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 2019

"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 que transforma de forma no lineal a las covariables para lograr detectar patrones más complejos. La transformación esta basada en polinomios por partes de continuidad arbitraria los cuales se estudian en detalle. Bajo el paradigma de aprendizaje bayesiano, se desarrolla un algoritmo eficiente para la estimación de los parámetros. Posteriormente, este algoritmo se implementa en un paquete para el software estadístico `R`. Usando este paquete, finalmente, se prueba y se valida el modelo, presentando una variedad de resultados que exponen de forma clara e intuitiva las capacidades de el modelo.

Palabras clave: modelos lineales generalizados, probit, modelos aditivos, bayesiano, no lineal, machine learning, splines, polinomios por partes

a mi mamá Irma

Agradecimientos

Creo que la mejor analogía a escribir una tesis es un maratón. Un maratón que me tomó más de dos años de vida, durante los cuales, la vida misma se llevó a mi abuela Teresa, mi padrino Jorge y mi tío abuelo del mismo nombre. Con ellos es con los que más estoy agradecido pues sé que sin sus bendiciones, no habría tenido la resistencia para terminar este trabajo.

Asimismo, quiero agradecer a un sinfín de personas pero principalmente a mi mamá Irma, pues no solo le debo la vida sino todo lo que soy y todo lo que tengo pues esta tesis también es suya. A mi papá Antonio, pues aunque no esté siempre lo he sentido presente. A mi otro papá Fernando, por su guía, apoyo incondicional y todo el amor que siempre me ha dado. A mi abuelo Carlos por inculcarme el gusto por el conocimiento y el valor del trabajo.

A Paulina, que es lo mejor que me ha pasado en la vida y con quién quiero pasar el resto. A Iñigo por su amistad sin medida, compañía e inteligencia fuera de este mundo. A los mejores amigos que alguien pudiera pedir, Pamela, Rodrigo, Brenda, Hector y Jorge, pues siempre me han inspirado a crecer y seguir adelante, así como ser las personas más exitosas que conozco. A Luis, Jime, Eli, Carlos, Mercy, Santiago y Tulio por acompañarme en la carrera y rompernos más de una vez la cabeza en demostraciones oscuras. A Toño, Chris, Edu y Mau por ser los mejores actuarios que conozco. A Jorge, por enseñarme de perseverancia y resiliencia así como por ser uno de mis más viejos amigos. A Pau C. y a Fernanda, por ser grandes amigas y

compañeras de este viaje.

A todos mis alumnos, porque más que un negocio, fueron un medio para consolidar mis conocimientos y seguir aprendiendo día a día. 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 a los profesores E. Barrios, M. Gregorio, J. Alfaro, R. Espinoza, G. Gravinsky y Z. Parada. A Beatriz Rumbos por darme esperanza cuando pensé que todo estaba perdido. A mis profesores de matemáticas y física del Green Hills que sembraron en mi la curiosidad por las matemáticas mientras creyeron en mi, impulsandome a perseguir mis sueños. A los grandes estadísticos que han dedicados sus vidas a estudiar los datos. En particular a T. Hastie, R. Tibshirani y J. Friedman que sin sus contribuciones, no tendría tesis alguna. Y finalmente, al Café Parabien y todo su personal por ser un espacio de trabajo y un hogar para mi los meses de más 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

$\lambda \in \mathbb{N}$: número total de términos del modelo.

$\mathcal{X}^d \subseteq \mathbb{R}^d$: 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 en los reales

Específicos del modelo

$z_i \sim \mathcal{N}(\cdot) \quad \forall i = 1 \ldots, n$: variables latentes del modelo cuya distribución es normal. En su forma vectorial: $\mathbf{z} = (z_1, \ldots, z_n)^t$

$\eta(\mathbf{x})$: función aditiva de predicción

$f_j(x_j) \quad \forall j = 1, \ldots, d$: polinomios por partes

$\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_\lambda)^t$: vector de parámetros por estimar. Si se le añade tilde entonces el contador comienza en uno y el vector no contiene el parámetro independiente, es decir: $\tilde{\boldsymbol{\beta}} = (\beta_1, \ldots, \beta_\lambda)^t$

$\Psi_l(\cdot) \quad \forall l = 1, \ldots, N^*$: funciones bases para la expansión en polinomios por partes de f_j. Ver (2.14) y (2.17). En ocasiones, todas las funciones base se organizan en una matriz $\widetilde{\Phi}$

N^*: número total de funciones base. Ver ecuación (2.22) para su expansión final. Si se usa N sin el asterisco denota de igual forma un número de funciones base arbitrario.

M: tamaño de la base para los polinomios por partes, por lo tanto, 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 tienen un total de $d(J-1)$ nodos acomodados en una matriz de igual tamaño.

Contadores e índices

$i = 1, \ldots, n$: contador, usado para denotar un conjunto de observaciones

$j = 1, \ldots, d$: contador, usado para denotar el conjunto de covariables. Usualmente se hace referencia a la dimensión arbitraria j

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

$l = 1, \ldots, N^*$: contador asociado al número de funciones base total en la expansión de polinomios por partes N^*.

$\hat{\imath} = 1, \ldots, M-1$: contador asociado al número de funciones base para cada subintervalo, M, en las expansione de polinomios truncados.

$\hat{j} = 1, \ldots, J-1$ contador asociado al número de funciones base para cada subintervalo (parámetro M) en las expansione de polinomios truncados

Probabilidad

$F(\cdot)$: Función de distribución arbitraria de la familia exponencial

$\mathcal{N}(\cdot\,|\mu, \sigma^2)$: distribución normal con su correspondiente parametrización de media y varianza. Se utiliza la misma notación para su forma vectorial añadiendo un subíndice para denotar su dimensionalidad: $\mathcal{N}_{\cdot}(\cdot|\boldsymbol{\mu}, \sigma)$ con su correspondiente vector de medias μ y vector de varianza covarianza Σ

$\Phi(\cdot) : \mathbb{R} \to (0, 1)$: la función de distribución acumulada de una distribución normal estándar $\mathcal{N}(\cdot\,|1, 0)$, 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 $\mathcal{N}(\epsilon|\mu, \sigma^2)$.

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

$\theta \in \Theta$ parámetros canónicos de distribuciones exponenciales, con Θ su correspondiente espacio

$\pi(\cdot)$: función de densidad

$\propto$: operador de proporcionalidad

ρ: correlación lineal de Pearson

L: función de perdida

Algoritmo

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

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

k_{thin}: parámetro de adelgazamiento

Otros

$h(\cdot)$: función arbitraria

$h^{(k)}$: (k)-ésima derivada de la función h.

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

I: función indicadora

$(\cdot)_+$: función parte positiva

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

ANOVA :*ANalysis Of VAriance*, modelos de análisis de varianza

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

GLM : *Generalized linear model*, modelo lineal generalizado

MCMC : *Markov Chain Monte Carlo*, cadena de Markov Montecarlo

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 plantea como objetivo: estudiar, explicar e implementar un modelo de clasificación supervisada con base en un modelo *probit* al que se le añade un componente no lineal de bases aditivas. Asimismo, se busca desarrollar un algoritmo asociado de aprendizaje para la estimación de parámetros con base en el paradigma bayesiano de inferencia.[2]

1. *machine learning (ML)*
2. Es común, hacer una distinción entre el aprendizaje estadístico y el aprendizaje de máquina pues, mientras que los modelos son los mismos, difieren en perspectiva. El aprendizaje estadistico presta mayor atención al aspecto inferencial e interpretación, cuando el aprendizaje de máquina coloca mayor énfasis en la implementación computacional y los resultados.

El modelo, buscará hacer inferencia sobre una base de datos y *aprender* sobre los patrones subyacentes que puedan contener para posteriormente, predecir el resultado de variables de respuesta. Este tipo de modelos, han resultado ser de enorme efectividad en ámbitos tan diversos, como lo son la medicina y las finanzas. Bajo esta óptica, se busca que el modelo sea práctico e útil, sin perder de vista el componente teórico subyacente. Por lo tanto, se busca explicar con el mayor detalle, cada componente del modelo para que este no sea tratado como una caja negra computarizada.

Los modelos probit son un tipo de regresión, que busca la clasificación de variables de respuesta y_i binarias (éxito o fracaso, positivo o negativo, etc).[3] Esta predicción, depende de información contenida en las covariables $\mathbf{x}_i$ para cada una de las observaciones $i = 1, \ldots, n$. Sin embargo, la relación entre y_i con $\mathbf{x}_i$ puede depender de estructuras complejas que no son necesariamente lineales; esto lleva a que la predicción de las respuestas con bsae en las covariables sea difícil. Para sobrepasar esto, al modelo se le agrega un componente no lineal en covariables que permite discernir estos patrones. Como se verá en el trabajo, el modelo induce fronteras no lineales de clasificación en el espacio donde $\mathbf{x}_i$ tome valores. En la figura 1.1, se tiene un ejemplo gráfico de tipo de clasificación que lleva a cabo el modelo. 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 proceso de aprendizaje busca *entrenar*, bajo el paradigma bayesiano, a una función η que logre separar este espacio de la mejor forma posible. Esta separación, induce una clasificación binaria (0 y 1 correspondiendo a rojo y azul respectivamen-

3. Es usual en la literatura, hablar de *clasificadores* cuando las respuestas son categorías (codificadas en variables discretas) y *regresiones* cuando las variables de respuestas son continuas.

te) a través de la función de distribución normal Φ. Con un modelo cuya frontera fuera lineal en covariables, llevar a cabo esta clasificación sería imposible.

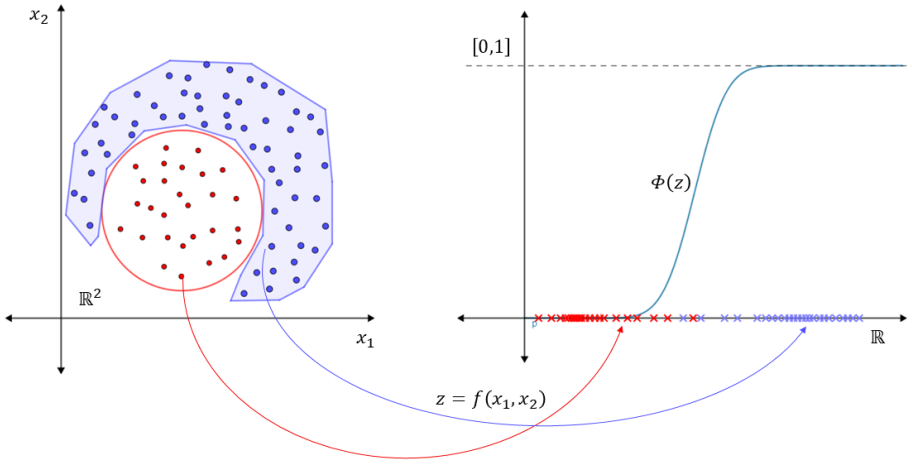


Figura 1.1: Diagrama explicativo de un modelo de clasificación probit no lineal

Para llevar a cabo la construcción del modelo, se comienza con una discusión teórica en el capítulo 2. Primeramente se estudian los modelos lineales generalizados (GLM), específicamente los modelos probit. Los GLM como su nombre lo indica, generalizan las regresiones tradicionales donde la respuesta y_i es escalar ($y_i \in \mathbb{R}$) a regresiones donde la respuesta puede ser discreta o restringida a cierto dominio (McCullagh y Nelder 1989). No obstante, los GLM siguen siendo lineales en las covariables pero se pueden flexibilizar usando diferentes ideas. Entre ellas, los modelos aditivos generalizado (GAM) presentadas en Hastie y Tibshirani (1986). En estos modelos, la flexibilización se logra transformando a las covariables $\mathbf{x}_i$, mediante la función de

predicción η, usando métodos no paramétricos con base en suavizadores. Para este trabajo, se toman esas ideas y se combinan con las de Denison, Mallick y Smith (1998) en las que se opta por darle una forma funcional concreta a η, correspondiente a una expansión de bases funcionales, particularmente, en polinomios por partes de continuidad y grado arbitrarios. La expansión resultante, tiene la peculiaridad que conectan muchas disciplinas y ramas de las matemáticas que han sido de mucha utilidad no sólo en el campo de la estadística. A lo largo del capítulo, se verá que con principios presentados, se abren las posibilidades en cuanto a modelos y datos sobre los que se pueden hacer regresiones o clasificaciones.

Desarrollado una vez el modelo, el capítulo 3 se concentra en su implementación computacional bajo el paradigma bayesiano de aprendizaje. Por lo tanto, se hace una breve introducción a la escuela bayesiana de la estadística, en particular al apredizaje bayesiano en un contexto de regresión. Este paradigma, responde a que, bajo las ideas de Albert y Chib (1993), el modelo se puede plantear de tal forma que se induce un algoritmo con base en el sampleo de Gibbs. La implementación final, se realiza en el paquete computacional para el lenguaje abierto de programación estadística `R`.[4]

En el capítulo 4, el modelo se prueba y se valida contra una serie de bases de datos. Primeramente, se hace una breve discusión sobre como evaluar la efectividad y precisión de un modelo como el presentado en este trabajo. Posteriormente, se ejecuta el algoritmo de aprendizaje contra cinco bases de datos simulados con dos covaria-

4. El desarrollo y explicación del paquete de cómputo se detalla en el Apéndice C. El paquete se puede descargar libremente de: `https://github.com/PaoloLuciano/bpwpm`

bles ($\mathbf{x}_i \in \mathbb{R}^2$). Estas pruebas preliminares, sirven para demostrar las capacidades predictivas del modelo y sobre todo, para hacer más concretas las matemáticas subyacentes, además de poder visualizar las diferentes fronteras flexibles obtenidas por el modelo. Asimismo, en este capítulo se discute la convergencia de las cadenas obtenidas por el muestreador de Gibbs. Para cerrar el capítulo, se replica un escenario real de análisis y modelado usando una base de datos médicos reales de cáncer.

Finalmente, se cierra la discusión en el capítulo 5 donde se revisan consideraciones finales y limitantes del modelo, sin embargo, se abre una discusión a posibles extensiones para mejorarlo. Posteriormente, se da un rápido vistazo a modelos relativamente más modernos los cuales han sido capaces de proezas computacionales que se creían imposibles hace algunas décadas. No obstante, se verá que muchos de estos modelos más complejos, son generalizaciones de modelos clásicos y extensiones análogas al trabajo presentado.

Capítulo 2

Modelo en su forma matemática

Como base fundamental de este trabajo, a continuación y de forma preliminar, se presenta una visión general modelo, mientras que el resto del capítulo se enfocará en profundizar en cada parta que lo compone.[1] Al modelo, se le titula *bayesian piecewise polynomial model (bpwpm)* por sus siglas en inglés. Se trata de seguir la notación usada en los libros Hastie, Tibshirani y Friedman (2008) y James y col. (2013). Asimismo, al comienzo de este trabajo se presenta un glosario de los símbolos y signos usados.

Se supone la siguiente estructura: $\{(y_i, \mathbf{x}_i)\}_{i=1}^{n}$ es el conjunto de datos observados independientes con n el tamaño de la muestra donde, $y_i \in \{0, 1\}$ son las variables

1. La versión completa del modelo se presenta en la sección 3.3.1 de la página 61.

de respuesta binarias, $\mathbf{x}_i \in \mathcal{X}^d \subseteq \mathbb{R}^d$ las covariables o regresores[2] y $d \in \mathbb{N}$ la dimensionalidad de las covariables.[3] Estos datos se organizan y se representan en una tabla (o matriz) como la presentada en la tabla 2.1. En ella, cada fila $i = 1, \ldots, n$ representa una observación. La primer columna corresponde al vector de respuestas y las columnas subsecuentes $j = 1, \ldots, d$ representan una covariable. Es útil pensar en estas columnas como d *dimensiones* que contienen información que induce la clasificación binaria.

$$\left[\begin{array}{c|c} y_1 & \mathbf{x}_1 \\ \vdots & \vdots \\ y_n & \mathbf{x}_n \end{array}\right] \quad = \quad \left[\begin{array}{c|ccc} y_1 & x_{1,1} & \ldots & x_{1,d} \\ \vdots & \vdots & & \vdots \\ y_n & x_{n,1} & \ldots & x_{n,d} \end{array}\right]$$

Tabla 2.1: Estructura de los datos

Asimismo, se define el espacio de covariables $\mathcal{X}^d$ como el producto cartesiano de los rangos de cada covariable j. Esta definición, está relacionada con los polinomios por partes f_j que se estudian en la sección: 2.3.

$$\begin{aligned} \mathcal{X}^d &= \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_d \\ &= [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_d, b_d] \subseteq \mathbb{R}^d \\ \text{con} \quad a_j &= \text{mín}\left\{x_{1,j}, \ldots, x_{n,j}\right\} \quad \forall j = 1, \ldots, d \\ b_j &= \text{máx}\left\{x_{1,j}, \ldots, x_{n,j}\right\} \quad \forall j = 1, \ldots, d. \end{aligned}$$

2. Se utiliza la convención de usar negritas para distinguir vectores $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,n})^t$

3. En el lenguaje de aprendizaje de máquina, es usual hablar de *outputs* e *inputs* para referirse a y_i y $\mathbf{x}_i$ respectivamente (Alpaydin 2014).

Definición 2.1. El modelo bpwpm (preliminar), $\forall i = 1, \ldots, n$:

$$y_i = \begin{cases} 1 & \iff \quad z_i > 0 \\ 0 & \iff \quad z_i \leq 0 \end{cases} \tag{2.1}$$

$$z_i \,|\, \mathbf{x}_i \sim \mathcal{N}(z_i \,|\, \eta(\mathbf{x}_i), 1) \tag{2.2}$$

$$\eta(\mathbf{x}_i) = f_0 + f_1(x_{i,1}) + f_2(x_{i,2}) + \ldots + f_d(x_{i,d}) \tag{2.3}$$

$$f_j(x_{i,j}) = \sum_{l=1}^{N^*} \beta_{j,l} \Psi_l(x_{i,j}, \mathcal{P}_j) \quad \forall j = 1, \ldots, d \tag{2.4}$$

Las expresiones (2.1) y (2.2) introducen n variables latentes z_i independientes entre si con distribución normal. Estas variables se relacionan de forma unívoca con las respuestas y_i formando una clase de equivalencia entre la probabilidad de dos eventos, permitiendo que se asocie el soporte binario de y_i con el soporte real de z_i. Es decir, (2.1) y (2.2) implican:

$$P(y_i = 1 | \mathbf{x}_i) = P(z_i > 0 \,|\, \mathbf{x}_i) = \Phi(\eta(\mathbf{x}_i)). \tag{2.5}$$

Esta definición, Albert y Chib (1993), es equivalente a la definición de un modelo probit pues la función liga resulta en la función de acumulación normal estándar $\Phi : \mathbb{R} \to (0, 1)$ (Sección 2.1). La identidad anterior (2.5) es inducida por la demostración de equivalencia entre definiciones que se detalla en el Teorema 2.3. Una de las razones para adoptar esta perspectiva es que Albert y Chib desarrollaron un método numérico vía simulación, bajo el paradigma bayesiano, para el cómputo exacto de

las distribuciones posteriores del vector de parámetros $\boldsymbol{\beta}$ el cual resultaba atractivo para los objetivos del trabajo.

Posteriormente, la ecuación (2.3) especifica la media de las variables latentes z_i, es decir, se le da forma funcional a $\mathbb{E}[z_i|\mathbf{x}_i] = \eta(\mathbf{x}_i)$. A esta función $\eta : \mathbb{R}^d \to \mathbb{R}$ se le conoce como función de predicción. La idea es suponer que la relación entre los componentes de las covariables $j = 1, \ldots, d$, es modelable como la suma de funciones (usualmente suaves) f_j más un término independiente f_0. Esta definición corresponde a losmModelos aditivos generalizados (GAM), una serie de modelos agrupados en Hastie y Tibshirani (1986). En la sección 2.2 se presenta una introducción a ellos.

Finalmente, la expresión (2.4) define a las funciones $f_j \quad \forall j$ en la parte más profunda del modelo. Estas funciones $f_j : \mathcal{X}_j = [a_j, b_j] \to \mathbb{R}$ realizan una transformación no lineal de las covariables $x_{i,j}$. Este proceso se lleva a cabo mediante una expansión en bases funcionales revisada en la sección 2.3. El objetivo de esta expansión es expresar cada f_j de una forma flexible, a través de la suma ponderada de funciones bases $\Psi_{j,l}(x_{i,j}, \mathcal{P}_j)$ y parámetros desconocidos $\beta_{j,l}$ los cuales se deben de estimar. Asimismo, las funciones bases dependen de tres componentes: las covariables $x_{i,j}$, una partición $\mathcal{P}_j$ para cada dimensión[4] y el número total de funciones base $N^* \in \mathbb{N}$. Sus formas funcionales, no son más que truncamientos de orden mayor en las covariables, por ejemplo: $(x_{i,j} - a)_+^b$ con a, b constantes definidas por N^* y $(\cdot)_+$ la función parte positiva, dando lugar a una expansión en polinomios por partes; particularmente, la presentada en Denison, Mallick y Smith (1998). Por el momento, se deja a las

4. Definida sobre el intervalo $\mathcal{X}_j = [a_j, b_j] \quad \forall j = 1, \ldots, d$

funciones bases $\Psi_{j,l}$ no especificadas por completo pues se decide presentarlas de forma constructiva en la sección 2.3.2, derivando en su forma funcional final en las ecuaciones (2.17) y (2.18).

Para esclarecer un poco más el trabajo, en la figura 2.1 se presenta un diagrama del modelo y sus componentes. De izquierda a derecha y para toda $i = 1, \ldots, n$: se busca transformar de forma no lineal a cada una de las covariables observadas $x_{i,j} \quad \forall j = 1, \ldots, d$ a través polinomios por partes condensados en las funciones f_j. Estas transformaciones dependen de parámetros desconocidos $\beta_{j,l}$ con $l = 1, \ldots, N^*$ y la partición de cada dimensión P_j. Una vez se tienen las covariables transformados, se suman las funciones f_j con un intercepto local f_0 para obtener una función de predicción η. Esta función actúa como la media de la variable latente z_i que relaciona a la respuesta y_i con $\mathbf{x}_i$. La relación se realiza a través de la función Φ para lograr la clasificación binaria en y_i.

