Índice general

Índice de figuras

Índice de tablas

Notación y abreviaciones

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, desarrollar y entender desde sus cimientos un modelo aplicable a esta categoría. El modelo, buscará hacer inferencia sobre una base de datos y *aprender* sobre los patrones subyacentes que estos puedan contener. Se busca profundizar en todos los aspectos de su desarrollo: consideraciones teóricas, paradigma de aprendizaje, implementación computacional y validación práctica.

Este tipo de modelos, han resultado ser de enorme efectividad en ámbitos tan diversos, como lo son la medicina y las finanzas. En ocasiones sin embargo, por su

1. *machine learning (ML)*

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

En particular, a continuación se presenta un modelo probit no lineal. El modelo probit es un tipo de regresión, que busca la predicción de variables de respuesta y_i binarias (éxito o fracaso, positivo o negativo, etc).[2] Esta predicción, depende de información contenida en las covariables $\mathbf{x}_i$ para cada una de las observaciones $i = 1, \ldots, n$. Sin embargo, esta información puede contener estructuras complejas que no son identificables por métodos lineales tradicionales, esto lleva a que la predicción de las respuestas y_i sea difícil. Para sobrepasar esto, al modelo se le agrega un componente no lineal en las covariables que permite discernir estos patrones. En el fondo, el modelo busca encontrar fronteras de segmentación tan *flexibles* como sean necesarias. En la Figura 1.1, se tiene un ejemplo gráfico de este tipo de modelos: se tienen observaciones del grupo azul y del grupo rojo con una clara separación no lineal en las covariables x_1 y x_2. El modelo busca *entrenar*, bajo el paradigma bayesiano, una función f (llamada predictor) que logre separar este espacio de la mejor forma posible. Esta separación, induce una clasificación binaria (0 y 1 correspondiendo a rojo y azul respectivamente) a través de la función de distribución normal

2. 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.

Φ. Con un modelo probit lineal, llevar a cabo esta clasificación sería imposible.

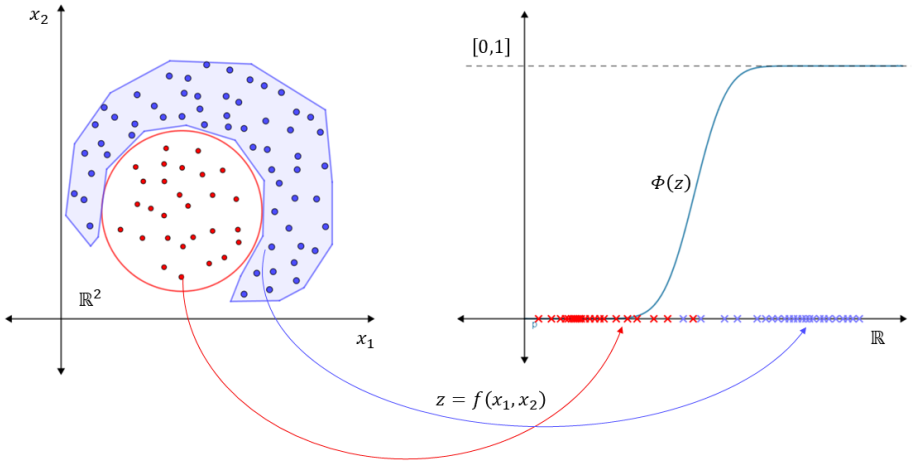


Figura 1.1: Diagrama explicativo de un modelo de clasificación probit no lineal

Para llevar a cabo la construcción, se comienza con una extensa discusión teórica y matemática en el Capítulo 2. Dada su estructura, el modelo se puede estudiar de *arriba hacia abajo*, es decir, de la parte más general a la parte más profunda. Por lo tanto, primero se estudian los Modelos Lineales Generalizados (GLM), específicamente los modelos probit asociados a la distribución normal. Los GLM dan el salto de una regresión donde la respuesta y_i es real, a regresiones donde la respuesta puede ser discreta o restringida a cierto dominio (MacCullagh y Nelder 1989). Los GLM, como su nombre lo indica, siguen siendo lineales en las covariables; sin embargo, se pueden flexibilizar usando las ideas de los Modelos Aditivos Generalizado (GAM) presentadas en Trevor Hastie y Robert Tibshirani (1986). En estos modelos,

la flexibilización se lleva a cabo transformando a las covariables $\mathbf{x}_i$, previamente a la regresión, mediante la función f usando métodos no paramétricos. Este trabajo, toma esas ideas y las combina con las de Denison, Mallick y Smith (1998) en las que se modifica la transformación antes mencionada al darle una forma funcional concreta a f, correspondiente a una serie de 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.

Desarrollado una vez el modelo, el Capítulo 3 se concentra en su implementación. Para ello, se hace una breve introducción al paradigma bayesiano de la estadística, en particular al apredizaje bayesiano en un contexto de regresión lineal. Este paradigma, responde a que, usando las ideas de Albert y Chib (1993), el algoritmo asociado al modelo recae en una técnica fundamental de la diciplina: el muestreador de Gibbs. Con esta poderosa herramienta, se presenta los detalles y lógica detrás de la implementación. Asimismo, se explica a detalle el algoritmo, el cual se implementa en un paquete computacional para el lenguaje abierto de programación estadística`R`.[3]

Una vez que el modelo es funcional y fácil de implementar, en el Capítulo 4 se

3. El desarrollo y explicación del paquete de cómputo se detalla en el Apéndice C, y corresponde a que, simplifica mucho el proceso de aprendizaje del modelo y fomenta su fácil uso y su validación por terceros. El paquete se puede descargar libremente de: `https://github.com/PaoloLuciano/bpwpm`

prueba y se valida contra una diversa 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 corre el modelo contra cinco bases de datos simulados con dos covariables ($\mathbf{x}_i \in \mathbb{R}^2$). Estas pruebas preliminares, sirven para demostrar las capacidades predictivas del modelo y sobre todo, para hacer más concretas las matemáticas subyacentes y poder visualizar las diferentes fronteras flexibles obtenidas por el modelo. Asimismo, en este capítulo se discute la convergencia de las cadenas obtenidas del muestreador de Gibbs, uno de los puntos cruciales y delicados del modelo.[4] Para cerrar el capítulo, se replica un escenario real de análisis y modelado, usando una base de datos médicos de cáncer de donde se obtienen excelentes resultados.

