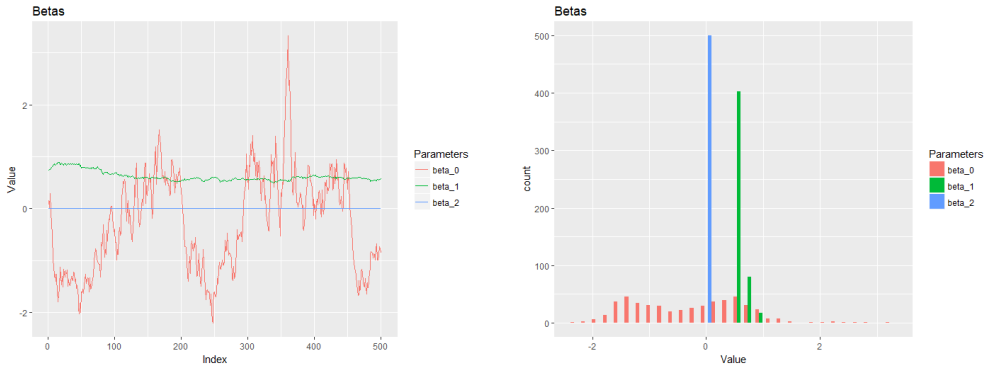
$$\begin{aligned} \theta_1^{(k)} &\sim \pi(\theta_1|\theta_2^{(k-1)}, \ldots, \theta_p^{(k-1)}) \\ \theta_2^{(k)} &\sim \pi(\theta_2|\theta_1^{(k)}, \theta_3^{(k-1)}, \ldots, \theta_p^{(k-1)}) \\ &\vdots \\ \theta_p^{(k)} &\sim \pi(\theta_p|\theta_1^{(k)}, \ldots, \theta_{p-1}^{(k)}) \end{aligned} \tag{3.7}$$

Este proceso se itera hasta tener una muestra de tamaño arbitrario, que haya alcanzado la región de probabilidad donde se encuentra la distribución estacionaria, en este caso la distribución posterior $\pi(\theta|\cdot)$.

La convergencia no es intuitiva, es decir, no es trivial derivar que al muestrear de las distribuciones condicionales, se llegue (eventualmente) a un la distribución conjunta. Sin embargo, la prueba formal, aunque compleja, recae en que se puede formar una matriz de transición con las distribuciones condicionales de θ_i, análoga a las matrices de las cadenas de Markov. Al dejar que $k \to \infty$, se llega a un resultado equivalente al de la ecuación (3.5), habiendo muestrado de la distribución posterior. Sin embargo, la ergodicidad, es un supuesto importante que se preserva y es necesario validad. Esto, pues se pueden dar casos donde la distribución posterior no exista.

El tener una muestra de la distribución final $\left\{\theta^{(k)}\right\}_{k=k^*}^{N_{\text{sim}}}$, donde N_{sim} es el número total de simulaciones arbitraria y k^* es el punto a partir del cual se cree que se obtiene la convergencia a $\pi(\theta|\cdot)$, tiene muchos beneficios en la practica. Se le pueden calcular momentos a la muestra, medidas de desviación y hacer representaciones gráficas para su análisis y evaluación. En la Figura 3.1, se tienen dos imágenes que de una muestra Gibbs para una simulación donde: $\theta = \beta \in \mathbb{R}^3$. Para esta figura, se toman y se grafican los últimos 500 valores de las cadenas. Se observan las *trazas* que se forman al ir simulando los parámetros y se hace un histograma, dando una idea de las distribuciones subyacentes[4].

4. Estas imágenes tienen como propósito ejemplificar el Gibbs sampler. Fueron obtenidas de una simulación del modelo que se analiza en Sección 4.2.2, donde se hará una exploración a fondo de la convergencia de este ejemplo en particular y se verá por que el parámetro β_2 es prácticamente cero. Asimismo, estas imágenes fueron generadas con la librería *ggplot2*, incorporada a las funcionalidades



(a) Trazas (b) Histogramas

Figura 3.1: Muestro Gibbs para el ejemplo 6 de la Sección 4.2.1

En la practica, muchos de los pormenores derivados del muestreo Gibbs, pueden ser mejorados. Dado que el valor inicial $\theta^{(0)}$ es dado por el estadista, en ocasiones el método tiene que explorar una región extensa de posibles valores para θ, por lo que podría tardar en converger. Esto deriva en que las primeras observaciones deban ser descartadas pues no son realizaciones de la distribución final buscada. A este periodo se le conoce como *burn-in*. En el ejemplo anterior, se utilizó k^* para designar la iteración a partir de cual se toman los valores simulados. En la practica, usualmente se guarda toda la cadena, se explora, tanto por resúmenes numéricos como con representaciones gráficas y se decide (de forma subjetiva) el corte k^*. Otro método ampliamente usado es el de adelgazamiento o *thinning*. Por las mismas características de la estimación por Gibbs, sobre todo en casos multivariados, los valores de las cadenas tienen correlaciones altas. Si se quieren muestras independientes, la bibliografía recomienda tomar cada (k_{thin})-ésimo valor de la cadena generada para

del paquete *bpwpm* desarrollado para este trabajo.

reducir la dependencia entre los parámetros. Usualmente se usan valores pequeños para k_{thin}. Estos sencillos pasos pasos para mejorar las cadenas, ya se encuentran implementados en el paquete *bpwpm* para `R` para el análisis rápido de las cadenas.

3.2.1. Algoritmo de Albert y Chib

El algoritmo particular del Gibbs sampler que se usa en este trabajo, es una versión modificada del presentado en Albert y Chib (1993). Este método ofrece varias ventajas para el modelo propuesto, pues se desarrolló específicamente para regresiones probit usando la variables latente z. Además es un método muy eficiente pues usa distribuciones conjugadas, por lo que las distribuciones condicionales, se pueden calcular directamente y la parte estocástica depende únicamente de simular distribuciones conocidas. Esto lleva a que los periodos de burn-in sean relativamente pequeños y que el adelgazamiento no sea fundamentalmente necesario.

A su algoritmo, ellos lo llaman *data augmentation for binary data* y son los pioneros en el uso de variables latentes para unir la respuesta y con las covariables $\mathbf{x}$ como se revisa en la Sección 2.1. En su exposición, ellos utilizan una función de proyección lineal $f(\mathbf{x}) = \beta^t\mathbf{x}$, diferente. Por lo mismo, (y por un breve momento) esta se utiliza para explicar la idea fundamental. Usando la misma notación, se introducen n variables latentes $\mathbf{z} = (z_1, \ldots, z_n)^t$, con $z_i \sim N(\beta^t\mathbf{x}_i, 1)$, sobre las que se definien se

redefinen las respuestas:

$$y_i = \begin{cases} 1, & \text{si } z_i > 0 \\ 0, & \text{si } z_i \leq 0. \end{cases} \tag{3.8}$$

La simplificación del modelo, obliga a $\theta = (\mathbf{z}, \beta)$. Por lo tanto, la derivación bayesiana queda:

$$\begin{aligned} \pi(\mathbf{z}, \beta|\mathbf{y}, \mathbf{X}) \quad &\alpha \quad \pi(\mathbf{y}|\mathbf{X}, \mathbf{z}, \beta)\ \pi(\mathbf{z}, \beta) \\ &\alpha \quad \pi(\mathbf{y}|\mathbf{z})\ \pi(\mathbf{z}|\beta, \mathbf{X})\ \pi(\beta) \\ &\alpha \quad \prod_{i=1}^{n} [I(y_i = 1)I(z_i > 0) + I(y_i = 0)I(z_i \leq 0)] \times \phi(z_i|\beta^t \mathbf{x}_i, 1) \times \pi(\beta). \end{aligned} \tag{3.9}$$

Ahora, bajo los fundamentos del Gibbs Sampler, en lugar de querer encontrar la distribución posterior (3.9), se busca encontrar las distribuciones condicionales. Para

β:

$$\pi(\beta|\mathbf{z},\mathbf{y},\mathbf{X}) = \frac{\pi(\mathbf{z},\beta|\mathbf{y},\mathbf{X})}{\pi(\mathbf{z})} \tag{3.10}$$

$$= \frac{\pi(\mathbf{y}|\mathbf{z})\ \pi(\mathbf{z}|\beta,\mathbf{X})\ \pi(\beta)}{\pi(\mathbf{y},\mathbf{X})\ \pi(\mathbf{z})}$$

$$= \frac{\pi(\mathbf{y}|\mathbf{z})}{\pi(\mathbf{y},\mathbf{X})\ \pi(\mathbf{z})} \overset{C}{\nearrow} \times \pi(\mathbf{z}|\beta,\mathbf{X})\ \pi(\beta) \tag{3.11}$$

$$= C\,\pi(\beta) \prod_{i=1}^{n} \phi(z_i|\beta^t\mathbf{x}_i, 1), \tag{3.12}$$

la densidad condicional es la misma que se derivaría si se tuviera una regresión lineal bayesiana con z de regresor, es decir: $z_i = \beta^t\mathbf{x}_i + e_i$ con $e_i \sim N(0,1)$. Esta es la utlidad de la variable latente, que convierte una regresión probit a una regresión lineal tradicional. Asimismo, para estos modelos, dependiendo de la distribucion *a priori* $\pi(\beta)$ se pueden conseguir resultados cerrados. Se hace notar que la ecuación (3.10) se toma de la definición de probabilidad condicional, y el paso de (3.11) a (3.12) se puede hacer ya que, al definir y como en la ecuación (3.8), sus representaciones son análogas y el cociente se desvanece, dejando únicamente la constante C que sale del término $\pi(\mathbf{y},\mathbf{X})$. Ahora, únicamente falta definir $\pi(\beta)$. Es común en la práctica usar distribuciones *no informativas* sobre los parámetros, cuando no se tiene experiencia sobre ellos. Sin embargo, para el modelo lineal bayesiano, existe una familia de distribuciones conjugadas, que son razonables para la aplicacion que se buscan (Banerjee 2008). En particular, al dejar la varianza fija y eligiendo de

distribución $\pi(\beta)$:

$$\beta \sim N_{d+1}(\beta|\mu_\beta, \Sigma_\beta) \tag{3.13}$$

donde $\mu_\beta \in \mathbb{R}^{d+1}$ es el hiperparámetro de media y $\Sigma_\beta \in \mathbb{R}^{(d+1)\times(d+1)}$ la matriz de covarianza, entonces, se tiene la distribución conjugada:

$$\beta|\mathbf{y}, \mathbf{z}, \mathbf{X} \sim N_{d+1}(\beta|\mu^*_\beta, \Sigma^*_\beta), \tag{3.14}$$

donde,

$$\mu^*_\beta = \Sigma^*_\beta \times (\Sigma^{-1}_\beta \mu_\beta + \mathbf{X}^t\mathbf{z})$$
$$\Sigma^*_\beta = (\Sigma^{-1}_\beta + \mathbf{X}^t\mathbf{X})^{-1}.$$

Esta distribución es conjugada pues preserva la estructura normal de β. Asimismo, es fácil de simular usando cualquier software estadístico, calculando previamente todos los parámetros y dando un valor (o iteración) para $\mathbf{z}$.[5] En el Apéndice A se hace un resumen de las distribuciones conjugadas y se completan algunos de los pasos tediosos de la derivación.

Ahora, condicionar sobre $\mathbf{z}$, es más sencillo, pues la derivación es similar, comenzando

5. Se hace notar, que este estimador, es relativamente parecido al estimador que se da en una regresión cordillera.

con la expresión (3.9) y reordenando términos:

$$\begin{aligned}
\pi(\mathbf{z}|\beta,\mathbf{y},\mathbf{X}) &= \frac{\pi(\mathbf{z},\beta|\mathbf{y},\mathbf{X})}{\pi(\beta)} \\
&= \frac{\pi(\mathbf{y}|\mathbf{z})\ \pi(\mathbf{z}|\beta,\mathbf{X})\ \pi(\beta)}{\pi(\mathbf{y},\mathbf{X})\ \pi(\beta)} \\
&= \frac{1}{\pi(\mathbf{y},\mathbf{X})} \overset{C}{\nearrow} \pi(\mathbf{y}|\mathbf{z}) \times \pi(\mathbf{z}|\beta,\mathbf{X}) \\
&= C \prod_{i=1}^{n} [I(y_i=1)I(z_i>0) + I(y_i=0)I(z_i\leq 0)] \times \phi(z_i|\beta^t\mathbf{x}_i,1).
\end{aligned} \tag{3.15}$$

De donde es claro ver que las z_i son independientes y tienen distribuciones normales truncadas en 0:

$$\begin{aligned}
&z_i|y_i=1,\beta \sim N(z_i|\beta^t\mathbf{x}_i,1) \text{ truncada a la izquierda,} \\
&z_i|y_i=0,\beta \sim N(z_i|\beta^t\mathbf{x}_i,1) \text{ truncada a la derecha,}
\end{aligned} \tag{3.16}$$

las cuales también son fáciles de simular usando los algoritmos de Devroye (1986).

Finalmente, una vez que se tienen las distribuciones condicionales (3.16) y (3.14), la simulación de esto dos primeros grupos de parámetros se realiza con un muestreo de Gibbs dado por las ecuaciones (3.7) de la página 56. En la Tabla 3.1, se puede revisar el pseudocódigo que se uso para la implementación.

```
SampleoGibbs(y, X, N_sim, beta(0), mu_beta, sigma_beta):
    sigma_beta <- ((sigma_beta)^-1 + X'X)^-1
    PARA k = 1 HASTA N_sim:
        z(k) <- SimNormTrunc(y, X, beta(k-1))
        mu_beta(k) <- sigma_beta*((sigma_beta)^-1*mu_beta
         + X'*z(k))
        beta(k) <- NormMulti(mu_beta(k), sigma_beta(k))
    REGRESAR beta
```

Tabla 3.1: Algoritmo de Albert y Chib para modelos probit

El valor de $\mathbf{z}^{(0)}$ en realidad no se tiene que dar pues se simula dependiendo de y y $\beta^{(0)}$. Este valor inicial $\beta^{(0)}$ es arbitario, pero se sugiere que sea dado por el estimador de máxima verosimilitud o el de mínimos cuadrados para las respuestas binarias $\beta^{(0)} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t y$. Sin embargo, en la práctica el algoritmo, por default, los inicializa en ceros y tardan poco tiempo en converger a la distribución límite.

Como comentario, el trabajo de Albert y Chib (1993), incluye muchos más métodos Gibbs sampler para diferentes modelos. Entre ellos, multinomiales o modelos tobit, los cuales usan como función liga g a la función de acumulación *t-student* en vez de la distribución normal.

3.3. Algoritmo *bpwpm*

Finalmente, se esta en posición para hablar de la estimación de los parámetros $\mathbf{w}$ en el corazón del modelo. Dado que este modelo se construyó desde cero, no se

encontró un algoritmo ya establecido. Sin embargo, usando las intuitivas ideas de los modelos GAM, se implemento un método para simular $\mathbf{w}$ que funciona muy bien en la practica. Una vez establecido eso, se juntan todos los pedazos y se arma el algoritmo final.

3.3.1. Algoritmo de *backfitting* para ajuste de modelos GAM

El *algoritmo de backfitting* fue desarollado para hacer la estimación (frecuentista) de las funciones f_j que componen los modelos GAM revisados en la sección 2.2.1 (Hastie y Tibshirani 1986). Como se mencionó, la idea central recae en que, cada dimensión $j = 1, \ldots, d$, más un término independiente f_0, capture de forma aditiva, toda la información posible del regresor, en este caso $z \in \mathbb{R}$, a través de transformaciones no lineales f_j, es decir:

$$z = f_0 + f_1(x_1) + \ldots + f_d(x_d) + e, \tag{3.17}$$

con ϵ el tradicional error aleatorio. Estas f_j son muy flexibles y se demuestra (bajo ciertos supuestos) que son splines cúbicos. En la práctica, usualmente se dejan indeterminados y son estimadas mediante procedimientos no paramétricos. Usar estas funciones en vez de un proyector lineal está justificado si $\mathbb{E}(z|\mathbf{x})$ no es lineal, lo cual se da muchas veces cuando se usan datos reales. Sin embargo, la estimación de f_j es diferente a la estimación tradicional de parámetros β pues se enfatiza, que son *arbitrarias* en su escala. Por ejemplo, sea $d = 1$ y $z_i = 1$ para alguna i. Se podría tener $z_i = \hat{f}_0 + \hat{f}_1(x_{i,1})$ con las estimaciones $\hat{f}_0 = 3$ y $\hat{f}_1(x_{i,1}) = -2$ ó $\hat{f}_0 = -1$ y

$\hat{f}_1(x_{i,1}) = 2$, sin importar el valor de $x_{i,1}$ y los dos serían modelos que (al menos para esta observación i) ajustarían perfectamente. Para controlar estas posibles fluctuaciones, el modelo trata de capturar la información de forma secuencial a través de los *residuales parciales*. Es decir, una vez dadas, todas las f_j menos una, se evalúa que tanto le falta al modelo por capturar y posteriormente, se hace un suavizamiento de estos usando una función genérica $S_j(\cdot|x_j)$ que depende de los datos. Más formalmente, los residuales parciales, reformulando la ecuación (2.13) con la notación de esta sección son, $\forall j^* = 1, \ldots, d$,

$$\hat{r}_{j^*} = z - f_0 - \sum_{\substack{j=1 \\ j \neq j^*}}^{d} f_j(x_j) = z - f_0 - \ldots - f_{j^*-1}(x_{j^*-1}) - f_{j^*+1}(x_{j^*+1}) - \ldots - f_d(x_d). \tag{3.18}$$

Posteriormente, se da un estimado de f_{j^*},

$$\hat{f}_{j^*}(x_{j^*}) = S(\hat{r}_{j^*}|x_{j^*}) = S[z - f_0 - \sum_{\substack{j=1 \\ j \neq j^*}}^{d} f_j(x_j)|x_{j^*}]. \tag{3.19}$$

Por lo tanto, $\hat{f}_{j^*}(x_{j^*})$ es un suavizamiento, de esa dimensión. De forma iterativa, una vez dada la estimación para la dimensión j^*, se puede mejorar, a su vez, la estimación de $f_{j^*\pm1}$, dando paso a un algoritmo secuencial hasta que no haya mucha variación en las f_j. Un paso importante que se debe hacer es *identificar* el modelo, esto es, centrar los residuales en 0 tomando el término independiente, como el promedio de las respuestas, es decir: $f_0 = \bar{z}$, de tal forma que todas las f_j queden alrededor de cero.