Las aparentemente complejas interacciones entre todos los componentes del modelo no son más que respuestas estructurales a un proceso de *síntesis* de la información. El modelo está buscando un patrón en las covariables $\mathbf{x}_i$ para la correcta clasificación de su respuesta binaria asociada y_i. Este proceso, se lleva a cabo mediante tres transformaciones $f_j(x_{i,j})\ \forall j$, $\eta(\mathbf{x}_i)$ y finalmente $\Phi(\eta(\mathbf{x}_i))$ las cuales cumplen el propósito de ir colapsando dimensiones. Se espera que este proceso, logre separar de forma flexible el espacio d-dimensional $\mathcal{X}^d$ a regiones más identificables (para la clasificación) que las regiones originales; donde finalmente, se le asigne una probabilidad a cada región de clasificación mediante Φ. El Capítulo 4 cuenta con visualizaciones

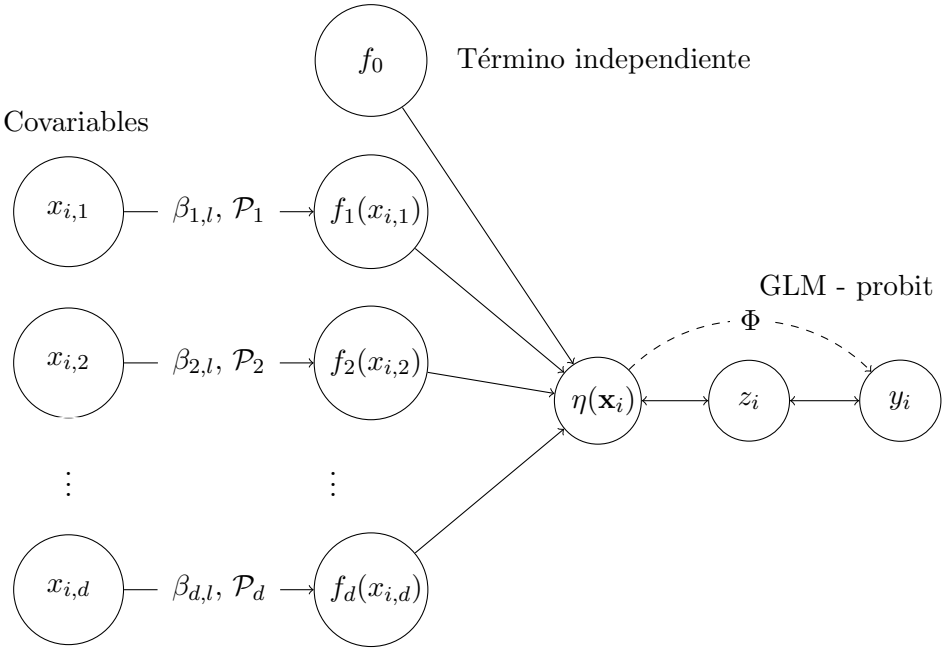


Figura 2.1: Diagrama del modelo

que esperan volver estos conceptos teóricos en algo más concreto.

Antes de continuar, vale la pena mencionar el precepto de Box (1976):

> *All models are wrong but some are useful*

En la perspectiva del autor, no se está tratando de construir un modelo que replique el proceso generador de los datos. Más bien, se está tratando de construir una útil abstracción de la realidad a través de un modelo matemático. Escoger cualquier

enfoque de modelado, es un proceso reduccionista y por ende, falible. Sin embargo, no significa que no se puedan discernir patrones en los datos y aprender de ellos.

2.1. Modelos lineales generalizados (GLM)

Los modelos lineales generalizados (GLM), McCullagh y Nelder (1989), surgen como una generalización del modelo de regresión lineal:

$$\begin{aligned} y_i|x_i &\sim \mathcal{N}(\mu(\mathbf{x}_i), \sigma^2) \quad \forall i = 1, \ldots, n \\ \mu(\mathbf{x}_i) &= \beta_0 + \tilde{\boldsymbol{\beta}}^t \mathbf{x}_i, \end{aligned}$$

donde $y_i \in \mathbb{R}$, $\tilde{\boldsymbol{\beta}} \in \mathbb{R}^d$ es un vector de parámetros y $\mu(\mathbf{x}_i) = \mathbb{E}[y_i|\mathbf{x}_i]$.[5] Las regresiones lineales, están acotadas a datos donde la variable de respuesta y_i tenga soporte real. En consecuencia se desarrollan los GLM, que busca flexibilizar este soporte a una mayor cantidad de respuestas. Esta modificación vuelve al modelo más complejo y deriva en diversas técnicas para la estimación de $\boldsymbol{\beta}$. Asimismo, la generalización del modelo lleva a que la intrepretación de los parámetros no sea trivial.[6]

5. Se usa $\tilde{\boldsymbol{\beta}}$ para distinguir al vector de dimensionalidad d y a $\boldsymbol{\beta}$ para distinguir al vector que contiene el término independiente, es decir, $\boldsymbol{\beta} \in \mathbb{R}^{d+1}$

6. Por ejemplo, en un modelo logit que busca la predicción de variables binarias, se logra expresar el logaritmo de la proporción de probabilidades (*Log-Odds-Ratio*) como una combinación lineal de las covariables. $\ln(p_1/p_0) = \boldsymbol{\beta}^t x$, donde p_k con $k = \{0, 1\}$, es la probabilidad de que la respuesta y sea 0 o 1 respectivamente.

Definición 2.2. El modelo lineal generalizado, Sundberg (2016):

$$\begin{aligned} y &\sim F(\mu(\mathbf{x})) \\ \eta &= \beta_0 + \boldsymbol{\beta}^t \mathbf{x} \\ g(\mu) &= \eta \end{aligned} \tag{2.6}$$

cuenta con los siguientes tres elementos:

F: distribución de la familia exponencial que describe el dominio de las respuestas y, cuya media $\mu(\cdot)$ es dependiente de las covariables.[7] Por ejemplo: Bernoulli si y es binaria, Poisson si $y \in \mathbb{Z}^+$ o una distribución Gamma si $y \in \mathbb{R}^+$

η: predictor lineal que explique (linealmente) la variabilidad sistemática de los datos.

g: función liga que une la media μ de la distribución con el predictor lineal,[8] es decir: $g(\mu(x)) = g(\mathbb{E}[y|x]) = \boldsymbol{\beta}^t x$. g puede ser cualquier función monótona que idealmente mapee de forma suave y biyectiva el dominio de la media μ con el rango del predictor lineal η (Härdle y col. 2004).

7. Al trabajar con distribuciones de la familia exponencial es usual parametrizar la distribución no con la media μ sino con el parámetro canónico θ.

8. Si la función g es tal que $\eta \equiv \theta$ entonces se dice que g es la función liga canónica.

2.1.1. El modelo probit

Por lo pronto, la discusión se centrará en la distribución Bernoulli pues resulta de forma natural dado que se busca construir un clasificador supervisado donde las respuestas observadas sean binarias, i.e. $y_i \in \{0, 1\}\ \forall i = 1, \ldots, n$. Esto es:

$$y_i \sim \text{Be}(y_i \,|\, p_i). \tag{2.7}$$

La distribución Bernoulli (2.7) tiene una estructura sencilla que puede ser resumida en las siguientes expresiones $\forall i = 1, \ldots, n$:

$$\begin{aligned} f(y_i|p_i) &= p_i^{y_i}(1-p_i)^{1-y_i} \\ \text{donde}\ \ y_i &\in \{0, 1\}, \\ \mathbb{E}[y_i] &= \mu_i = P(y_i = 1) = p_i \\ \mathbb{V}[y_i] &= p_i(1-p_i). \end{aligned} \tag{2.8}$$

En (2.8) se observa la función de masa de probabilidad Bernoulli en su forma tradicional que puede ser reexpresada para que cumpla la definición de la familia exponencial.[9] Dado el soporte y la definición de la distribución Bernoulli, la media de la distribución $\mu = p$ coincide con la probabilidad de que la variable aleatoria tome el valor de uno. Asimismo, la varianza queda especificada por el mismo parámetro p.

9. Una distribución (de un solo parámetro) se dice que pertenece a la familia exponencial si se puede expresar de la forma: $f(y;\theta) = h(y)\exp\{y \cdot \theta - A(\theta)\}$ con $h(y)$, $A(\theta)$ funciones conocidas y θ el parámetro canónico, en el caso Bernoulli: $\theta(p) = \ln p/(1-p)$.

El que la media conocida con la probabilidad de éxito en una distribución Bernoulli es de gran utilidad en un contexto de clasiicación por varias razones. Primero, al modelar la media $\mu = p$, se está caracterizando por completo la distribución y la predicción de la variable y. Segundo, se restringen las posibles funciones liga a las funciones que mapeean de forma biyectiva $\mathbb{R}$, el dominio del predictor lineal η, al el rango de la media, el intervalo $(0, 1)$. Dadas las propiedades buscadas, es usual usar como función liga a las inversas de funciones *sigmoidales*. Las funciones sigmoidales, son funciones $s : \mathbb{R} \to (0, 1)$ estrictamente monótonas y por ende, biyectivas. Algunos ejemplos son la ya mencionada logit, la función probit que concierne a este trabajo o la curva de Gompertz. Estas funciones cumplen un papel de activación, es decir, una vez que el predictor lineal rebase cierto umbral, crecen rápidamente y toman valores más cercanos a uno, *activando* así la probabilidad de que y sea uno.[10]

En particular, en este trabajo se escoge como función liga a la función probit dándole nombre al modelo, en consecuencia al trabajo de Albert y Chib (1993). La función probit la inversa de la función de acumulación normal estándar $\Phi(\cdot)$, i.e. $g(\mu) = g(p) = \text{probit}(p) = \Phi^{-1}(p)$. Dado que la notación puede ser confusa, en la figura 2.2 se presenta una representación gráfica de la función liga para un modelo probit.[11]

Juntando todos los componentes, se está en posibilidades de detallar el modelo

10. En un contexto de aprendizaje de máquina, se les conoce como funciones de activación a las inversas de la funciones liga g^{-1} que no necesariamente tienen que ser biyectivas (Bishop 2006). Por ejemplo, en redes neuronales es común utilizar la función $ReLu(x) := \text{máx}\{0, x\}$ la cual no es suave (Sanderson 2017).

11. Para no caer en redundancia de notación se tiene a partir de ahora: $s(x) = g^{-1}(x) = \Phi(x)$ la función de acumulación normal estándar. Asimismo, se deja de usar μ para referirse a la media y se utiliza únicamente p.

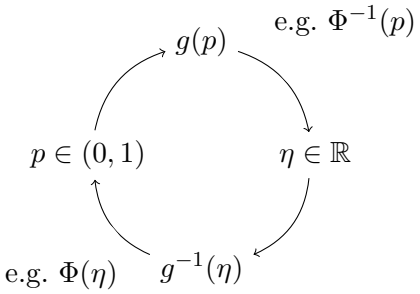


Figura 2.2: Esquema de función liga g para un modelo probit

probit en su forma más rigurosa. Rescatando la notación de un GLM (2.6) con sus respectivas covariables $\mathbf{x}_i$:

$$y_i|\mathbf{x}_i \sim \text{Be}(y_i|p_i) \qquad \forall i = 1, \ldots, n \tag{2.9}$$

$$\eta_i = \eta(\mathbf{x}_i) \tag{2.10}$$

$$p_i = \Phi(\eta_i) = \Phi(\eta(\mathbf{x}_i)) \tag{2.11}$$

Equivalencia en las definiciones del modelo

El lector notará, que la especificación del modelo probit en las ecuaciones anteriores no corresponde a la definición mostrada al principio del capítulo. No obstante, las definiciones son equivalentes y se prueba a continuación.

Teorema 2.3. *Un modelo probit especificado en* (2.9) *y* (2.11)*, es equivalente a un modelo de variable latente como el presentado en* (2.2) *y* (2.1)*.*

Demostración. Dado un modelo probit se tiene, sin perdida de generalidad $\forall i = 1, \ldots, n$:

$$\begin{aligned}
\mathbb{E}[y_i \,|\, \mathbf{x}_i] &= p_i \\
&= P(y_i = 1 \,|\, \mathbf{x}_i) \qquad \text{por (2.9)} \\
&= \Phi(\eta(\mathbf{x}_i)) \qquad \text{por (2.11)}
\end{aligned}$$

Lo cual, es equivalente a introducir n variables aleatorias $\tilde{z}_i \sim \mathcal{N}(\tilde{z}_i|0,1)$ tales que:

$$\begin{aligned}
\Phi(\eta(\mathbf{x}_i)) &= P(\tilde{z}_i \leq \eta(\mathbf{x}_i) \,|\, \mathbf{x}_i) \quad \text{por definición de la función de acumulación} \\
&= P(\tilde{z}_i > -\eta(\mathbf{x}_i) \,|\, \mathbf{x}_i) \quad \text{por simetría de la distribución normal} \\
&= P\left(\frac{\tilde{z}_i + \eta(\mathbf{x}_i)}{1} > 0 \,\middle|\, \mathbf{x}_i\right) \\
&= P(z_i > 0 \,|\, \mathbf{x}_i).
\end{aligned}$$

Donde $z_i = \tilde{z}_i + \eta(\mathbf{x}_i)$ es una transformación biyectiva de $\tilde{z}_i$ tal que:

$$z_i|\mathbf{x}_i \sim \mathcal{N}(z_i|\eta(\mathbf{x}_i), 1),$$

lo cual es idéntico a la expresión (2.2). Asimismo, al tener la igualdad

$$P(y_i = 1|\mathbf{x}_i) = P(z_i > 0|\mathbf{x}_i)$$

y por ende su probabilidad complementaria $P(y_i = 0|\mathbf{x}_i) = P(z_i \leq 0|\mathbf{x}_i)$, se define una clase de equivalencia entre probabilidades. Es decir, se puede definir y_i en términos de z_i y viceversa, dando lugar a la definición (2.1).

El argumento es casi idéntico si la demostración se comienza asumiendo la definición de un modelo de variable latente como en (2.2) y (2.1) y se construye hasta llegar a un GLM como en (2.9) y (2.11). Sin embargo, se tiene la peculiaridad de que la varianza debe de ser igual a uno para ser que la correspondencia entre definiciones sea exacta.[12] Q.E.D.

La ecuación (2.10) realmente no influye en la prueba pues esta puede tener la forma funcional que se requiera para la aplicación especifica, ya sea lineal $\eta_i = \beta_0 + \boldsymbol{\beta}^t\mathbf{x}_i$ como en (2.6) o algo diferente como se opta en este trabajo.

Liga entre la variable latente z y η

Para entender como se conectan las n variables latentes z_i con sus respectivos predictores lineales $\eta(\mathbf{x}_i)$, se necesita profundizar un poco más en el objetivo del modelo. Recapitulando, mediante la función liga Φ se une la media p_i, la probabilidad de que la respuesta y_i sea uno con los datos $\mathbf{x}_i$. Esto se logra, a través de una variable latente z_i definida con distribución normal cuya media $\eta(\mathbf{x}_i)$, la función de predicción,

12. Comenzar con $z_i|\mathbf{x}_i \sim \mathcal{N}(z_i|\eta(\mathbf{x}_i), \sigma^2)$ con $\sigma^2 \neq 1$ deriva en que $p_i = \Phi(\eta(\mathbf{x}_i)/\sigma)$ lo cual es diferente a lo que se tiene en (2.11).

es una transformación de las covariables $\mathbf{x}_i$. Es decir,

$$P(z_i > 0|\mathbf{x}_i) = P(y_i = 1|\mathbf{x}_i) = p_i(\mathbf{x}_i) = \mathbb{E}[y_i|\mathbf{x}_i] = g^{-1}(\eta(\mathbf{x}_i)) = \Phi(\eta(\mathbf{x}_i)). \quad (2.12)$$

Este enfoque funciona, además de por el componente algorítmico, por la siguiente idea. Si se quiere crear una regla de decisión que clasifique observaciones en categorías binarias con base en cierta información, parecería intuitivo condensar esa información de forma que proporcione suficiente evidencia para inducir la clasificación. Traduciendo en términos matemáticos, la información $\mathbf{x}_i$ se condensa en la función $\eta(\mathbf{x}_i)$ la cual induce la clasificación de y_i a través de la ecuación (2.12). Por ejemplo, si se tiene una $\eta(\mathbf{x}_i) >> 0$ para alguna observación i, implicaría que $P(z_i > 0|\mathbf{x}_i)$ es cercano a uno (por el dominio de Φ) y por lo tanto, es muy probable que y_i sea un uno. El argumento es idéntico para la probabilidad complementaria.

Al final, como se menciona anteriormente, el modelo está resumiendo información al ir colapsando dimensiones. El siguiente paso en el modelo consiste en detallar la transformación que debe realizar el predictor lineal $\eta(\mathbf{x}_i)$. Tradicionalmente como se mencionó en (2.6), esta transformación era lineal tanto en parámetros como en covariables, dando lugar a fronteras de decisión lineales. Sin embargo, el siguiente paso lógico es modificar estos modelos para que las fronteras puedan ser más flexibles, rompiendo la linealidad en las covariables para lograr encontrar patrones más complejos.

2.2. La función de predicción η

2.2.1. Una breve introducción a los GAM

Como se detalla 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 las covariables. En particular, Hastie y Tibshirani se agrupan una clase de modelos a los que se les da el nombre de modelos aditivos generalizados (GAM). Estos modelos logran identificar relaciones no lineales utilizando, usualmente, métodos no paraméticos de suavizamiento en los datos adoptando así, un enfoque de *dejar que los datos hablen por si mismos.*[13]

Definición 2.4. Un GAM, tiene la forma $\forall i = 1, \ldots, n$:

$$\mathbb{E}[y_i|\mathbf{x}_i] = g^{-1}\left[f_0 + f_1(x_{i,1}) + \ldots + f_d(x_{i,d})\right], \tag{2.13}$$

con g^{-1} la inversa de la función liga definida en (2.1) y el predictor lineal $\eta(\mathbf{x}_i) = f_0 + f_1(x_{i,1}) + \ldots + f_d(x_{i,d})$.

La idea fundamental de los GAM, es asumir que los efectos en las covariables se pueden modelar como una suma de funciones por componentes, es decir, cada covariable $x_j \quad \forall j = 1, \ldots, d$ está siendo transformada de forma no lineal e independiente por una función asociada $f_j \quad \forall j$. De esta forma, se retiene algo de la interpretabili-

13. Página 1 de Hastie y Tibshirani (1990)

dad del modelo lineal. La funciones f_j que ahora componen el predictor lineal η se busca que sean tan flexibles como sea necesario, permitiendo que el estadista pueda hacer menos suposiciones rígidas sobre los datos. Estas funciones f_j son suaves y no especificadas (*no paramétricas*), es decir, no tienen una forma funcional concreta y representable algebraicamente. Sin embargo, es justamente ahí donde recae la fuerza de los GAM: al dejar a las funciones f_j ser no especificadas, se permite que estas capturen los efectos necesarios en los datos para hacer el mejor ajuste posible, a este proceso se le llama suavizamiento.

Un suavizador, se puede definir de forma general, como una herramienta que resume la tendencia de la respuesta y como función de las covariables $\mathbf{x}$ y produce un estimador f que es menos variable (ruidoso) que la respuesta en si. Como se mencionó con anterioridad, estos suavizadores son de naturaleza no paramétrica pues no se asume una dependencia rígida de y en $\mathbf{x}$.[14] Como ejemplos prácticos de métodos no paramétricos, se encuentran los ajustes de medias móviles y el suavizamiento LOESS[15] (Cleveland y Devlin 1988).

En un GAM la estimación de las funciones f_j, se lleva a cabo por el algoritmo de ajuste hacia atrás (*backfitting algorithm*), Hastie y Tibshirani (1986). Este procedimiento, busca dar estimadores de cada f_j de forma iterativa por componentes, utilizando como regresores los residuales parciales. Por ejemplo, sea $d = 2$ y $g^{-1}(w) = w$

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

15. El suavizamiento LOESS, *locally estimated scatterplot smoothing*, es un tipo de regresión local que ajusta modelos más simples a subconjuntos de los datos para construir una función global que describa de forma no lineal la variabilidad intriseca que se presenta.

la función identidad, quedando así el modelo:

$$\mathbb{E}[y_i|\mathbf{x}_i] = f_0 + f_1(x_1) + f_2(x_2).$$

Dados estimadores preliminares $\hat{f}_0$ y $\hat{f}_1$ de las respectivas funciones f_0 y f_1, se definen los residuales parciales: $\mathbb{E}[y_i|\mathbf{x}_i] - (\hat{f}_0 + \hat{f}_1(x_1))$ sobre los cuales se busca suavizar f_2. Este proceso resulta en una mejor estimación de la función f_2, con la cual, se puede mejorar el estimador de f_1. Ese proceso se lleva a cabo de forma iterativa, hasta que el cambio en las funciones funciones f_j sea menor que un umbral especificado.[16] Este algoritmo, se puede extender para d y g arbitrarias y es bastante flexible a modificaciones. En un GAM, las curvas resultantes de las funciones f_j son suaves y lejos de ser lineales. Asimismo, sus formas, pueden ayudar a entender el fenómeno subyacente.

Los GAM en el contexto de este trabajo

Sin dudar la elegancia y practicalidad de los métodos no paramétricos, para este trabajo, se opta modificar el enfoque original de los GAM y darles una forma rígida a las funciones f_j, regresando a los dominios de la estadística paramétrica. Esta decisión, pues se busca profundizar en los polinomios por partes estudiados en la siguiente sección 2.3 que componen a las funciones f_j. Aunque pareciera una desviación considerable del trabajo original de Hastie y Tibshirani, en realidad en el Apéndice A se detalla como los polinomios por partes son el resultado de plantear

16. La demostración de convergencia de un GAM se encuentra en Stone (1985)

la idea de suavizamiento como un problema de optimización. Asimismo, los GAM son tan flexibles en su definición (y concepto) que es usual restringir las funciones f_j con formas funcionales concretas.[17]

Bajo esta óptica, para este trabajo se retienen dos de las ideas fundamentales de los GAM: aditividad y las transformaciones por componentes de las covariables. Es decir, la definición de un GAM (2.13) sustituye el predictor lineal tradicional de los GLM (2.6), $\eta(\mathbf{x}_i) = \beta_0 + \tilde{\boldsymbol{\beta}}^t \mathbf{x}_i$, con una suma de funciones $\sum_j^d f_j(x_j)$ más un intercepto constante f_0 que juega el papel de β_0, dando lugar a la ecuación (2.3) definida a inicios de este capítulo:

$$\eta(\mathbf{x}_i) = f_0 + f_1(x_{i,1}) + f_2(x_{i,2}) + \ldots + f_d(x_{i,d}). \tag{2.3}$$

Se hace notar, que a diferencia de los modelos lineales donde se tiene a los parámetros $\boldsymbol{\beta}$ incluidos en la expansión de η, en los GAM los parámetros se incluyen dentro de cada una de las f_j pues, los efectos de cada covariable son resumidos dentro de las mismas transformaciones. Aunque se pueden agregar parámetros que ponderen cada f_j, sobre-parametrizar puede llevar a la incorrecta especificación del modelo y caer en problemas de identificabilidad de los parámetros.

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 modelo. Es decir, cada $f_j(x_{i,j})$ es el efecto que tiene la covariable j, para una observación i, en la clasificación. Por lo tanto y heredado de la ecuación

17. Capitulo 9.1 y Ejemplo 5.2.2 de Hastie, Tibshirani y Friedman (2008)

(2.12) si f_j es más positiva para esta observación i, se tiene mayor evidencia (en el componente j) de que la respuesta binaria asociada y_i sea uno. En la peculiaridad de que $d = 2$, se podrá visualizar, no solo las funciones f_j de manera independiente, sino toda $\eta(\mathbf{x}_i)$ en $\mathbb{R}^3$ como una serie de picos y valles donde será positiva en caso de que y_i sea clasificada como uno y negativa en caso de que sea cero. La imagen de la página 96.

La inclusión de un término independiente f_0 es importante en los GAM pues es uno de los resultados de la derivación mencionada en el apéndice A. Asimismo, se debe considerar el caso tal que $f_j(x_j) = 0 \quad \forall j$ de donde se necesita un término independiente f_0. Para este trabajo al término f_0 se le da el mismo tratamiento que el de un parámetro independiente convencional, por lo tanto, se estima usando el mismo procedimiento que todos los demás parámetros. Este hecho se esclarecerá en las secciones subsecuentes. Las imágenes 4.2c y 4.2c de la página 75, son solo algunos ejemplos de las posibles formas finales que pueden adoptan las funciones f_j. Para esta realización particular del modelo, están compuestas por segmentos de recta que no son suaves.