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 avanzados y usados hoy en día, son generalizaciones de modelos tradicionales.

4. Como se verá en los capítulos subsecuentes, dada la expansión de f en polinomios por partes, la *identificabilidad* de los parámetros es uno de los puntos débiles del modelo.

Capítulo 2

Modelo en su forma matemática

Como base fundamental de este trabajo, a continuación y de forma preliminar,[1] se presenta una visión general modelo; mientras que el resto del capítulo se enfocará en profundizar en cada parta de este. Se trata de seguir la notación usada en los libros de 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 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 di-

1. La versión completa del modelo se presenta en la sección 3.3.1 de la página 62
2. Se utiliza la convención de usar negritas para distinguir vectores $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,n}) \quad \forall i = 1, \ldots, n$

mensionalidad 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 es el vector de respuestas y las columnas subsecuentes $j = 1, \ldots, d$ representan una covariable. Es útil pensar en estas columnas como *d variables o dimensiones* que contienen información que induce la clasificación binaria. Asimismo, se define el espacio de covariables $\mathcal{X}^d$ como el pro-

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

Tabla 2.1: Estructura de los datos

ducto 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.

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

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 probit bayesiano no lineal (preliminar), $\forall i = 1, \ldots, n$:

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

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

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

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

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 este enfoque es que Albert y Chib desarrollaron un método numérico vía simulación, bajo el paradigma bayesiano, para el cómputo exacto de

las distribuciones posteriores de los parámetros $\beta_{j,l}$ 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 $\eta(\cdot) : \mathbb{R}^d \to \mathbb{R}$ 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 los Modelos aditivos generalizados (GAM) introducidos en Trevor Hastie y Robert Tibshirani (1986); en la Sección 2.2 se estudia el detalle de este tipo de modelos.

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 a detalle 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,

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

la presentada en Denison, Mallick y Smith (1998). Por el momento, se deja a las 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).

Antes de continuar y para esclarecer el trabajo un poco más, 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 covariable 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 a su vez, conecta a la respuesta y_i con la información adicional contenida en las covariables $\mathbf{x}_i$, a través de un modelo probit 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 $f_j(x_{i,j}) \quad \forall j$, $\eta(\mathbf{x}_i)$ 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

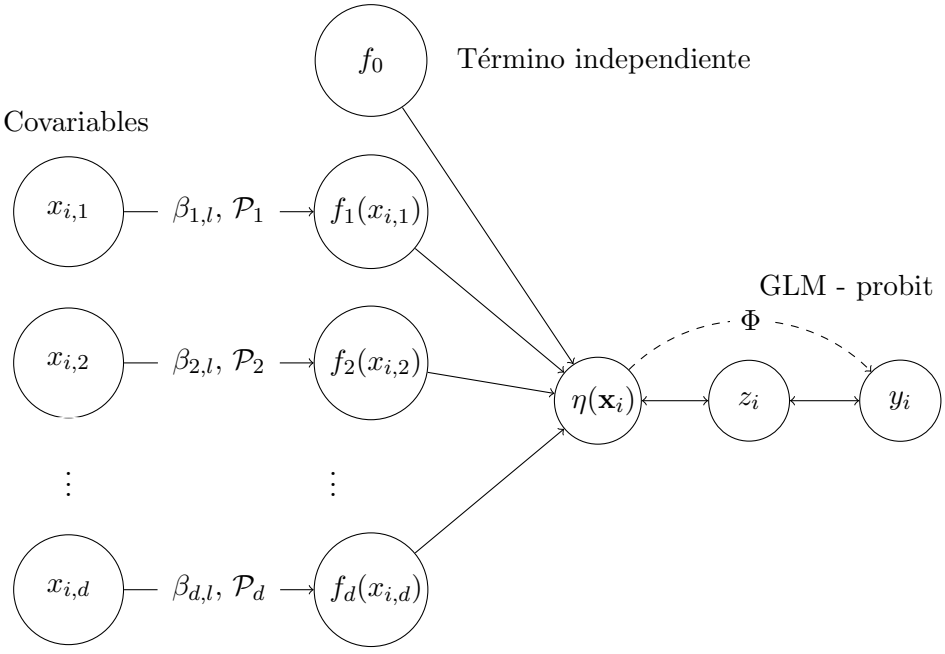


Figura 2.1: Diagrama del modelo

visualizaciones que esperan volver estos conceptos teóricos en algo más concreto.