El algoritmo de backfitting en pseudocódigo y usando una función de suavizamiento arbitraria, se presenta en la Tabla 3.2:

```
AlgoritmoBackfitting(z, X, f_init):
    f(0) <- Promedio(z)
    f(j) <- f_init(j) para toda j = 1,...,d
    MIENTRAS convergencia:
        PARA j = 1,...,d,1,...,d,...:
            f(j) <- Suavizar(z - f(0) - suma[f(k),
             para toda k != j], x(j))
```

Tabla 3.2: Algoritmo de backfitting para modelos GAM

Aplicando los métodos GAM al modelo

Este principio de suavizamiento iterativo de los residuales parciales, es justamente lo que se buscaba para poder finalizar el algoritmo. Recordando la función de projección (2.3) del modelo, una vez estimada z, se tiene una representación análoga a un GAM,

$$\hat{z} = \beta_0 + \beta_1 f_1(x_1) + \ldots + \beta_d f_d(x_d) + e = \beta^t \mathbf{f}(\mathbf{x}) + e,$$

con la adición de las β_j. De donde se pueden obtener, usando la misma idea los residuales parciales, $\forall j^* = 1, \ldots, d$,

$$r_{j^*} = 1/\beta_{j^*}[\hat{z} - \beta_0 - \sum_{\substack{j=1 \\ j \neq j^*}}^{d} \beta_j f_j(x_j)]. \tag{3.20}$$

Estos residuales, calculándose para toda observación $i = 1, \ldots, n$, son vectores. El paso fundamental, para poder hacer la estimación de $\mathbf{w}$ recae sobre el suavizamiento. En la sección anterior, se denotó por la función arbitraria $S_j(\cdot|x_j)$ sin embargo, para este modelo se tiene una representación funcional concreta para S_j, los polinomios por partes flexibles dados por la ecuación (2.16) y (2.17) de donde se hace la igualdad,

$$r_{j^*} = \sum_{l=1}^{N^*} w_{j^*,l}\ \Psi_l(x_{j^*}, \mathcal{P}_{j^*}) = w_{j^*}^t \Psi(x_{j^*}, \mathcal{P}_{j^*}). \tag{3.21}$$

Lo cual, no es más que un modelo de regresión lineal *hacia abajo*,[6] usando a w_{j^*} como coeficientes y a las transformaciones $\Psi(x_{j^*}, \mathcal{P}_{j^*})$ como covariables. Este hecho, se debe a la *dualidad* en expansiones en bases funcionales, de las que tanto se habló en el capítulo pasado. Existe una correspondencia uno a uno, entre β y w_j, así como entre $\mathbf{f}$ y Ψ. Por lo tanto, usando las técnicas que se discutieron en la Sección 3.2, las w_j pueden ser estimadas exactamente de misma forma que β.

Aunque intricado y notacionalmente pesado, el modelo en si, está compuesto únicamente por una serie de regresiones lineales: una regresión para β y d regresiones para $w_j(\quad j = 1, \ldots, d)$, donde se está ligando la respuesta binaria a través de la z. Con esta simplificación, se espera que la Figura 2.1 de la página 10, haya adquirido una nueva luz. En el siguiente capítulo, se busca esclarecer con ejemplos toda esta maraña de notación y se probará su efectividad.

6. y no *a lo largo*, como sería para β.

3.3.2. Implementacion final

Una vez conectados todos componentes del modelo, este se puede presentar en su versión final y más completa, aunque definitivamente, menos clara. Se recuerda que existe un compendio notacional al inicio de este trabajo. Juntando las expresiones (2.1) a (2.4), (2.16), (2.17), (3.13) y agregando las distribuciones a priori para w_j: $\pi(w_j)$; se tiene el modelo *bpwpm*:

$$
\begin{aligned}
y \,|\, z &\sim \mathrm{Be}(y \,|\Phi(z)), \\
z \,|\, x &\sim \mathrm{N}(z \,|f(\mathbf{x}), 1), \\
f(\mathbf{x}) &\approx \sum_{j=0}^{d} \beta_j f_j(x_j) = \beta^t \mathbf{f}(\mathbf{x}), \\
f_j(x_j) &\approx \sum_{l=1}^{N^*} w_{j,l} \Psi_{j,l}(x_j, \mathcal{P}_j) = w_j^t \Psi(x_j, \mathcal{P}_j), \quad \forall j = 0, 1, \ldots, d, \\
&= \sum_{i=1}^{M} w_{j,i,0}\; x_j^{i-1} + \sum_{i=K}^{M-1} \sum_{k=1}^{J-1} w_{j,i,k}\; (x_j - \tau_{j,k})_+^i
\end{aligned}
$$

donde,

$$
\begin{aligned}
\beta &\sim N_{d+1}(\beta|\mu_\beta, \Sigma_\beta), \\
w_j &\sim N_{N^*}(w_j|\mu_{w_j}, \Sigma_{w_j}) \qquad \forall j = 0, 1, \ldots, d.
\end{aligned} \tag{3.22}
$$

A diferencia de su exposición en el Capítulo 2, el algoritmo final, debe de construir de abajo hacia arriba, pues se necesita tener una estimación puntual de los parámetros para poder calcular las funciones intermedias y que todo quede definido de forma numérica.

Como paso final del algoritmo, se deben definir las particiones $\mathcal{P}_j \quad \forall j = 1, \ldots, d$. Estas particiones, quedan determinadas al estimar las posiciones los $J-1$ nodos τ_j. De forma intuitiva, se necesitarán más nodos donde se tengan más datos pues son las regiones donde se podrían dar un mayor número de saltos. Por lo tanto, el algoritmo calcula automáticamente las posiciones usando los cuantiles correspondientes para cada dimensión.[7]

Dado que el modelo tiene muchos componentes y pasos intermedios, la Figura 3.2, hace un resumen gráfico del algoritmo. Una vez más, el supeíndice $^{(k)}$ denota la iteración. $\Psi_{N^*}(\mathbf{X}, \mathcal{P})$ denota la expansión en bases truncadas para los datos $\mathbf{X}$ y la partición $\mathcal{P}$ (de todas las dimensiones)[8] dependiendo del los tres parámetros $M,\ J$ y K (resumidos en N^*). Finalmente, $F \in \mathbb{R}^{n \times d}$ denota las matriz de transformaciones no lineales, es decir: $F = [f_1(x_1), \ldots, f_d(x_d)]$ donde cada $f_j(x_j)$ es un vector para todas las observaciones. El ciclo de adentro, se da pues al ir calculando y actualizando cada dimensión j, cambia la transformación f_j y se deben de actualizar los residuales parciales. Este proceso se repite d veces y una vez calculados todos los vectores w_j, se regresa al bucle principal donde se simula$z$ y β una vez más.

7. Se calculan los cuartiles con saltos de probabilidad $1/J$, donde J es el número de intervalos.
8. El objeto Ψ en realidad es un arreglo 3-dimensional pues cada $x_{i,j}$ tiene una expansión de tamaño N^*. Su implementación, se basa en el diagrama 2.1 de la página 39.

Dados valores iniciales para $\pi(\beta)$ y $\pi(\mathbf{w})$, se itera $k = 1, 2, 3, \ldots$

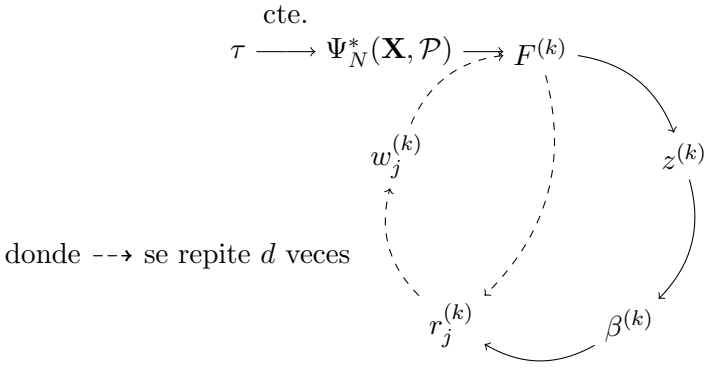


Figura 3.2: Esquema del algoritmo

Dado que los datos y los nodos son fijos, la expansión en bases de polinomios truncados únicamente se tiene que calcular una vez y es constante. Si se usara el enfoque de Denison, Mallick y Smith (1998), esta parte aislada del diagrama, debería de ser incluida en el ciclo.

En forma de pseudocódigo el algoritmo final se encuentra en la Tabla 3.3 de la página 72. Sin embargo, inclusive en esta forma, el modelo sigue sufriendo de una notación pesada debido a la gran cantidad de índices y variables,[9] sin embargo, la implementación del algoritmo final, se separó en una serie de funciones relativamente sencillas y vectorizadas que hacen de el manejo de los objetos una tarea más sencilla. El código que se desarrolló es de dominio publico disponible en `https://github.com/PaoloLuciano/BPWPM`, si se quiere hacer una exploración más

9. Como dato curioso, si se expusieran todas las ecuaciones para cada dato real $x_{i,j}$ y la expansión en bases completa, en vez de su representación vectorial, se tendrían que usar 5 índices.

```
bpwpm(y, X, M, J, K, N_sim, beta(0), mu_beta(0), sigma_beta(0),
  w(0), mu_w(0), sigma_w(0)):
    tau <- Cuantiles(X, 1/J)
    Phi <- CalculaPhi(X, M, J, K, tau)
    F(0) <- CalculaF(Phi, w(0))

    PARA k = 1 HASTA Nsim:
        z(k) <- SimNormTrunc(y, F, beta(k-1))

        sigma_beta(k) <- ((sigma_beta(0))^-1 + F(k-1)'F(k-1))^-1
        mu_beta(k) <- sigma_beta(k)*((sigma_beta(k))^-1*mu_beta(k-1)
                    + F'*z(k))
        beta(k) <- NormMulti(mu_beta(k), sigma_beta(k))

        PARA j = 1 HASTA d
            r(j) <- ResidualesParciales(z, beta, F(k))

            sigma_w(j,k) <- (sigma_w(j,k-1)^-1 + Phi'Phi)^-1
            mu_w(j,k) <- sigma_w(j,k)*((sigma_w(j,k))^-1*mu_w(j,k-1)
                   + Phi'r(j))
            w(j,k) <- NormMulti(mu_w(j,k), sigma_w(j,k))

            F <- ActualizarDimensionjDeMatrizF(Phi, w(j,k))

    REGRESAR beta, w
```

Tabla 3.3: Algoritmo *bayesian piecewise polinomial model*

profunda. Asimismo, se desarrolló mucha funcionalidad adicional para visualizar e imprimir información útil de los posibles modelos que se puedan hacer. En el Apéndice B, se hace un compendio de las funciones y una breve descripción de lo que hacen. La documentación completa se encuentra también en el paquete.

Capítulo 4

Ejemplos y resultados

El modelo presentado en este trabajo, teóricamente pesado y con una implementación tediosa que recae en técnicas de simulación, resultó ser realmente efectivo en la práctica. A lo largo de este capítulo, se hará una exploración intuitiva y visual de sus capacidades. Todas las gráficas presentadas, se generaron con el mismo paquete que realiza la estimación, pues los mismos objetos que las funciones de `R` arrojan, pueden ser utilizadas para hacer gráficas que reflejan la intuición subyacente del modelo.

En particular, se simularon cinco bases de datos en dos dimensiones, es decir, se tienen dos covariables $\mathbf{X} \in \mathbb{R}^2$, con diferentes patrones para la respuesta y tanto lineales como no lineales. Esto, con el objetivo de poder *visualizar* la clasificación y los diferentes tipos de fronteras no lineales. Asimismo, al usar bases simuladas en $\mathbb{R}^2$,

se puede visualizar la función $f(\mathbf{x})$ en tres dimensiones. Posteriormente, se aplica el modelo a una base de datos reales de cáncer, donde, al aumentar la dimensionalidad, estos no se pueden visualizar. Sin embargo, se dan una serie de resúmenes numéricos y medidas que evalúan la precisión del modelo, abriendo la discusión a limitaciones de este.

4.1. Evalución del modelo

Dos buenas medidas de evaluar la efectividad (y precisión) de un modelo de clasificación binaria, son las *matrices de confusión* y la función *log-loss* (ll).

Sea $\mathbf{y} = (y_1, \ldots, y_n)^t$ el vector de respuestas verdaderas; $\hat{\mathbf{p}} = (\hat{p_1}, \ldots, \hat{p_n})^t$ el vector de probabilidades ajustadas, donde $\hat{p}_i = \hat{P}_{\text{modelo}}(y_1 = 1|\mathbf{x}_i)$ es la probabilidad estimada por el modelo de que la observación y_i sea igual a 1, definiendo el vector de respuestas ajustadas $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_n)^t$, donde, $\hat{y}_i = 1 \iff \hat{p}_i > .5$. La función *log-loss* $ll : \{0,1\}^n \times [0,1]^n \to \mathbb{R}^+$ es,

$$ll(\mathbf{y}, \hat{\mathbf{p}}) = -\sum_{i=1}^{n} [y_i \ln(\hat{p}_i) + (1 - y_i) \ln(1 - \hat{p}_i)]. \tag{4.1}$$

La ventaja de usar la función ll, es que da una métrica que, no sólo para mide que tan buena es la clasificación binaria, sino, que toma en cuenta la precisión de la predicción. Esto se debe a la función es convexa y se penaliza cuando las

probabilidades ajustadas están muy lejos de la real. Asimismo, si la predicción fue incorrecta pero la probabilidad fue cercana a 0.5 no se penaliza tanto. Idealmente $ll = 0$ si se da una clasificación perfecta y conforme crezca, el modelo es peor. En la práctica y bajo un enfoque frecuentista, la función LL es la que usualmente se utiliza para entrenar y comparar modelos de clasificación como redes neuronales.

El segundo método, la matriz de confusión, no es más que un método descriptivo, con base en *tablas de contingencia* que calcula las frecuencias para aciertos y errores, separando en grupos. Esto es:

	$\hat{y} = 0$	$\hat{y} = 1$	
$y = 0$	#0's ✓	#0's clasificados como 1	# de observaciones cero
$y = 1$	#1's clasificados como 0	#1's ✓	# de observaciones uno
	# de estimados cero	# de estimados uno	Total de obs. $= n$

Tabla 4.1: Matriz de confusión

De donde se puede ver la exactitud en las predicciones del modelo. Estos dos métodos, serán los usados para evaluar cada ejemplo.