2.3. Funciones f_j

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 media η 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 el tipo de polinomios por partes presentados en Denison, Mallick y Smith (1998). Toda la siguiente sección se concentra en darle forma funcionales a las sub-funciones Ψ para definir por completo f_j y por ende η.

2.3.1. Expansión en bases funcionales

Saliendo por un momento del domino de la estadística, se definen las expansiones en bases funcionales. Sin entrar mucho en los detalles técnicos, dado un espacio funcional[18] se puede representar cualquiera de sus elementos, en este caso una función arbitraria h, como la combinación lineal de los elementos de la base Ψ y constantes β. 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:

$$h(\mathbf{x}) = \sum_{l=1}^{N} \beta_l \Psi_l(\mathbf{x}) = \tilde{\beta} b f^t \Psi(\mathbf{x}). \tag{2.14}$$

Bajo esta definición, $\mathbf{\Psi}(\mathbf{x}) = (\Psi_1(\mathbf{x}), \ldots, \Psi_N(\mathbf{x}))^t$ es un vector cuyos elementos $\Psi_l(\mathbf{x})$ son llamados funciones base y tienen el mismo mapeado que h. De la misma

18. Espacio vectorial cuyos elementos son funciones.

forma $\tilde{\boldsymbol{\beta}} = (\beta_1, \ldots, \beta_N)^t$ es un vector de coeficientes constantes. Finamente, $N \in \mathbb{N}$ es un entero mayor o igual a la dimensión del espacio funcional que se maneja.[19]

En un contexto estadístico de regresión, se definen los modelos lineales de bases funcionales,[20] capitulo 3 de Bishop (2006), como:

$$h(\mathbf{x}) = \beta_0 + \sum_{l=1}^{N} \beta_l \Psi_l(\mathbf{x}) = \beta_0 + \tilde{\boldsymbol{\beta}}^t \boldsymbol{\Psi}(\mathbf{x}), \tag{2.15}$$

lo cual es idéntico a (2.14) con la adición del término independiente β_0. Bajo este contexto, se busca representar una transformación g de la media condicional de la respuesta por una función dependiente de los datos, es decir: $h(\mathbf{x}) = g(\mathbb{E}[y\,|\mathbf{x}]) = \eta(\mathbf{x})$. Por lo tanto se puede pensar que esta función h es análoga a la función de predicción η, también puede ser expresada como su expansión en bases funcionales.[21]

La idea, es que se remplace (o se aumente) la cantidad de covariables $\mathbf{x}$ con transformaciones de estas, capturadas en el vector $\Psi(\mathbf{x})$. Como ejemplos se pueden tener:

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

$\Psi_l(\mathbf{x}) = \ln x_l$ ó $x_l^{1/2}$ para alguna $l = 1, \ldots, N = d$, donde se tienen transformaciones no lineales en cada una (o algunas) de las covariables.

19. Dependiendo de el espacio funcional y la complejidad de la función real por estimar h, en ocasiones se requiere que $N = \infty$ para que se de la igualdad estricta (Bergstrom 1985).
20. *Linear basis function models.*
21. Un supuesto fuerte pero útil.

$\Psi_l(\mathbf{x}) = \exp\left\{-\dfrac{(x_l - \mu_l)^2}{2s^2}\right\} \quad l = 1, \ldots, d$ una expansión en bases gaussianas con μ_j el parámetro que gobierna la ubicación y s la escala de las funciones bases.

$\Psi_l(\mathbf{x}) = x_j^a I(\tau_b \leq x_j < \tau_c)$ para alguna j y $\forall l = 1, \ldots, N$ con $a \in \mathbb{N}$ y τ_b, τ_c nodos fijos. Dando lugar a una expansión en bases polinómicas como la que se usa en este trabajo (sección 2.3.2).

$\Psi_l(\mathbf{x}) = x_j x_k \quad \forall l = 1, \ldots, N$, para alguna j, k. Da lugar a un modelo con interacciones.

Como se ve, esta representación es tan flexible que engloba muchos de los modelos y transformaciones posibles en el mundo de las regresiones, uniendo temas de análisis funcional con estadística aplicada. Asimismo esta representación ha resultado ser de gran utilidad en la práctica. Se hace notar que el último ejemplo rompe con la aditividad inherente de los modelos que se han estudiado hasta ahora, mostrando que esta generalización no está restringida a ser completamente aditiva en covariables. Sin embargo h, por su construcción, siempre es lineal en los parámetros $\boldsymbol{\beta}$ pero no lineal en las covariables, dependiendo de la forma de $\Psi(\mathbf{x})$.

Dependiendo del tipo de datos y el propósito del modelo, puede ser conveniente usar algún tipo de funciones base sobre otras. Sin embargo, sobre todo cuando se tiene poca o ninguna experiencia con los datos, se busca una representación 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. No obstante,

una desventaja de estos métodos 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*[22]. Algunos de los métodos más comunes para lograrlo son los siguientes, Hastie, Tibshirani y Friedman (2008):

Métodos de restricción: 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 de esto.

Métodos de selección de variables: como lo son los modelos CART y MARS,[23] 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*.[24]

Para los objetivos de este trabajo, lo que se busca expresar en su expansión de bases funcionales no es la función de predicción η, sino sus componentes aditivos f_j. Al aislar cada función f_j que dependen únicamente de una variable real $x_j \quad \forall j$, es decir: $f_j : \mathbb{R} \to \mathbb{R} \quad \forall j$, se puede simplificar la exposición y reducir el número de índices pues sus expresiones algebraicas son idénticas.

22. Seguir los datos tan de cerca que se pierda la señal entre el ruido
23. *Classification & regression tree* (Breiman y col. 1984) y *multivariate adaptive regression splines* (Friedman 1991) respectivamente.
24. *Least absolute shrinkage and selection operator* (Hoerl y Kennard 1970; Tibshirani 1996)

2.3.2. Polinomios por partes y *splines*

Los polinomios por partes, por su flexibilidad, ha resultado ser de gran utilidad en diversas ramas de de las matemáticas. En particular, el mundo de la estadística surgen de forma natural como solución a varios problemas de modelado (ver apéndice A). Antes de exponer la representación final de las funciones f_j, se da una exposición constructiva de los polinomios por partes. Se usa como referencia las primeras dos secciones de el Capitulo 5 de Hastie, Tibshirani y Friedman (2008) y 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 $\mathcal{P} = \{\tau_1, \ldots, \tau_{J-1}\}$ tal que $a \leq \tau_1 < \ldots < \tau_{J-1} \leq b$. Las constantes τ son llamadas *nodos*.[25] Con los nodos seleccionados, se puede representar a diferentes niveles de precisión una función arbitraria h, a través de su expansión análoga a la ecuación (2.14), donde cada Ψ_j será una función que depende, tanto de la partición $\mathcal{P}$ como de la variable real x.

Para ejemplificar se presenta un ejemplo sencillo. Primero, se parte el intervalo en tres pedazos ($J = 3$) definiendo una partición con dos nodos, es decir: $\mathcal{P} = \{\tau_1, \tau_2\}$.

25. En la definición, se puede incluir o no la frontera dependiendo de si se busca hacer inferencia fuera del intervalo acotado de los datos.

Posteriormente, a cada subintervalo se le asocia una función Ψ_j tales que:

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

con $I(\cdot)$ la función indicadora que vale uno si x se encuentra en la región y cero en otro caso. Bajo esta definición, se construye una función h:

$$\begin{aligned} h(x) &= \sum_{l=1}^{J} \beta_l \Psi_l(x) \\ &= \beta_1 I(x < \tau_1) + \beta_2 I(\tau_1 \leq x < \tau_2) + \beta_3 I(\tau_2 \leq x). \end{aligned}$$

Esta función h es una función escalonada, en el sentido de que cada región de x tiene un nivel β_j.[26]

Con este ejemplo, al partir el intervalo y construir funciones más sencillas sobre ellos, se ilustra a grandes rasgos como funcionan los polinomios por partes. Sin embargo, los polinomios por partes pueden ser mucho más flexibles pues a cada intervalo se puede ajustar un polinomio de grado arbitrario $(M-1)$.[27] Adicionalmente, se puede añadir restricciones de continuidad en los nodos y no sólo continuidad entre los polinomios, sino continuidad en las derivadas. Esta es la flexibilidad de los polinomios

26. Dado un conjunto de observaciones $\{(y_i, x_i)\}_{i=1}^n$, si se buscara estimar los parámetros β usando una función de perdida cuadrática, se puede demostrar que cada $\hat{\beta}_j = \bar{y}_j$, es decir, para cada región el mejor estimador constante, es el promedio de los puntos de esa región.

27. Se usa esta convención pues para representar un polinomio de grado $M-1$ se necesitan M términos.

por partes, que se les puede pedir cuanta *suavidad*, o no, se requiera, entendida como la continuidad de la $(\tilde{K})$-ésima derivada.

Número total de funciones bases N

Formalizando la idea anterior, al tomar una expansión de bases para cada intervalo, el número de funciones base aumenta en J por cada grado que se agregue, dando un total de JM bases funcionales. Esto ocurre porque se necesita definir una base de tamaño M para cada subintervalo $j = 1, \ldots, J$, es decir, $\mathcal{B} = \left\{1, x, x^2, \ldots, x^{M-1}\right\}$ con $\mathcal{B}$ la base. Esta definición, deriva en 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 en las derivadas que se le pida al polinomio, lo restringe y por ende, devuelve el mismo número de funciones bases, se denota por $\tilde{K}$ este número. Sin embargo, es más intuitivo pensar en un parámetro $K = \tilde{K} + 1$ como el número de restricciones que se imponen en los nodos. Es decir, $K = 0$ implica intervalos independientes, $K = 1$, implica que los polinomios se conectan, $K = 2$ implica continuidad en la primera derivada $(\tilde{K} = 1)$ y así sucesivamente. Bajo esta definición los polinomios por partes tienen un total de:

$$N(M, J, K) = JM - K(J - 1) \tag{2.16}$$

bases funcionales y por ende, el mismo número de parámetros $\boldsymbol{\beta}$ por estimar. Dada la construcción y las características de M, J y K se derivan de forma trivial las restricciones para estos parámetros: $M > K \geq 0$ y $J > 1$.

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. 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, es decir, se impone la restricción adicional $K = M-1$. Se hace notar, que existen muchos tipos de *splines*, como lo son los B-Splines. Dependiendo de la aplicación, se pueden construir más o menos flexibles o más rápidos en su implementación computacional. En **deboor1978splines** y más recientemente Wahba (1990) se hacen tratados extensivos sobre ellos y sus generalizaciones. Los *splines* cúbicos se han popularizado en la literatura, pues resultan en curvas suaves al ojo humano, reteniendo suficiente flexibilidad para aproximar una gran cantidad de funciones.

Polinomios por parte flexibles

Habiendo definido M, J y K y por ende el número de funciones bases N, finalmente se le puede dar forma funcional a las funciones base Ψ que se usan en este trabajo. Se define primero, la función auxiliar *parte positiva*; sea $a \in \mathbb{R}$:

$$a_+ = \text{máx}\left\{0, a\right\}.$$

Esta función, ayuda a que se puedan escribir un gran número de polinomio por partes usando una notación relativamente sencilla, evitando separar la función en varias lineas.

Definición 2.5. Expansión en bases truncada, Denison, Mallick y Smith (1998):

$$h(x) = \sum_{l=1}^{N} \beta_l \ \Psi_l(x, \mathcal{P}) = \tilde{\boldsymbol{\beta}}^t \boldsymbol{\Psi}(x, \mathcal{P}) \tag{2.17}$$

$$\text{donde, } \ N = JM - K(J-1)$$

$$= \underbrace{\sum_{\hat{\imath}=0}^{M-1} \beta_{\hat{\imath},0} \ x^{\hat{\imath}}}_{\text{polinomio base}} + \underbrace{\sum_{\hat{\imath}=K}^{M-1} \sum_{\hat{\jmath}=1}^{J-1} \beta_{\hat{\imath},\hat{\jmath}} \ (x - \tau_{\hat{\jmath}})_+^{\hat{\imath}}}_{\text{parte truncada}} \tag{2.18}$$

Esta representación, tiene muchas propiedades atractivas que se detallan a continuación. Asimismo, es prácticamente la expansión de bases implementada en el modelo final.

Al primer sumando de (2.18) se le conoce como polinomio base (*baseline polynomial*), pues afecta a todo el intervalo de definición $[a, b]$. El segundo sumando, conocido como la parte truncada, controla la suavidad entre los nodos. Es decir, por cada nodo $\hat{\jmath} = 1, \ldots, J-1$ se tienen $M - K$ funciones parte positivas $(\cdot)_+$ que se activan (se vuelven positivas) a medida que x recorre el su dominio $[a, b]$ hacia la derecha y va pasando por los nodos τ_j. Estas funciones parte positiva, van capturando los efectos de los intervalos anteriores que, al combinarlos con el primer sumando definen

un polinomio de grado $M-1$ en todo el intervalo.[28] La principal utilidad de esta expansión, es que engloba todas las ideas antes mencionadas en tres parámetros: M, J y K, al escogerlos, se pueden representar un gran número polinomios por partes. Por ejemplo, si $M=3$, $J=5$ y $K=0$ se tiene un polinomio por partes en 5 subintervalos (4 nodos) donde cada subintervalo es una parábola independiente de la anterior, es decir, las parábolas no son continuas entre si. Por el contrario, si $K=M-1$ se devuelve a la definición de *splines*, o por último, si $M=1$, $J=3$ y $K=0$, se tienen constantes por segmentos como en el ejemplo introductorio de los polinomios por parte.

Para la facilitar la interpretación de los parámetros y la expansión de (2.18), los parámetros β cuentan dos índices: $\hat{\imath}$ y $\hat{\jmath}$. El índice $\hat{\imath}$ siempre estará asociado al grado de su función base asociada, es decir, si $\hat{\imath}=2$ se está hablando de un término de grado 2. En el segundo sumando (la parte truncada) el índice $\hat{\imath}$ comienza en K para codificar las restricciones de continuidad.[29] El segundo índice $\hat{\jmath}=1,\ldots,J-1$ describe el nodo al que está asociado el parámetro. Como convención, si $\hat{\jmath}=0$, se hace referencia al primer sumando (el polinomio base) que siempre está activo sobre el intervalo.

La ecuación (2.17) es una expansión en bases arbitrarias análoga a definición de

28. Esta expansión, surge de integrar un polinomio por partes, constante en cada subintervalo, $M-1$ veces, pues las constantes de integración se pueden agrupar en el polinomio base.

29. Esta codificación es sutil pues, al hacer la demostración de continuidad, hay que considerar los límites izquierdos y derechos. Los límites izquierdos siempre coinciden con la función en el nodo. Sin embargo, los términos $(x-\tau)_+^K$ se desvanecen únicamente hasta la (K)-ésima derivada. Para la $(K+1)$-derivada, el coeficiente correspondiente se suma a la función y rompe la continuidad pues no corresponde el límite derecho.

(2.14). Se hace notar, que en (2.17) se hace referencia a β con un solo índice $l = 1, \ldots, N^*$ mientras que en (2.18) con dos. Esta disparidad surge de la necesidad de una doble interpretación de la expresión; como una expansión de bases arbitrarias Ψ_l y su correspondiente expansión en bases truncadas. Sin embargo, existe una biyección notacional entre los elementos β_l, $\beta_{i,j}$ y Ψ_l presentada en la tabla 2.2 de la página 36. Esta tabla ayuda no sólo a esclarecer la notación, sino a expresar todo de forma matricial que posteriormente se implementará en el código.

Los nodos τ: el trabajo de Denison, Mallick y Smith

Las ideas de Denison, Mallick y Smith (1998), van más allá de la ecuación (2.18). En su trabajo, los autores presentan un método automático y bayesiano para estimar con un alto grado de precisión relaciones funcionales complejas. 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 $\{(y_i, x_i)\}_{i=1}^n$:

$$y_i = h(x_i) + e_i \quad i = 1, \ldots, n \tag{2.19}$$

donde e_i son variables aleatorias con media cero ($\mathbb{E}[e_i] = 0 \; \forall i$).

Para lograrlo, usan el polinomio definido en (2.18) y desarrollan un método bayesiano para la estimación de los nodos τ que son tradicionalmente fijos. Asimismo, su método permite modelar a la vez J aumentando o disminuyendo la cantidad de nodos, desarrollando un algoritmo de muestre Gibbs trans-dimensional, es decir, el

β_l	$\beta_{\hat{\imath},\hat{\jmath}}$	$\Psi_l(x,\mathcal{P})$		
Subíndice l	Subíndices $\hat{\imath},\hat{\jmath}$	Función Base		
1	$0,0$	1	M elementos	
2	$1,0$	x		
$\vdots$	$\vdots$	$\vdots$		
M	$M-1,0$	x^{M-1}		
$M+1$	$K,1$	$(x-\tau_1)_+^K$	$M-K$	$J-1$ veces
$M+2$	$K+1,1$	$(x-\tau_1)_+^{K+1}$		
$\vdots$	$\vdots$	$\vdots$		
$M+(M-K)$	$M-1,1$	$(x-\tau_2)_+^{M-1}$		
$M+(M-K)+1$	$K,2$	$(x-\tau_2)_+^K$	$M-K$	
$M+(M-K)+2$	$K+1,2$	$(x-\tau_2)_+^{K+1}$		
$\vdots$	$\vdots$	$\vdots$		
$M+2(M-K)$	$M-1,2$	$(x-\tau_2)_+^{M-1}$		
$\vdots$	$\vdots$	$\vdots$		
$M+(J-2)(M-K)+1$	$K,J-1$	$(x-\tau_{J-1})_+^K$	$M-K$	
$M+(J-2)(M-K)+2$	$K+1,J-1$	$(x-\tau_{J-1})_+^{K+1}$		
$\vdots$	$\vdots$	$\vdots$		
$M+(J-1)(M-K)$	$M-1,J-1$	$(x-\tau_{J-1})_+^{K+1}$		

Tabla 2.2: Biyección entre β_l, $\beta_{i,j}$ y sus correspondientes funciones base Ψ_l.

Se tiene un total de $N = M + (J-1)(M-K) = JM - K(J-1)$ términos, ecuación (2.16). Por construcción, se es consistente con la definición de *spline* si $K = M-1$.

algoritmo cambia el número de parámetros en cada iteración. 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. Con lo anterior, se demuestra que la suavidad, aunque útil, no siempre es necesaria. Muchas funciones discontinuas no se podrían estimar del todo usando polinomios continuos como los *splines*. Al final, todo depende de los datos y el propósito del modelo.

La ventaja de que nodos sean parámetros por estimar, es que se estos pueden concentrar en los lugares donde la función varia más. Y al contrario, si la función es relativamente suave para algún intervalo, se necesitan usan pocos nodos. Sin embargo y para propósitos de este trabajo, los nodos se toman determinados desde el principio. Su número $J-1$ es definido por el estadista y su localización se escoge en los cuantiles del rango de las covariables.[30] 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 observará que para fines prácticos, el modelo funciona muy bien y finalmente en el capítulo 5 se discute que habría cambiado de haberse implementado.

30. Es decir, si se tiene J intervalos, se toman los nodos como los cuantiles que acumulan probabilidad $1/J$ en el rango $[a, b]$.

2.3.3. Consideraciones matemáticas adicionales

Bajo la óptica de la implementación del modelo, se hace énfasis en la linealidad de los parámetros. Al sustituir (2.4) en (2.3) este hecho se hace aún más evidente:

$$\begin{aligned}
\eta(\mathbf{x}_i) &= f_0 + \sum_{j=1}^{d} f_j(x_{i,j}) \\
&= f_0 + \sum_{j=1}^{d} \boldsymbol{\beta}_j^t \boldsymbol{\Psi}(\mathbf{x}_i, \mathcal{P}_j) \\
&= f_0 + \sum_{j=1}^{d} \left[\sum_{l=1}^{N^*} \beta_{j,l} \Psi_l(x_{i,j}, \mathcal{P}_j) \right] \qquad (2.20) \\
&= \boldsymbol{\beta}^t \widetilde{\boldsymbol{\psi}}_i(\mathbf{x}_i). \qquad (2.21)
\end{aligned}$$

En donde cada sumando interior de (2.20) tiene una expansión de bases funcionales definida por la ecuación (2.18).[31] Esta representación, permite visualizar la linealidad en parámetros del modelo, asimismo, la función f_0, al ser constante puede ser interpretada como otro parámetro adicional, es decir: $f_0 \equiv \beta_0$. La linealidad en los parámetros de la función de predicción η, derivan en que (2.20) pueda ser re-expresada simplemente como el producto punto de un largo vector de parámetros $\boldsymbol{\beta} \in \mathbb{R}^{\lambda}$ y un renglón $\widetilde{\boldsymbol{\psi}}_i(\mathbf{x}_i)$, ec. (2.21), de una matriz de diseño más grande $\widetilde{\Psi}$ que incorpora la doble suma de las correspondientes expansiones en bases y todas las observaciones $i = 1, \ldots, n$.

31. No se hace la sustitución pues la notación resulta innecesaria.

No obstante, bajo las definiciones anteriores aún se tiene un problema de confusión en los parámetros. Al ya tener un término independiente $f_0 = \beta_0$ en (2.20), para preservar la identificabilidad de los parámetros se deben realizar unas pequeñas modificaciones a (2.18). Los parámetros confundidos pueden tener dos orígenes. Primero, si se permite que $K = 0$ el segundo sumando tendría términos independientes no deseados. Esto se arregla fácilmente imponiendo la restricción de continuidad en los polinomios, es decir, $K > 0$.[32] Segundo, se debe retirar el término independiente inherente en el polinomio base, es decir, comenzar el primer sumando de (2.18) en uno en vez de cero. Esta modificación retira una función base modificando N y convirtiendola en N^*.[33] Juntando estas cambios (2.18) se transforma en:

$$
\begin{aligned}
h(x) &= \sum_{l=1}^{N^*} \beta_l \, \Psi_l(x, \mathcal{P}) \\
\text{con } \; N^* &= J \times M - K(J-1) - 1 \\
\text{donde: } \; M &> K > 0 \text{ y } J > 1 \\
&= \sum_{\hat{\imath}=1}^{M-1} \beta_{\hat{\imath},0} \, x^{\hat{\imath}} + \sum_{\hat{\imath}=K}^{M-1} \sum_{\hat{\jmath}=1}^{J-1} \beta_{\hat{\imath},\hat{\jmath}} \, (x - \tau_{\hat{\jmath}})_+^{\hat{\imath}}. \qquad (2.22)
\end{aligned}
$$

Lo cual, es finalmente la expansión que se implementa en el modelo. Solamente basta dejar que $h(x)$ sea igual a $f_j(x_j)$ para toda $j = 1, \ldots, d$ y se regresa a la ecuación

32. De manera preeliminar, se implementó una versión del algoritmo que permitía esta confusión. El ajuste no mejoraba cuando $K = 0$ y solamente causaba que las cadenas simuladas de los parámetros no convergieran debidamente. Sin embargo, en los polinomios por partes si se observaba la discontinuidad.

33. Denison, Mallick y Smith resuelven este problema de identificabilidad al solamente usar una covariable para la estimación de las curvas, permitiendo retirar uno de los parámetros independientes sin penalización. Asimismo, su algoritmo automático trans-dimensional les permitía tener polinomios por partes discontinuos.

canónica del modelo (2.4).