Antes de continuar, vale la pena mencionar el precepto de Box (1979):

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), MacCullagh y Nelder (1989), surgen como una generalización del modelo de regresión lineal:

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

donde $y_i \in \mathbb{R}$, $\beta \in \mathbb{R}^d$ es un vector de parámetros y $\mu(\mathbf{x}_i) = \mathbb{E}[y_i|x_i]$. Las regresiones lineales, como su nombre lo indica, 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 β. Asimismo, la generalización del modelo lleva a que la intrepretación de los parámetros no sea trivial.[5]

5. 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. $\ln(\pi_1/\pi_0) = \beta^t x$, donde π_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):

$$\begin{aligned} y &\sim F(\mu(\mathbf{x})) \\ \eta &= \beta_0 + \boldsymbol{\beta}^t \mathbf{x} \\ \mu &= g^{-1}(\eta) \end{aligned} \tag{2.6}$$

que 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.[6] 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.[7]

g: función liga que une la media μ de la distribución con el predictor lineal,[8] es decir: $\mu(x) = \mathbb{E}[y|x] = g^{-1}(\boldsymbol{\beta}^t x)$. 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).

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

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

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

Dado que para este trabajo se busca construir un clasificador supervisado donde las respuestas observadas sean binarias, i.e. $y_i \in \{0, 1\}\ \forall i = 1, \ldots, n$; la discusión se centrará en la distribución Bernoulli pues resulta de forma natural. 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 $\forall i = 1, \ldots, n$:

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

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 regresión. 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 intervalo $(0, 1)$, el dominio de la media. 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 éxito.[10]

En particular, en este trabajo se escoge como función liga a la función probit, la inversa de la función de acumulación normal estándar $\Phi(\cdot)$, i.e. $g(\mu) = g(p) = \text{probit}(p) = \Phi^{-1}(p)$, dándole nombre al modelo, en consecuencia al trabajo de Albert y Chib (1993). 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 probit en su forma más rigurosa, rescatando la notación de un GLM (2.6) con sus

10. En un contexto de aprendizaje de máquina, se les conoce como funciones de activación a las inversas de la función liga g^{-1} (Bishop 2006). Recientemente se utiliza mucho la función $ReLu(x) := \text{máx}\{0, x\}$ la cual no es suave ni biyectiva (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

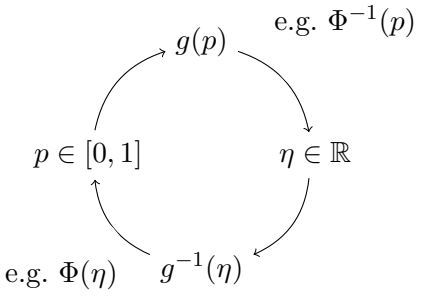


Figura 2.2: Esquema de función liga g para un modelo probit

respectivas covariables $\mathbf{x}_i$:

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

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

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

Equivalencia en las definiciones del modelo

El lector notará, sin embargo, que la especificación del modelo probit en las ecuaciones anteriores no corresponde a la definición mostrada al principio del Capítulo. Sin embargo, a continuación se prueba de forma muy sencilla la equivalencia entre ellas.

Teorema 2.3. *Un modelo probit especificado en* (2.9)*,* (2.10) *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 $\tilde{z}_i \sim N(\tilde{z}_i|0, 1)$ tales que:

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

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

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

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 da una correspondencia uno a uno entre espacios de probabilidad y se

puede definir y_i en terminos de z_i y viceversa, dando lugar a la definición (2.1).

El argumento es similar 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 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 $\eta_i = \eta(\mathbf{x}_i)$.

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. En la Sección 3.3 se detalla el algoritmo, derivado del paradigma bayesiano, de las variables latentes z_i y su conexión con los parámetros β.

Recapitulando, mediante la función liga Φ se une la media p_i, la probabilidad de éxito de la respuesta y_i con los datos $\mathbf{x}_i$. Esto se logra, a través de una variable latente z_i con distribución normal cuya media $\eta(\mathbf{x}_i)$, la función de predicción, es una

12. Comenzar con $z_i|\mathbf{x}_i \sim N(z_i|\eta(\mathbf{x}_i), \sigma^2)$ 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)

transformación de las covariables $\mathbf{x}_i$.

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

Este enfoque funciona, además de por el componente algorítmico, por la siguiente idea altamente intuitiva. Si se quiere crear una regla de decisión que clasifique observaciones en categorías binarias con base en cierta información, parecería intuitivo condensar esta información de forma que proporcione suficiente evidencia para inducir la clasificación. Es decir y 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 $f(\mathbf{x}_i)$ muy positiva 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 éxito ($y_i = 1$). 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, T.J. Hastie y R.J. 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 aún 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

13. Página 1 de T.J. Hastie y R.J. Tibshirani (1990)

por una función asociada $f_j \quad \forall j$. De esta forma, se retiene algo de la interpretabilidad 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 estimado 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).

La estimación de las funciones f_j, se lleva a cabo por el algoritmo de ajuste hacia atrás (*backfitting algorithm*), Trevor Hastie y Robert Tibshirani (1986). Este procedimiento, busca dar estimadores de cada f_j de forma iterativa por componen-

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.

tes, utilizando como regresores los residuales parciales. Por ejemplo, sea $d = 2$ y $g^{-1}(w) = w$ la función identidad, quedando así el modelo:

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

Se toman estimadores preliminares $\hat{f}_0$ y $\hat{f}_1$, para suaviza $f_2(x_2)$ sobre los residuales parciales: $\mathbb{E}[y_i|\mathbf{x}_i] - (\hat{f}_0 + \hat{f}_1(x_1))$. Dado entonces $\hat{f}_2$ se puede mejorar el estimador de $f_1(x_1)$. 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 Sección 2.3 que componen a las funciones f_j. Aunque pareciera una desviación considerable del trabajo original de T.J. Hastie y R.J. Tibshirani, en realidad en el Apéndice A se detalla como los polinomios por partes son el resultado de plantear la idea de suavizamiento como un problema de optimización. Asimismo, los GAM

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

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), $\eta(\mathbf{x}_i) = \beta_0 + \boldsymbol{\beta}^t\mathbf{x}_i$, 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 β 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 (2.12) si f_j es más positiva para esta observación i, se tiene mayor evidencia (en la

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

covariable 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 ...