4.2. Análisis a fondo de una modelo sencillo

El primer ejemplo que se analizará, es un ejemplo muy sencillo, que busca ejemplificar cada componente del modelo. Se simularon un total de $n = 350$ observaciones separadas en dos grupos, cada uno con tamaños $n_0 = 200$ y $n_1 = 150$ respectivamente ($n = n_0 + n_1$). Los puntos se muestrearon de distribuciones normales bivariadas, esto es:

$$\text{Grupo 0: } \mathbf{x}_i \sim N_2\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \middle| \mu_0 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \Sigma_0 = \begin{pmatrix} 0.25 & 0.35 \\ 0.35 & 1 \end{pmatrix}\right) \qquad i = 1, \ldots, 200$$

$$\text{Grupo 1: } \mathbf{x}_i \sim N_2\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \middle| \mu_1 = \begin{pmatrix} 4 \\ 4 \end{pmatrix}, \Sigma_1 = \begin{pmatrix} 1 & .24 \\ .24 & .64 \end{pmatrix}\right) \qquad i = 201, \ldots, 350$$

Las medias μ se toman diferentes para dar una clara separación y las covarianzas corresponden a las correlaciones $\rho_0 = 0.7$ y $\rho_1 = .3$ respectivamente. Codificando el grupo 0 con $y = 0$ de color rojo y el grupo 1 con $y = 1$ de color azul. Se tienen los datos presentados en la Figura 4.1.[1] Los parámetros se escogieron con un proceso de *prueba y error* para dar estructura pero a la vez separación en el espacio de covariables $\mathcal{X}^2 \approx [0.3, 7.5] \times [-0.5, 5.9]$. Esto, para que se tuviera una pequeña región donde las distribuciones se traslaparan y exista cierto grado de confusión. El objetivo del modelo, es poder hacer una separación de estas dos regiones sin sobreajustar, identificando a grandes rasgos dónde se encuentran los puntos rojos y

1. Vale la pena mencionar, que todas las gráficas y presentadas en este Capítulo, fueron generadas usando las capacidades de la librería *ggplot2*, donde se incorporó su funcionalidad al paquete *bpwpm* para poder generar estos gráficos de una forma fácil y rápida para este tipo de modelos.

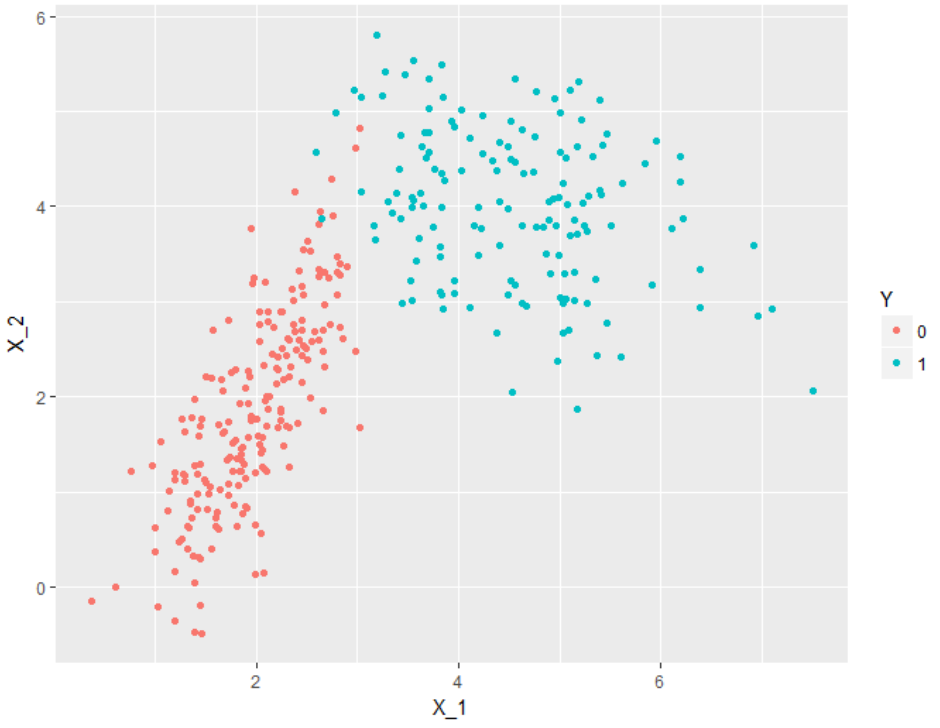


Figura 4.1: Ejemplo 1, Poco traslape entre grupos

dónde se encuentran los puntos azules.

Modelo probit frecuentista para comparar

En un modelo tradicional, la función de proyección lineal, rígida, y por ende la frontera de clasificación es lineal. Para comparar, se corrió el siguiente modelo probit fre-

cuentista en `R`, usando la función `glm(..., family = binomial(link = 'probit'))`:

$$p_i = P(y_i = 1) = \mathbb{E}[y|\mathbf{x}_i] = \Phi(f(\mathbf{x}_i)) \quad \Rightarrow$$
$$\Phi^{-1}(p_i) = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} \qquad \forall i = 1, \ldots, n \tag{4.2}$$

De donde se obtuvieron los resultados presentados en la tabla 4.2

Parámetro	Valor Estimado
$\hat{\beta_0}$	-17.29
$\hat{\beta}_1$	4.43
$\hat{\beta}_2$	1.08
Métricas	Valor
ll	0.0399

	$\hat{y} = 0$	$\hat{y} = 1$	
$y = 0$	198	2	200
$y = 1$	2	148	150
	200	150	350

Tabla 4.2: Resultados para modelo probit

Dada la simplicidad de los datos, el modelo lineal probit, presentado en la ecuación (4.2) resulta ser una excelente forma de hacer la clasificación. Como se ve en la Figura: 4.2, los datos son fácilmente separables por una linea recta que cruza exactamente donde se empiezan a traslapar. Únicamente, existen 4 datos que quedan mal clasificados, pero que, dadas sus coordenadas, parecerían pertenecer a los grupos opuestos y se consideran como datos atípicos. El modelo presenta una precisión de 98.85 %, todos los parámetros fueron significativos[2] y se tiene un valor de la función *log-loss* muy bajo. Por lo tanto, se puede concluir que este modelo probit tradicional, es un muy buen modelo para este conjunto de datos.

2. Usando las pruebas t clásicas de modelos lineales frecuentistas.

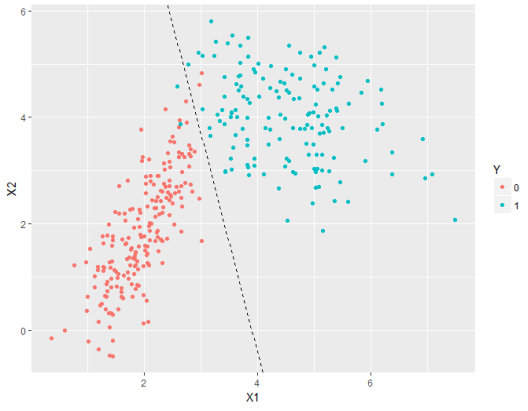


Figura 4.2: Separación de los grupos por medio de un modelo probit lineal frecuentista

Ejemplo 1: el modelo bpwpm

Ahora, esta base de datos se prueba contra el modelo *bpwpm* de este trabajo. Se esperaría que, al menos, se comportara como el probit anterior, pues, dada su estructura tan flexible, el modelo podría colapsar en uno lineal, replicando el probit tradicional anterior.

Como un primer ejemplo sencillo de comprender, se corre un modelo donde los polinomios por partes son rectas disjuntas en cada segmento con un solo nodo. Se realizan mil simulaciones del muestreo Gibbs, de donde se descartan las primeras 500 observaciones y posteriormente se toma cada segunda observación. Estas especificaciones se resumen en la siguiente tabla que se presentará antes de cada modelo.

Parámetros		Parámetro Sim.
$M = 2$	$N^* = 4$	$N_{\text{sim}} = 1000$
$J = 2$	$d = 2$	$k^* = 500$
$K = 0$	$n = 350$	$k_{\text{thin}} = 2$

Tabla 4.3: Ejemplo 1, rectas disjuntas, un solo nodo

Dado que este es un modelo extremadamente sencillo y que se tiene un número pequeño de bases para los polinomios por partes, $N^* = 4$, se presenta por única ocasión, la expansión completa para poderla comparar contra el modelo anterior (4.2). Esto es,[3]

$$
\begin{aligned}
p_i = P(y_i = 1) = \mathbb{E}[y|\mathbf{x}_i] = \Phi(f(\mathbf{x}_i)) \quad &\Rightarrow \\
\Phi^{-1}(p_i) = \beta_0 + \beta_1 f_1(x_{i,1}) + \beta_2 f_2(x_{i,2}) \qquad &\forall i = 1, n, \ldots, \qquad (4.3) \\
= \beta_0 & \\
+ \beta_1[w_{1,1} + w_{2,1}x_{i,1} + w_{3,1} + w_{4,1}(x_{i,1} - \tau_{1,1})_+] \qquad &(4.4) \\
+ \beta_2[w_{1,2} + w_{2,2}x_{i,2} + w_{3,2} + w_{4,2}(x_{i,2} - \tau_{1,2})_+]. \qquad &(4.5)
\end{aligned}
$$

Contrastando la ecuación (4.2) del modelo probit contra (4.3) del modelo *bpwpm*, se

3. Para $w_{l,j}$, se usa la convención de subíndices $l = 1, \ldots, N^*$ de la biyección de la Tabla 2.1 y $j = 1, \ldots, , d$ para indicar la dimensión

puede ver la introducción del componente no lineal a través de las funciones f_j desglosadas en (4.4) y (4.5). El modelo , tiene un total de $1+d+dN^* = 11$ parámetros, recordando que los nodos son fijos. Las expansiones truncadas de polinomios son relativamente sencillas y se ve claramente que se les da estructura de recta, permitiendo discontinuidades entre ellas pues $(\cdot)_+$ es una función por partes que se activa si $x_{i,j}$ es mayor que el nodo $\tau\cdot, j$. Aunque esta expansión es aparatosa, el modelo logra hacer excelentes predicciones en cuanto a las regiones. Cabe mencionar que, dado este es un ejemplo introductorio con un número relativamente bajo de observaciones, la estimación de los parámetros, se realizó *dentro de la muestra* (*in-sample*) esto quiere decir, que el modelo se entrena con las mismas observaciones contra las que se busca predecir.[4]

Previo al análisis de convergencia de las cadenas, se hace una exploración preliminar para explicar todos los detalles del modelo. Usando la función de perdida cuadrática, se obtienen los resultados presentados en la Tabla 4.4.

Numéricamente, el modelo se ve bien; la matriz de confusión es idéntica a la del probit anterior y, por ende, la precisión. Sin embargo, se tiene un valor de la función *log-loss* un poco más bajo, indicando que se tiene un mejor modelo una vez consideradas las probabilidades ajustadas $\hat{\mathbf{p}}$. Sin embargo, a diferencia del modelo anterior, los parámetros estimados $\hat{\beta}$ y $\hat{\mathbf{w}}$ no se pueden interpretar de la misma manera. En

4. El efecto que esto podría tener es que se sobre-ajuste o se hagan predicciones demasiado acertadas, sin embargo, es normal hacerlo para este tipo de modelos dado n. Además, por lo pronto el objetivo final es dar predicciones a través de las regiones formadas y no tanto para observaciones nuevas. De cualquier forma, se podría hacer separando, antes del análisis, la base de datos en dos, una para entrenar el modelo y otra para probarlo.

Info. predicción	
Est. Puntual	Media posterior
Precisión	98.85 %
log-loss	0.03702

$\hat{\beta}$	
$\hat{\beta}_0$	-0.79
$\hat{\beta}_1$	3.35
$\hat{\beta}_2$	0.65

$\hat{\mathbf{w}}$	
-0.52	-1.48
0.39	-0.38
0.09	0.66
1.05	1.32

	$\hat{y}=0$	$\hat{y}=1$	
$y=0$	198	2	200
$y=1$	2	148	150
	200	150	350

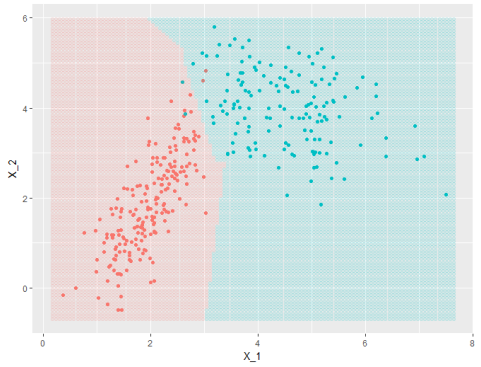
Tabla 4.4: Ejemplo 1, resultados

modelos de ML, debido a la complejidad, es mejor tratar a los parámetros como partes funcionales del modelo, que como números con significado. Sin embargo en especial el vector $\hat{\beta}$ puede considerarse como los *pesos* que se le dan a cada transformación no lineal, y su magnitud corresponde a que tanta fuerza tiene esa dimensión en el modelo. De este ejemplo se ve claramente que la primer dimensión, es con la que mejor se están explicando los datos.[5]

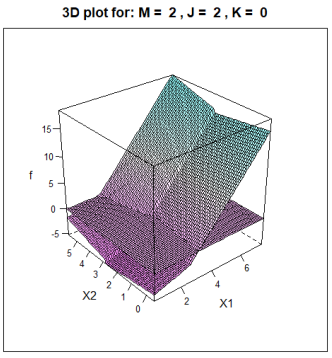
La Figura 4.3a, es clave para entender el modelo, en ella se presentan las dos regiones (de colores) que el modelo detecta para hacer las predicciones.

A diferencia de los modelos lineales donde la frontera de predicción binaria es, por ende, lineal, el modelo presentado en este trabajo permite tener fronteras tan comple-

5. Dado que se tiene una muestra arbitrariamente grande de los parámetros y se conoce su distribución, se podría probar la significancia de los parámetros.



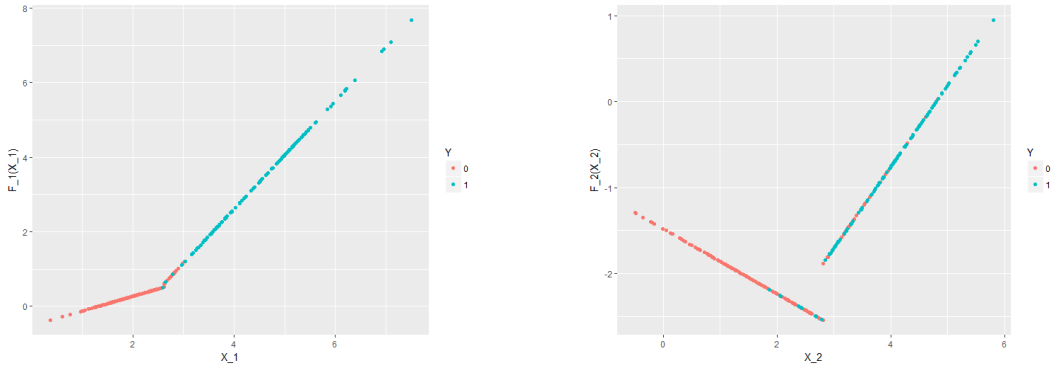
(a) Regiones de predicción para modelo con $M = 2$, $J = 2$ y $K = 0$



(b) Representación en 3D de $\hat{f}$

Figura 4.3: Ejemplo 1 con M = 2, J = 2, K = 0, modelo lineal discontinuo

jas como se quiera (aunque no siempre necesarias). Para este ejemplo en particular, la frontera refleja que se está usando un solo nodo y polinomios lineales discontinuos, es por ello que se da la rugosidad. Encontrar la frontera como tal, resulta una tarea mucho más complicada, pues correspondería a resolver la ecuación $\hat{f}(\mathbf{x}) \equiv 0$; en los GLM, esta frontera es perfectamente lineal y el despeje se puede hacer. Sin embargo, cuando se tienen modelos no lineales, se puede hacer una proyección de ella pues, recordando, lo que interesa es discernir cuando la función $\hat{f}$ sea positiva o negativa. Gracias al hecho que $d = 2$ para este ejemplo, se puede visualizar $\hat{f}(\mathbf{x})$ en la Figura 4.3b. En esta gráfica, se marca con un plano el *corte* cuando $\hat{f}(\mathbf{x})$ se vuele positiva. Además. se detectan los *pliegues* de discontinuidades derivados del nodo y de la especificación en los parámetros M, J y K. Se hace notar, que esta representación, no es más que la suma ponderada (por $\hat{\beta}$) de las transformaciones no lineales $f_j \quad j = 1, 2$, por lo cual vale la pena visualizarlas en la Figura 4.4. Aunque



(a) $\hat{f}_1(x_1)$ (b) $\hat{f}_2(x_2)$

Figura 4.4: Transformaciones no lineales para cada dimensión

la escala vertical de f_j es arbitraria, pues en realidad están ajustando a los residuales parciales, las funciones están cumpliendo su propósito de detectar y capturar el patrón que deriva en la separación de los grupos. Para valores izquierdos de ambas variables, se tiene el grupo rojo, para valores mayores, se tiene el grupo azul. Ese efecto, es capturado en el salto que dan las rectas en el nodo, mediante una mayor pendiente en las rectas del lado derecho (mayoritariamente azules), pues, al ser más positivas conforme se avanza en el rango de x, estas tendrán más peso en la función de proyección $\hat{f}$, volviéndola más positiva y por ende, más probable que esa región contenga observaciones del grupo azul.

Con estas gráficas, se espera dar claridad a todos los componentes del modelo. Este ejemplo trivial, donde $d = 2$, tiene la ventaja que todo es visualizable gráficamente. Se hace énfasis en que muchas de las variables presentadas o métodos, son puramente estructurales y que cumplen una función meramente técnica en el algoritmo, como

lo son las variables auxiliares **z**. Al final, y como se presentó en el la Figura 2.1, el modelo tiene cierta coherencia y simplicidad fácilmente representable por regiones de dos colores. Posteriormente, se verá la flexibilidad de estas regiones, ahí donde recae la fuerza del modelo.