Juntando estas observaciones, el predictor lineal η se puede re-expresar en su forma vectorial compacta $\boldsymbol{\eta}$:

$$\boldsymbol{\eta}(\mathbf{X}) = \widetilde{\Psi}(\mathbf{X})\boldsymbol{\beta}, \tag{2.23}$$

donde $\mathbf{X} \in \mathbb{R}^{n \times d}$ es la matriz de covariables, $\boldsymbol{\beta}$ el vector de parámetros con un total de $\lambda = 1 + d \times N^*$ elementos y $\widetilde{\Psi}(\mathbf{X}) \in \mathbb{R}^{n \times \lambda}$ la transformación no lineal definida con anterioridad. Vistos en sus correspondientes formas matriciales, se tienen las siguiente estructuras:

$$\boldsymbol{\eta}(\mathbf{X}) = \begin{bmatrix} \eta(\mathbf{x}_1) \\ \eta(\mathbf{x}_2) \\ \vdots \\ \eta(\mathbf{x}_n) \end{bmatrix} \qquad \boldsymbol{\beta} = \left[\begin{array}{c} \beta_0 \\ \hdashline \beta_1 \\ \vdots \\ \beta_{N^*} \\ \hdashline \beta_{N^*+1} \\ \vdots \\ \beta_{2N^*} \\ \hdashline \vdots \\ \beta_{d \times N^*} \end{array}\right] \begin{array}{l} \text{término independiente} \\ \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} \boldsymbol{\beta}_1 :\ N^* \text{ términos} \\ \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} \boldsymbol{\beta}_2 :\ N^* \text{ términos} \\ \\ \\ \\ \\ \end{array} \left.\begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}\right\} d \text{ veces}$$

$$\widetilde{\Psi}(\mathbf{X}) = \left[\begin{array}{c:c:c:c} 1 & f_1(x_{1,1}) & \dots & f_d(x_{1,d}) \\ \vdots & \vdots & & \vdots \\ 1 & f_1(x_{n,1}) & \dots & f_d(x_{n,d}) \end{array}\right] \tag{2.24}$$

$$= \left[\begin{array}{c:c:c:c:c:c:c} 1 & \Psi_1(x_{1,1},\mathcal{P}_1) & \dots & \Psi_{N^*}(x_{1,1},\mathcal{P}_1) & \dots & \Psi_1(x_{1,d},\mathcal{P}_d) & \dots & \Psi_{N^*}(x_{1,d},\mathcal{P}_d) \\ \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ 1 & \Psi_1(x_{n,1},\mathcal{P}_1) & \dots & \Psi_{N^*}(x_{n,1},\mathcal{P}_1) & \dots & \Psi_1(x_{n,d},\mathcal{P}_d) & \dots & \Psi_{N^*}(x_{n,d},\mathcal{P}_d) \end{array}\right]$$

Bajo esta definición, el modelo se simplifica y se observa que en realidad cada f_j es una expansión de cada covariable x_j en más términos que se le añaden al predictor lineal. Asimismo, aunque el modelo no sufra de problemas de identificabilidad en los parámetros, no se puede asegurar ya la no-colinealidad entre las columnas de $\widetilde{\Psi}$ por construcción, por lo que se podrían dar problemas en la estimación.[34]

A pesar de la utilidad de estos polinomios por parte, todos sufren de problemas más allá del rango de definición $\mathcal{X}_j = [a_j, b_j] \quad \forall j = 1, \dots, d$. Pues, su naturaleza global hace que fuera de la región con nodos los polinomios crezcan o decrezcan rápidamente. Por lo tanto, extrapolar con polinomios es peligroso y podría llevar a predicciones 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. Es razonable que esta modificación mejore la fuerza predictiva fuera de el dominio de

34. Bajo el paradigma frecuentista y esta forma funcional, los parámetros también se podrían estimar por un procedimiento de mínimos cuadrados, en donde serían evidentes los problemas en la matriz de covarianzas $\widetilde{\Psi}^t\widetilde{\Psi}$.

entrenamiento. Sin embargo, en un contexto de regresión (o clasificación) general, se recomienda no hacer inferencia fuera de el espacio de covariables $\mathcal{X}^d$, pues en realidad, no se tiene evidencia para tomar conclusiones en esta región.

Al estar trabajando en espacios funcionales, la elección de base es relativamente arbitraria y se podría modificar 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 $\boldsymbol{\beta}$ es inmediata. Sin embargo, no es buena computacionalmente hablando cuando J es grande. En la practica, usualmente se implementan B-Splines[35] o bases ortogonales que se derivan de lo estudiados. No obstante, para no complicar más la exposición (y el algoritmo en si) se implementó una versión vectorizada de (2.22) con base en la tabla 2.2 y (2.24) que se ejecuta bastante rápido inclusive cuando n y J son grandes.

En la practica, los parámetros M, J y K se calibran comparando diferentes alternativas de modelos pues, como ya se mencionó anteriormente, hacer J variable y automático hubiera escapado de los objetivos del trabajo.[36]

35. Vease el Capítulo 5.5 de Wasserman (2007) o el Apéndice del Capítulo 5 en Hastie, Tibshirani y Friedman (2008).
36. En el capítulo 4 y 5 se discute la selección del modelo.

Capítulo 3

Paradigma bayesiano e implementación

> *[...], 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.*
>
> Barber (2010)

Dado el modelo tan estructurado que se desarrolla, pasar de su forma matemática a su implementación computacional no resulta fácil. Sin embargo, con base en las ideas de Albert y Chib (1993), se desarrolla un algoritmo que logra un buen grado

de precisión en la predicción de las respuestas de una forma computacionalmente eficiente. En el fondo, la implementación recae en el método de muestreo de Gibbs, por lo que se hace una breve introducción a la escuela de inferencia bayesiana. Al algoritmo también se le titula: *bayesian piece wise polinomial model (bpwpm)* y puede ser revisado en la página 64. 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 C.

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 útil, no está del todo axiomatizada 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, desarrollando un paradigma completo para la inferencia y la toma de decisiones bajo incertidumbre. Asimismo, 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, (Bernardo y Smith 2001) y (Mendoza y Regueiro 2011).

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*. Este paradigma es el que más corresponde con el sentido que usualmente se le da a la palabra. La inferencia o predicción sobre eventos, se realiza mediante una *actualización* de la información que se tiene bajo la luz de nueva evidencia, modificando así la medida de incertidumbre. El teorema de Bayes es el mecanismo que permite realizar este proceso de actualización. De manera informal el teorema (3.1) explica que 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.

Teorema 3.1. *El teorema de Bayes (informal):*

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

Donde, el término central $P(C|E)$ es una medida descriptiva de las condiciones (usualmente datos) llamada *verosimilitud*, $P(E)$ es la probabilidad previa (*a priori*) que se tiene del evento E y $P(E|C)$ es la probabilidad posterior (actualizada).

En un contexto de estadistica paramétrica más formal, los eventos E se abstraen en una serie de parámetros θ que usualmente son desconocidos. Asimismo las condiciones C quedan resumidas en datos observados $\mathbf{X}$ que son interpretados como *evidencia*. Bajo este paradigma antes de poder hacer cualquier intento de inferencia sobre θ, se debe especificar el *modelo probablistico* que se asume describe el fenómeno observado, pues es a través de este modelo que se da una medida concreta

para cuantificar la incertidumbre. Primero, se tienen ciertas creencias, hipótesis u conocimiento previo, *a priori*, sobre los parámetros θ, los cuales se representan por una medida de probabilidad $\pi(\theta)$. Segundo, se tienen datos $\mathbf{X}$ a los que se asigna un modelo de probabilidad dependiente de los parámetros $\pi(\mathbf{X}|\theta)$, a la que se le conoce como *verosimilitud* (Bernardo 2003).

Teorema 3.2. *El teorema de Bayes:*

$$\pi(\theta|\mathbf{X}) \propto \pi(\mathbf{X}|\theta)\,\pi(\theta) \tag{3.2}$$

Habiendo especificado el modelo, el teorema de Bayes (3.2) describe el proceso de actualización de conocimiento sobre los parámetros θ. La idea es que este proceso de actualización sea, de la misma forma, un *proceso de aprendizaje*, en el cual los parámetros capturen la información contenida en los datos.

Bajo el paradigma frecuentista, se adopta un enfoque diferente para el aprendizaje. Se asume que no hay incertidumbre inherente en los parámetros dado los datos por lo que simplemente son desconocidos y se deben de estimar. 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 suma de residuales cuadrados (RSS por sus siglas en inglés) de los modelos ANOVA, se busca la $\hat{\theta}$ que minimice los residuales, así, el modelo logra capturar toda la variabilidad posible de los datos.

Independientemente del paradigma estadístico que se escoja, siempre es importante la validación del modelo y de sus supuestos. No obstante, tanto la teoría bayesiana como la frecuentista han resultado de infinita utilidad en la práctica 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 $\propto$. 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.[1] 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. Muchos de estos métodos recaen en la teoría de las *cadenas de Markov*, como lo es, el muestreador Gibbs a presentarse en la sección 3.2.

1. En el Apéndice B se detallan las distribuciones conjugadas y se realiza más a fondo la derivación de los resultados de este trabajo.

Estimadores Bayesianos

Una vez realizado el proceso de actualización, se cuenta con una distribución posterior de probabilidad para los parámetros de interés.[2] No obstante, por practicaldiad y utilidad, en ocasiones se busca dar un *estimador puntual* de los parámetros. La teoría de la decisión dicta que para medir la deseabilidad de escoger cierto parámetro en particular, se debe definir una función de perdida (L) o utilidad que optimice esta elección. Particularmente, las funciones de perdida logran medir las consecuencias incurridas, al tomar $\hat{\theta}$ como el valor puntual del parámetro. Lo hacen, penalizando la distancia entre el valor real θ y su estimador puntual $\hat{\theta}$. Por lo tanto y sin entrar mucho en los detalles técnicos, para dar un estimador puntual se resuelve el problema de optimización:

$$\hat{\theta} = \min_{\theta \in \Theta} \mathbb{E}[L(\hat{\theta}, \theta)] \tag{3.3}$$

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 la distribución posterior.

2. Es común tener, no es la distribución analítica, sino una muestra de ella.

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.[3]. En el paquete, se implementa una forma sencilla de obtener estimadores puntuales con cualquiera de las dos primeras funciones de pérdida (cuadrática y valor absoluto). Para la aplicación de este modelo, sin embargo, dado el uso de familias conjugadas, las distribuciones posteriores resultantes tienen la característica que la media, la mediana y la moda coinciden facilitando la elección por parte del analista.

3.2. Herramientas de simulación

Una vez establecida el proceso de actualización, se estudian las técnicas para simular de la distribución posterior $\pi(\theta|\mathbf{X})$. Desde principios de los años noventa, se han desarrollado algoritmos y paquetería estadística que permiten plantear modelo de una forma sencilla y obtener una muestra arbitrariamente grande de θ. Sin embargo, la gran mayoría de estos algoritmos recaen 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 simula-

3. Excepto la moda muestra para los casos continuos.

ción, como tal es un tema que merece un estudio más profundo, no obstante, sus aplicaciones prácticas son muy intuitivas (Robert y Casella 2004). Las técnicas de simulación, permiten que los estadísticos y experimentadores puedan hacer el menor número de supuestos posibles sobre los modelos, puesto que ya no se buscan resultados analíticos sino más bien, describir el fenómeno de la forma más precisa posible y dejar los cálculos a una computadora.

Breve introducción a las cadenas de Markov

Definición 3.3. Una cadena de Markov, es una secuencia de variables aleatorias: $X^{(1)}$, $X^{(2)}$, ... que cumplen la *propiedad Markoviana*:

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

con t interpretado como *tiempo* y $x^{(t)}$ el estado en el que se encuentra la variable aleatoria $X^{(t)}$.

Esta definición, implica que 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 explicada 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 Ross (2014).

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 ($X^{(t)}$ 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*,[4] entonces existe una *distribución límite* que es igual a la *distribución estacionaria*: $\exists\, \pi$, un vector de estados, 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.[5] Esta idea se puede extender a casos más complejos donde se relajan o se cambian algunos de los supuestos. Incluso, se extiende a casos donde el número de estados es no finito, pero el concepto fundamental es el mismo. En el contexto de este trabajo, la idea es poder simular *secuencialmente* cadenas de parámetros θ que eventualmente converjan a la distribución estacionaria.

3.2.1. Muestreador de Gibbs

El el muestreador de Gibbs (*Gibbs sampler*) es método, para simular variables aleatorias de una *distribución conjunta* sin tener que calcularla directamente, (Gelfand y Smith 1990). Usualmente, el muestreo de Gibbs se usa dentro de un contexto

4. Aperiódica, irreducible y recurrente positiva. Para efectos dse 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.

5. 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)

bayesiano, aunque también funciona para otras aplicaciones. A primera vista, pareciera complejo, 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 los parámetros $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_\lambda)^t$ que provienen de la distribución conjunta $\pi(\boldsymbol{\theta})$. Esta distribución usualmente no es conocida analíticamente, sin embargo el muestreador de Gibbs permite simular una muestra arbitrariamente grande de la distribución con la que se puede aproximar empíricamente $\hat{\pi}(\boldsymbol{\theta}) \approx \pi(\boldsymbol{\theta})$. Posteriormente esta muestra se estudia con medidas de centralidad y dispersión, gráficos, cuantiles, etcétera.

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:

$$\begin{aligned} \theta_1 &\sim \pi(\theta_1|\theta_2, \ldots, \theta_\lambda) \\ \theta_2 &\sim \pi(\theta_2|\theta_1, \theta_3, \ldots, \theta_\lambda) \\ &\vdots \\ \theta_\lambda &\sim \pi(\theta_\lambda|\theta_1, \ldots, \theta_{\lambda-1}) \end{aligned} \tag{3.4}$$

Se comienza con un valor inicial arbitrario $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \ldots, \theta_\lambda^{(0)})^t$, 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 inicia-

les. En este caso, para $k = 1, 2, 3, \ldots$ se tiene:

$$\begin{aligned}
\theta_1^{(k)} &\sim \pi(\theta_1|\theta_2^{(k-1)}, \ldots, \theta_\lambda^{(k-1)}) \\
\theta_2^{(k)} &\sim \pi(\theta_2|\theta_1^{(k)}, \theta_3^{(k-1)}, \ldots, \theta_\lambda^{(k-1)}) \\
&\vdots \\
\theta_\lambda^{(k)} &\sim \pi(\theta_\lambda|\theta_1^{(k)}, \ldots, \theta_{\lambda-1}^{(k)})
\end{aligned} \tag{3.5}$$

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 $\pi(\boldsymbol{\theta})$.

La convergencia no es intuitiva, es decir, no es trivial derivar que al muestrear de las distribuciones condicionales, se obtenga eventualmente una muestra de la distribución conjunta Sin embargo, la prueba formal recae en las mismas ideas de las cadenas de Markov. Definido el problema, se puede formar una kernel de transición, generalización de las matrices de transición, derivado de las distribuciones condicionales de $\theta_i \; \forall i = 1, \ldots, \lambda$. A la larga ($k \to \infty$) y dadas las propiedades de ergodicidad, los valores de la cadena corresponden a valores muestreados de la distribución conjunta. En Casella y George (1992) y Tierney (1994) se presentan versiones más rigurosas de el porqué las cadenas Markov de un muestreador de Gibbs convergen.

En la práctica, una vez obtenida la cadena $\left\{\theta^{(k)}\right\}_{k=0}^{N_{\text{sim}}}$, donde N_{sim} es el número total de elementos simulados, es importante revisar si esta ya ha alcanzado la distribución posterior. Para ello, es usual revisar la media ergódica (media acumulada) de cada

parámetro, de donde se esperaría ver que la variación hacia el final de la cadena ser mínima. Asimismo, se suele analizar la traza de la cadena en sí y los histogramas de ella. Para ejemplificar, en la figura 3.1 se tienen tres imágenes de las cadenas simuladas por el mustreador Gibbs implementado en este trabajo,[6] en particular, las cadenas de la realización 1 del ejemplo 1 en la página la página 75. Para esta realización en praticular, se escogen los parámetros $M = 2$, $J = 2$ y $K = 1$, implicando que se tienen rectas continuas en tres nodos ($N^* = 2$), derivando en un total de $\lambda = 5$ parámetros por estimar ($\boldsymbol{\beta} \in \mathbb{R}^5$). La imagen 3.1a presenta la media ergódica de todos los parámetros que se empiezan a estabilizar conforme avanzan el número de iteraciones del algoritmo. En 3.1b, se grafican las trazas de los primeros 3 parámetros (β_0, β_1 y β_2) y en 3.1c sus correspondientes histogramas.[7] Se observa como los primeros valores de los parámetros aún no se estabilizan del todo y sus medias fluctúan, asimismo, se puede observar claramente, como los histogramas tienen formas similares a la de una distribución normal; este hecho se esclarecerá en la sección 3.3.

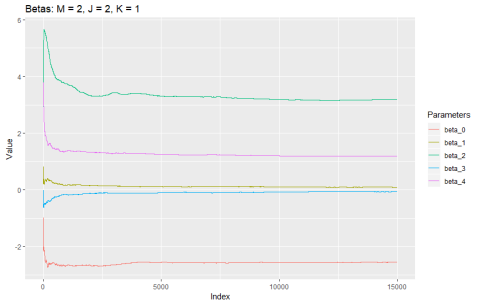
Mejoras a las cadenas

Como se observó en las imágenes previas, el muestreador de Gibbs aunque útil, no es infalible.[8] No obstante, las cadenas pueden ser mejoradas de dos formas sencillas. La primera se conoce como *burn-in* y consiste en eliminar los primeros (k^*)-esimos

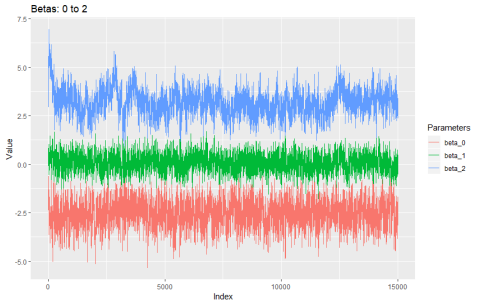
6. Las imágenes fueron generadas con la librería *ggplot2*, incorporada a las funcionalidades del paquete desarrollado para este trabajo.

7. Solamente se muestran los primeros tres parámetros para evitar tener gráficos muy saturados.

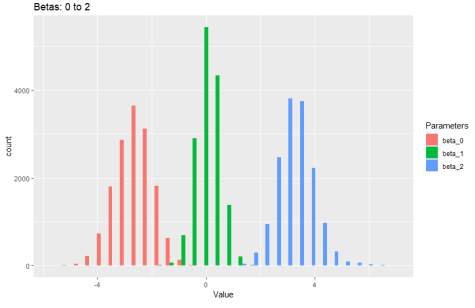
8. En el sentido que no genera una muestra de v.a.i.i.d.



(a) Media ergódica



(b) Trazas de la cadena



(c) Histogramas

Figura 3.1: Muestro Gibbs para el ejemplo 1 (sección 4.2)

valores simulados de la cadena. Esto dado que el valor inicial $\theta^{(0)}$ es fijado por el estadista, por lo que en ocasiones el algoritmo tiene que explorar una región extensa de posibles valores de θ para converger. Por lo tanto, si se busca una muestra de distribución posterior $\pi(\theta)$ los primeros valores pueden ser descartados. El corte $0 < k^* < N_{\text{sim}}$ es decidido de forma subjetiva una vez que se explora la cadena entera, ya sea por resúmenes numéricos o por representaciones gráficas. El segundo método es conocido como adelgazamiento (*thinning*) y consiste en tomar cada (k_{thin})-ésimo valor de la cadena para reducir (más no desaparecer) la dependencia entre los parámetros. Esto ocurre porque las cadenas de Markov, sobre las que de-

pende el muestreador de Gibbs, son generadas de forma secuencial con base en el valor actual actual de la cadena (propiedad markoviana). Por lo tanto, los valores simulados están altamente correlacionados. Sin embargo, estos sencillos pasos para mejorar las cadenas logran mejorar las muestras y ya se encuentran implementados en el paquete.

3.3. El modelo *bpwpm*

Habiendo estudiado el muestreador de Gibbs, resta únicamente definir el algoritmo usado en el modelo.

Aumentación de datos para respuestas binarias

En Albert y Chib (1993), los autores desarrollan un método bayesiano para el análisis de respuestas binarias y policotómicas.[9] En el caso binario, su enfoque resultaba muy atractivo para los objetivos del trabajo. Su modelo titulado *aumentación de datos para respuestas binarias*,[10] propone una definición del modelo probit como la presentada en (2.1) y (2.2); bajo esta definición, la derivación de las distribuciones marginales de los parámetros es fácil.[11] Asimismo, proponen usar distribuciones con-

9. Una respuesta policotómica es una respuesta que perteneces a más de dos categorías, por ejemplo, partidos políticos; usualmente se modelan con distribuciones multinomiales.

10. *data augmentation for binary data*

11. Albert y Chib también proponen un modelo con función liga t-student dando lugar a un modelo *tobit*.

jugadas normales para los parámetros $\boldsymbol{\beta}$ derivando en un algoritmo relativamente rápido pues 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.

Entrando en el detalle, el planteamiento es casi idéntico al presentado en la definición 2.1, es decir, se introducen n variables latentes $\mathbf{z} = (z_1, \ldots, z_n)^t$ tales que:

$$y_i = \begin{cases} 1 & \iff \quad z_i > 0 \\ 0 & \iff \quad z_i \leq 0 \end{cases} \tag{2.1}$$

$$z_i \,|\, \mathbf{x}_i \sim \mathcal{N}(z_i \,|\, \eta(\mathbf{x}_i), 1) \tag{2.2}$$

$$\eta(\mathbf{x}_i) = \boldsymbol{\beta}^t \widetilde{\boldsymbol{\psi}}_i(\mathbf{x}_i) \tag{2.21}$$

Donde $\widetilde{\boldsymbol{\psi}}_i(\mathbf{x}_i)$ es el renglón i de la matriz de transformación (2.24) presentada en la página 41. Sin embargo, se busca estudiar el modelo desde el paradigma bayesiano. Dado que el modelo recae en la definición de las variables latentes $\mathbf{z}$, las cuales son desconocidas pero modeladas con una distribución normal, estas pasan a ser parte de los parámetros en el sentido de que deben ser simuladas también, pues son la liga entre todos los componentes del modelo. Siendo consistentes con la notación de (3.2) se tienen entonces dos grupos de parámetros: $\boldsymbol{\theta} = (\mathbf{z}, \boldsymbol{\beta})$. Por lo tanto, la derivación

de la densidad posterior resulta en:

$$\begin{aligned}
\pi(\mathbf{z}, \boldsymbol{\beta}|\mathbf{y}, \mathbf{X}) &\propto \pi(\mathbf{y}|\mathbf{X}, \mathbf{z}, \boldsymbol{\beta})\ \pi(\mathbf{z}, \boldsymbol{\beta}) && \text{por (3.2)} \\
&\propto \pi(\mathbf{y}|\mathbf{z})\ \pi(\mathbf{z}|\boldsymbol{\beta}, \mathbf{X})\ \pi(\beta) && \text{por definición} \\
&= \prod_{i=1}^{n} [I(y_i = 1)I(z_i > 0) + I(y_i = 0)I(z_i \leq 0)] \\
&\qquad \times \phi(z_i|\eta(\mathbf{x}_i), 1) \times \pi(\boldsymbol{\beta}). && (3.6)
\end{aligned}$$

Donde $\pi(\mathbf{y}|\mathbf{z})$ es la función de verosimilitud, $\phi(\cdot|\mu, \sigma^2)$ es la función de densidad de una variable aleatoria distribuida $\mathcal{N}(\cdot|\mu, \sigma^2)$ y $\pi(\boldsymbol{\beta})$ la densidad *a priori* de $\boldsymbol{\beta}$.