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.

Sin embargo, para este trabajo al termino 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 su 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 por componentes, 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:

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

Bajo esta definición, $\Psi(\mathbf{x}) = (\Psi_1(\mathbf{x}), \ldots, \Psi_{N^*}(\mathbf{x}))^t$ 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 $\beta = (\beta_1, \ldots, \beta_{N^*})^t$ 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:

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

lo cual es idéntico a (2.14) con la adición del término independiente β_0.[21] Bajo este contexto, lo que se busca es representar una transformación g de la media condicional de la respuesta y 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, análoga a la función de predicción η, también puede ser expresada como su expansión en bases funcionales.[22]

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:

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

$\Psi_l(\mathbf{x}) = \ln x_l$ ó $x^{1/2}$ 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. β_0 es fundamental para el correcto ajuste en el caso que $\mathbf{x}_i = 0$ para alguna i.
22. Un supuesto fuerte pero útil.

$\Psi_l(\mathbf{x}) = \exp\left\{-\frac{(x_l - \mu_l)^2}{2s^2}\right\} \quad l = 1, \ldots, d$ 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 donde se incluyen términos de interacción.

Como se ve, esta representación engloba muchos de los modelos y transformaciones posibles en el mundo de las regresiones, uniendo temas de análisis funcional con estadística aplicada. Además de que en general han resultado ser de gran utilidad en la práctica. Se hace notar que el último ejemplo rompe con la aditividad inherente de 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 β pero no lineal en las covariables, dependiendo de la forma de $Psi(\mathbf{x})$.

De igual forma, 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 más flexible (por no decirla ingenua) de éstos. El método más común es tomar una familia grande de funciones que logre representar una gran variedad de patrones. No

obstante, una desventaja de estos métodoses que al contar con una cantidad muy grande de funciones base y por ende parámetros, se requiere controlar la complejidad del modelo para evitar el *sobre-ajuste*. Algunos de los métodos más comunes para lograrlo son los siguientes (Hastie, Tibshirani y Friedman 2008):

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

Se hace notar que en la Sección anterior, se modelaba η, sin embargo 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 los cuales depende únicamente de una variable real $x_j \quad \forall j$.[25] Por lo tanto, se trabaja por el momento únicamente con funciones que mapeen reales a reales.

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)
25. Más adelante se verá que la exposición es análoga pero se simplifica mucho el número de índices.

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 dar la expresión final de las funciones f_j, se da una exposición constructiva de estos. 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 $\mathcal{P} = \{\tau_1, \tau_2, \ldots, \tau_{J-1}\}$ tal que $a \leq \tau_1 < \ldots < \tau_{J-1} \leq b$, las constantes τ son llamadas *nodos*.[26] Con los nodos seleccionados, se puede hacer una representación de una función arbitraria h en su expansión de bases como en 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.

Un ejemplo sencillo: sea $J = 3$, separando el intervalo en tres pedazos, es decir la partición tiene dos nodos $\mathcal{P} = \{\tau_1, \tau_2\}$. A cada subintervalo se le asocia una función

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

Ψ_j,

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

con $I(\cdot)$ la función indicadora que vale uno si x se encuentra en la región y cero en otro caso. De donde 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.[27]

Con este ejemplo, se ilustra a grandes rasgos como funcionan los polinomios por partes. Sin embargo, los polinomios por partes son mucho más flexibles pues a cada intervalo se puede ajustar un polinomio de grado arbitrario $(M-1)$.[28] 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 por partes, que se les puede pedir cuanta *suavidad* (o no) se requiera,

27. 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, es fácil 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.

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

entendido 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 $J \times M$ bases funcionales. Esto ocurre porque se necesita definir una base de tamaño M para cada subintervalo $j = 1, \ldots, J$, es decir, $\mathcal{B}_j = \left\{1, x, x^2, \ldots, x^{M-1}\right\}$. Esta definición, lleva a polinomios que se comportan de forma independiente en cada intervalo y no se conecten. Naturalmente, la primera condición en la que se piensa, es imponer continuidad en los nodos lo cual devuelve $J - 1$ parámetros que corresponden a los $J - 1$ nodos. De la misma forma, cada grado de continuidad 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:

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

bases funcionales y por ende, el mismo número de parámetros β por estimar. Dada la construcción que se hace y las características de M, J y K se tienen las restricciones:

$M > K \geq 0$ 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 ($K = M-1$). Se hace notar, que existen muchos tipos de *splines*, además de que pueden ser más flexibles o más rápidos en su implementación computacional como los B-Splines. En Boor (1978) y más recientemente Wahba (1990) se hacen tratados extensivos sobre ellos 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 delimitado el número de funciones bases N^* y por ende sus componentes M, J y K, finalmente se le puede dar forma funcional a las funciones base Ψ. 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 pueda representar un polinomio por partes de una forma relativamente sencilla evitando separar la función en varias lineas.

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

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

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

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

Esta expansión aunque pesada en notación, tiene muchas propiedades atractivas. Además, 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 $j = 1, \ldots, J-1$ se tienen $M - K$ funciones parte positivas que se van activando 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.[29] 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 $M = 3$, $J = 5$, $K = 0$ se tiene un polinomio por partes en 5 subintervalos

29. 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.

(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*.