Aunque las funciones f_j sean relativamente sencillas presentadas en una dimensión como en la Figura 4.4, una vez colapsadas en la función de proyección f, se pueden tener efectos inesperados o relativamente extraños. Esto se debe a que la interacción entre los nodos en más de una dimensión puede ser complicada de visualizar y al juntar todo, se dan efectos como los pliegues de la figura 4.3b. Para solucionar esto, usualmente se pide cierta suavidad en los polinomios por partes haciéndolos splines para que f sea también suave. Sin embargo, dependiendo de la aplicación los parámetros M, J y K se van calibrando hasta que el modelo sea aceptable y las cadenas hayan convergido.

4.2.1. Diferentes tipos de fronteras - análisis de sensibilidad

Toda la flexibilidad del modelo, depende de los parámetros M, J y K que recaen en las manos del estadista. Estos parámetros controlan la suavidad de la frontera y es importante que se entienda como cada uno afecta la estimación de los polinomios. Por lo tanto y para pasar de esta base de datos sencilla a algunas más retadoras, se juega un poco con ellos para ver sus efectos en el modelo.

Ejemplo 2: polinomios constantes

A principio, mientras se desarrollaba el paquete, parecía intuitivo que, si se está construyendo una clasificación binaria, a cada observación bidimensional $\mathbf{x}_i = (x_{i,1}, x_{i,2})^t$ se transformara en una especie de *observación de diseño*,[6] la cual codificaría si su correspondiente respuesta y_i era 0 o 1. Esta linea de pensamiento, llevo a pensar que escoger los polinomios por partes como constantes sería la mejor opción para la predicción y separación de las regiones. Por lo tanto, se corre el modelo con los parámetros presentados en la Tabla 4.5.

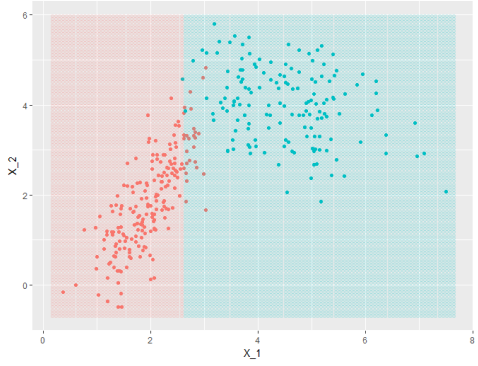
Parámetros		Parámetro Sim.
$M = 1$	$N^* = 2$	$N_{\text{sim}} = 1000$
$J = 2$	$d = 2$	$k^* = 500$
$K = 0$	$n = 350$	$k_{\text{thin}} = 2$

Tabla 4.5: Ejemplo 2, rectas constantes, un solo nodo

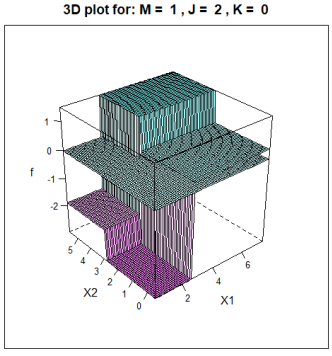
De donde se obtiene los resultados presentados en la Tabla 4.6. De ahora en adelante, y hasta llegar al análisis de convergencia, dado que la complejidad de los modelos comenzará a aumentar, se opta por no mostrar los estimadores $\hat{\beta}$ y $\hat{\mathbf{w}}$ pues no aportan nada a la discusión y son difícilmente interpretables. Se opta mejor por presentar las gráficas.

Claramente, esta no es una mejora al modelo. Tanto en precisión como en métrica *log-*

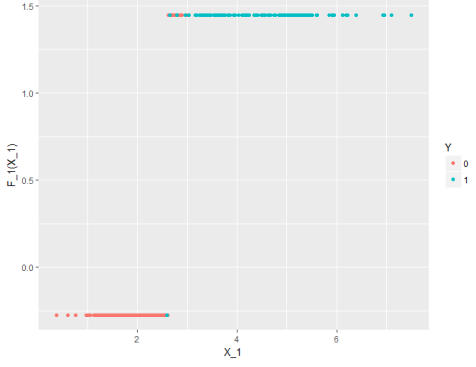
6. Derivado de las *matrices de diseño* que se construyen para los modelos ANOVA; si se pudiera mapear cada observación a un espacio más sencillo codificando la respuesta, se podría hacer una predicción aún mejor.



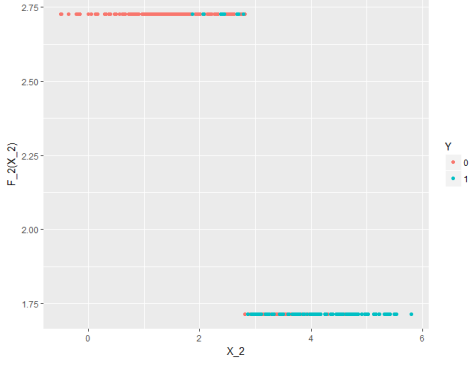
(a) Frontera



(b) Representación 3D de $\hat{f}$



(c) $\hat{f}_1(x_1)$



(d) $\hat{f}_2(x_2)$

Figura 4.5: Ejemplo 2 con $M = 1$, $J = 2$, $K = 0$, función escalonada

Info. predicción	
Est. Puntual	Media posterior
Precisión	92.3 %
log-loss	0.2081

	$\hat{y}=0$	$\hat{y}=1$	
$y=0$	174	26	200
$y=1$	1	149	150
	175	175	350

Tabla 4.6: Ejemplo 2, resultados

loss el modelo empeoró significativamente. En la Figura4.5, se visualiza el porqué. Al tener únicamente un nodo y polinomios por partes constantes, la región de predicción es la más sencilla posible: dos planos que separan las regiones rojas y azules. En la imagen 4.5b, se visualiza la función escalonada resultante con 4 *mesetas* producto de las interacciones entre los dos nodos. Se hace notar que la región inferior izquierda es más negativa pues se tiene menor probabilidad de ser del grupo azul. En las imágenes 4.5c y 4.5d se ven los polinomios constantes que siguen el diseño del experimento, esto es, codificar de forma más sencilla la variable real $\mathbf{x}$, sin embargo, al considerar las interacciones entre los nodos, la intuición se pierde. Se hace notar que en la Figura 4.5d los polinomios están invertidos, esto se debe a que β_2 se estimó negativo, por lo tanto, a mayores valores de $\hat{f}_2$, la f global aumenta, indicando que los parámetros están captando bien los patrones.

Ejemplo 3: aumento del número de nodos

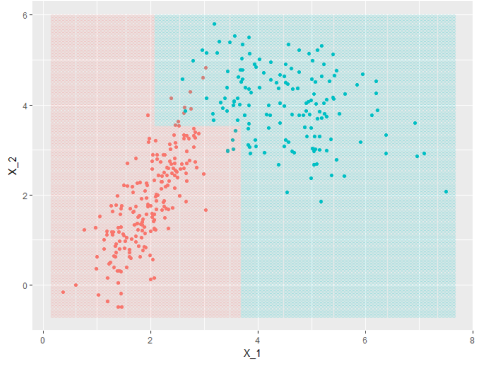
Como un tercer ejemplo, se podría pensar que aumentando el número de nodos y dejando todo lo demás constante mejoraría la predicción, sin embargo, lo hace de

forma marginal. Tomando $J = 3$ (equivalente a dos nodos) se obtiene una precisión del 95.7 %, en la Figura 4.6 se presentan los resultados visuales. Se hace notar, que en este caso en especifico, si la posición del segundo nodo en x_1 fuera ligeramente menor ($x_1 \approx 3$) la región de predicción mejoraría. De igual forma, se hace notar que se tienen $J^2 = 9$ mesetas en la representación 3D y que, en las regiones de confusión, las f_j tienen valores más cercanos a cero, identificando esta incertidumbre.

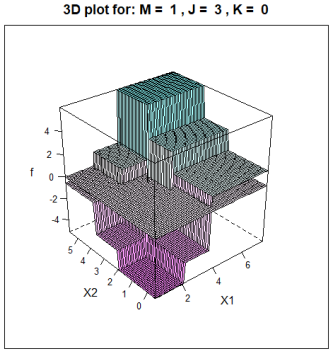
Seguir aumentando el número de nodos, no mejora sustancialmente la estimación. Por lo tanto, lo que se debe hacer es romper la linealidad aumentando M y K.

Ejemplo 4: linealidad + continuidad

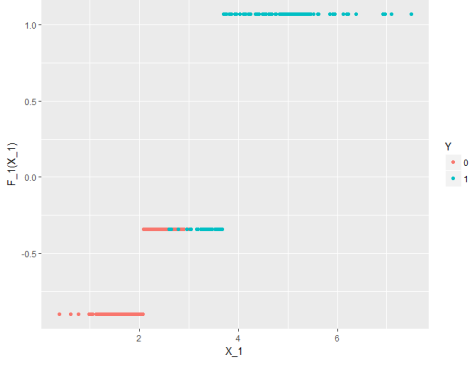
Para el primer ejemplo de la Sección 4.2, se usaron los parámetros, $M = 2$, $J = 2$ y $K = 0$. Esto corresponde a expansiones polinómicas lineales pero discontinuas. Parecería contra-intuitivo usar polinomios continuos en los nodos pensando en que se buscan predicciones binarias, sin embargo, resulta que al aumentar la *suavidad* en los polinomios, se logra al menos igualar el nivel de precisión para este ejemplo. En otras bases de datos se verá que esta suavidad es inclusive necesaria para dar mejores predicciones. Asimismo, se ve empiricamente, que al aumentar el numero de parámetros y la suavidad, se tienen estimaciones más robustas que convergen mejor a distribuciones estacionarias. Esto se debe a que al aumentar el número de variables (en este caso dos) las fronteras flexibles, logran aproximar de mejor manera la frontera real.



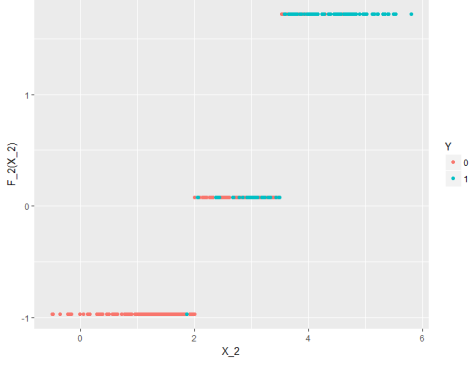
(a) Frontera



(b) Representación 3D de $\hat{f}$



(c) $\hat{f}_1(x_1)$



(d) $\hat{f}_2(x_2)$

Figura 4.6: Ejemplo 3 con $M = 1$, $J = 3$, $K = 0$

Para este nuevo ejemplo, se aumenta $K = 1$ haciendo que la doble suma en la expansión de bases de 2.17 de la página 2.17 se desvanezca regresando a la definición de splines lineales. Se tiene el modelo resumido en la Tabla 4.7, con sus correspondientes resultados presentados en la Tabla 4.8.

Parámetros		Parámetro Sim.
$M = 2$	$N^* = 3$	$N_{\text{sim}} = 1000$
$J = 2$	$d = 2$	$k^* = 500$
$K = 1$	$n = 350$	$k_{\text{thin}} = 2$

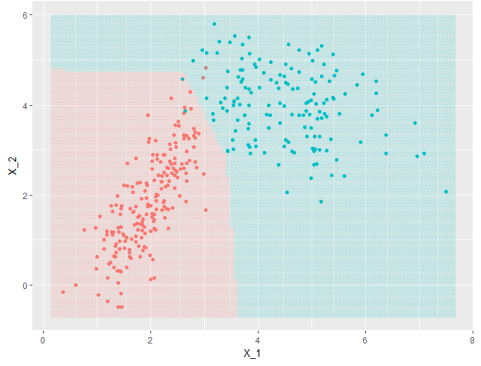
Tabla 4.7: Ejemplo 4, rectas continuas, un nodo

Info. predicción	
Est. Puntual	Media posterior
Precisión	98.85 %
log-loss	0.04109

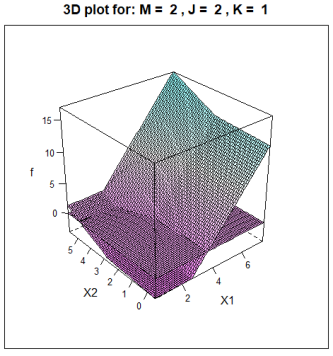
	$\hat{y} = 0$	$\hat{y} = 1$	
$y = 0$	198	2	200
$y = 1$	2	148	150
	200	150	350

Tabla 4.8: Ejemplo 4, resultados

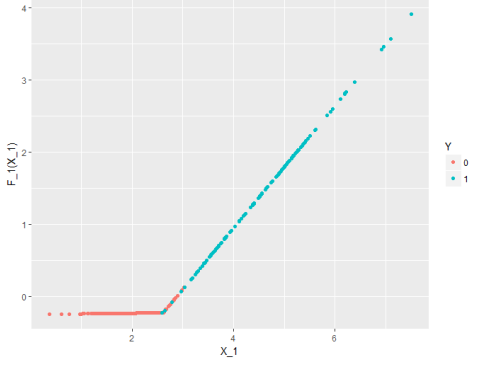
Este modelo es igual de bueno que el primer ejemplo, pero con un valor de *log-loss* marginalmente mayor. Una vez más, se presentan las cuatro imágenes habituales para evaluarlo en la Figura 4.7. Otra característica contra-intuitiva de este modelo en particular, es que se tiene un parámetro menos para cada expansión de bases, ahora $N^* = 3$, pues al añadir la restricción de continuidad por nodo se elimina uno de los dos términos independientes en las expansiones de las ecuaciones (4.4) y (4.5). Con estas imágenes, es claro ver que se está buscando mayor suavidad progresivamente. En 4.7b, se ve que la *sabana* ya no da brincos discontinuos y, aunque aún no sea suave, si retiene la estructura de predicción. Esta frontera se aprecia mejor



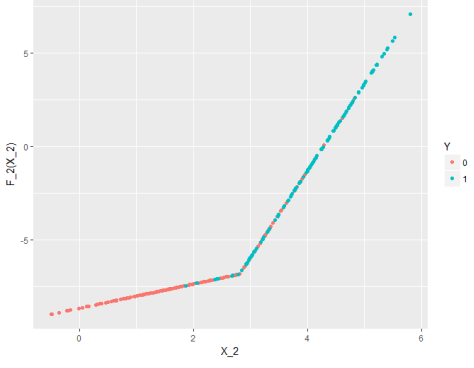
(a) Frontera



(b) Representación 3D de $\hat{f}$



(c) $\hat{f}_1(x_1)$



(d) $\hat{f}_2(x_2)$

Figura 4.7: Ejemplo 4 con $M = 2$, $J = 2$, $K = 1$, modelo lineal continuo

Parámetros		Parámetro Sim.
$M = 3$	$N^* = 9$	$N_{\text{sim}} = 1000$
$J = 3$	$d = 2$	$k^* = 500$
$K = 0$	$n = 350$	$k_{\text{thin}} = 2$

(a) Ejemplo 5, parábolas discontinuas

Parámetros		Parámetro Sim.
$M = 4$	$N^* = 6$	$N_{\text{sim}} = 2000$
$J = 3$	$d = 2$	$k^* = 1000$
$K = 3$	$n = 350$	$k_{\text{thin}} = 3$

(b) Ejemplo 6, splines cúbicos

en 4.7a, donde se ve claramente como el modelo logra detectar perfectamente bien la región problemática. Finalmente, de 4.7c y 4.7d, se pueden apreciar las rectas ya continuas. Analizándolas, se ve claramente que existe muy poca confusión en cuanto a la primera región roja por lo que $\hat{f}_1(x_1)$ es plana al principio y después comienza a crecer rápidamente. Sucede lo mismo con $\hat{f}_2(x_2)$, sin embargo, la escala es un poco diferente, ya que $\hat{\beta}_2$ es cercana a cero, confirmando la creencia de que la región se puede separar de manera excelente, únicamente con la información de x_1.[7] Asimismo, aumentar el número de nodos ya no mejora significativamente el modelo y sólo le agrega más parámetros por estimar.

Ejemplo 5 y 6: polinomios de orden mayor

Para cerrar las pruebas con esta base de datos y empezar a probarlo en regiones más interesantes, se corren dos últimos modelos con polinomios de orden mayor. En particular para el ejemplo 5, se usan parábolas discontinuas con 2 nodos para ver las capacidades del modelo. Finalmente para el ejemplo 6 se usan los famosos splines cúbicos para ver que tan suave puede llegar a ser el modelo.

7. Aunque no se implementaron, técnicas de *selección de variables* también podrían ser útiles.

Se hace notar, que para el ejemplo 6 se corre una cadena relativamente más larga y con periodo de *burn-in* k^* también mayor. Esto se debe a el ejemplo 6 es el elegido para analizar su convergencia en la Sección 4.2.2 pues se dan resultados interesantes.