Bajo los fundamentos del muestreador de Gibbs, dado que muestrear de (3.6) es complejo, se busca derivar entonces las distribuciones condicionales de $\mathbf{z}$ y $\boldsymbol{\beta}$. Para $\boldsymbol{\beta}$, la densidad marginal condicional esta entonces dada por:

$$\begin{aligned}
\pi(\boldsymbol{\beta}|\mathbf{z}, \mathbf{y}, \mathbf{X}) &= \frac{\pi(\mathbf{z}, \boldsymbol{\beta}|\mathbf{y}, \mathbf{X})}{\pi(\mathbf{z})} && (3.7) \\
&= \frac{\pi(\mathbf{y}|\mathbf{z})\ \pi(\mathbf{z}|\boldsymbol{\beta}, \mathbf{X})\ \pi(\boldsymbol{\beta})}{\pi(\mathbf{y}, \mathbf{X})\ \pi(\mathbf{z})} \\
&= \frac{\pi(\mathbf{y}|\mathbf{z})}{\pi(\mathbf{y}, \mathbf{X})\ \pi(\mathbf{z})} \overset{C}{\nearrow} \times \pi(\mathbf{z}|\boldsymbol{\beta}, \mathbf{X})\ \pi(\boldsymbol{\beta}) && (3.8) \\
&= C\, \pi(\boldsymbol{\beta}) \prod_{i=1}^{n} \phi(z_i|\eta(\mathbf{x}_i), 1), && (3.9)
\end{aligned}$$

Esta expresión es la misma que se derivaría si se tuviera una regresión lineal bayesiana con z de regresor, es decir, el modelo $z_i = \boldsymbol{\beta}^t \widetilde{\boldsymbol{\psi}}_i(\mathbf{x}_i) + e_i$ con $e_i \sim \mathcal{N}(0, 1)$ y z_i

conocidas. De lo anterior, se observa la utlidad de la variable latente: convierte una clasificación probit a una regresión lineal, haciendo uso de las variables latentes $\mathbf{z}$ como el regresor lineal. Se hace notar que la ecuación (3.7) se toma de la definición de probabilidad condicional, y el paso de (3.8) a (3.9) se puede hacer ya que, al definir y como en la ecuación (2.1), sus representaciones son análogas y el cociente se desvanece, dejando únicamente la constante C que sale del término $\pi(\mathbf{y}, \mathbf{X})$.

Únicamente falta definir $\pi(\boldsymbol{\beta})$. En la práctica es común 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 aplicación que se busca, además, derivan en resultados cerrados. En particular, si se elige la distribución $\pi(\boldsymbol{\beta})$ como:

$$\boldsymbol{\beta} \sim \mathcal{N}_\lambda(\boldsymbol{\beta} \,|\, \mu_{\boldsymbol{\beta}}, \Sigma_{\boldsymbol{\beta}}), \tag{3.10}$$

con el hiper-parámetro de media $\boldsymbol{\mu}_{\boldsymbol{\beta}} \in \mathbb{R}^\lambda$ y la matriz de covarianza $\Sigma_{\boldsymbol{\beta}} \in \mathbb{R}^{\lambda \times \lambda}$. Sustituyendo (3.10) en (3.9) y usando resultados estándar de modelos lineales (Banerjee 2008), se deriva que la densidad marginal conjugada para los parámetros es:

$$\boldsymbol{\beta} \,|\, \mathbf{y}, \mathbf{z}, \mathbf{X} \sim \mathcal{N}_\lambda(\boldsymbol{\beta} \,|\, \mu^*_{\boldsymbol{\beta}}, \Sigma^*_{\boldsymbol{\beta}}), \tag{3.11}$$

donde,

$$\mu^*_{\boldsymbol{\beta}} = \Sigma^*_{\boldsymbol{\beta}} \times (\Sigma^{-1}_{\boldsymbol{\beta}} \mu_{\boldsymbol{\beta}} + \widetilde{\Psi}(\mathbf{X})^t \mathbf{z})$$
$$\Sigma^*_{\boldsymbol{\beta}} = \left[\Sigma^{-1}_{\boldsymbol{\beta}} + \widetilde{\Psi}(\mathbf{X})^t \widetilde{\Psi}(\mathbf{X})\right]^{-1}.$$

Esta distribución es conjugada pues preserva la estructura normal de los parámetros, es decir, tanto la distribución inicial como la distribución posterior de $\boldsymbol{\beta}$ son normales. Asimismo, es fácil simular de esta distribución usando cualquier software estadístico, calculando previamente la media y covarianza y dando un valor (o iteración) para $\mathbf{z}$.[12] Con base en Banerjee (2008), en el Apéndice B se hace un resumen de las distribuciones conjugadas y se completan algunos de los pasos de esta derivación.

Ahora, condicionar sobre $\mathbf{z}$ es más sencillo y la derivación resulta similar. Comenzando con la expresión (3.6) y re-ordenando términos se tiene:

$$\begin{aligned}
\pi(\mathbf{z} \,|\, \boldsymbol{\beta}, \mathbf{y}, \mathbf{X}) &= \frac{\pi(\mathbf{z}, \boldsymbol{\beta} \,|\, \mathbf{y}, \mathbf{X})}{\pi(\boldsymbol{\beta})} \\
&= \frac{\pi(\mathbf{y} \,|\, \mathbf{z})\ \pi(\mathbf{z} \,|\, \boldsymbol{\beta}, \mathbf{X})\ \pi(\boldsymbol{\beta})}{\pi(\mathbf{y}, \mathbf{X})\ \pi(\boldsymbol{\beta})} \\
&= \cancel{\frac{1}{\pi(\mathbf{y}, \mathbf{X})}}^{\,C} \ \pi(\mathbf{y} \,|\, \mathbf{z}) \times \pi(\mathbf{z} \,|\, \beta, \mathbf{X}) \\
&= C \prod_{i=1}^{n} [I(y_i = 1) I(z_i > 0) + I(y_i = 0) I(z_i \leq 0)] \\
&\qquad \times \phi(z_i \,|\, \eta(\mathbf{x}_i), 1). \qquad (3.12)
\end{aligned}$$

De donde se observa que cada z_i es independiente (por el teorema de factorización)

12. Se hace notar, que este estimador, es relativamente similar al estimador que se usa en una regresión *Ridge*, (Tibshirani 1996).

con con distribución normal truncada en 0, es decir $\forall i = 1, \ldots, n$:

$$
\begin{aligned}
z_i|y_i, \boldsymbol{\beta} &\sim \mathcal{N}(z_i|\boldsymbol{\beta}^t\widetilde{\boldsymbol{\psi}}_i(\mathbf{x}_i), 1)_{I(z_i>0)I(y_i=1)} \quad \text{truncamiento a la izquierda} \\
z_i|y_i, \boldsymbol{\beta} &\sim \mathcal{N}(z_i|\boldsymbol{\beta}^t\widetilde{\boldsymbol{\psi}}_i(\mathbf{x}_i), 1)_{I(z_i\leq 0)I(y_i=0)} \quad \text{truncamiento a la derecha.}
\end{aligned}
\tag{3.13}
$$

Estas distribuciones también son fáciles de simular usando los algoritmos de Devroye (1986).

3.3.1. Implementación algorítmica final

Finalmente, al haber definido todos componentes del modelo, este se puede presentar en su versión final y más completa (aunque más pesada en notación).[13]

13. Se recuerda que existe un compendio de notación al inicio de este trabajo.

Definición 3.4. El modelo *bpwpm* (final),[14] $\forall i = 1, \ldots, n$:

$$y_i = \begin{cases} 1 & \iff \quad z_i > 0 \\ 0 & \iff \quad z_i \leq 0 \end{cases} \tag{2.1}$$

$$z_i \,|\, \mathbf{x}_i \sim \mathcal{N}(z_i \,|\, \eta(\mathbf{x}_i), 1) \tag{2.2}$$

$$\eta(\mathbf{x}_i) = f_0 + f_1(x_{i,1}) + f_2(x_{i,2}) + \ldots + f_d(x_{i,d}) \tag{2.3}$$

$$f_j(x_{i,j}) = \sum_{l=1}^{N^*} \beta_{j,l} \Psi_l(x_{i,j}, \mathcal{P}_j) \qquad \forall j = 1, \ldots, d \tag{2.4}$$

$$= \sum_{\hat{\imath}=1}^{M-1} \beta_{j,\hat{\imath},0}\, x_{i,j}^{\hat{\imath}} + \sum_{\hat{\imath}=K}^{M-1} \sum_{\hat{\jmath}=1}^{J-1} \beta_{j,\hat{\imath},\hat{\jmath}}\, (x_{i,j} - \tau_{j,\hat{\jmath}})_+^{\hat{\imath}}. \tag{3.14}$$

con las restricciones: $M > K > 0$ y $J > 1$,

$$N^* = JM - K(J-1) - 1 \tag{3.15}$$

$$\boldsymbol{\beta} \sim \mathcal{N}_\lambda(\boldsymbol{\beta} \,|\, \boldsymbol{\mu}_{\boldsymbol{\beta}}, \Sigma_{\boldsymbol{\beta}}) \qquad (\lambda = 1 + d \times N^*) \tag{3.10}$$

La ecuación (3.14) no es más que la expansión (2.22) presentada en la página 39 sobre toda $x_{i,j}$. Asimismo, el modelo se puede presentar en su forma vectorial más

14. Aumentando sobre la definición 2.1

compacta:

$$y_i = \begin{cases} 1 & \iff \quad z_i > 0 \\ 0 & \iff \quad z_i \leq 0 \end{cases} \tag{2.1}$$

$$z_i \,|\, \mathbf{x}_i \sim \mathcal{N}(z_i \,|\, \eta(\mathbf{x}_i), 1) \tag{2.2}$$

$$\boldsymbol{\beta} \sim \mathcal{N}_\lambda(\boldsymbol{\beta} \,|\, \mu_{\boldsymbol{\beta}}, \Sigma_{\boldsymbol{\beta}}) \qquad (\lambda = 1 + d \times N^*) \tag{3.10}$$

$$\boldsymbol{\eta}(\mathbf{X}) = \widetilde{\Psi}(\mathbf{X})\boldsymbol{\beta} \tag{2.23}$$

De estas expresiones y juntandolo con el muestreador de Gibbs (3.5) definido por las distribuciones marginales de $\boldsymbol{\beta}$ y $\mathbf{z}$, (3.11) y (3.13) respectivamente, se presenta el algoritmo final en la págona 64. El valor inicial $\mathbf{z}^{(0)}$ en realidad no se tiene que proporcionar pues se simula dependiendo de $\mathbf{y}$ y $\beta^{(0)}$. Este valor inicial $\beta^{(0)}$ es arbitrario, pero se sugiere en Albert y Chib (1993) 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 inizializa los parámetros en ceros por defecto. En la primera iteración, se esparcen por el espacio y van convergiendo a la distribución límite en relativamente poco tiempo.

El código que se desarrolló es de dominio publico y está disponible en `https://github.com/PaoloLuciano/BPWPM2`. Asimismo, se desarrolló mucha funcionalidad adicional para visualizar e imprimir información de los posibles modelos. En el Apéndice C se hace un compendio de las funciones y una breve descripción de su uso.

Algoritmo 1: *Bayesian piece-wise polynomial model* (bpwpm)

Datos: $\mathbf{y}$, $\mathbf{X}$, M, J, K, N_{sim}, $\boldsymbol{\beta}^{(k)}$, $\boldsymbol{\mu_\beta}$ y $\Sigma_{\boldsymbol{\beta}}$

Resultado: Objeto que contiene las cadenas simuladas de $\boldsymbol{\beta}$

$N^* \leftarrow J \times M - K(J-1) - 1$

$\lambda \leftarrow 1 + d \times N$

$\mathcal{P} \leftarrow$ cálculo de la partición con base en cuantiles de probabilidad $1/J$ para toda covariable sobre $\mathcal{X}^d$

$\widetilde{\Psi} \leftarrow$ expansión de polinomios por partes, con base en $\mathbf{X}$, $\mathcal{P}$, M, J y K

$\Sigma^*_{\boldsymbol{\beta}} = \left[\Sigma^{-1}_{\boldsymbol{\beta}} + \widetilde{\Psi}^t\widetilde{\Psi}\right]^{-1}$

Inicializar un vector de tamaño λ que contendrá las las cadenas $\tilde{\boldsymbol{\beta}} \leftarrow \boldsymbol{\beta}^{(0)}$

para $k = 1, \ldots, N_{sim}$ **hacer**

  $\boldsymbol{\eta}^{(k)} \leftarrow \widetilde{\Psi}\boldsymbol{\beta}^{(k)}$

  Simular $\mathbf{z}^{(k)}$ dado $\mathbf{y}$ y $\boldsymbol{\eta}^{(k)}$ con distribuciones normales truncadas

  $\mu^{*(k)}_{\boldsymbol{\beta}} = \Sigma^*_{\boldsymbol{\beta}} \times (\Sigma^{-1}_{\boldsymbol{\beta}}\mu_{\boldsymbol{\beta}} + \widetilde{\Psi}^t\mathbf{z}^{(k)})$

  Simular $\boldsymbol{\beta}^{(k)}$ de una distribución normal con media $\mu^{*(k)}_{\boldsymbol{\beta}}$ y matriz de varianza $\Sigma^*_{\boldsymbol{\beta}}$

  $\tilde{\boldsymbol{\beta}} \leftarrow \tilde{\boldsymbol{\beta}} + \boldsymbol{\beta}^{(k)}$

fin

Ya que el modelo tiene muchos componentes y pasos intermedios, la figura 3.2 hace un resumen gráfico del algoritmo. El superíndice $^{(k)}$ denota el número de la iteración, $\widetilde{\Psi}$ denota la expansión en bases truncadas para los datos $\mathbf{X}$, definida por los parámetros fijos M, J y K y la partición $\mathcal{P}$ que contiene los nodos τ.[15] 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. Posteriormente, se calcula $\boldsymbol{\eta}^{(0)}(\mathbf{X}) = \widetilde{\Psi}(\mathbf{X})\boldsymbol{\beta}^{(0)}$ con lo que queda definida la simulación de $\mathbf{z}^{(0)}$ como variables aleatorias normales truncadas. Finalmente, se aumenta el contador contador en uno, se calcula $\mu_{\boldsymbol{\beta}}^{*(k)}$ y se simulan los parámetros $\boldsymbol{\beta}$ que tienen distribución normal condicionada en $\mathbf{z}$. En cada iteración los parámetros se guardan en un objeto que regresa la rutina.

15. La implementación computacional de $\widetilde{\Psi}$, se basa en el diagrama 2.2 de la página 36 y la expresión (2.24). La subrutina que realiza la expansión tiene el nombre de `calculate_Psi` en el paquete y está vectorizada para que su ejecución sea veloz.

Dado un valor inicial $\boldsymbol{\beta}^{(0)}$ y los parámetros $M,\ J$ y K, se itera $k = 0, 1, \ldots, N_{\text{sim}}$:

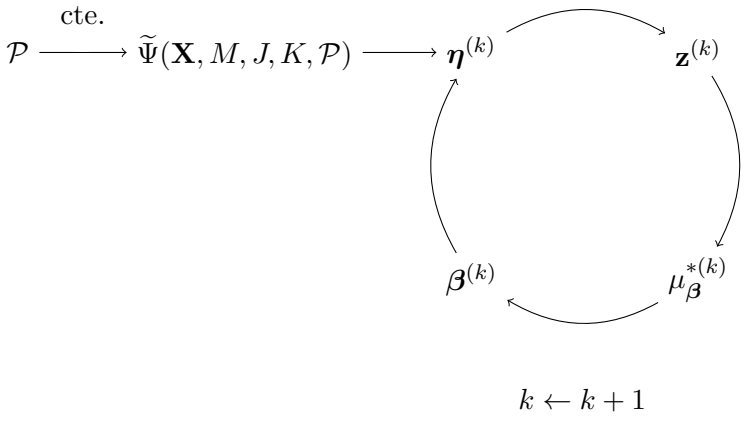


Figura 3.2: Esquema del algoritmo

Capítulo 4

Ejemplos y resultados

El modelo general presentado en este trabajo, aunque pesado en notación, resultó ser muy efectivo al llevarlo a la práctica. A lo largo de este capítulo, se hará una exploración intuitiva y visual de sus capacidades. Se remarca que todas las gráficas presentadas, se generaron con el mismo paquete `bpwpm` que realiza la estimación de los parámetros $\boldsymbol{\beta}$. Pues, los mismos objetos que las funciones devuelven, pueden ser utilizados para hacer gráficas que evalúan el modelo y reflejen la intuición subyacente.

Para mostrar los resultados y las capacidades del modelo se presentan seis ejemplos breves. Los primeros cinco, corresponden a bases de datos simuladas en dos dimensiones, es decir, se tienen dos covariables ($\mathbf{x}_i \in \mathbb{R}^2 \ \forall i$), con diferentes patrones para las respuestas $\mathbf{y}$ tanto lineales como no lineales. El objetivo, es poder visualizar lo

flexibles que son las fronteras de clasificación: la parte no lineal del modelo. Asimismo, al trabajar con bases de datos donde $\mathcal{X}^2 \subseteq \mathbb{R}^2$, se puede visualizar la función $\eta(\mathbf{x})$ en tres dimensiones. El último ejemplo corresponde a una base de datos médicos reales donde cada observación representa un tumor que pueden o no ser cancerígeno, las covariables representan ciertas características sobre este. Al aumentar la dimensionalidad, el modelo ya no es representable visualmente pero se siguen obteniendo buenos resultados.

A todos los modelos presentados a lo largo de este capítulo se les realizó un análisis de convergencia mirando las medias ergódicas de las cadenas. Sin embargo, únicamente se estudia a detalle para el ejemplo 2 de forma que no se saturara más la presentación.

4.1. Evaluación del modelo

Antes de poder presentar los ejemplos, se definen las dos métricas que se usarán para probar la efectividad (y precisión) de los modelos. Al trabajar con modelos de clasificación binaria, una forma intuitiva de medir su efectividad es a través de un simple conteo de *errores y aciertos*. Este conteo, se presenta en una *matriz de confusión* que desglosa la clasificación en sus respectivas categorías binarias. Asimismo, se presenta la función *log-loss* (ll) que no solo pondera la clasificación sino la *precisión* de esta, medida a través de las probabilidades ajustadas $\hat{p}_i$ que se le asigna a cada observación i.

Las matrices de confusión (tabla 4.1), son un método descriptivo con base en las tablas de contingencia que calcula la frecuencia de los aciertos y errores separando por grupos. Donde $\hat{y}$ es la variable predicha de la respuesta y y # el símbolo que denota *número*. Asimismo, se define la precisión del modelo como:

$$\text{precisión} = \frac{\text{Número de clasificacioens correctas}}{\text{Número total de observaciones}}$$

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

Tabla 4.1: Matriz de confusión

Sin embargo, la matriz de confusión resulta deficiente para comprar modelos completamente diferentes que resulten en la misma clasificación, por ello, se define una métrica más formal. Sea $\mathbf{y} = (y_1, \ldots, y_n)^t$ el vector de respuestas observadas y $\hat{\mathbf{p}} = (\hat{p_1}, \ldots, \hat{p_n})^t$ el vector de probabilidades ajustadas, donde $\hat{p}_i = \hat{P}_{\text{modelo}}(y_i = 1|\mathbf{x}_i)$ es la probabilidad estimada por el modelo de que la observación y_i sea igual a uno. Con lo anterior, se define un vector de respuestas ajustadas $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_n)^t$, haciendo la predicción en el corte $\hat{y}_i = 1 \iff \hat{p}_i > 0.5$.[1]

1. Este corte, es resultado de la simetría en cero de la función de acumulación normal estándar Φ, derivado de la ecuación (2.12).

Definición 4.1. La función *log-loss* $ll : \{0,1\}^n \times [0,1]^n \to \mathbb{R}^+$:

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

La ventaja de usar la función ll, es que resulta en una métrica que, no sólo mide que tan buena es la clasificación binaria, sino, que toma en cuenta la precisión de la predicción. Idealmente $ll = 0$ si se da una clasificación perfecta y conforme tome valores más positivos, el modelo realiza una peor predicción. Esto se debe a que 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. En la práctica y bajo un enfoque frecuentista, la función ll puede ser vista como una función de costos y más recientemente se ha utilizado para para entrenar y comparar modelos de clasificación como lo son las redes neuronales (Nielsen 2015).

4.2. Ejemplo 1 - las capacidades del modelo `bpwpm`

El primer ejemplo que se analizará, busca ejemplificar los componentes del modelo en general y sus capacidades. Para ello, 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 datos se muestrearon de dos distribuciones normales bivariadas:

$$\underset{i=1,\ldots,200}{\text{Grupo 0:}} \quad \mathbf{x}_i \sim \mathcal{N}_2 \left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \middle| \boldsymbol{\mu}_0 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \Sigma_0 = \begin{pmatrix} 0.25 & 0.35 \\ 0.35 & 1 \end{pmatrix} \right)$$

$$\underset{i=201,\ldots,350}{\text{Grupo 1:}} \quad \mathbf{x}_i \sim \mathcal{N}_2 \left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \middle| \boldsymbol{\mu}_1 = \begin{pmatrix} 4 \\ 4 \end{pmatrix}, \Sigma_1 = \begin{pmatrix} 1 & -0.24 \\ -0.24 & 0.64 \end{pmatrix} \right)$$

Las medias $\boldsymbol{\mu}_j$ $j = \{0, 1\}$ se toman relativamente alejadas y las covarianzas corresponden a las correlaciones $\rho_0 = 0.7$ y $\rho_1 = 0.3$ respectivamente. Estos parámetros de simulación se escogen a través de un proceso empírico resultando en una estructura simple donde los grupos están claramente separados y hay poco traslape. Asimismo, el espacio de covariables queda definido: $\mathcal{X}^2 \approx [0.3, 7.5] \times [-0.5, 5.9]$. Se codifica el grupo 0, $(y = 0)$, de color rojo y el grupo 1, $(y = 1)$, de color azul.[2] La base de datos final se presenta en la figura 4.1.