Para la facilitar la interpretación de los parámetros y la expansión de (2.18), las β cuentan dos índices: i y j. El índice i siempre está asociado al grado de su función base asociada, es decir, si $i = 2$ se está hablando de un término de grado 2. En el segundo sumando (la parte truncada) el índice i comienza en K para codificar las restricciones de continuidad.[30] El segundo índice $j = 1, \ldots, J - 1$ describe el nodo al que está asociado el parámetro. Como convención, si $j = 0$, 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 igual a (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 35. Esta Tabla ayuda no sólo a esclarecer la notación, sino a expresar todo de forma matricial que posteriormente se implementará en el código.

30. 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.

β_l	$\beta_{i,j}$	$\Psi_l(x, \mathcal{P})$		
Subíndice l	Subíndices i, j	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 tienen $N^* = M + (J-1)(M-K) = J \times M - K(J-1)$ términos, ecuación (2.16). Por construcción, se es consistente con la definición de *spline* si $K = M - 1$.

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 las e_i son variables aleatorias con media cero.

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. Además permiten aumentar o disminuir la cantidad de estos nodos desarrollando un algoritmo Gibbs sampler trans-dimensional, es decir, que 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 alguna sección, 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.[31] En el Capítulo 3 se detalla como la simplificación de no incorporar los nodos como parámetros ayuda bastante a la velocidad del algoritmo. Posteriormente en el Capítulo 4, se ve que para fines prácticos, el modelo funciona muy bien y finalmente en el Capítulo 5 se discute que habría cambiado de haberse implementado.

2.3.3. Consideraciones finales para el modelo

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 más evidente:

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

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

En donde cada sumando interior tiene una expansión de bases funcionales definida por la ecuación (2.18).[32] Asimismo, f_0 puede ser pensado como otro parámetro adicional, es decir: $f_0 = \beta_0$. Este hecho de linealidad en los parámetros de η, lleva a que (2.20) pueda ser re-expresada simplemente como el producto punto de un largo vector de parámetros $\boldsymbol{\beta}$ y una matriz de diseño $\widetilde{\Psi}$ que incorpora la doble suma con sus correspondientes expansiones en bases y todas las observaciones $i = 1, \ldots, n$. Sin embargo, 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 el modelo, 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, $K > 0$.[33] Segundo, se debe retirar el término independiente inherente en el polinomio base, es decir, comenzar el primer sumando en uno en vez de cero. Esta modificación retira una función base modificando N^*.[34] Juntando estas cambios

32. No se hace la sustitución pues la notación resultaría demasiado pesada.

33. 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. No obstante, en los polinomios por partes si se observaba la discontinuidad.

34. 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.

(2.18) se transforma en:

$$h(x) = \sum_{l=1}^{N^*} \beta_l \; \Psi_l(x, \mathcal{P})$$

$$\text{con } \; N^* = J \times M - K(J-1) - 1$$

$$\text{donde: } \; M > K > 0 \text{ y } J > 1$$

$$= \sum_{i=1}^{M-1} \beta_{i,0} \; x^i + \sum_{i=K}^{M-1} \sum_{j=1}^{J-1} \beta_{i,j} \; (x - \tau_j)_+^i. \tag{2.21}$$

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 canónica del modelo (2.4).

Juntando todo lo anterior, el predictor lineal η se puede re-expresar en su forma vectorial compacta:

$$\boldsymbol{\eta}(\mathbf{X}) = \widetilde{\Psi}(\mathbf{X})\boldsymbol{\beta}, \tag{2.22}$$

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

estructuras:

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

$$\widetilde{\Psi}(\mathbf{X}) = \left[\begin{array}{c:c:c:c} 1 & f_1(x_{1,1}) & \dots & f_d(x_{1,d}) \\ \vdots & \vdots & & \vdots \\ 1 & f_1(x_{n,1}) & \dots & f_d(x_{n,d}) \end{array}\right] \tag{2.23}$$

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

Bajo esta definición, el modelo se simplifica. Se observa que en realidad cada f_j es justamente 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 la no-colinealidad entre las columnas

de $\widetilde{Psi}$ por construcción, por lo que se podrían dar problemas en la estimación.[35]

A pesar de la utilidad de estos polinomios por parte, todos sufren de problemas más allá del rango de definición $\mathcal{X}_j = [a_j, b_j] \quad \forall j = 1, \ldots, d$. 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 $2 \times (M-2)$ 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 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 cambiar como lo hace una transformación de coordenadas en un espacio euclidiano. Cada base tiene sus beneficios y desventajas. Para esta exposición, se escoge la expansión en bases truncadas pues es explicada fácilmente y tiene una forma funcional relativamente sencilla. Además, la interpretación de los coeficientes β es inmediata. Sin embargo, no es buena computacionalmente hablando cuando J es grande. En la practica, usualmente se implementan B-Splines[36] o bases

35. 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 $\Psi^t\Psi$.

36. Vease el Capítulo 5.5 de Wasserman (2007) o el Apéndice del Capítulo 5 en Hastie, Tibshirani

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 (2.21) con base en la Tabla 2.2 y (2.23) 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[37] pues, como ya se mencionó anteriormente, hacer J variable y automático es muy complejo.

y Friedman (2008).

37. En el Capítulo 4 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 teórica 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 de una forma relativamente 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 se le titula: *bayesian piece wise polinomial model (bpwpm)* para reflejar los componentes del modelo 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. (Mendoza y Regueiro 2011), (Bernardo y Smith 2001)

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 sobre creencias, se realiza mediante una *actualización* de estas 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. En menos palabras, el teorema de Bayes está actualizando la probabilidad de ocurrencia de un evento ponderando esta probabilidad por la información que se tiene sobre el.

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. 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 θ que minimice estos errores, así, el modelo logra

capturar toda la variabilidad presentan los datos.