Para el ejemplo 5, se obtuvieron los resultados presentados en la Tabla 4.10 con sus respectivas imágenes presentadas en la Figura 4.8.

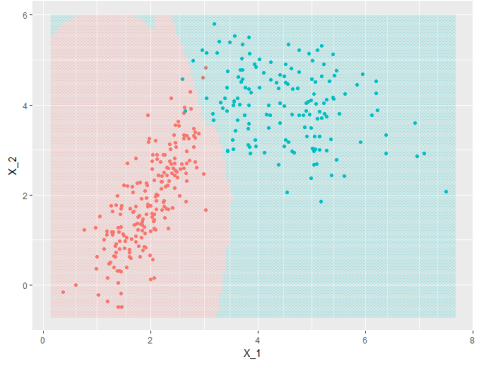
Info. predicción	
Est. Puntual	Media posterior
Precisión	98.85 %
log-loss	0.03189

	$\hat{y}=0$	$\hat{y}=1$	
$y=0$	198	2	200
$y=1$	2	148	150
	200	150	350

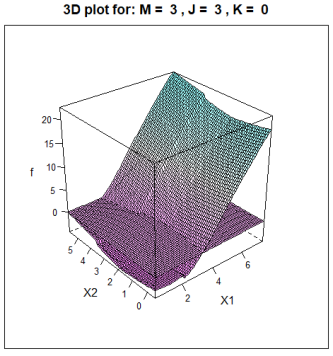
Tabla 4.10: Ejemplo 5, resultados

De donde lo único diferente es la forma de las funciones $\hat{f}_j$. La Figura 4.8c, presenta un comportamiento interesante: se conserva algo de la continuidad, casi como si las parábolas *quisieran* ser continuas pues esa es la mejor opción para la predicción. Asimismo 4.8b y 4.8a presentan claramente los efectos de las discontinuidades derivados de elegir $K = 0$. Este ejemplo, funciona mejor como un referente de las capacidades del modelo preservando la precisión, en la práctica tener 21 parámetros para un ejemplo tan sencillo no es nada útil.

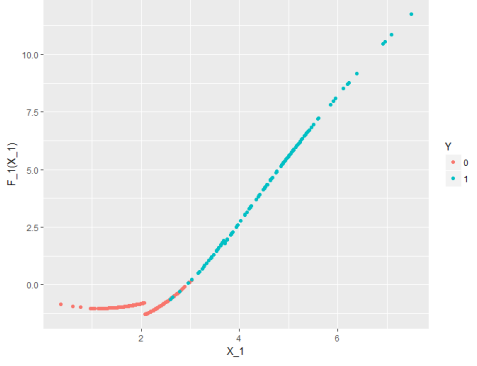
Finalmente, el ejemplo 6, es más de lo mismo, los resultados son prácticamente iguales que los obtenidos con los demás modelos. Sin embargo, se presentan las imágenes en la Figura 4.9 donde se puede ver claramente la *suavidad* obtenida en la



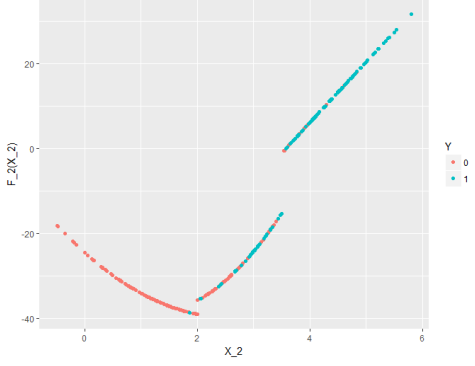
(a) Frontera



(b) Representación 3D de $\hat{f}$



(c) $\hat{f}_1(x_1)$



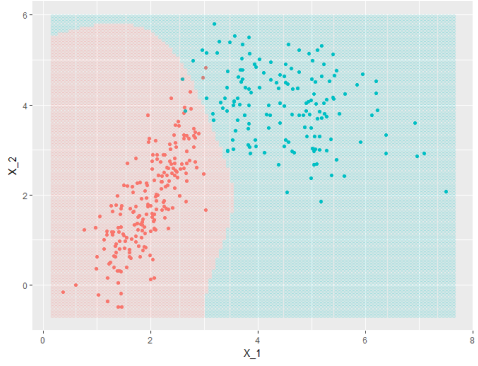
(d) $\hat{f}_2(x_2)$

Figura 4.8: Ejemplo 5 con $M = 3$, $J = 3$, $K = 0$, parábolas discontinuas

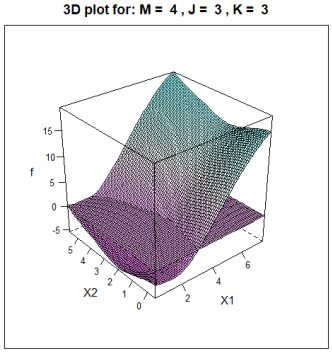
frontera, en la función $\hat{f}$ y las correspondientes $\hat{f}_j$. Al tener una cadena más larga, la escala de las $\hat{f}_j$, empieza a ser relativamente arbitraria, sobre todo para la segunda pues, conforme converge el modelo los parámetros $\mathbf{w}$ compensan la escala de β y viceversa. Se continúa con este ejemplo en la siguiente sección.

En la practica, escoger el *mejor* modelo es subjetivo pues depende del proceso de calibración de los parámetros y los resultados que se busquen obtener. Dado que el algoritmo es rápido, sobre todo para d pequeñas, se puede jugar mucho con el, y explorar una serie de modelos. En casi todos los ejemplos presentados en esta base de datos se obtuvo una precisión de 98.85 %, sería inverosímil tratar de mejorarla pues obligaría al modelo a sobreajustar y hacer regiones pequeñas para los 4 puntos que quedan en regiones de predicción opuestas. Asimismo, el número de nodos, al menos en este ejemplo, parece no importar mucho, aumentarlo, únicamente aumenta la complejidad del modelo y no aporta mucho, se ve claramente que con pocos parámetros y nodos, se tiene excelentes resultados.

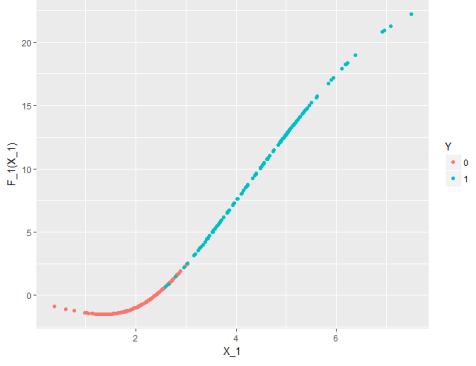
Como comentario final, se hace notar que dado el algoritmo es *estocástico* y depende de la simulación de parámetros aleatorios, replicar exactamente las cadenas es una tarea casi imposible; mas, los resultados y las regiones son consistentes. Al menos para este ejemplo, los valores de k^* y k_{thin} también parecieran no importar mucho pues las cadenas convergen muy rápido por las propiedades de las distribuciones conjugadas usadas.



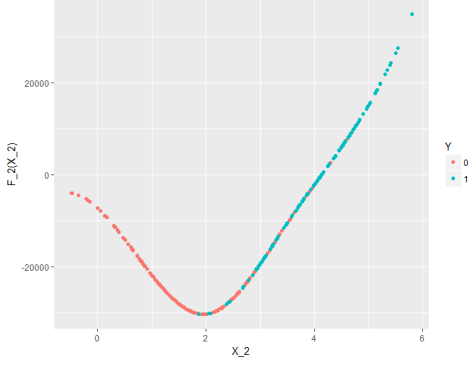
(a) Frontera



(b) Representación 3D de $\hat{f}$



(c) $\hat{f}_1(x_1)$



(d) $\hat{f}_2(x_2)$

Figura 4.9: Ejemplo 6 con $M = 4$, $J = 3$, $K = 3$, splines cúbicos

4.2.2. Análisis de convergencias

El ejemplo 6 es interesante, pues ya se había explorado, sin saberlo, cuando se hablo de mustreo Gibbs en la sección 3.2. En la Figura 3.1 de la página 58, se aprecian las trazas y los histogramas de los parámetros β derivados de la simulación del modelo. Analizando la imagen 3.1a, a primera vista, se observan varias cosas; la linea azul, que representa el parámetro $\hat{\beta}_2$ es prácticamente cero, la linea verde del parámetro $\hat{\beta}_1$ es muy consistente, casi sin variar, y la traza de $\hat{\beta}_0$ sube y baja sin aparente patrón. La imagen 3.1a únicamente confirma esta creencia.

Este es el inicio de un análisis de convergencia más profundo que se debe realizar independientemente de los resultados tan positivos que se mostraron anteriormente. Pues, se recuerda, que estos fueron derivados de estimaciones puntuales para los parámetros usando la media posterior, la cual, no se sabe si es exacta o si depende de la sección que se tome de la cadena. Asimismo, este proceso se debe realizar para todos los parámetros del modelo, es decir, además de β, se deben estudiar los vectores $w_j \quad j = 1, 2$. Por lo tanto, se realiza un análisis exploratorio de estas cadenas usando tres fuentes de información. Primero, se presentan resúmenes numéricos en la Tabla 4.11 para todos los parámetros del modelo una vez *corregidas* las cadenas.[8] Después se contrastan estos números con la Figura 4.10, donde se analizan gráficamente las cadenas por si mismas. Por último, en la Figura 4.11, se presentan las *medias ergódicas* de las cadenas. Esta medida, no es más que la media acumulada de la cadena, la cual, se espera coincidan de forma muy puntual al valor de la media

8. Descartando las observaciones antes de k^* y adelgazando cada k_{thin}-ésimo valor

posterior. Con toda esta información, se puede formar una pintura completa de la estimación y convergencia de los parámetros del modelo.

Para los parámetros $\hat{\beta}$, las medias y las medianas son en general similares como se aprecia en la imágenes 4.10a y 4.10b. Por un lado, $\hat{\beta}_0$ fluctúa bastante, teniendo un rango máximo de 4 unidades y una desviación estándar de casi una unidad, sin embargo, el rango intercuartílico queda consistentemente en los números negativos. Análizando sus medias ergódicas, tanto la total en 4.11a como la parcial 4.11b, se confirma que $\hat{\beta}_0$ está convergiendo a valores negativos alrededor de -1. Aunque su valor fluctúa y no es el más consistente, lo importante es que sea negativo y su media es un valor que funciona en la práctica. Por el otro lado, $\hat{\beta}_1$ tiene resultados más precisos confirmado por todas sus gráficas. La traza casi no varia, y tomando la segunda mitad de la cadena, su valor converge a algo cercano de 0.8. La estabilidad de $\hat{\beta}_1$, se debe a que es fundamental para el modelo, es el parámetro más significante, pues, junto con $\hat{\beta}_0$ logra capturar toda la información de los datos y resultar en los niveles de precisión presentados. Se enfatiza la importancia de el periodo de *calentamiento*, al estimar la media posterior, se necesitan únicamente observaciones de la distribución posterior; si se toma el principio de la cadena se estaría metiendo ruido al cálculo.

Algo que queda claro tanto en tablas como en imágenes, es que el parámetro $\hat{\beta}_2$ es idénticamente cero. Aunque pareciera raro, este fenómeno está perfectamente explicado también por la importancia que tiene $\hat{\beta}_1$, al haber explicado los datos únicamente en la primera dimensión con algo de ayuda de $\hat{\beta}_0$, la contribución de $\hat{\beta}_2$

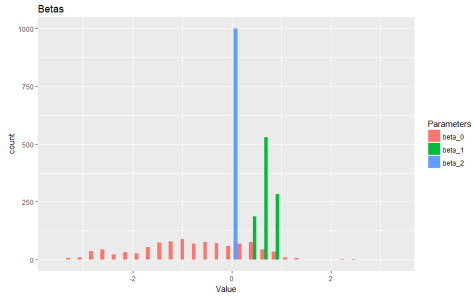
Métrica	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$
Mínimo	-3.34	0.63	0.00
Primer Cuartíl	-2.50	0.79	0.00
Media	-1.65	0.82	0.00
Mediana	-1.74	0.81	0.00
Tercer Cuartíl	-0.93	0.85	0.00
Máximo	1.01	0.99	0.00
Desviación Estandar	0.93	0.06	0.00

	$\hat{w}_1$					
Métrica	$\hat{w}_{1,1}$	$\hat{w}_{1,2}$	$\hat{w}_{1,3}$	$\hat{w}_{1,4}$	$\hat{w}_{1,5}$	$\hat{w}_{1,6}$
Mínimo	-12.60	-9.18	-23.62	-0.14	-16.06	-1.59
Primer Cuartíl	-2.44	-2.87	-2.27	0.45	-3.31	0.28
Media	-1.55	-1.94	-1.64	0.97	-2.27	1.46
Mediana	-1.24	-1.56	-1.20	0.69	-1.53	0.88
Tercer Cuartíl	-0.32	-0.72	-0.53	1.30	-0.90	2.29
Máximo	15.22	4.67	4.10	7.74	0.60	11.55
Desviación Estandar	2.10	1.74	1.94	0.78	1.98	1.84

	$\hat{w}_2$					
Métrica	$\hat{w}_{2,1}$	$\hat{w}_{2,2}$	$\hat{w}_{2,3}$	$\hat{w}_{2,4}$	$\hat{w}_{2,5}$	$\hat{w}_{2,6}$
Mínimo	0.00	-61080	-46710	-7448	-25550	0.00
Primer Cuartíl	0.00	-7509	-5396	77.7	-4444	-43.9
Media	0.00	-4693	-3515	1543	-2917	160.07
Mediana	0.00	-2846	-1887	1227	-2320	355.6
Tercer Cuartíl	0.00	-69	-82	2293	-103.2	3185
Máximo	0.00	29420	36720	4110	14410	3092
Desviación Estandar	0.00	7147	5691	1799	3562	452

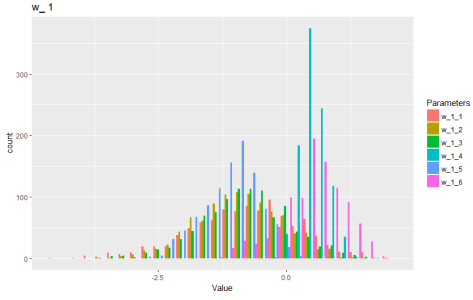
Tabla 4.11: Resúmenes numéricos para los parámetros del modelo presentado en el ejemplo 6

(a) Traza de $\hat{\beta}$



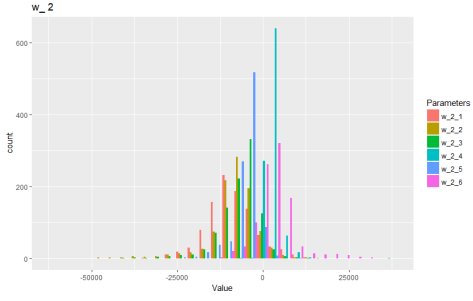
(b) Histograma de $\hat{\beta}$

(c) Traza de $\hat{w}_1$



(d) Histograma de $\hat{w}_1$

(e) Traza de $\hat{w}_2$



(f) Histograma de $\hat{w}_2$

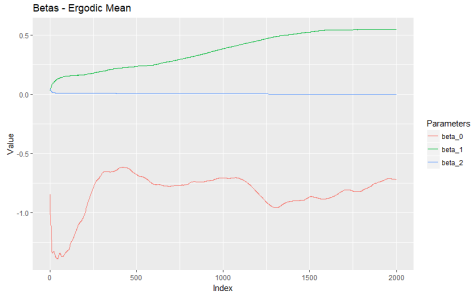
Figura 4.10: Trazas e histogramas del ejemplo 6

Se grafican los últimos 1000 valores de las cadenas del muestreo Gibbs

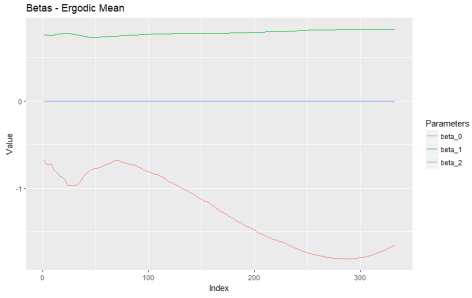
y toda su correspondiente expansión polinómica se puede obviar.[9] Asimismo, dada la falta de ortogonalidad entre las $\hat{\beta}$ y $\hat{\mathbf{w}}$, es que los resúmenes numéricos de $\hat{w}_2$ no tienen sentido. Al tener que el parámetro que controla toda la expansión de $j = 2$ es cero, los parámetros $\hat{w}_2$ pueden tomar escalar arbitrarias y fluctuar todo lo quieran sin afectar al modelo. Estas escalas y comportamiento no convergente se ilustra en las imágenes 4.10e y 4.11e. Sin embargo, al revisar de cerca la figura 4.10f, se nota que prácticamente todos los parámetros de $\hat{w}_2$ parecieran tener distribuciones normales con media en cero y lo que están haciendo es subir y bajar. La creencia es confirmada por la imagen 4.11f. Este comportamiento, aunque indeseado, no es atípico. En la práctica, llega a causar problemas de *condicionamiento* de las matrices si las cadenas son muy largas. Por ello, vale la pena *monitoriear* y hacer exploraciones preeliminares para ir descartando parámetros o probando modelos diferentes hasta lograr uno que satisfaga cierto umbral subjetivo del estadista. Para este ejemplo, no tiene mucha relevancia dado que las cadenas convergen rápido, por lo que se toma el segundo parámetro $\hat{\beta}_2$ con motivos ilustrativos.