Tres realizaciones del modelo

El objetivo principal de esta simple base de datos es ejemplificar el tipo de fronteras alcanzables por η, mostrando una clara separación entre los dos grupos sin sobreajustar, para ello, se corren tres realizaciones del modelo. Para la primera se escoge el modelo más sencillo, una frontera lineal con un solo nodo (M = 2, J = 2 y K = 1). La segunda realización, consta de parábolas continuas más no suaves sobre cuatro

2. Se recomienda visualizar la versión digital de este trabajo donde se aprecian con más claridad los colores. Disponible en `https://github.com/PaoloLuciano/Tesis-Latex/`

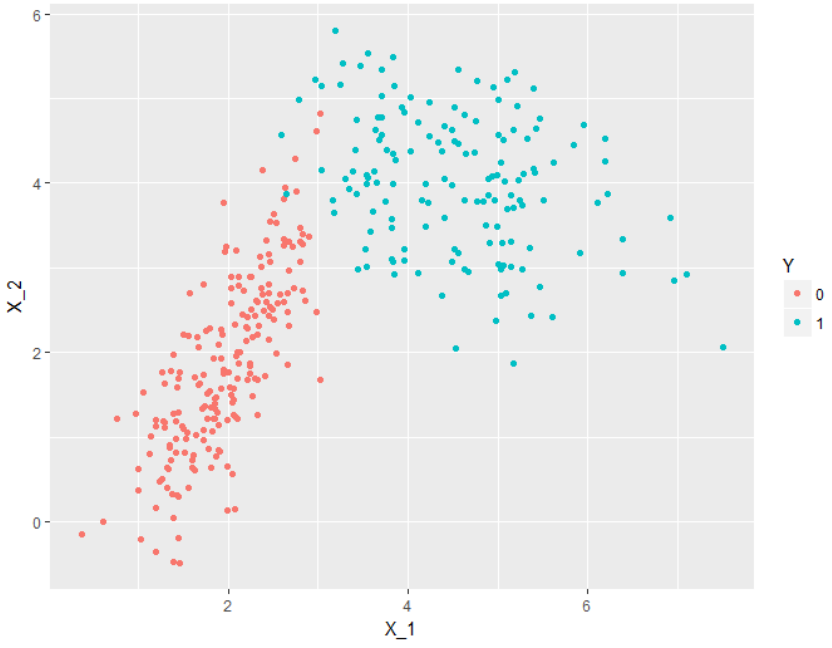


Figura 4.1: Ejemplo 1 - datos normales bivariados

nodos (M = 3, J = 5 y K = 1). Finalmente la tercera realización consta de splines cúbicos en 3 nodos (M = 4, J = 3 y K = 3).[3] En la tabla 4.2 se resume lo anterior.[4]

3. Polinomios por partes cúbicos suaves hasta la segunda derivada

4. Se recuerda que $M-1$ corresponde al grado de los polinomios, $J-1$ es el número de nodos, K el parámetro que controla la suavidad, $N^* = JM - K(J-1) - 1$ el número de funciones base (por expansión de cada covariable) y $\lambda = 1 + d * N^*$ el número total de parámetros.

Parámetro	Realización 1	Realización 2	Realización 3
M	2	3	4
J	2	5	3
K	1	1	3
N^*	2	10	5
λ	5	21	11

Tabla 4.2: Ejemplo 1 - tres realizaciones del modelo

Para las tres realizaciones se simularon $N_{\text{sim}} = 15{,}000$ valores de $\boldsymbol{\beta}$ y se opta por no usar periodo de *burn-in* ni suavizamiento para las cadenas, es decir: $k^* = 0$ y $k_{\text{thin}} = 0$, esto para hacer a las cadenas más comparables entre si. La única modificación que se realiza entre las tres realizaciones es que, para la tercera, se estandarizan las covariables.[5] Al usar polinomios de orden mayor, en este caso polinomios de tercer grado, el algoritmo puede caer en problemas numéricos pues $\hat{\eta}$ puede crecer muy rápido fuera de $\mathcal{X}^d$; se expande sobre este tema en el capitulo 5.

En las figuras 4.2, 4.3 y 4.4 se presentan imágenes que ejemplifican las tres realizaciones del modelo respectivamente. En las imágenes 4.2a, 4.3a y 4.4a se visualizan las diferentes tipos de fronteras que el modelo logra estimar. Con estas fronteras, se nota claramente como es determinante la elección de M, J y K en sus formas. El modelo logra estimar tanto fronteras relativamente rígidas (imágen 4.2a) como fronteras más suaves en las imágenes subsecuentes. Asimismo, para cada realiza-

5. Se resta la media y se divide entre la desviación estándar muestral de cada covariable.

ción, se tiene la representación en 3D de cada función $\hat{\eta}$ que preserva la suavidad (o no) de sus componentes. Rescatando las ideas de los GAM, se puede colapsar cada expansión de polinomios por partes en sus correspondientes $\hat{f}_j$ y visualizarla como la transformación no lineal de cada covariables. Por ejemplo, en la imagen 4.2c se observa que $\hat{f}_1(x_1)$ está compuesta por rectas que se conectan en el nodo, mientras que 4.4d representa $\hat{f}_2(x_2)$, un polinomio cúbico suave hasta la segunda derivada.

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

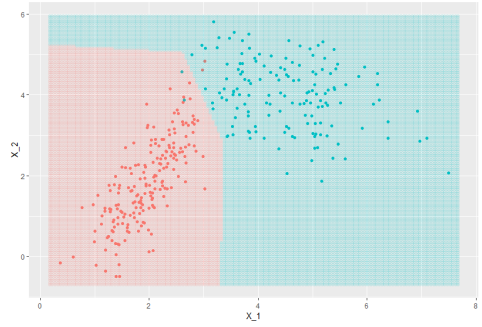
(a) Matriz de confusión para todas las realizaciones

Realización	*ll*
1	0.04088
2	0.03464
3	0.03498

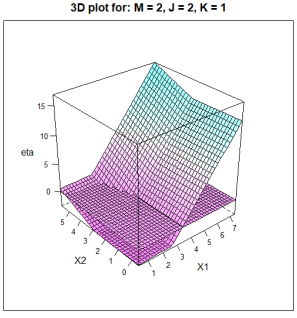
(b) *log-loss*

Tabla 4.3: Ejemplo 1 - resultados

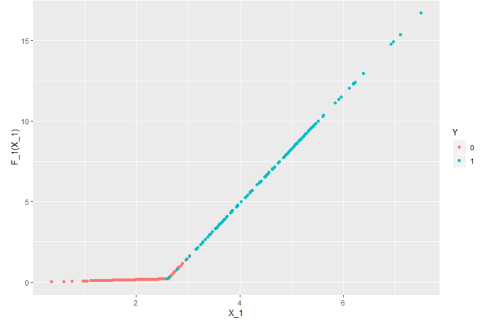
Al estar tratando con una base de datos tan sencilla, no es el enfoque comparar los resultados de estas realizaciones entre si pues las tres logran exactamente la misma clasificación desglosada en la tabla 4.3a. Al compartir la matriz de confusión, por ende, las realizaciones también comparten una precisión de 98.9 %. De la matriz y las imágenes, se observa que se clasifican de forma incorrecta solo cuatro observaciones. Sería inverosímil tratar de alcanzar una precisión del 100 % pues implicaría sobre-ajsutar el modelo. Para estas cuatro observaciones incorrectamente clasificadas, no se tiene la suficiente evidencia como para clasificarlas en su categoría contraría. Sin embargo, los modelos se pueden comparar más a fondo por medio de la métrica *ll* presentada en la tabla 4.3b. Aunque muy similares, la definición de la métrica indica que la realización dos es la mejor por un pequeño margen pues es la más cercana a cero. Claro está, bajo esta comparación no se toma en cuenta el número



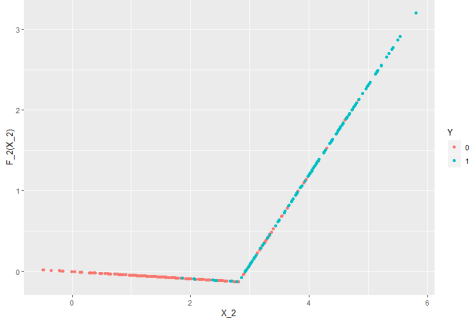
(a) Frontera de predicción



(b) Representación 3D de $\hat{\eta}$

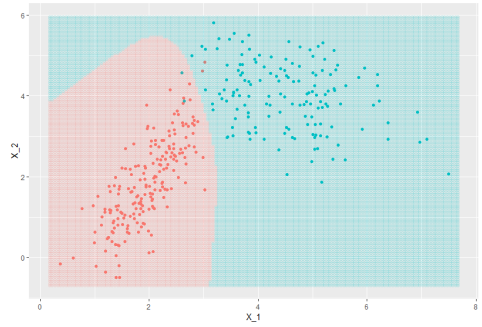


(c) $\hat{f}_1(x_1)$

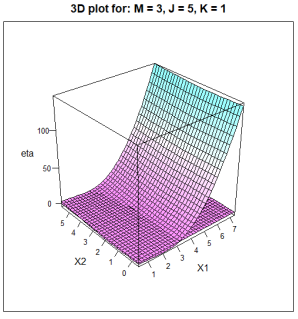


(d) $\hat{f}_2(x_2)$

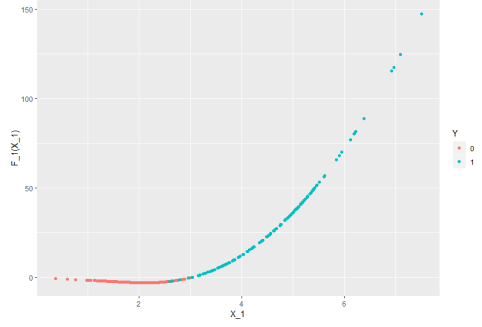
Figura 4.2: Realización 1 - fronteras lineales con un nodo (M = 2, J = 2 y K = 1)



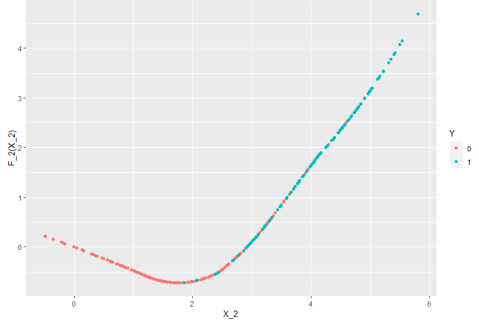
(a) Frontera de predicción



(b) Representación 3D de $\hat{\eta}$

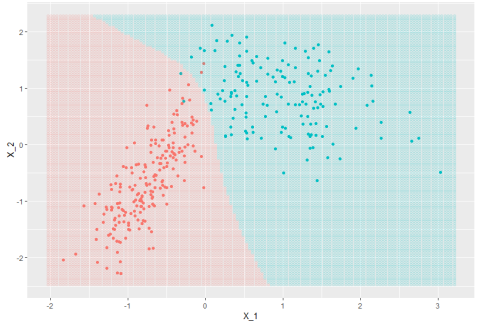


(c) $\hat{f}_1(x_1)$

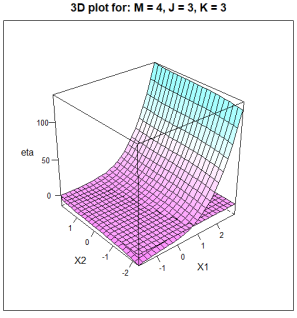


(d) $\hat{f}_2(x_2)$

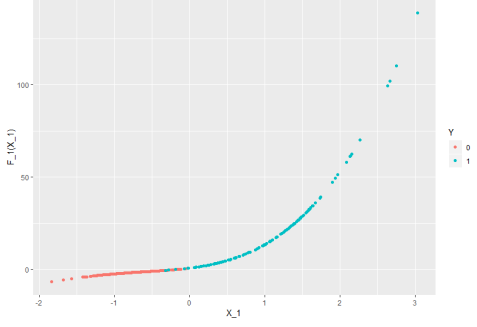
Figura 4.3: Realización 2 - parábolas continuas mas no suaves (M = 3, J = 5 y K = 1)



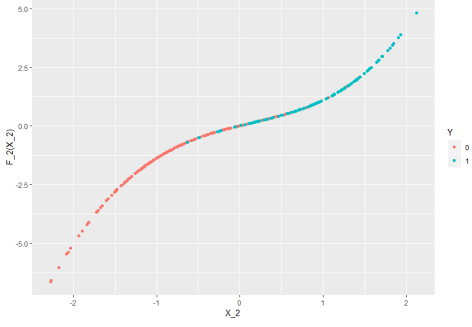
(a) Frontera de predicción



(b) Representación 3D de $\hat{\eta}$



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



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

Figura 4.4: Realización 3 - *splines* cúbicos (M = 4, J = 3 y K = 3)

de parámetros y la complejidad del modelo en si.

Dado que este es un ejemplo introductorio 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.[6] Cabe mencionar que para esta sencilla base de datos en particular, usar un modelo complejo como el *bpwpm* no es del todo necesario pues la base podría ser clasificada con la misma precisión por un modelo que use un predictor lineal en covariables. Sin embargo, se usa la base de datos para ejemplificar los tipos de fronteras flexibles. Asimismo, presentar las formas funcionales que toman las funciones f_j tampoco aportaría mucho pues están compuestas de muchos términos aditivos que no vale la pena desglosar.

4.3. Ejemplo 2 - comparación contra un GLM

Aprovechando la familiaridad de la base de datos anterior, se decide modificarla para que existan dos regiones de clasificación separadas. Se tomaron aproximadamente trece puntos, más allá de $x_1 \approx 5.5$ y se cambia su clasificación. En la imagen 4.6a se presenta esta base de datos modificada.

Con afán de comparar las predicciones del modelo *bpwpm* presentado en este trabajo contra uno más convencional, primeramente se corre un modelo probit lineal en

6. El efecto que esto puede tener es que se sobre-ajuste o se hagan predicciones demasiado acertadas. De cualquier forma el paquete incluye funcionalidad como para permitir un entrenamiento previo y una predicción *fuera de muestra* (*out-of-sample).*

covariables. Es decir, se estiman los parámetros $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)^t$ del modelo:[7]

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

De donde se obtienen los resultados presentados en la tabla 4.4 y la figura 4.5. De la imagen anterior, todo lo que quede por arriba de recta será clasificado como uno y todo lo que quede por debajo como cero.

Figura 4.5: Frontera de predicción para modelo probit lineal en covariables

7. La estimación se realiza bajo el paradigma frecuentista usando el método de mínimos cuadrados a través de la función `glm(..., family = binomial(link = 'probit'))` en `R`.

Parámetro	Estimado
$\hat{\beta}_0$	-4.67
$\hat{\beta}_1$	0.45
$\hat{\beta}_2$	0.91

Info. predicción	
Est. Puntual	No aplica
Precisión	90 %
log-loss	0.28072

	$\hat{y}=0$	$\hat{y}=1$	
$y=0$	194	19	213
$y=1$	16	121	137
	210	140	350

Tabla 4.4: Resultados para modelo probit lineal

Por ende, el modelo resultante es:

$$\Phi^{-1}(\hat{p}_i) = \hat{\eta}(\mathbf{x}) = -4.67 + 0.45x_{i,1} + 0.91x_{i,2}, \tag{4.3}$$

de donde se puede obtener explicitamente la ecuación de la frontera de predicción igualando (4.3) a 0.5, es decir:

$$\begin{aligned} \Phi(\hat{\eta}(\mathbf{x}_i)) \equiv 0.5 \qquad &\iff \\ \hat{\eta}(\mathbf{x}_i) = 0 \qquad &\iff \\ 0.45x_{i,1} + 0.91x_{i,2} = 4.67 \end{aligned} \tag{4.4}$$

Ahora, se corre el modelo *bpwpm* especificando $M = 3, J = 3$ y $K = 2$ (resumen en la tabla 4.5). Para este ejemplo, se opta por analizar a fondo todos los componentes y hacer un análisis más detallado de su convergencia, por lo tanto, se presentan los resultados de los estimadores en la tabla 4.6 e imágenes generadas en la figura 4.6.

Parámetros		Parámetro Sim.
$M = 3$	$N^* = 4$	$N_{\text{sim}} = 10{,}000$
$J = 3$	$\lambda = 9$	$k^* = 7{,}500$
$K = 2$	$n = 350$	$k_{\text{thin}} = 0$

Tabla 4.5: Ejemplo 2 - regiones disjuntas de clasifcación

Juntando todo, el modelo final como forma funcional a:

$$\begin{aligned}\Phi^{-1}(\hat{p}_i) = \hat{\eta}(\mathbf{x}) &= \hat{f}_0 + \hat{f}_1(x_{i,1}) + \hat{f}_2(x_{i,2}) \\ &= \overbrace{\hat{\beta}_0}^{\hat{f}_0} \\ &+ \overbrace{\hat{\beta}_1 x_{i,1} + \hat{\beta}_2 x_{i,1}^2 + \hat{\beta}_3 (x_{i,1} - \hat{\tau}_{1,1})_+^2 + \hat{\beta}_4 (x_{i,1} - \hat{\tau}_{1,2})_+^2}^{\hat{f}_1(x_{i,1})} \\ &+ \overbrace{\hat{\beta}_5 x_{i,2} + \hat{\beta}_6 x_{i,2}^2 + \hat{\beta}_7 (x_{i,2} - \hat{\tau}_{2,1})_+^2 + \hat{\beta}_8 (x_{i,2} - \hat{\tau}_{2,2})_+^2}^{\hat{f}_2(x_{i,2})} \quad ,\end{aligned} \tag{4.5}$$

la cual queda perfectamente definida si se sustituyen los valores de $\boldsymbol{\beta}$ y nodos contenidos en $\mathcal{P}$ presentados en la tabla 4.6. La ecuación (4.5) permite observar la expansión en bases final de $\eta(\cdot)$ para esta realización y elección de parámetros. Asimismo, en esta expansión se observa su forma aditiva y el desglose de las funciones $\hat{f}_j$. Es necesario remarcar que la transformación que realizan las funciones no lineales f_j, se observa contrastando las imágenes y 4.6b.

Es decir, el espacio inicial de covariables $\mathcal{X}^2$ (imagen 4.6a) tiene una forma que no puede ser separable por una frontera lineal. Sin embargo al llevar a cabo la

transformación no lineal de esta base (imagen 4.6a), se deriva en un espacio que si puede ser separable por una recta. En consecuencia, la frontera de clasificación es disjunta, pues el modelo identifica dos regiones donde los datos deben ser clasificados como cero (imágenes 4.6c y 4.6d). 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 esta base de datos en particular, se debe usar un nodo adicional cerca de la segunda región, ya que la curvatura, deriva de él. Asimismo, la suavidad de las funciones $\hat{f}$, deriva de la elección de K.

Contrastando los resultados del modelo probit lineal (tabla 4.4) contra el modelo *bpwpm* (tabla 4.6), se observa que se tiene una mejora en precisión sustancial pues se enfatiza que la flexibilidad en la frontera viene derivada del número relativamente grande de parámetros. Uno de los beneficios es que para el modelo probit lineal, esta frontera se puede derivar de forma explicita la ecuación (4.4), mientras que para el modelo *bpwpm* implicaría resolver numéricamente la ecuación derivada de la expresión no lineal (4.5). Asimismo, al compara la métrica *log-loss*, se observa que se tiene una mejora importante. En cuanto a tiempo de estimación computacional, no se tiene una diferencia significativa entre los dos modelos.

Info. predicción	
Est. Puntual	Media posterior
Precisión	98.6 %
log-loss	0.04505

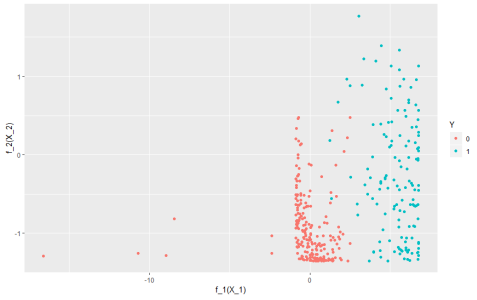
	$\hat{y}=0$	$\hat{y}=1$	
$y=0$	210	2	200
$y=1$	2	135	137
	212	138	350

$\boldsymbol{\beta}$	Valor	
$\hat{\beta}_0$	-2.03	$\hat{f}_0$
$\hat{\beta}_1$	-1.74	$\hat{f}_1(x_{i,1})$
$\hat{\beta}_2$	0.90	
$\hat{\beta}_3$	-0.07	
$\hat{\beta}_4$	-3.68	
$\hat{\beta}_5$	-1.01	$\hat{f}_1(x_{i,1})$
$\hat{\beta}_6$	0.13	
$\hat{\beta}_7$	0.31	
$\hat{\beta}_8$	-0.25	

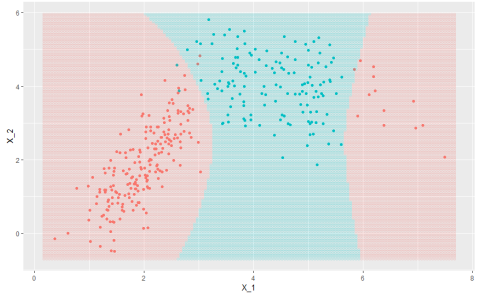
$\mathcal{P}$	Valor	
$\hat{\tau}_{1,1}$	2.07	Nodos
$\hat{\tau}_{1,2}$	3.69	
$\hat{\tau}_{2,1}$	2.00	
$\hat{\tau}_{2,2}$	3.52	

Tabla 4.6: Ejemplo 2 - resultados

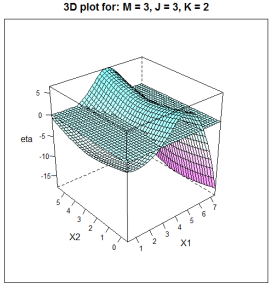
(a) Base del ejemplo 1 modificada



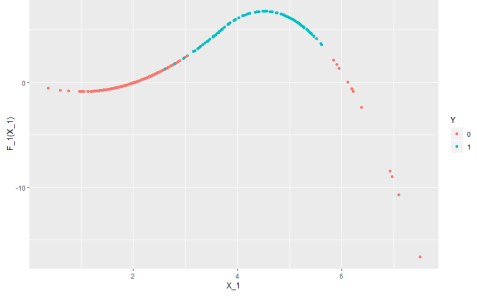
(b) Transformación no lineal



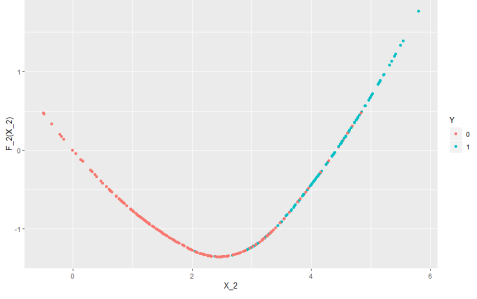
(c) Frontera de predicción



(d) Representación 3D de $\hat{\eta}$



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



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

Figura 4.6: Ejemplo 2 - regiones disjuntas de clasificación ($M = 3$, $J = 3$ y $K = 2$)

4.3.1. Análisis de convergencia

Para finalizar este ejemplo, se busca realizar un análisis detallado de la convergencia de las cadenas pues es parte fundamental del estudio de un modelo bayesiano. Por lo tanto en la tabla 4.7 se presentan resúmenes numéricos de los parámetros $\boldsymbol{\beta}$ y las cadenas completas en la figura 4.7.[8]

Al analizar las gráficas y los resúmenes, se nota como ciertos parámetros como $\hat{\beta}_4$ fluctúan mucho en su estimación en un comienzo, sin embargo de 4.7e se observa como el modelo converge a la larga. Asimismo, de los histogramas y trazas de las cadenas, se observa que estas tienden a estar bien formadas y presentan vagamente una distribución normal multivariada, estabilizandose conforme el algoritmo de muestreo Gibbs converge al espacio de probabilidad buscado. El periodo de burn-in, se escoge en $k* = 7,500$ pues se busca dar estimaciones puntuales de la media posterior lo más exactas posibles y pareciera que a partir de $k*$ se logra esto pues tanto las cadenas como la media ergódica no fluctúa demasiado. Si únicamente se mostraran los datos de las cadenas recordadas, los histogramas estarían perfectamente formados y tendrían una desviación estándar de aproximadamente uno por construcción.