Independientemente del paradigma estadístico que se escoja, siempre es importante la validación del modelo y de sus supuestos. No obstante, tanto teoría bayesiana como 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. En el Apéndice B se detallan las distribuciones conjugadas y se realiza más a fondo la derivación de los resultados de este trabajo. Sin embargo, con los avances en el poder computacional disponible y técnicas numéricas para resolver integrales (Robert y Casella 2004), se han desarrollado muchos métodos para aplicar el proceso de aprendizaje independientemente de que tan complejo sea el modelo. 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.

Estimadores Bayesianos

Una vez realizado el proceso de actualización, se cuenta con una distribución posterior de probabilidad para los parámetros de interés.[1] 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 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:

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

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.

1. 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.[2]. 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-

2. 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 $X^{(k)}$.

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 expresada como: el futuro, condicionando al presente, es independiente del pasado. El ejemplo canónico que se presenta es la caminata aleatoria: $X^{(t+1)} = X^{(t)} + e^{(t)}$, con $e^{(t)}$ error aleatorio generado de forma independiente. De esta idea se desarrolla toda una rica teoría revisada en cursos de procesos estocásicos (Ross 2009).

Una de las ideas más relevantes para lo que concierne este trabajo, es la de *matrices de transición*. Dada una cadena con n posibles estados ($X^{(t)}$ 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*,[3] entonces existe una *distribución límite* que es igual a la *distribución estacionaria*: $\exists\, \pi$ tal que $\pi P = \pi$. Sin entrar en los detalles técnicos, la ergodicidad es la propiedad que asegura que eventualmente se alcanza la convergencia de la cadena sin importar el estado inicial tras repetidas aplicaciones de la matriz de transición P.[4] 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 misma. 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

3. 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.

4. 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 $\theta = (\theta_1, \ldots, \theta_\lambda)$ que provienen de la distribución conjunta $\pi(\theta)$. 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 $\hat{\pi}(\theta) \approx \pi(\theta)$. En la práctica usualmente más que aproximar la distribución, se busca estudiar la muestra 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:

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

Se comienza con un valor inicial arbitrario $\theta^{(0)} = (\theta_1^{(0)}, \ldots, \theta_p^{(0)})^t$, donde el superíndice $^{(k)}$ corresponde a la iteración k. Se comienza a simular de las correspondientes distribuciones condicionales, las cuales quedan especificadas para los valores inicia-

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

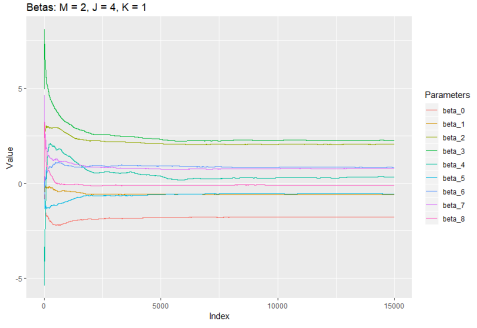
$$\begin{aligned}
\theta_1^{(k)} &\sim \pi(\theta_1|\theta_2^{(k-1)}, \ldots, \theta_p^{(k-1)}) \\
\theta_2^{(k)} &\sim \pi(\theta_2|\theta_1^{(k)}, \theta_3^{(k-1)}, \ldots, \theta_p^{(k-1)}) \\
&\vdots \\
\theta_p^{(k)} &\sim \pi(\theta_p|\theta_1^{(k)}, \ldots, \theta_{p-1}^{(k)})
\end{aligned} \tag{3.5}$$

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 $\pi(\theta)$.

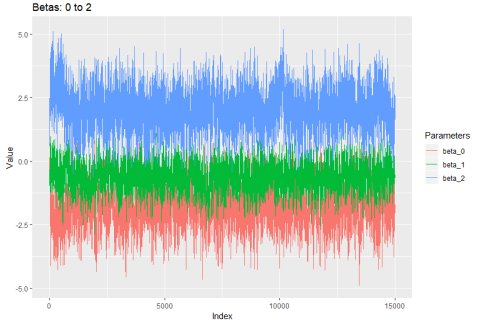
La convergencia no es intuitiva, es decir, no es trivial derivar que al muestrear de las distribuciones condicionales, se obtenga eventualmente una muestra de la distribución conjunta Sin embargo, la prueba formal recae en las mismas ideas de las cadenas de Markov. Definido el problema, se puede formar una kernel de transición, generalización de las matrices de transición, derivado de las distribuciones condicionales de θ_i. 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, es importante revisar si esta ya ha alcanzado la distribución posterior. Para ello, es usual revisar la media ergódica (media acumulada) de cada

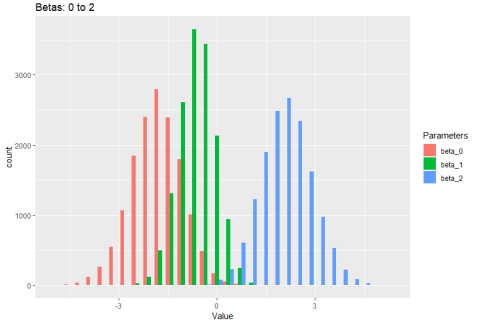
parámetro, de donde se esperaría ver que la variación hacia el final de la cadena es mínimo. Asimismo, se suele revisar la traza de la cadena en sí y los histogramas de ella. En la figura 3.1 se tienen tres imágenes de las cadenas simuladas por el mustreador Gibbs implementado en este trabajo[5], en particular, las cadenas del ejemplo... de la página ... . Para el modelo se escogen los parámetros $M = 2$, $J = 4$ y $K = 1$, implicando que se tienen rectas continuas en tres nodos, derivando en un total de $\lambda = 9$ parámetros por estimar ($\boldsymbol{\beta} \in \mathbb{R}^9$). La imagen 3.1a presenta



(a) Media ergódica



(b) Trazas de la cadena



(c) Histogramas

Figura 3.1: Muestro Gibbs para el ejemplo 1 de la Sección ...