Finalmente, se nota la tabla de $\hat{w}_1$. Aunque pareciera se tiene mucha variabilidad en la mayoría de los parámetros, en realidad, se forman cadenas bastante estables que derivan en estimaciones puntuales buenas. Además, la estimación es robusta pues se respeta la normalidad de la distribución posterior derivada de los cálculos bayesianos. Este efecto se ve claramente en la imagen 4.10d. De esta misma, se nota una clara separación en los parámetros que al final se tomarán negativos y

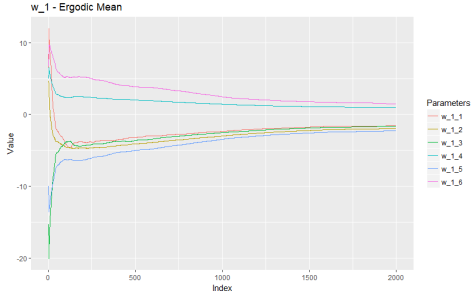
9. Este efecto también se debe a la forma que está implementado el algoritmo, los residuales parciales se van capturando de forma ascendente en las dimensiones. Si se hiciera al revés, el parámetro significativo sería $\hat{\beta}_2$ y $\hat{\beta}_1$ sería cero.



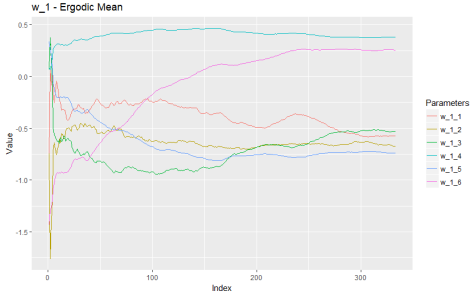
(a) Media ergódica para $\hat{\beta}$



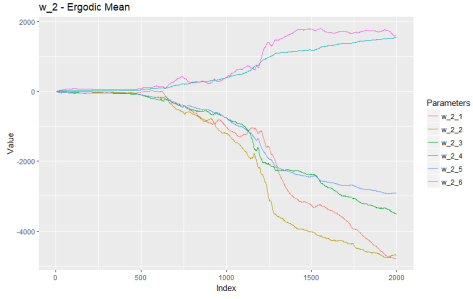
(b) Media ergódica para $\hat{\beta}$ corregida



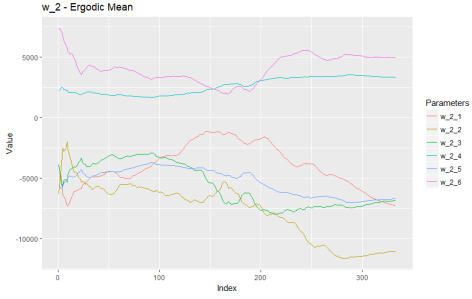
(c) Media ergódica para $\hat{w}_1$



(d) Media ergódica para $\hat{w}_1$ corregida



(e) Media ergódica para $\hat{w}_2$



(f) Media ergódica para $\hat{w}_2$ corregida

Figura 4.11: Medias ergódicas del ejemplo 6

En la primera columna, se muestran las medias ergódicas para toda la cadena, es decir, los $N_{\text{sim}} = 2000$ parámetros. En la segunda, se muestran las medias ergódicas para las cadenas corregidas con $k^* = 1000$ y $k_{\text{thin}} = 1000$

en los positivos, incluso, con procedimientos frecuentistas de pruebas de hipótesis, se podría concluir cuales de ellos pueden ser estadísticamente cero, reduciendo el número de parámetros. Se hace una mención especial a la excelente convergencia de estos parámetros que se nota en las figuras 4.11c y 4.11d, al ser realmente importantes para el modelo, controlando la curvatura de la frontera, vale la pena que sean estimados con precisión.

Es interesante notar, que la mayoría de los ejemplos presentados en este trabajo alcanzaron convergencia, pero, esta convergencia no fue tan precisa como gustaría. Sin embargo, los resultados son, sorprendentemente, tan buenos que posiblemente existan relaciones no consideradas que hacen que los parámetros capturen información adicional y hagan excelentes fronteras de predicción. Adicionalmente, estas gráficas son generadas por rutinas del paquete desarrollado para este trabajo, cuya función es automatizar, hasta cierto punto, el análisis de convergencia.

4.3. Otros resultados interesantes

Los ejemplos presentados a continuación, son más expositivos que analíticos, es decir, se enfatizan los resultados y las características fundamentales que los detalles tediosos y técnicos del modelo como se hizo en la sección anterior. Estos ejemplos y bases de datos simuladas, buscan sobre todo, poner a prueba las capacidades no lineales del modelo haciendo predicciones que serían imposibles para un GLM.

Una ligera modificación

Aprovechando la familiaridad de la base de datos anterior, se decidió modificarala para que existieran dos regiones separadas con observaciones del primer grupo. Se tomaron aproximadamente 13 puntos, más allá de $x_1 = 6$ y se cambió su clasificación. Se corre un modelo usando ahora 4 nodos y parábolas continuas resumindo en la Tabla 4.12. Los resultados se presentan en la Tabla 4.13 y la Figura 4.12.

Parámetros		Parámetro Sim.
$M = 3$	$N^* = 9$	$N_{\text{sim}} = 1000$
$J = 4$	$d = 2$	$k^* = 500$
$K = 1$	$n = 350$	$k_{\text{thin}} = 2$

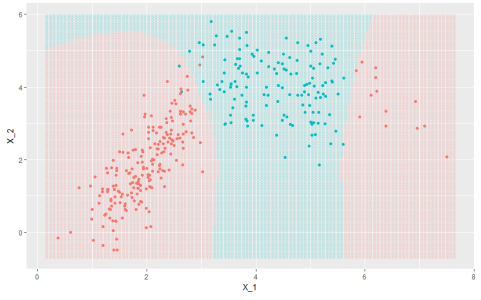
Tabla 4.12: Ejemplo 7, datos normales bivariados modificados

Info. predicción	
Est. Puntual	Media posterior
Precisión	98.6 %
log-loss	0.04217

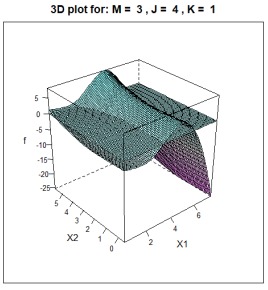
	$\hat{y} = 0$	$\hat{y} = 1$	
$y = 0$	211	2	213
$y = 1$	3	134	137
	214	136	350

Tabla 4.13: Ejemplo 7, resultados

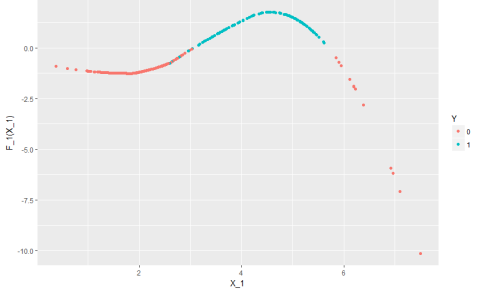
Este ejemplo es interesante pues, como se ve en la imágen 4.12b, la *sabana* que antes era creciente a medida que x_1 crecía, ahora se vuelve a curvar, volviéndose negativa otra vez y clasificando bien la segunda sección roja. Una vez más, se tienen esos pocos puntos que no quedan bien clasificados, incluyendo uno nuevo cerca de las coordenadas cartesianas $(5.8, 2.3)$. Para estos datos, se debe usar un nodo adicional



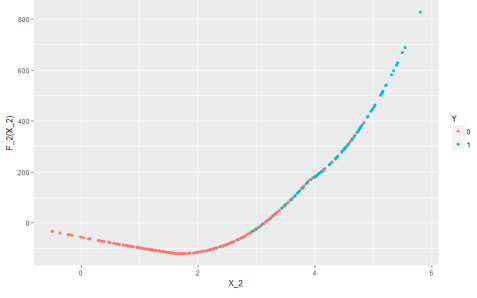
(a) Frontera



(b) Representación 3D de $\hat{f}$



(c) $\hat{f}_1(x_1)$



(d) $\hat{f}_2(x_2)$

Figura 4.12: Ejemplo 7 con $M = 3$, $J = 4$, $K = 1$

cerca de la segunda región, ya que la curvatura, deriva de él. El parámetro K en este ejemplo no es muy relevante como se ve en la imagen 4.12c, nuevamente porque $\hat{f}_1(x_1)$ pareciera ser suficientemente suave sin tener que restringir el modelo. Finalmente, se enfatiza que vuelve a suceder lo mismo que pasó con el ejemplo 6, donde la información se podía resumir únicamente con las primeras dos dimensiones.

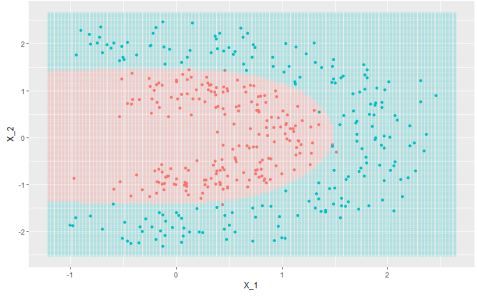
Regiones curvas

Como siguientes dos ejemplos, se buscó estresar la interacción entre las dimensiones buscando regiones más complejas. En particular, se buscó replicar algo similar a la imagen del capítulo introductorio 1.1 de la página 3. Para el ejemplo 8, se generaron datos con coordenadas polares para ángulos con rango entre $(-1, 1)$ y tomando diferentes radios para cada grupo. Posteriormente, se les sumó ruido blanco a los puntos para que existiera una región de confusión. Dadas las características curvas de los datos, piensa que usar parábolas continuas es una buena opción para modelarlos. El modelo final termina con los parámetros presentados en la Tabla 4.14. Los resultados e imágenes se presentan en la Tabla 4.15 y la Figura 4.13 respectivamente.

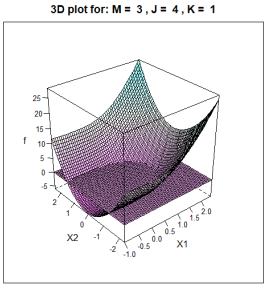
Parámetros		Parámetro Sim.
$M = 3$	$N^* = 9$	$N_{\text{sim}} = 1000$
$J = 4$	$d = 2$	$k^* = 500$
$K = 1$	$n = 400$	$k_{\text{thin}} = 1$

Tabla 4.14: Ejemplo 8, datos parabólicos anidados

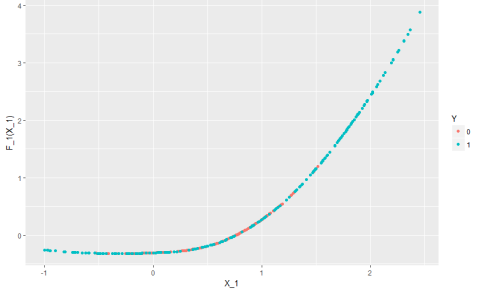
Este es un modelo particularmente interesante de forma gráfica. Se ve claramente



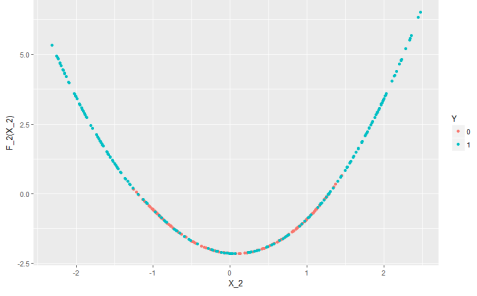
(a) Frontera



(b) Representación 3D de $\hat{f}$



(c) $\hat{f}_1(x_1)$



(d) $\hat{f}_2(x_2)$

Figura 4.13: Ejemplo 8 con $M = 3$, $J = 4$, $K = 1$

Info. predicción	
Est. Puntual	Media posterior
Precisión	98.8 %
log-loss	0.04217

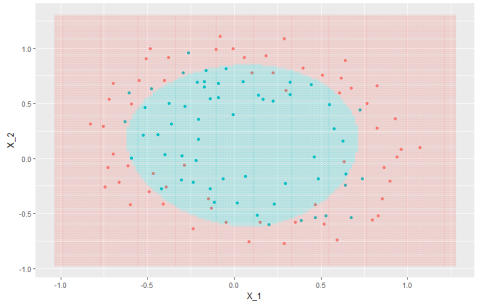
	$\hat{y}=0$	$\hat{y}=1$	
$y=0$	196	4	200
$y=1$	1	199	200
	197	203	400

Tabla 4.15: Ejemplo 8, resultados

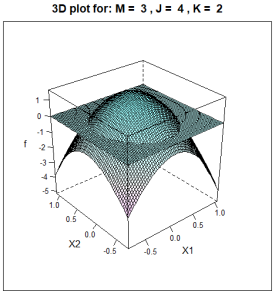
que la segunda dimensión 4.13d captura la parte parabólica y la primera 4.13d, la región donde se confunden los grupos pero posteriormente hay certidumbre. A diferencia de los modelos presentados hasta ahora, todos los parámetros β son altamente significantes, pues sin su interacción, claramente no se habría detectado el patrón. Sin embargo, estos datos siguen teniendo una clara separación por lo que el modelo sigue logrando detectar la frontera de forma correcta y tener precisión.

Continuando con las regiones no lineales, se obtuvo una base de datos pequeña del curso online de Machine Learning de NG (2018).[10] Esta base de datos se usa para entrenar modelos saturados logit con regularización, logrando predecir fronteras circulares con modelos lineales. Se decidió, probarlo también con el modelo a ver si se obtenían resultados comparables. Efectivamente se logró y con un menor número de parámetros por entrenar. El modelo, una vez más, fue ajustado con parábolas continuas las cuales resultaron ser excelentes herramientas. Se tiene el ejemplo 9 resumido en la Tabla 4.16, con resultados e imágenes en la Tabla 4.17 y Figura 4.14 respectivamente.

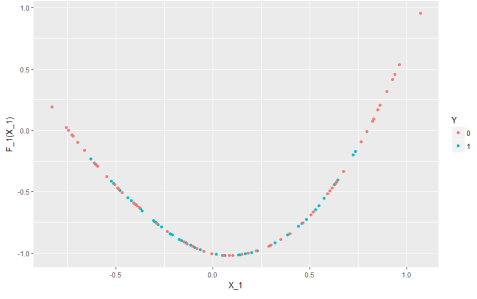
10. Este curso, se ofrece de forma gratuita en la página de Coursera



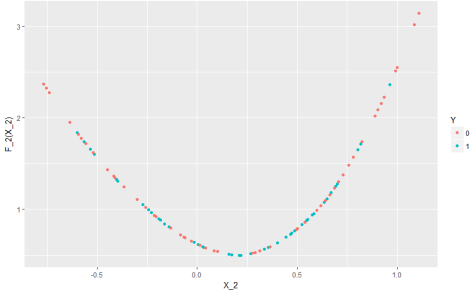
(a) Frontera



(b) Representación 3D de $\hat{f}$



(c) $\hat{f}_1(x_1)$



(d) $\hat{f}_2(x_2)$

Figura 4.14: Ejemplo 9 con $M = 3$, $J = 4$, $K = 2$

Parámetros		Parámetro Sim.
$M = 3$	$N^* = 6$	$N_{\text{sim}} = 1000$
$J = 4$	$d = 2$	$k^* = 500$
$K = 2$	$n = 118$	$k_{\text{thin}} = 1$

Tabla 4.16: Ejemplo 9, datos circulares

Info. predicción	
Est. Puntual	Media posterior
Precisión	78.8 %
log-loss	0.4532

	$\hat{y} = 0$	$\hat{y} = 1$	
$y = 0$	48	12	60
$y = 1$	13	45	58
	61	44	118

Tabla 4.17: Ejemplo 9, resultados

Todo el poder del modelo, recae en esta forma de romper la linealidad y poder estimar regiones irregularmente curvas, incluso con muy pocas observaciones. El modelo tiene un total de 15 parámetros,[11] cuando en el curso, se entrenaba con 28 parámetros.[12] La forma en la que interactún los nodos, combinando las dos parábolas que se forman en 4.14c y 4.14c es muy interesante, pues, aunque se tienen 3 nodos, lo optimo para estas figuras es formar dos parábolas continuas y suaves al aumentar K.

11. 3 en beta β más 2×6 de los vectores w_j
12. Cabe mencionar que, dada la regularización, muchos de estos términos ser desvanecían.