Para este modelo flexible, aunque los parámetros no estén confundidos, existe la posibilidad de que algunos de ellos convergan a cero pues son innecesarios para la estimación, por ejemplo $\hat{\beta}_3$ y $\hat{\beta}_6$. Para identificar estos parámetros, se pueden aplicar

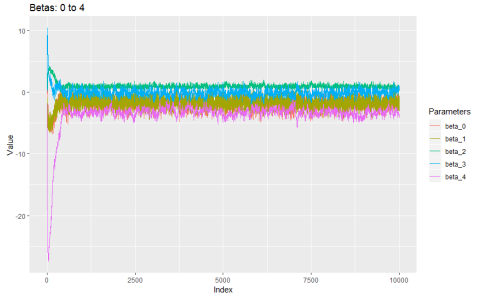
8. Tanto las imágenes como los resúmenes, aún no han sido ajustados por el periodo de burn-in, de ahí la disparidad contra las estimaciones puntuales de 4.6.

pruebas de hipótesis o procedimientos de selección de variables ya que se cuenta con toda la información que se tendría en un modelo tradicional; sin embargo, se enfatizan los resultados de predicción del modelo completo y no la interpretabilidad de los parámetros individuales.

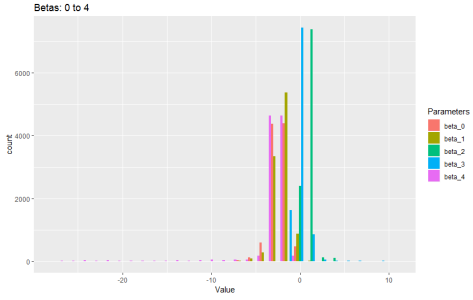
Métrica	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$
Mínimo	-6.79	-6.63	-0.39	-2.11	-27.40
Primer Cuartíl	-2.54	-2.22	0.66	-0.48	-3.72
Media	-2.03	-1.73	0.90	-0.07	-3.68
Mediana	-1.98	-1.69	0.85	-0.09	-3.28
Tercer Cuartíl	-1.44	-1.17	1.05	0.27	-2.86
Máximo	0.85	1.56	4.21	10.38	-1.07
Desviación Estandar	0.89	0.87	0.45	0.72	2.61

Métrica	$\hat{\beta}_5$	$\hat{\beta}_6$	$\hat{\beta}_7$	$\hat{\beta}_8$
Mínimo	-5.59	-1.64	-1.89	-2.47
Primer Cuartíl	-1.49	-0.04	-0.05	-0.69
Media	-1.00	0.13	0.31	-0.21
Mediana	-0.97	0.13	0.27	-0.27
Tercer Cuartíl	-0.44	0.31	0.63	0.13
Máximo	2.44	1.47	6.67	11.0
Desviación Estandar	0.87	0.28	0.60	0.94

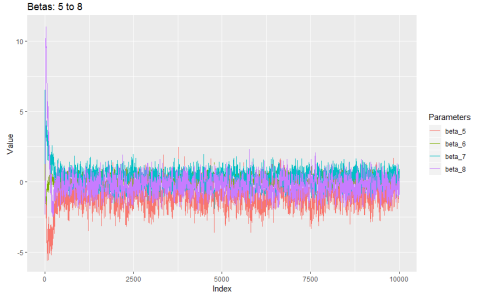
Tabla 4.7: Resúmenes numéricos para las cadenas de $\boldsymbol{\beta}$



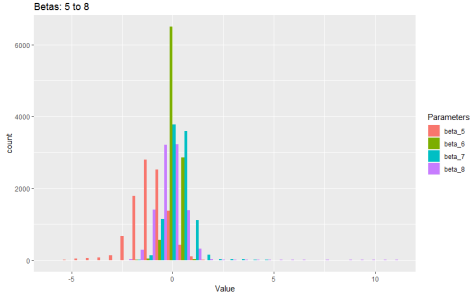
(a) Trazas de $\hat{\beta}_0$ a $\hat{\beta}_4$



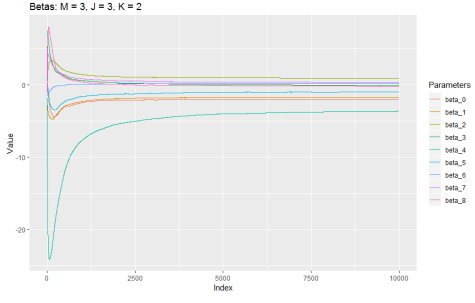
(b) Histogramas de $\hat{\beta}_0$ a $\hat{\beta}_4$



(c) Trazas de $\hat{\beta}_5$ a $\hat{\beta}_8$



(d) Histogramas de $\hat{\beta}_5$ a $\hat{\beta}_8$



(e) Media ergódica

Figura 4.7: Ejemplo 2 - análisis de convergencia

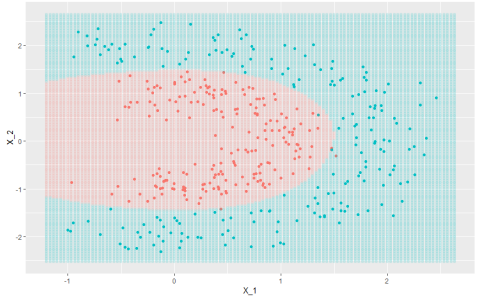
4.4. Ejemplos 3 a 5 - otros resultados interesantes

Los ejemplos presentados a continuación, son más expositivos que analíticos, es decir, se enfatizan los resultados más que los detalles matemáticos 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 y estresar las interacciones entre las dimensiones. Al estar tratando con regiones de clasificación más complejas, la predicción correcta sería imposible para un GLM.

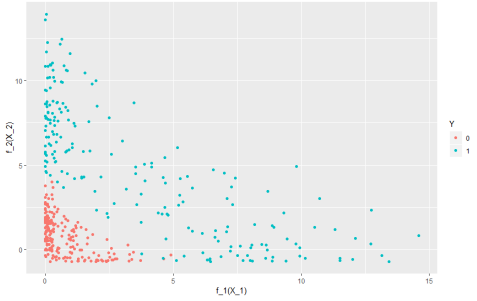
Ejemplo 3 - región parabólicos

Para este ejemplo, se generaron $n = 400$ datos en $\mathbb{R}^2$ usando coordenadas polares al tomar ángulos con un rango entre $[-1, 1]$. Posteriormente se tomaron diferentes radios para diferenciar cada grupo y finalmente se les sumó ruido blanco a los puntos para que existiera una región de confusión. Las simulación derivó en un patrón de datos cuya frontera es curva, casi parabólica. Dadas las características de los datos, se piensa que usar polinomios por partes parabólicos y suaves ($M = 3$ y $K = 2$) es una buena opción para modelarlos. Los parámetros escogidos para la realización final del modelo, se presentan en la tabla 4.8. Asimismo, los resultados e imágenes se presentan en la tabla 4.9 y la figura 4.8 respectivamente.

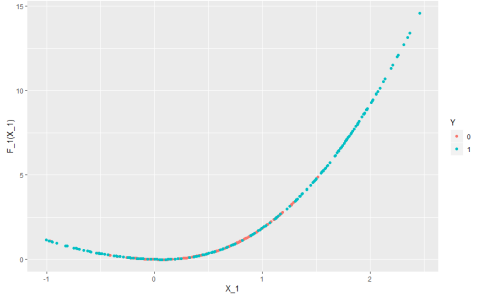
Esta es una realización particularmente interesante pues con un total $\lambda = 11$ parámetros se logra una precisión alta además de obtener convergencia relativamente rápido



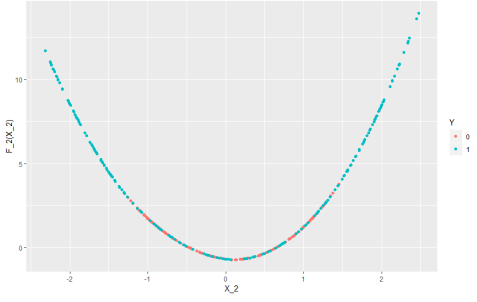
(a) Frontera



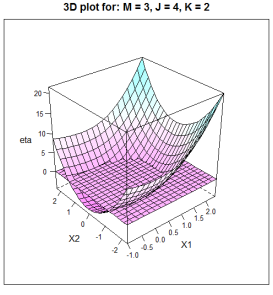
(b) Transformación no lineal



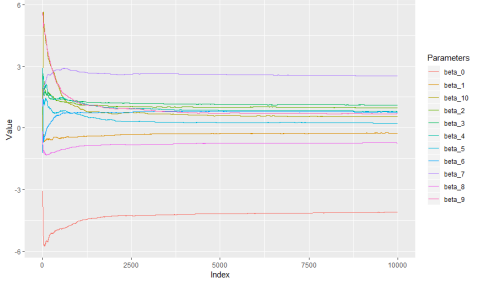
(c) $\hat{f}_1(x_1)$



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



(e) Representación 3D de $\hat{\eta}$



(f) Medias ergódicas

Figura 4.8: Ejemplo 3 - parábolas suaves ($M = 3$, $J = 4$ y $K = 2$)

Parámetros		Parámetro Sim.
$M = 3$	$N^* = 5$	$N_{\text{sim}} = 10{,}000$
$J = 4$	$\lambda = 11$	$k^* = 2{,}500$
$K = 2$	$n = 400$	$k_{\text{thin}} = 0$

Tabla 4.8: Ejemplo 3 - región parabólica

Info. predicción	
Est. Puntual	Media posterior
Precisión	99.2 %
log-loss	0.04352

	$\hat{y} = 0$	$\hat{y} = 1$	
$y = 0$	198	2	200
$y = 1$	1	199	200
	199	201	400

Tabla 4.9: Ejemplo 3 - resultados

($N_{\text{sim}} = 10{,}000$ y $k^* = 2{,}500$). Analizando el modelo de forma gráfica, se observa claramente que la segunda transformación $\hat{f}_2(x_2)$ (imagen 4.8d) captura la parte parabólica. A la vez, la primera transformación $\hat{f}_1(x_1)$ (imagen 4.8c) le da poco peso a la región donde hay confusión entre los los grupos pero posteriormente crece en donde hay certidumbre. Asimismo, se presenta el espacio de la transformación no lineal en la imagen 4.8b en donde se observa que el grupo rojo cero, se concentra en la esquina inferior izquierda, representando la posible separación lineal en este espacio transformado.

Ejemplo 4 - región ovalada

Para esta base de datos en particular se busca replicar algo similar a la imagen del capítulo introductorio 1.1 de la página 3. Se obtuvo una base de datos pequeña del curso en linea de ML de Ng (2018) que presenta una frontera de clasificación ovalada.[9] Esta base de datos se usa para entrenar modelos saturados logit con regularización, Hastie, Tibshirani y Friedman (2008), logrando predecir fronteras curvas con modelos tradicionalmente lineales al incluir interacciones de orden mayor entre covariables. Por lo tanto, se decidió probarlo también con el modelo presentado para contrastar.

El modelo una vez más, fue ajustado usando parábolas suaves las cuales resultaron ser excelentes herramientas. La parámetros escogidos para la realización final del modelo, se presenta en la tabla 4.10 con resultados e imágenes en la tabla **??** y figura 4.9 respectivamente.

Parámetros		Parámetro Sim.
$M = 3$	$N^* = 3$	$N_{\text{sim}} = 2{,}000$
$J = 2$	$\lambda = 7$	$k^* = 500$
$K = 2$	$n = 118$	$k_{\text{thin}} = 0$

Tabla 4.10: Ejemplo 4 - región ovalada

Para esta realización del modelo, se buscó estresar su flexibilidad al incluir el menor número de términos posibles usando un solo nodo ($\lambda = 7$ y $J = 2$) y cadenas

9. Este curso, se ofrece de forma gratuita en la plataforma de MOOC's Coursera.

Info. predicción	
Est. Puntual	Media posterior
Precisión	78.8 %
log-loss	0.4714

	$\hat{y}=0$	$\hat{y}=1$	
$y=0$	48	12	60
$y=1$	13	45	58
	61	57	118

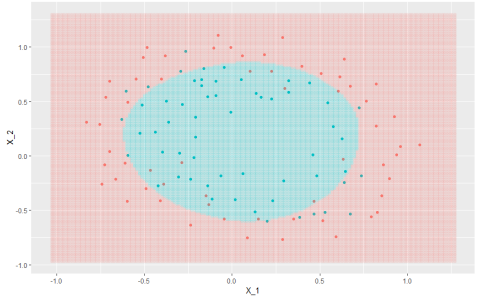
Tabla 4.11: Ejemplo 4 - resultados

cortas ($N_{\text{sim}} = 2{,}000$). Aunque una precisión de 78.8 % no resulte tan atractiva, es la precisión que se presenta en el curso en línea y permanece constante aún si se aumenta λ. La métrica *ll* mejora (marginalmente) sobre la presentada en el curso (≈ 0.5), sin embargo, se logra una reducción significativa en el número de parámetros pues el modelo saurado de Ng (2018) inicia con 28 parámetros.[10]. Asimismo, como se observa en la figura 4.9f las cadenas convergen rápidamente. Todo el poder del modelo, recae en la forma funcional de las funciones $\hat{f}_j$ al poder estimar regiones irregularmente curvas, con pocas observaciones y parámetros.

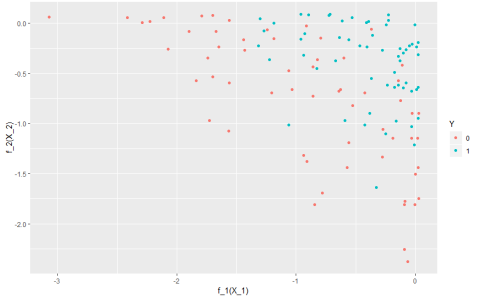
Ejemplo 5 - *yin-yang*, limitaciones del modelo

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 difícil de identificar por un algoritmo. Los datos tratan de simular un *yin-yang* que se puede observar en la figura 4.10a. La simple simulación de la base de datos representó un reto donde se conjuntaron varias áreas de la matemática aplicada. En el software

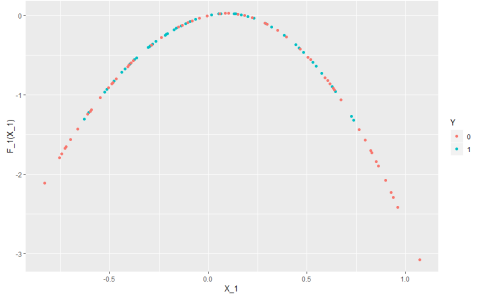
10. Dada la regularización, muchos de estos parámetros se desvanecían



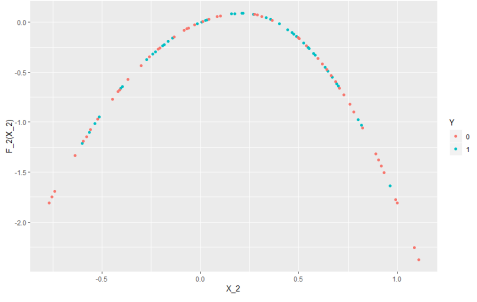
(a) Frontera



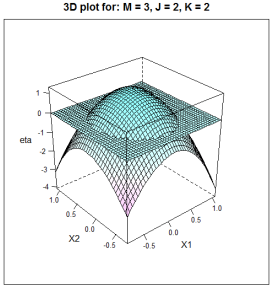
(b) Transformación no lineal



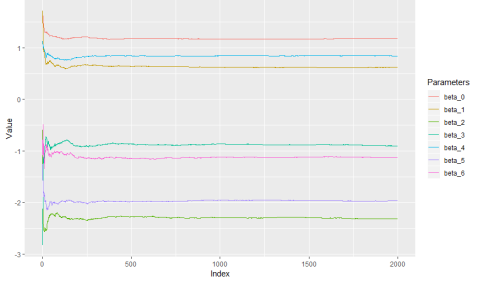
(c) $\hat{f}_1(x_1)$



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



(e) Representación 3D de $\hat{\eta}$

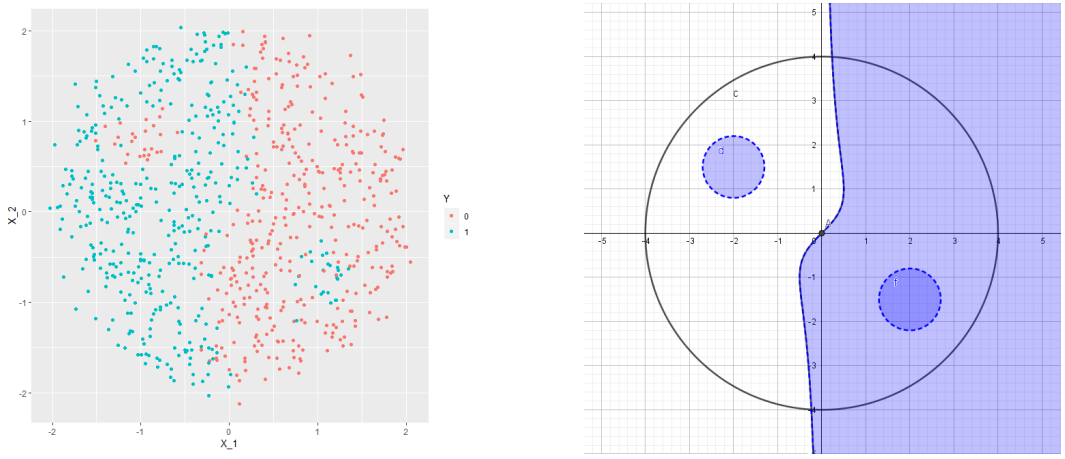


(f) Medias ergódicas

Figura 4.9: Ejemplo 4 - parábolas suaves en un nodos ($M = 3$, $J = 2$ y $K = 2$)

`GeoGebra`, se generó el diagrama presentado en la figura 4.10b que consiste de las siguientes desigualdades cartesianas:

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



(a) Datos simulados

(b) Salida del software `GeoGebra`

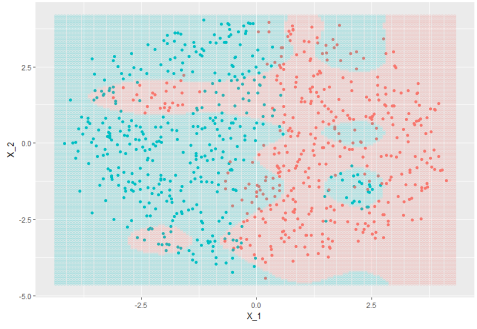
Figura 4.10: Ejemplo 5 - patrón yin-yang

Una vez dibujadas las ecuaciones, se generó una base de datos de aproximadamente $n \approx 800$ observaciones con una distribución uniforme dentro del círculo usando coordenadas polares. A estos puntos se les asignó la categoría cero, posteriormente se cambió la categoría a los puntos que cumplieran con las desigualdades. Después, se le añadió algo de ruido normal a cada punto para darle aleatoriedad a la base de

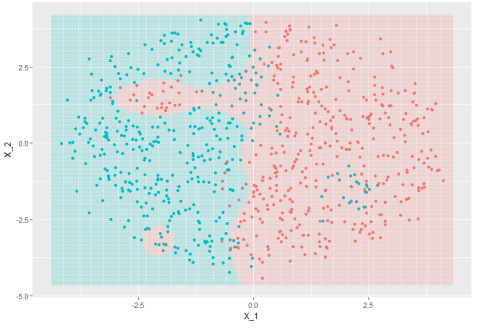
datos pero manteniendo el patrón y finalmente, se escala la base para centrarla en cero.

Se corrieron un sinfín de realizaciones del modelo, tratando de calibrar los parámetros M, J y K para captar de la mejor manera posible el patrón. Sin embargo y aunque el modelo casi siempre lograba una precisión de cerca de 85 %, no se logra la clasificación esperada identificando los puntos de color dentro de las áreas contrarias. De cualquier forma se observa como el algoritmo está tratando de encontrar este patrón. En la figura 4.11 se pueden ver fronteras de algunos de los mejores modelos.[11] Para las imágenes 4.11c y 4.11b, se observa como el modelo está tratando de encontrar las regiones anidadas, sin embargo, nunca se logra de forma precisa. Finalmente la imagen 4.11d, muestra una, de las muchas representaciones 3D que se hicieron al tratar de ajustar esta base de datos. Precisamente en esta última imagen 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 malla cuadriculada 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

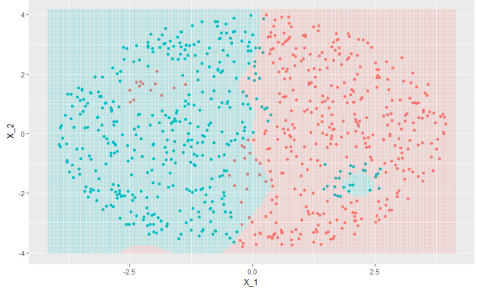
11. En la imagen 4.11a, el modelo detecta relativamente bien la curva que separa las regiones y detecta de forma aislada el circulo azul de la esquina inferior derecha.



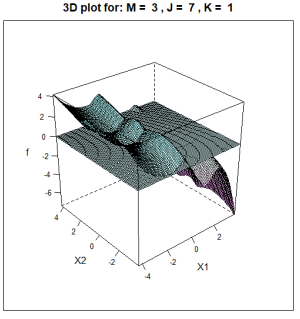
(a) Sobre-ajuste



(b) Mejor modelo



(c) Falta de precisión



(d) Gráfico 3D para uno de los modelos

Figura 4.11: Fronteras de varios modelos para datos yin-yang

tuvieron que correr con un número grande de nodos $J \approx 20$, derivando en un número de parámetros aún mayor.

4.5. Ejemplo 6 - el modelo en la práctica

Hasta ahora, todos los resultados de este trabajo han sido sobre bases de datos simuladas. Claramente se forman imágenes atractivas por construcción, sin embargo, no se está prediciendo nada en realidad pues se utiliza una metodología *dentro de muestra* para enfatizar las posibles fronteras del modelo. Por lo tanto y como último ejemplo, se presenta la base de datos de cáncer de mama de la Universidad 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) y Bennett y Mangasarian (1992).