5. Las imágenes fueron generadas fácilmente con la librería *ggplot2*, incorporada a las funcionalidades del paquete *bpwpm2* desarrollado para este trabajo.

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.[6] 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 perfecto.[7] No obstante, las cadenas pueden ser mejoradas de dos formas sencillas. La primera se le conoce como periodo de *burn-in* y consiste en eliminar los primeros k^*-esimos valores simulados de la cadena. Esto sucede, ya que el valor inicial $\theta^{(0)}$ es dado 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 o *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 depende el

6. Solamente se muestran los primeros tres parámetros para evitar tener gráficos muy saturados.
7. En el sentido que no genera una muestra v.a.i.i.d.

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 *bpwpm2* desarrollado para el rápido análisis de los modelos.

3.3. El modelo *bpwpm*

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

Método de Albert y Chib

En Albert y Chib (1993), los autores desarrollan un método bayesiano para el análisis de respuestas binarias y policotómicas.[8] En el caso binario, su enfoque resultaba muy atractivo para los objetivos del trabajo pues uno de los resultados es la definición de una regresión probit bayesiana con el uso de variable latente z.[9]

En resumen, su modelo titulado *data augmentation for binary data*, propone una definición del modelo probit como la presentada en (2.1) y (2.2). Bajo esta definición,

8. Una respuesta policotómica es una respuesta que perteneces a más de dos categorías, por ejemplo: partidos políticos. Usualmente se modelan con distribuciones multinomiales.

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

se calculan las distribuciones marginales de los parámetros. Asimismo, se propone usar distribuciones conjugadas normales para $\boldsymbol{\beta}$ derivando en que el algoritmo sea 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:

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

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

$$\eta(\mathbf{x}_i) = \boldsymbol{\beta}^t \widetilde{\boldsymbol{\psi}}_i(\mathbf{x}_i) \tag{3.6}$$

Donde $\widetilde{\boldsymbol{\psi}}_i(\mathbf{x}_i)$ es un renglón de la matriz de transformación (2.23) presentada en la página 40. Sin embargo, ahora 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: $\theta = (\mathbf{z}, \boldsymbol{\beta})$. Por lo

tanto, la derivación de la densidad posterior resulta en:

$$\begin{aligned}
\pi(\mathbf{z}, \boldsymbol{\beta}|\mathbf{y}, \mathbf{X}) &\propto \pi(\mathbf{y}|\mathbf{X}, \mathbf{z}, \boldsymbol{\beta})\ \pi(\mathbf{z}, \boldsymbol{\beta}) \qquad \text{por (3.2)} \\
&\propto \pi(\mathbf{y}|\mathbf{z})\ \pi(\mathbf{z}|\boldsymbol{\beta}, \mathbf{X})\ \pi(\beta) \qquad \text{por definición} \\
&= \prod_{i=1}^{n} [I(y_i = 1)I(z_i > 0) + I(y_i = 0)I(z_i \leq 0)] \\
&\qquad \times \phi(z_i|\eta\mathbf{x}_i), 1) \times \pi(\boldsymbol{\beta}). \qquad (3.7)
\end{aligned}$$

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 $N(\cdot|\mu, \sigma^2)$ y $\pi(\boldsymbol{\beta}$ la densidad *a priori* de $\boldsymbol{\beta}$.

Bajo los fundamentos del muestreador de Gibbs, dado que encontrar (3.7) es complejo, se busca derivar mejor las distribuciones condicionales. Para $\boldsymbol{\beta}$, la densidad marginal condicional esta dada por:

$$\begin{aligned}
\pi(\boldsymbol{\beta}|\mathbf{z}, \mathbf{y}, \mathbf{X}) &= \frac{\pi(\mathbf{z}, \boldsymbol{\beta}|\mathbf{y}, \mathbf{X})}{\pi(\mathbf{z})} \qquad (3.8) \\
&= \frac{\pi(\mathbf{y}|\mathbf{z})\ \pi(\mathbf{z}|\boldsymbol{\beta}, \mathbf{X})\ \pi(\boldsymbol{\beta})}{\pi(\mathbf{y}, \mathbf{X})\ \pi(\mathbf{z})} \\
&= \frac{\pi(\mathbf{y}|\mathbf{z})}{\pi(\mathbf{y}, \mathbf{X})\ \pi(\mathbf{z})} \overset{C}{\nearrow} \times \pi(\mathbf{z}|\boldsymbol{\beta}, \mathbf{X})\ \pi(\boldsymbol{\beta}) \qquad (3.9) \\
&= C\, \pi(\boldsymbol{\beta}) \prod_{i=1}^{n} \phi(z_i|\eta(\mathbf{x}_i), 1), \qquad (3.10)
\end{aligned}$$