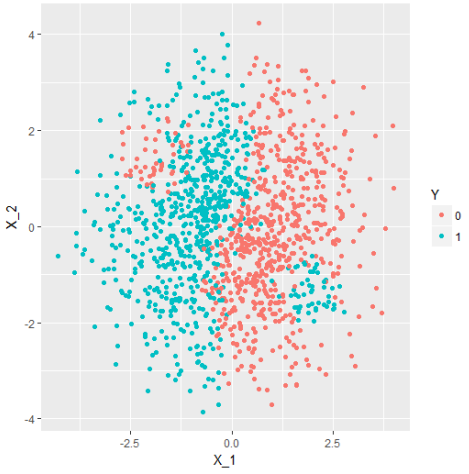
Ultimo ejemplo con datos simulados - limitaciones del modelo

Para finalizar con las bases de datos simulados, el modelo se llevó al límite de sus capacidades sobre un patrón de puntos, intuitivo al ojo humano, pero realmente difícil de identificar por un algoritmo.[13] Los datos tratan de simular un *yin-yang* que se puede ver en la Figura 4.15a. La simple simulación de la base de datos representó un reto donde se conjuntaron varias áreas de la matemática aplicada. En el software `GeoGebra`, se generó el diagrama presentado en la Figura 4.15b que consiste de las siguientes desigualdades cartesianas:

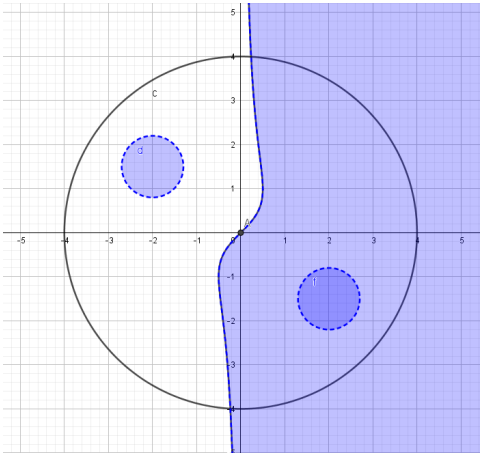
$$\begin{aligned} x^2 + y^2 &< 16, \\ (x+2)^2 + (y-1.5)^2 &< 0.49, \\ (x-1.5)^2 + (y+2)^2 &< 0.49, \\ x &< \frac{y}{(1+y^2).} \end{aligned}$$

Una vez *dibujadas* las ecuaciones, se generaron dos bases de datos de aproximadamente $n \approx 1500$ observaciones. La primera, con distribución uniforme dentro del círculo,[14] y otra usando una distribución normal bivariada simétrica ($\rho = 0$) pero con desviación estándar $\sigma = 2.5$ para abarcar todo el circulo. A todos estos puntos se les asignó la categoría 0, posteriormente, se asignó la categoría 1 a los puntos que cayeran en las regiones deseadas. Al final, se le añadió algo de ruido normal a

13. O al menos el presentado en este trabajo
14. Usando coordenadas polares



(a) Datos simulados representando un yin-yang



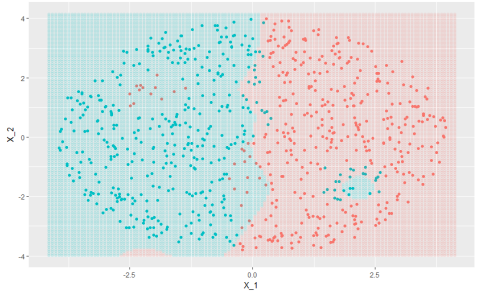
(b) Salida del software donde se construyeron las ecuaciones para generar los datos.

Figura 4.15: Patrón yin-yang

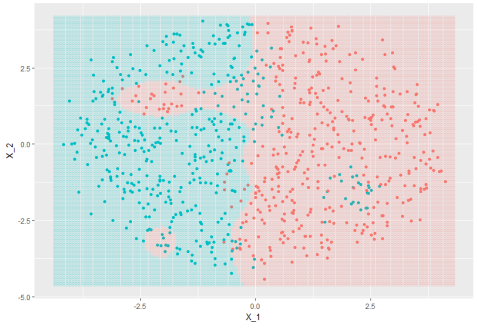
cada punto para darle aleatoriedad a la base de datos pero manteniendo el patrón terminando así la simulación.

El modelo se corrió, con muchas esperanzas, un sinfín de veces, tratando de calibrar los parámetros y captar exactamente el patrón. Sin embargo y aunque el modelo casi siempre lograba una precisión de cerca de 85 %, no se lograron los resultados esperados. De cualquier forma, el modelo y el algoritmo, claramente están haciendo su mejor trabajo y los parámetros convergen. En la Figura 4.16 se pueden ver las fronteras de algunos de los mejores modelos.

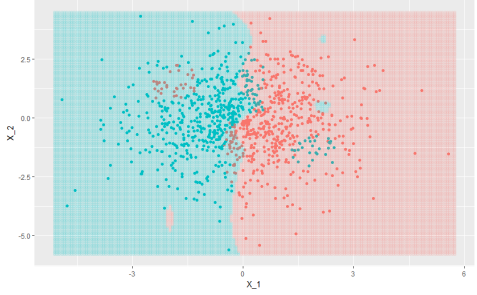
Para las dos primeras imágenes 4.16a y 4.16b, se usa la base de datos uniformes, la primera sin ruido y la siguiente con. El agregarle ruido, hace que los datos no sean



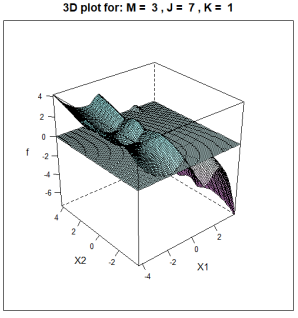
(a) Datos uniformes sin ruido, buen ajuste



(b) Datos uniformes con ruido, mejor modelo



(c) Datos normales con ruido



(d) Gráfico 3D para uno de los modelos

Figura 4.16: Fronteras de varios modelos para datos yin-yang

tan uniformes en el espacio y que ciertas características se vuelvan más prominentes que otras. Es por ello, que en la segunda imagen, se logró detectar la zona roja de la izquierda, a costa de perder algunas observaciones azules en el *puente* que se forma para llegar a ella. Sin embargo, en la primera imagen, el modelo no sólo detectó relativamente bien la curva de en medio, sino que detecta de forma aislada, el circulo azul de la esquina inferior derecha. En la tercera imagen 4.16c, se usan los datos normales con ruido de donde se ve claramente que el modelo detecta que existen regiones que debería de estar estimando dentro de las categorías opuestas. Finalmente 4.16d, muestra una, de las muchas representaciones 3D que se hicieron al tratar de ajustar esta base de datos.

Precisamente en esta última imagen se esconde el porqué no se logró hacer la estimación correcta: la dependencia implícita entre los nodos. Estos nodos, en realidad están dividiendo el espacio bi-dimensional en una cuadrícula donde las interacciones son difíciles de discernir. Conforme aumenta el número de nodos, más complejo se vuelve el modelo. Es por ello, que los picos y valles se repiten en un patrón uniforme. Asimismo, dada la naturaleza global de los polinomios y esta interacción, el modelo tiene esta estructura decreciente siempre, derivando que los picos y los valles nunca alcancen las regiones extremas en polos opuestos. De igual forma, la uniformidad y simetría impar, inherente a esta base de datos, llevó a que la estimación de los parámetros fuera óptima dentro de las capacidades del modelo. Otra desventaja de esta base, es que estos modelos se tuvieron que correr con un número grande de nodos $J \approx 20$, derivando en un número de parámetros aún más alto $N^* \approx 50$. Sin embargo, el tiempo de estimación para cadenas de más de 4000 observaciones y

alrededor de 100 parámetros, nunca excedió el minuto.

4.4. Prueba con datos médicos reales

Aunque interesantes, hasta ahora, todos los resultados de este trabajo han sido sobre bases de datos simuladas sin trascendencia alguna. Claramente forman imágenes atractivas por construcción, pero no se está prediciendo nada ni formando modelos aplicables en la vida real. Por lo tanto, y para hacer una última prueba de el modelo, se presenta una base de datos de cáncer de mama de la Univeridad de Wisconsin. Esta base de datos, es citada en varios trabajos de los años noventa, donde se tratan de hacer clasificaciones binarias usando una serie de procedimientos más robustos que los tradicionales GLM (Mangasarian, Setiono y Wolberg 1990), (Bennett y Mangasarian 1992).

De manera general y sin entrar en la parte médica de las variables como tal, se presenta un análisis exploratorio preliminar que se lleva a cabo para seleccionar las que se consideren relevantes. La base de datos cuenta con $n = 699$ observaciones de las cuales el 34.5 % representan pacientes infectados con tumores malignos representados por el color rojo. Se cuenta con diez variables (dimensiones) médicas sobre las características de los tumores como lo son: el tamaño, la uniformidad de la pared celular, etcétera. En la Figura 4.17, se muestran los gráficos de puntos *pareados* para todas las posibles combinaciones, además de información adicional. Este proceso, se lleva a cabo para tratar de seleccionar las variables relevantes y/o, discernir alguna región

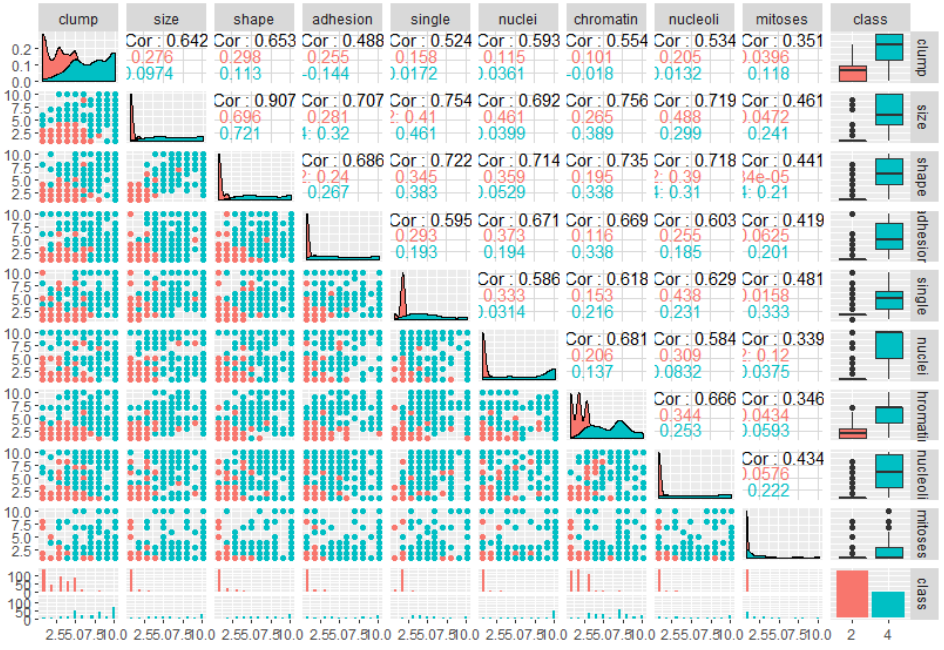
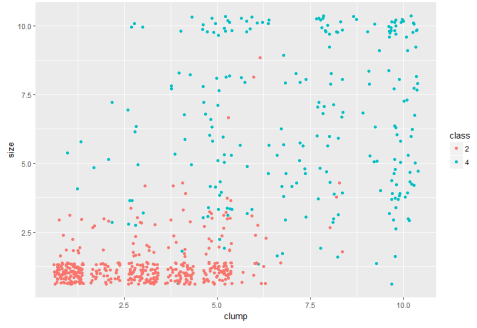


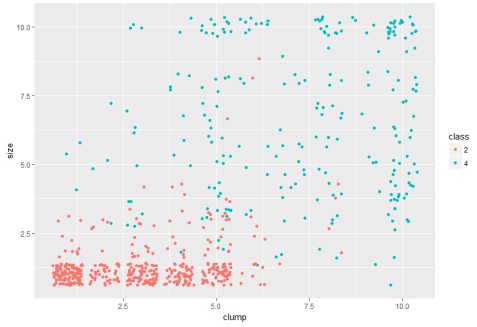
Figura 4.17: Análisis exploratorio para selección de variables

que se pueda separar por proyectores no lineales. Se hace notar que las variables, están codificadas en una escala a 10 puntos, por lo tanto, la representación gráfica de los datos se ve más como una cuadrícula que como un espacio real de variables. Se seleccionan las variables *clump, size* y *chromatin*[15] debido a que pareciera ser las que mejor separan el espacio. En la Figura 4.18 se presentan dos gráficos de puntos con algo de ruido para hacer notar que las regiones son un poco más complejas de lo que podría parecer a simple vista, además de que se tienen puntos idénticos con clasificaciones contrarias. Sin embargo, si se detecta cierto patrón en los datos.

15. Estas variables corresponden a el espesor de los tumores, su tamaño y la textura de la cromatina en las células respectivamente



(a) Variables *clump* y *chromatin*



(b) Variables *clump* y *size* con algo de ruido para separar las observaciones.

Figura 4.18: Gráficos con ruido para separar las observaciones

Para poder hablar de *predicción* como tal, tiene que existir una base de datos contra la cual probar las estimaciones del modelo. Por lo tanto, la base original se decide partir en dos, un conjunto de entrenamiento con el 60 % de las observaciones ($n = 274$) y un conjunto de prueba, con las observaciones restantes sobre las que se evaluará el modelo.[16] El modelo final, se resume en la Tabla 4.18, y consta de parábolas continuas con cuatro nodos. Como de costumbre, los resultados numéricos se presentan en la Tabla 4.19.

Parámetros		Parámetro Sim.
$M = 3$	$N^* = 11$	$N_{\text{sim}} = 4000$
$J = 5$	$d = 3$	$k^* = 3000$
$K = 1$	$n = 409$	$k_{\text{thin}} = 0$

Tabla 4.18: Prueba con datos médicos reales

16. La diferencia de 16 observaciones entre la suma de entrenamiento y prueba, contra las 699 originales, se debe a que estas estaban incompletas y por lo tanto se descartan.

Info. predicción	
Est. Puntual	Media posterior
Precisión	96 %
log-loss	0.2199

	$\hat{y}=0$	$\hat{y}=1$	
$y=0$	170	8	178
$y=1$	3	93	96
	173	101	274

Tabla 4.19: Datos médicos, resultados

Estos resultados son excelentes pues, incluso haciendo una predicción *fuera de muestra* se logra una precisión del 96 %, significando, que inclusive en $d = 3$ el modelo logra hacer una buena separación. Sin embargo, derivado también de lo mismo, es que no se pueden hacer visualizaciones como en los ejemplos anteriores. La convergencia es clara aunque, si se revisan los resultados numéricos, las escalas son relativamente arbitrarias. El algoritmo, aunque bueno en general, sufre de un problema de estabilidad numérica. Al aumentar el número de parámetros, sobre todo a través de d, las cadenas empiezan a divergir mucho y muy rápido, haciendo que la estimación de los parámetros sea errónea. Por lo pronto y para $d \leq 4$, el algoritmo funciona bien para cadenas cortas y, aunque pueda ser inestable a la larga, da resultados predictivos muy buenos. De igual manera, para esta base de datos, la codificación de las variables usando una escala de 10 puntos, no es óptima para el modelo pues los nodos se asumen reales.

Capítulo 5

Conclusiones

El desarrollo de un modelo de *machine learning*, terminó por derivar en el estudio y aplicación de múltiples áreas de las matemáticas, profundizar en temas como la simulación, modelos estructurados y probabilidad bayesiana, fue una tarea altamente edificante. Fue muy interesante el reto que representó el entendimiento de un modelo tan complejo como el presentado en este trabajo y fue aún más gratificante que el modelo funcionara tan bien como lo hizo.

5.1. Consideraciones finales del modelo

Este trabajo, como todo modelo estadístico, no está exento de contratiempos, limitaciones y consideraciones que se deben tomar en cuenta a la hora de aplicarlo. Aunque, en general, es un buen modelo de clasificación supervisada, siempre hay que tomarse los resultados con cautela crítica y ponerlos a prueba. Los modelos estadísticos, tanto por sus características como por el uso de datos muestrales, son aproximaciones a la realidad y deben ser usados con criterio. Sin embargo, es innegable que se estén convertido en herramientas, útiles y necesarias, en la mayoría de los ámbitos de la civilización contemporánea, como lo son las finanzas, la ciencia y la salud.