De manera general y sin entrar en el detalle biológico de las variables como tal, se presenta un análisis exploratorio preliminar que se lleva a cabo para seleccionar, de forma completamente subjetiva, 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 (etiqueta cero). Se cuenta con diez variables (dimensiones) médicas sobre las características de los tumores como lo son: el tamaño, la uniformidad de la pared celular y la cromatina.[12]

12. Forma en la que se presenta la cadena de ADN en el núcleo celular.

En la figura 4.12, se muestran los gráficos de puntos pareados para todas las posibles combinaciones de covariables además de cierta información adicional. Se hace notar

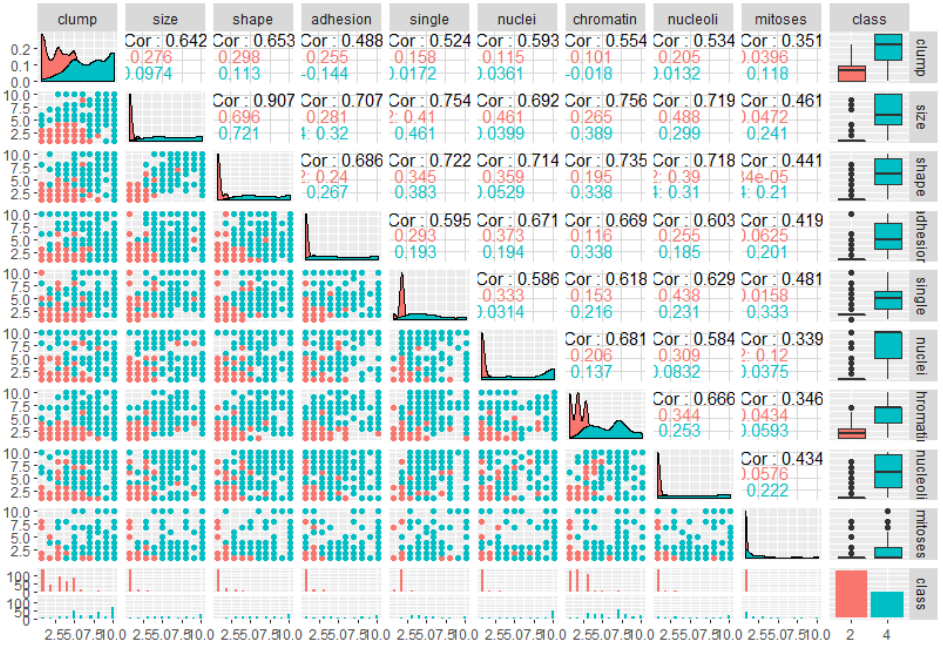
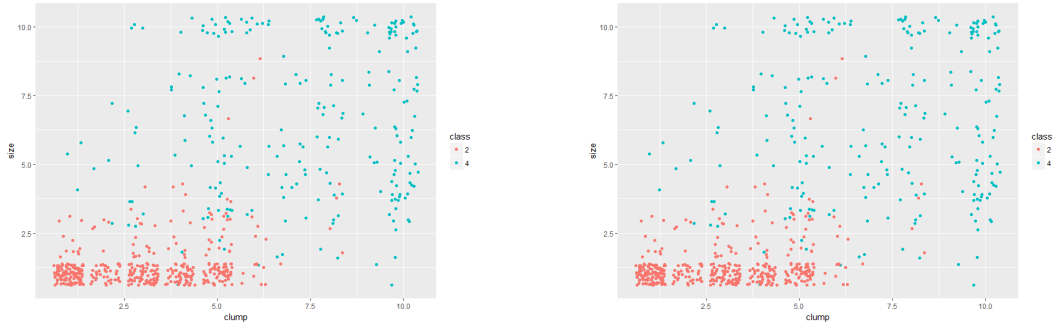


Figura 4.12: Análisis exploratorio para selección de variables

que las covariables 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. Derivado de esta exploración previa, se seleccionan las covariables *clump, size* y *chromatin* debido a que parecieran ser las que mejor separan el espacio.[13] En la figura 4.13 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

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

en una primera exploración, además se tienen puntos idénticos con clasificaciones contrarias. Sin embargo, a simple vista se detecta cierto patrón en los datos.



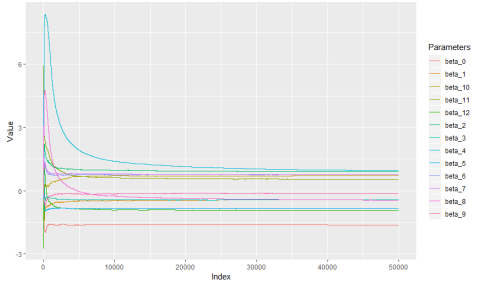
(a) Variables *clump* y *chromatin*

(b) Variables *clump* y *size*

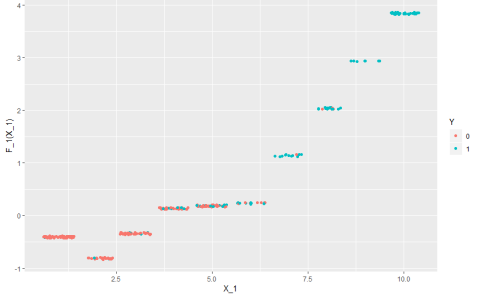
Figura 4.13: Gráficos de puntos con ruido para separar las observaciones

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 parte en dos: un conjunto de entrenamiento con el 60 % de las observaciones ($n_{\text{train}} = 409$) y un conjunto de prueba con las observaciones restantes ($n_{\text{test}} = 274$) sobre las que se evaluará el modelo.[14] La realización final de entrenamiento del modelo se resume en la tabla 4.12, se escogen segmentos de recta continuos sobre tres nodos. Los resultados numéricos sobre la base de datos de prueba se presentan en la tabla 4.13 y el análisis de convergencia a través de las medias ergódicas en la figura 4.14a. Asimismo, se presenta cada $\hat{f}_j \quad j = 1, 2, 3$ en las figuras 4.14b, 4.14c y 4.14d respectivamente.

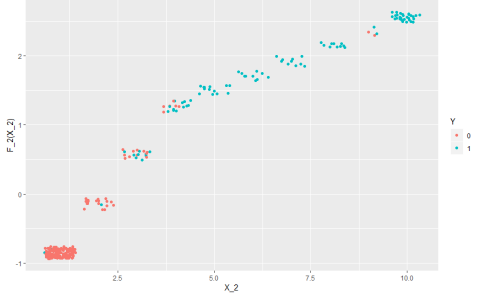
14. La diferencia de 16 observaciones entre la suma de entrenamiento y prueba, contra las 699 originales, se debe a que estas estaban incompletas y por lo tanto se descartan.



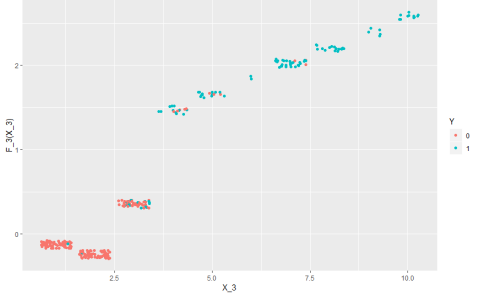
(a) Medias ergódica



(b) $\hat{f}_1(x_1)$ con ruido



(c) $\hat{f}_2(x_2)$ con ruido



(d) $\hat{f}_3(x_3)$ con ruido

Figura 4.14: Media ergódica y funciones $\hat{f}_j(x_j) \quad j = 1, 2, 3$

Parámetros		Parámetro Sim.
$M = 2$	$N^* = 4$	$N_{\text{sim}} = 50{,}000$
$J = 4$	$\lambda = 13$	$k^* = 10{,}000$
$K = 1$	$n = 409$	$k_{\text{thin}} = 0$

Tabla 4.12: Ejemplo 6 - datos médicos reales

Info. predicción	
Est. Puntual	Media posterior
Precisión	95.6 %
log-loss	0.1561

	$\hat{y} = 0$	$\hat{y} = 1$	
$y = 0$	169	9	178
$y = 1$	3	93	96
	172	102	274

Tabla 4.13: Ejemplo 6 - resultados

Haciendo una predicción fuera de muestra los resultados son buenos logrando una precisión del 95.6 %. Asimismo, se resalta que inclusive en dimensiones ($d = 3$) más altas si se escogen los parámetros adecuados M, J y K, el número total de parámetros ($\lambda = 13$) no necesita aumentar demasiado para lograr una buena separación del espacio.[15] Sin embargo, derivado también del número de covariables es que no se pueden hacer una visualización en el plano cartesiano $\mathbb{R}^2$ como en los ejemplos anteriores. No obstante, la convergencia es clara y los resultados buenos, incluso usando segmentos de recta y un número de nodos pequeño. Se hace notar que la codificación de las covariables usando una escala de diez puntos no es óptima para un modelo que se construye pensando en un espacio real de covaraibles $\mathcal{X}^d$, sin embargo, no parece afectar la estimación de los parámetros.

Capítulo 5

Conclusiones

El desarrollo de un modelo de aprendizaje de máquina, derivar en el estudio y aplicación de múltiples áreas de la estadística y las matemáticas. Es interesante el reto que representa el entendimiento de un modelo tan estructurado como el presentado en este trabajo y es gratificante observar los buenos resultados. Sin embargo, el modelo se puede mejorar significativamente, además de que restan algunos temas en los que vale la pena profundizar. Este capítulo busca revisar las limitaciones y contratiempos que podrían surgir en la aplicación del modelo.

5.1. Consideraciones finales sobre el modelo

Convergencia y sus implicaciones

Lograr cadenas siempre convergentes, estables y que tengan la distribución posterior buscada es complejo por dos razones. Primeramente pues, aunque el muestreador de Gibbs garantice la ergodicidad, si se tienen regiones de baja probabilidad, el algoritmo podría tomar un tiempo casi infinito en alcanzarlas, Robert y Casella (2004). Segundo por el componente numérico del algoritmo, esto pues la mayoría de los métodos bayesianos de simulación dependen de generadores de números aleatorios, además, se podrían dar errores de estimación por problemas de desbordamiento binario. Por ejemplo, no es raro que el algoritmo caiga en errores de precisión de máquina al tener términos de orden mayor ($M >> 0$) que crecen rápidamente. Sin embargo, todos los modelos presentados en este trabajo son replicables al fijar la semilla del algoritmo generador, e incluso, si este se cambia, los parámetros siguen convergiendo a los valores puntuales presentados, lo cual indica que efectivamente existe un patrón que el modelo está encontrando y los resultados no se debe a la aleatoriedad del algoritmo.

Calibración de los parámetros y velocidad del algoritmo

Aunque se podría pensar que fijar M, J y K a discreción del estadista sesga los resultados, en realidad es sólo una consecuencia de haber escogido un modelo tan

estructurado como se hizo. Prácticamente en ningún modelo estadístico, incluso en los no paramétricos, se puede abandonar toda decisión al algoritmo para que este encuentre el modelo perfecto. Siempre existirá un parámetro o variable que se deba de afinar lo cual introduce una dimensión subjetiva al modelo, Wasserman (2007). Inclusive, la misma selección del modelo introduce variabilidad no sistemática en los datos lo cual podría sesgar los resultados. Sin embargo, en el caso de los parámetros del modelo, existe un proceso de calibración que se puede realizar de forma analítica y no por fuerza bruta. Siempre se busca entender el porqué esa selección puntual de parámetros no funciona y así, modificar el modelo en respuesta. En particular para los ejemplos presentados la selección de M, J y K, para los casos sencillos, era prácticamente trivial y el modelo lograba capturar el patrón con diferentes tipos de fronteras. Como ejemplo se tienen las realizaciones del primer ejemplo de la sección 70.

Es curioso notar, que aunque el modelo sea complejo y pueda crecer rápidamente en el número de parámetros al modificar M, J y K o aumentar el número de covariables d, la velocidad del algoritmo es relativamente 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 trivial. Prácticamente en todos los modelos probados, se terminaron de simular las cadenas en un minuto o menos. Aquello que hace que el algoritmo sea más lento, usualmente es escalar n ordenes de magnitud. Otro factor importante que influye en la velocidad del algoritmo es el uso de un paradigma bayesiano para 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 para este modelo, habría logrado que el algoritmo de estimación, fuera casi instantáneo para un número grande de parámetros.

Asimismo, gracias a la fácil disponibilidad y uso 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, tanto el modelo como el algoritmo se puede ir mejorando con contribuciones de terceros. Dado que el algoritmo se implementó en el software estadístico `R`, el cual tiene múltiples ventajas, se reconoce que no es el lenguaje más veloz computacionalmente pues corre a un nivel muy alto. Si se pensara usar el algoritmo para aplicaciones más robustas, se recomendaría adaptar las funciones a un lenguajes de nivel más bajo como lo puede ser `C++`.

5.2. Posibles mejoras y actualizaciones

Como se mencionó, la fuerza del modelo recae en la combinación de todos sus componentes pues le otorga flexibilidad a las estructuras contenidas en η. No obstante, como se estudió en el ejemplo 5, el modelo tiene limitantes que se pueden mejorar.

La primer y más urgente mejora que se propone explorar, es la de incorporación de un método para la selección de covariables. Bajo el enfoque de la estadística frecuentista para modelos de regresión, en ocasiones se trata de buscar aquellas

covariables más significativas para la predicción correcta 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, Bishop (2006).[1] Los métodos de aprendizaje de máquina más recientes son especialmente efectivos en este ámbito; sus algoritmos recaen en usar cantidades enormes de información con múltiples covariables ($d >> 1$) para hacer predicciones robustas al entrenar miles de parámetros, Nielsen (2015). Bajo el paradigma bayesiano la selección de covariables también se puede manear. Los métodos más usados, incorporan otra serie nueva serie de variables auxiliares (usualmente funciones indicadoras) cuyo trabajo es detectar cuando una variable es relevante o no. A estas variables, también se les da un tratamiento bayesiano y son estimadas por los mismos algoritmos MCMC a la par de todos los demás, O'Hara, Sillanpää y col. (2009).

Para este trabajo como se observa en el ejemplo 6, la selección de covariables se hizo de manera manual (y subjetiva) 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 covariables, no sólo se realiza en afán de simplificar los modelos, sino por una razón computacional de convergencia pues al no tener covariables adicionales los parámetros asociados a las variables relevantes serían más significativos. Asimismo, se podría reducir la colinealidad entre covariables.

1. Usualmente el criterio de Akaike

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 de muestreo pues la dimensión del número de parámetros fluctúa como se agregan o se eliminan nodos. 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.

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 (3.6). Esto permite que el muestreo sea sencillo, requiriendo únicamente álgebra lineal y simulaciones de variables con distribución normal multivariada. Aunque el supuesto de que $\boldsymbol{\beta} \sim \mathcal{N}_{d+1}$ 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, requeriría de cambiar sustancialmente 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`.

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 funciona muy bien, sin embargo, es computacionalmente lenta. Por ejemplo, si se incorporara el cambio en la posición de los nodos sería forzoso recalcular la matriz $\tilde{\Phi}$ múltiples veces. Haciendo un cambio de bases, se puede usar un conjunto de b-splines que representen exactamente el mismo polinomio pero se calculen más rápido. Asimismo, esta modificación permitiría incorporar *splines naturales* que son menos globales y no fluctúan tan rápido más allá de la frontera, Wahba (1990).

Estas capacidades adicionales, robustecerían en gran forma al modelo. Si se pensara en usarlo para aplicaciones a gran escala, con miles de datos y aplicaciones concretas, sería importante incorporarlas. Sin embargo, para efectos de este trabajo y para problemas menos trascendentes de clasificación binaria, estas consideraciones se puede obviar. Asimismo, para todos los ejemplos y bases de datos probadas, no resultaron en un contratiempo.

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, entre ellos los citados, que han expandido sobre los métodos tradicionales. 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,

por ejemplo Madan, Saluja y Zhao (2015) y Shah y Zhang (2014). Con este trabajo, además de desarrollar el modelo, se busca sembrar una base teórica y técnica de las posibles extensiones del aprendizaje de máquina, disciplina la cual no es más que estadística computacional llevada al límite.

Algunos de los métodos de aprendizaje de máquina, no son más que modelos GLM como el presentado, que se corre un gran número de veces sobre bases de datos con miles de observaciones, 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 de activación, después de colapsar dimensiones, rebasa cierto umbral. Este proceso se repite miles de veces logrando detectar patrones cada vez más complejos. Al final, fuera de las capacidades de estos modelos y su complejidad, la gran mayoría, son regresiones aumentadas que se basan en los mismos principios que el modelo presentado en este trabajo. Por lo mismo, valía hacer una exploración a fondo de uno modelo análogo y de autoria propia.

La fuerza que han adquirido los métodos de aprendizaje de máquina en los últimos años se debe a que han logrado romper con muchos de los paradigmas tradicionales. Esto pues se han comenzado a aplicar a datos poco tradicionales como lo podrían ser las imágenes y los sonidos; asimismo, han extendido las capacidades de predicción a conjuntos enormes de categorías. Su utilidad es tan grande que dispositivos de de uso diario, utilizan estos métodos y modelos para clasificar fotos, recopilar información

o entender el lenguaje hablado. Sin embargo, vale la pena recordar que *el que una computadora aprenda* es básicamente encontrar patrones, usualmente complejos y no lineales, en datos que expresan algún fenómeno de la realidad. Al final, los modelos estadísticos han sido, y siguen siendo, clave para el desarrollo de la ciencia y la tecnología. Por ello, se considera que es más vital que nunca poder entenderlos y analizarlos de forma correcta y bien fundamentada.

Apéndice A

Splines: orígenes y justificación de su uso

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 (**??**) se puede plantear como encontrar la función h que minimice:

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

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 (A.1) 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.[1]

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 (A.1) 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:

$$\sum_{i=1}^{n} h_j(x_{ij}) = 0 \quad \forall j \tag{A.2}$$

1. 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 $\mathcal{L}_2([a, b])$.

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 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*:

$$\hat{h}_{j^*} = y - h_0 - \sum_{\substack{j=1 \\ j \neq j^*}}^{d} h_j \tag{A.3}$$

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 **??**.

Apéndice B

Distribuciones conjugadas

Teorema B.1. *Dado el modelo bpwpm (definición 3.4, página 61) donde se tienen las siguientes distribuciones:*

$$z_i \,|\, \mathbf{x}_i \sim \mathcal{N}(z_i \,|\, \eta(\mathbf{x}_i), 1) \tag{2.2}$$

$$\boldsymbol{\beta} \sim \mathcal{N}_\lambda(\boldsymbol{\beta} \,|\, \boldsymbol{\mu}_{\boldsymbol{\beta}}, \Sigma_{\boldsymbol{\beta}}), \tag{3.10}$$

la distribución posterior de $\boldsymbol{\beta}$ *dado* $\mathbf{y}, \mathbf{z}$ *y* $\mathbf{X}$ *es conjugada. Es decir:*

$$\boldsymbol{\beta} \,|\, \mathbf{y}, \mathbf{z}, \mathbf{X} \sim \mathcal{N}_\lambda(\boldsymbol{\beta} \,|\, \mu^*_{\boldsymbol{\beta}}, \Sigma^*_{\boldsymbol{\beta}}), \tag{3.11}$$

con los parámetros actualizados,

$$\mu^*_{\boldsymbol{\beta}} = \Sigma^*_{\boldsymbol{\beta}} \times (\Sigma^{-1}_{\boldsymbol{\beta}} \mu_{\boldsymbol{\beta}} + \widetilde{\Psi}(\mathbf{X})^t \mathbf{z})$$
$$\Sigma^*_{\boldsymbol{\beta}} = \left[\Sigma^{-1}_{\boldsymbol{\beta}} + \widetilde{\Psi}(\mathbf{X})^t \widetilde{\Psi}(\mathbf{X})\right]^{-1}.$$

Demostración. Se retoma la derivación comenzada en la página 58.

$$\begin{aligned}\pi(\boldsymbol{\beta}|\mathbf{z}, \mathbf{y}, \mathbf{X}) &= \frac{\pi(\mathbf{z}, \boldsymbol{\beta}|\mathbf{y}, \mathbf{X})}{\pi(\mathbf{z})} \\ &= C\, \pi(\boldsymbol{\beta}) \prod_{i=1}^{n} \phi(z_i|\eta(\mathbf{x}_i), 1),\end{aligned}$$

utilizando un argumento de proporcionalidad sobre $\boldsymbol{\beta}$ y expandiendo la función de densidad de una variable aleatoria normal ϕ se tiene:

$$\propto \quad \pi(\boldsymbol{\beta}) \times \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(z_i - \eta(\mathbf{x}_i))^2\right\}.$$

Ahora, se sustituyendo la identidad (2.21), $\eta(\mathbf{x}_i) = \boldsymbol{\beta}^t \widetilde{\boldsymbol{\psi}}_i(\mathbf{x}_i)$ y se desglosa $\pi(\boldsymbol{\beta})$ con la correspondiente densidad normal multivariada,

$$\propto \quad \pi(\boldsymbol{\beta}) \times \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(z_i - \eta(\mathbf{x}_i))^2\right\}.$$

Q.E.D.

Apéndice C

Paquete en R: desarrollo y lista de funciones

Bibliografía

Albert, James H., y Siddhartha Chib. 1993. «Bayesian Analysis of Binary and Polychotomous Response Data». *Journal of the American Statistical Association:* 669-679.

Alpaydin, Ethem. 2014. *Introduction to Machine Learning.* MIT press.

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

Barber, David. 2010. *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, Albert R. 1985. «The Estimation of Nonparametric Functions in a Hilbert Space». *Econometric Theory* 1 (01): 7-26.

Bernardo, José M. 2003. *Bayesian Statistics. Encyclopedia of Life Support Systems (EOLSS). Probability and Statistics.*

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

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

Box, George E. P. 1976. «Science and Statistics». *Journal of the American Statistical Association* 71 (356): 791-799.

Breiman, Leo, Jerome Friedman, Charles J. Stone y Richard A. Olshen. 1984. *Classification and Regression Trees.* CRC press.

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

Cleveland, W. S., y S. J. Devlin. 1988. «Locally Weighted Regression: an Approach to Regression Analysis by Local fitting». *Journal of the American Statistical Association:* 596-610.

Denison, David G. T., Bani K. Mallick y Adrian F. M. 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 Random Variate Generation.* Volumen 4. Springer-Verlag New York.

Friedman, Jerome. 1991. «Multivariate Adaptive Regression Splines». *The Annals of Statistics:* 1-67.

Gelfand, Alan E., y Adrian 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 New York.

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

———. 1990. *Generalized Additive Models.* Chapman & Hall, London.

Hastie, Trevor, Robert Tibshirani y Jerome Friedman. 2008. *The Elements of Statistical Learning.* Springer, Series in Statistics.

Hoerl, Arthur E., y Robert W. Kennard. 1970. «Ridge Regression: Biased Rstimation for Nonorthogonal Problems». *Technometrics* 12 (1): 55-67.

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

Madan, Isaac, Shaurya Saluja y Aojia Zhao. 2015. «Automated bitcoin trading via machine learning algorithms». 20. `http://cs229.stanford.edu/proj2014/Isaac%20Madan,%20Shaurya%20Saluja,%20Aojia%20Zhao,Automated%20Bitcoin%20Trading%20via%20Machine%20Learning%20Algorithms.pdf`.

Mangasarian, Olvi L., Rudy Setiono y William H. Wolberg. 1990. «Pattern Recognition Via Linear Programming: Theory and Application to Medical Diagnosis». *Large-scale Numerical Optimization:* 22-31.

McCullagh, Peter, y John A. Nelder. 1989. *Generalized Linear Models.* Chapman & Hall, London.

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].

Nielsen, Michael A. 2015. *Neural Networks and Deep Learning.* Determination Press. `http://neuralnetworksanddeeplearning.com/`.

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, Sheldon M. 2014. *Introduction to Probability Models.* 11.ª edición. Academic Press.

Sanderson, Grant. 2017. *But what *is* a Neural Network? — Deep learning, Chapter 1.* `https://www.youtube.com/watch?v=aircAruvnKk`.

Schoenberg, Isaac 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.

Shah, Devavrat, y Kang Zhang. 2014. «Bayesian Regression and Bitcoin». En *Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on,* 409-414. IEEE.

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.

Tibshirani, Robert. 1996. «Regression Shrinkage and Selection via the Lasso». *Journal of the Royal Statistical Society: Series B (Methodological)* 58 (1): 267-288.

Tierney, Luke. 1994. «Markov Chains for Exploring Posterior Distributions». *The Annals of Statistics:* 1701-1728.

Wahba, Grace. 1990. *Spline Models for Observational Data.* Volumen 59. Society for Industrial & Applied Mathematics.

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