Convergencia y sus implicaciones

En particular, este capítulo busca revisar las limitaciones y contratiempos que podrían surgir. Como primer punto, repetido a lo largo del trabajo, se revisa la convergencia del modelo pues esta es fundamental. Lograr cadenas siempre convergentes, estables y que tengan la distribución posterior deseada es difícil. Esto, pues los métodos de simulación bayesianos, dependen de parámetros, variables, algoritmos y generadores de números aleatorios, y sería inútil pedir que todos los modelos convergencia a la perfección. Los algoritmos MCMC aunque complejos y hasta cierto punto misteriosos, se deben entender y *afinar* para la aplicación en concreto.[1] Sin embargo, no por

1. Al principio de este trabajo, se consideró usar un algoritmo de cadenas de Markov *Hamiltonianas*, que combina ideas de física para lograr estimaciones más robustas y con menor correlación

la dificultad, se debe obviar la convergencia, de otra forma, se estaría tratando de acertar (dejando todo a la suerte) a los valores subjetivos y sesgados del estadista. Se debe de tener cierto criterio para aceptar desviaciones y mejor aún, entenderlas y tratar de corregirlas. En general todos los ejemplos presentados en este trabajo convergieron de forma relativamente buena, pero sobre todo *replicables*, lo cual indica que si existe un patrón que el modelo está encontrando y no fue un golpe de suerte estadístico el encontrar las regiones de separación.

La convergencia del modelo, ahora, vista desde el punto de vista computacional y no tanto bayesiano, es uno problema más importantes de los que sufre aún el algoritmo. No es raro, que al aumentar d, algunos parámetros empiecen a tener problemas de escala y terminen por divergir. El ejemplo 6, sobre el que se realizó todo el análisis de convergencia, sufre justamente de este problema. Si la cadena fuera más grande, el parámetro $\hat{w}_2$ hubiera seguido creciendo y el algoritmo (que depende de inversión de matrices) hubiera terminado por caer en errores de condicionamiento numéricos. Sin embargo, la fuente del error es bien conocida; si se revisa la ecuación de los residuales parciales 3.20 aplicados al modelo, se ve claramente cuando si $\beta_{j^*} \to 0$, los residuales parciales de esta variable o dimensión $r_{j^*} \to \infty$ y por lo tanto el vector w_{j^*}. Esta falta de ortogonalidad entre parámetros β y $\mathbf{w}$ es complicada de corregir, usualmente, se deja de usar β enteramente y se trata de capturar toda la información a través de $\mathbf{w}$ como en los GAM tradicionales. Sin embargo, ningún algoritmo es mejor que otro y los resultados que se lograron fueron suficientemente aceptables.

entre los parámetros. Sin embargo, dada la complejidad en su aplicación al modelo, se optó por usar algoritmos más sencillos y fáciles de implementar directos del trabajo de Albert y Chib (1993)

Calibración de los parámetros y velocidad del algoritmo

Aunque se podría pensar que al mover $M,\ J$ y K a discreción del estadista se están sesgando los resultados, en realidad es sólo una consecuencia de haber escogido un modelo tan complejo y estructurado. En prácticamente ningún modelo estadístico, incluso en los no paramétricos, se puede dejar todo al algoritmo y que esta encuentre el modelo perfecto. Siempre habrá un parámetro o variable que se debe de *afinar*, lo cual introduce una dimensión subjetiva al modelo. La diferencia para este trabajo, es que se tienen que afinar algunos parámetros más. Sin embargo, este proceso de calibración, se puede hacer de tal forma que no sea por *fuerza bruta*, solamente buscando obtener resultados; por el contrario, la calibración debe ser un proceso analítico, que analiza el porqué esa selección particular de parámetros no funcionó y modificarlos en respuesta. En la practica sin embargo, y con excepciones contadas,[2] la selección de $M,\ J$ y K para las bases de datos sencillas era prácticamente trivial y el modelo siempre capturaba el patrón, con diferentes tipos de fronteras; como se vio en los primeros ejemplos del análisis de sensibilidad de la sección 4.2.1.

Es curioso notar, que aunque el modelo sea complejo y pueda crecer rápidamente en el número de parámetros modificando $M,\ J$ y K, la velocidad del algoritmo es bastante buena. Gracias a las optimizaciones realizadas en los cálculos parciales y el uso de distribuciones conjugadas, la simulación de un gran número de parámetros es relativamente trivial. Fuera de esos casos, prácticamente todos los modelos corridos, se terminaron en un minuto o menos. Aquello que hace que el algoritmo sea más

2. Usualmente para casos límite cuando $K = 0$

lento, usualmente es aumentar d o escalar n varias ordenes de magnitud. Gracias a la fácil disponibilidad y aplicación del paquete *bpwpm*, se exhorta al lector probarlo sobre diferentes datos y problemas, ya que sería interesante verlo aplicado en otros contextos y datos. Además, el algoritmo se puede ir mejorando con contribuciones externas.

Otro factor importante que influye en la velocidad del algoritmo es el uso de un paradigma bayesiano en el entrenamiento. Esta decisión se toma más que nada por cuestiones personales, ya que la filosofía bayesiana de *actualización del conocimiento* resuena mucho con aquella del autor. Sin embargo, el paradigma frecuentista es muy valioso por si mismo y en este caso (usando métodos tradicionales de estimación) hubiera logrado que el algoritmo, fuera casi instantáneo par aun número enorme de parámetros, sin embargo, este enfoque hubiera requerido hacer un trabajo completamente diferente, con sus pros y sus contras.

5.2. Posibles mejoras y actualizaciones

La fuerza del modelo recae en el gran número de componentes que tiene, sin embargo, este número también le otorga cierta *flexibilidad*, no tanto en la estimación, sino en su estructura. Cada parte que contiene, se puede modificar de una infinidad de formas, haciendo el modelo más complejo o más sencillo, más robusto o para otras aplicaciones. Al final, este no es infalible y siempre hay espacio para mejorar.

La primer y más urgente mejora que se propone explorar, es la de incorporación de un método para la *selección de variables*. El enfoque de la estadística frecuentista, especialmente para modelos de regresión, es buscar aquellas variables *más significativas* para la predicción de la respuesta. Existen procedimientos iterativos *hacia adelante y hacia atrás*, que exploran el espacio de 2^d modelos posibles y encuentran el mejor usando criterios análogos al de la función log-loss usada en este trabajo.[3] Los métodos de ML más recientes son especialmente efectivos en este ámbito; sus algoritmos recaen en usar cantidades enormes de información con múltiples variables ($d >> 0$) para hacer predicciones robustas al entrenar miles de parámetros. Bajo un paradigma bayesiano la selección de variables también se puede tratar bajo esta óptica. Los métodos más usados, incorporan otra serie nueva serie de variables auxiliares (usualmente indicadoras), cuya función es *detectar* cuando una variable es relevante o no. A estas variables, también se les da un tratamiento bayesiando y son estimadas por los mismos algoritmos MCMC a la par de todas las demás (O'Hara, Sillanpää y col. 2009).

Para este trabajo sin embargo, esta selección de variables se hizo de manera manual (y subjetiva hasta cierto punto) tomando únicamente aquellas que se consideraban importantes o útiles, derivado de una exploración a priori de los datos. La urgencia de incorporar esto al modelo, se debe a que la selección de variables, no sólo se realiza en afán de simplificar los modelos, sino por una razón computacional de convergencia. Por lo mismo que se discutió arriba, cuando una β_j era cercana a cero, las cadenas tendían a diverger, haciendo la estimación imposible. Asimismo, la colinealidad entre

3. Usualmente el criterio de Akaike

variables puede exacerbar este problema, volviendo la identificación de variables relevantes una cuestión todavía más urgente para el modelo. Por lo pronto, para d entre 1 y 4, el modelo funciona bien, solamente se debe tener en mente la longitud de las cadenas.

La siguiente modificación interesante está en la selección automática de posiciones nodales. La principal razón por la que no se logró estimar perfectamente el ejemplo del *yin-yang* se debe a que los nodos se concentraban hacia el centro donde hay más datos y no en los pequeños círculos donde se necesitaban. Esto viene derivado de que hasta el momento, sus posiciones se eligen en los cuantiles de los datos. Como se mencionó, el mismo trabajo rector de este trabajo Denison, Mallick y Smith (1998), considera un método para realizar esto, pero implicaría usar métodos más avanzados en el algoritmo MCMC pues las dimensiones cambian de forma constante. Balancear esa capa adicional con la estimación de todos los parámetros, latentes y no latentes, salía del enfoque de este trabajo y hubiera mejorado marginalmente las estimaciones presentadas. Sin embargo, vale la pena tomarlo en cuenta para el futuro.

Otra modificación considerada es volver el algoritmo de muestreo Gibbs en algo menos rígido. Como se menciona en el Capítulo 3, se toman distribuciones conjugadas para el proceso de aprendizaje bayesiano pues simplifica mucho la derivación de la ecuación (3.4), conviriendola en (3.9) lo cual permite que el muestreo sea sencillo, requiriendo únicamente álgebra lineal y simulaciones de distribuciones normales multivariadas. Aunque el supuesto no es malo, sería bueno poder incorporar distribuciones a priori arbitrarias, para poder reflejar conocimiento previo de la base de

datos o información de expertos. Hacer esta modificación sin embargo, si requeriría de cambiar sustancialmente todo el algoritmo, y por ende las derivaciones, Asimismo, se estaría obligando a usar paquetes de software que permitan estimaciones más generales como las librerías `STAN` o `BUGGS` que, aunque son excelente herramientas bayesianas, no eran el lenguaje que se planeaba usar para este trabajo.

Se hace notar que el algoritmo se implementó en el software estadístico `R`. Aunque `R` tiene múltiples ventajas en el uso de estructuras y cálculo de medidas estadisticos, no es el lenguaje más veloz pues corre a un nivel muy alto. Si se pensara usar el algoritmo para aplicaciones más robustas, se recomendaría usar lenguajes de nivel más bajo como `C++`.

Como última modificación, se considera que si se usara una expansión de bases diferente, sería posible mejorar tanto la velocidad, como la precisión del algoritmo más allá de los nodos. La expansión en bases truncadas es buena y en la práctica funciono muy bien, sin embargo, es computacionalmente lenta. Si se incorporara el cambio en la posición de los nodos sería forzoso recalcular las matrices Φ_j múltiples veces, haciendo que el algoritmo se volviera lento. Haciendo un cambio de bases, se puede usar un conjunto de b-splines que representen exactamente el mismo polinomio sustancialmente más rápido. Asimismo, esta modificación permitiría incorporar los *splines naturales* que no fluctúan tan rápido, más allá de la frontera.

Estas capacidades adicionales, robustecerían en gran forma al modelo y lo harían una herramienta muy poderosa. Si se pensara en usar el modelo para aplicaciones a gran escala, con miles de datos más, sería vital incorporarlas. Sin embargo, para

efectos de este trabajo, muchas de estos problemas, se pueden superar de formas sencillas y no fueron en realidad contratiempos para los ejemplos presentados.

5.3. El aprendizaje de una máquina

El mundo de la estadística computacional ha sido revolucionado en las últimas décadas. Gracias a los grandes estadistas como los citados, que han visto más allá de los métodos tradicionales, es que se han dado avances astronómicos en las posibilidades. Eso, aunado al aumento exponencial en las capacidades de cómputo, los modelos, se han vuelto cada vez más poderosos y útiles en la vida real.

Algunos de los métodos de ML no son más que modelos GLM como el presentado, que se corre miles de veces sobre bases de datos gigantezcas, donde existen capas de regresiones y un sinfín de parámetros por estimar. Las redes neuronales por ejemplo, son regresiones sucesivas entre *neuronas* de información, que no son otra cosa más que variables latentes z intermedias. Cada capa de neuronas, va captando patrones subyacentes de los datos. Las neuronas, se dice que se activaron cuando la función liga, después de colapsar dimensiones, rebasa cierto umbral. Este proceso se corre miles de veces entre miles de neuronas[4] logrando detectar patrones cada vez más complejos. Si para este trabajo se usan muchos índices, en los textos de ML se usan incluso más. Al final, fuera de las capacidades de estos modelos y su complejidad, la gran mayoría, son *regresiones glorificadas* que se basan en los mismos principios que

4. Usualmente de manera frecuentista.

el presentado en este trabajo. Por lo mismo, valía hacer una exploración a fondo de uno modelo análogo.

La fuerza que han adquirido las técnicas de ML en los últimos años, es que han logrado romper con muchos de los paradigmas tradicionales. Estos responden preguntas como: ¿se pueden aplicar modelos estadísticos a imágenes y sonidos? ¿por qué restringirse a dos categorías en la respuesta? y ¿a cuantos datos y variables se puede aplicar?. En general, las respuestas son más que positivas, tanto, que dispositivos de de uso diario, usan estos modelos para clasificar fotos, recopilar información o entender el lenguaje hablado. Los modelos han sido clave para el desarrollo de un mundo cada vez más futurista y probablemente seguirán avanzando en sus capacidades. Entenderlos y poder analizarlos, se vuelve clave pues, al final, se le está dando un nuevo sentido a lo que implica *que una máquina aprenda*.

Con este trabajo, además de desarrollar el modelo, se buscó dar una base teórica y técnica de las posibles extensiones del *aprendizaje de máquina*. El autor, espera que se le haya dado un mejor contexto a la también llamada *Inteligencia Artificial*, lo cual, se espera se haya visto, no es más que estadística computacional llevada al límite.

Apéndice A

Distribuciones Conjugadas

Apéndice B

Paquete en R. Desarrollo y Lista de Funciones

Bibliografía

Albert, J.H., y S. Chib. 1993. «Bayesian analysis of binary and polychotomous response data». *Journal of the American Statistical Association:* 669-679.

Banerjee, Sudipto. 2008. *Bayesian Linear Model: Gory Details.* `http://www.biostat.umn.edu/~ph7440/pubh7440/BayesianLinearModelGoryDetails.pdf`. [En Linea; accedido el 10 de Mayo, 2018].

Barber, D. 2012. *Bayesian Reasoning and Machine Learning.* Cambridge University Press.

Bennett, Kristin P, y Olvi L Mangasarian. 1992. «Robust linear programming discrimination of two linearly inseparable sets». *Optimization methods and software* 1 (1): 23-34.

Bergstrom, A. R. 1985. «The estimation of nonparametric functions in a hilbert space». *Econometric Theory* 1 (01): 7-26.

Bernardo, José M, y Adrian FM Smith. 2001. *Bayesian Theory.* John Wiley & Sons.

Bishop, C M. 2006. *Pattern Recognition and Machine Learning.* Springer.

Boor, C De. 1978. *A Practical Guide to Splines.* 346. New York, Springer-Verlag.

Box, George E. P. 1979. *Robustness in the Strategy of Scientific Model Building.* p. 74. May. RL Launer / GN Wilkinson.

Casella, George, y Edward I George. 1992. «Explaining the Gibbs sampler». *The American Statistician* 46 (3): 167-174.

Denison, DGT, BK Mallick y AFM Smith. 1998. «Automatic bayesian curve fitting». *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 60 (2): 333-350.

Devroye, Luc. 1986. *Non-Uniform Eandom Variate Generation.* Volumen 4. Springer-Verlag New York.

Friedman, Jerome H. 1991. «Multivariate adaptive regression splines». *The Annals of Statistics:* 1-67.

Gelfand, A E, y A F M Smith. 1990. «Sampling-Based Approaches to Calculating Marginal Densities». *Journal of the American Statistical Association* 85 (410): 398-409.

Härdle, Wolfgang, Marlene Müller, Stefan Sperlich y Axel Werwatz. 2004. *Nonparametric and Semiparametric Models.* Springer Verlag.

Hastie, T., R. Tibshirani y J. Friedman. 2008. *The Elements of Statistical Learning.* Springer, Series in Statistics.

Hastie, Trevor, y Robert Tibshirani. 1986. «Generalized additive models». *Statistical Science:* 297-310.

James, Gareth, Daniela Witten, Trevor Hastie y Robert Tibshirani. 2013. *An Introduction to Statistical Learning.* Springer.

MacCullagh, P., y J. A. Nelder. 1989. *Generalized Linear Models.* Chapman & Hall, London.

Mangasarian, Olvi L, R Setiono y WH Wolberg. 1990. «Pattern recognition via linear programming: Theory and application to medical diagnosis». *Large-scale numerical optimization:* 22-31.

Mendoza, Manuel, y Pedro Regueiro. 2011. *Estadística Bayesiana.* Instituto Tecnológico de México.

NG, Andrew. 2018. *Machine Learning.* Coursera Online Course, `https://www.coursera.org/learn/machine-learning`. [En Linea; accedido en primavera del 2018].

O'Hara, Robert B, Mikko J Sillanpää y col. 2009. «A review of bayesian variable selection methods: what, how and which». *Bayesian Analysis* 4 (1): 85-117.

Robert, Christian P., y George Casella. 2004. *Monte Carlo Statistical Methods.* Springer.

Ross, S.M. 2009. *Introduction to Probability Models.* Academic Press.

Schoenberg, I J. 1964. «Spline interpolation and the higher derivatives». *Proceedings of the National Academy of Sciences of the United States of America* 51, número 1 (): 24-8.

Stone, Charles J. 1985. «Additive regression and other nonparametric models». *The Annals of Statistics:* 689-705.

Sundberg, Rolf. 2016. *Statistical Modelling by Exponential Families, Lecture Notes.* Stockholm University.

Wahba, G. 1990. *Spline Models for Observational Data.* Society for Industrial & Applied Mathematics.

Wasserman, Larry. 2007. *All of Nonparametric Statistics.* Springer.