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

conocidas. De lo anterior, se observa la utlidad de la variable latente: convierte una regresión probit a una regresión lineal. Asimismo, para estos modelos, dependiendo de la distribucion *a priori* $\pi(\boldsymbol{\beta})$ se pueden conseguir resultados cerrados. Se hace notar que la ecuación (3.8) se toma de la definición de probabilidad condicional, y el paso de (3.9) a (3.10) se puede hacer ya que, al definir y como en la ecuación (**??**), 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. En particular, se elige la distribución $\pi(\boldsymbol{\beta})$ como:

$$\boldsymbol{\beta} \sim N_\lambda(\boldsymbol{\beta} \,|\, \mu_{\boldsymbol{\beta}}, \Sigma_{\boldsymbol{\beta}}) \tag{3.11}$$

donde $\mu_{\boldsymbol{\beta}} \in \mathbb{R}^\lambda$ es el hiper-parámetro de media y $\Sigma_{\boldsymbol{\beta}} \in \mathbb{R}^{\lambda \times \lambda}$ la matriz de covarianza. Sustituyendo (3.11) en (3.10) y usando resultados estándar de modelos lineales (Banerjee 2008), se deriva que la densidad marginal conjugada para los parámetros es:

$$\boldsymbol{\beta} \,|\, \mathbf{y}, \mathbf{z}, \mathbf{X} \sim N_\lambda(\boldsymbol{\beta} \,|\, \mu^*_{\boldsymbol{\beta}}, \Sigma^*_{\boldsymbol{\beta}}), \tag{3.12}$$

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, esta distribución es fácil de simular usando cualquier software estadístico, calculando previamente la media y covarianza y dando un valor (o iteración) para $\mathbf{z}$.[10] 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. Comen-

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

zando con la expresión (3.7) y re-ordenando términos se tiene:

$$\begin{aligned}
\pi(\mathbf{z} \,|\, \boldsymbol{\beta}, \mathbf{y}, \mathbf{X}) &= \frac{\pi(\mathbf{z}, \boldsymbol{\beta} \,|\, \mathbf{y}, \mathbf{X})}{\pi(\boldsymbol{\beta})} \\
&= \frac{\pi(\mathbf{y} \,|\, \mathbf{z})\ \pi(\mathbf{z} \,|\, \boldsymbol{\beta}, \mathbf{X})\ \pi(\boldsymbol{\beta})}{\pi(\mathbf{y}, \mathbf{X})\ \pi(\boldsymbol{\beta})} \\
&= \frac{1}{\pi(\mathbf{y}, \mathbf{X})} \overset{C}{\nearrow} \pi(\mathbf{y} \,|\, \mathbf{z}) \times \pi(\mathbf{z} \,|\, \beta, \mathbf{X}) \\
&= C \prod_{i=1}^{n} [I(y_i = 1) I(z_i > 0) + I(y_i = 0) I(z_i \leq 0)] \\
&\qquad \times \phi(z_i \,|\, \eta(\mathbf{x}_i), 1). \qquad (3.13)
\end{aligned}$$

De donde se observa que cada z_i es independiente con con distribución normal truncada en 0, es decir $\forall i = 1, \ldots, n$:

$$\begin{aligned}
z_i | y_i, \boldsymbol{\beta} &\sim N(z_i | \beta^t \mathbf{x}_i, 1)_{I(z_i > 0) I(y_i = 1)} \quad \text{truncamiento a la izquierda} \qquad (3.14) \\
z_i | y_i, \boldsymbol{\beta} &\sim N(z_i | \beta^t \mathbf{x}_i, 1)_{I(z_i \leq 0) I(y_i = 0)} \quad \text{truncamiento a la derecha.}
\end{aligned}$$

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

3.3.1. Implementación algorítmica final

Una vez conectados todos componentes del modelo, este se puede presentar en su versión final y más completa (aunque definitivamente más pesada en notación).[11]

Definición 3.4. El modelo probit bayesiano no lineal (final),[12] $\forall i = 1, \ldots, n$:

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

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

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

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

$$= \sum_{\hat{\imath}=1}^{M-1} \beta_{j,\hat{\imath},0}\, x_{i,j}^{\hat{\imath}} + \sum_{\hat{\imath}=K}^{M-1} \sum_{\hat{\jmath}=1}^{J-1} \beta_{j,\hat{\imath},\hat{\jmath}}\, (x_{i,j} - \tau_{j,\hat{\jmath}})_+^{\hat{\imath}}. \tag{3.15}$$

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

$$\boldsymbol{\beta} \sim N_\lambda(\boldsymbol{\beta} \,|\, \mu_{\boldsymbol{\beta}}, \Sigma_{\boldsymbol{\beta}}) \qquad (\lambda = 1 + d \times N^*) \tag{3.11}$$

La ecuación (3.15) no es más que la expansión 2.21 presentada en la página 39 sobre

11. Se recuerda que existe un compendio de notación al inicio de este trabajo.
12. Aumentando sobre la definición 2.1

toda $x_{i,j}$. Asimismo, el modelo se puede presentar en su forma vectorial:

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

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

$$\boldsymbol{\beta} \sim N_\lambda(\boldsymbol{\beta} \,|\, \mu_{\boldsymbol{\beta}}, \Sigma_{\boldsymbol{\beta}}) \qquad (\lambda = 1 + d \times N^*) \tag{3.11}$$

$$\boldsymbol{\eta}(\mathbf{X}) = \widetilde{\Psi}(\mathbf{X})\boldsymbol{\beta} \tag{2.22}$$

De estas expresiones y juntandolo con el muestreador de Gibbs (3.5) definido por las distribuciones marginales de $\boldsymbol{\beta}$ y $\mathbf{z}$, (3.12) y (3.14) respectivamente, se presenta el algoritmo final en la tabla 1. 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 default. 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}, $\boldsymbol{\beta}^{(k)}$, $\mu_{\boldsymbol{\beta}}$ 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.[13] 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 $\mathbf{z}$. Todas las iteraciones de los parámetros se guardan en un objeto que regresa la rutina.

13. La implementación computacional de $\widetilde{\Psi}$, se basa en el diagrama 2.2 de la página 35 y la expresión (2.23). La subrutina que realiza la expansión tiene el nombre de `calculate_Psi` en el paquete y está vectorizada para que su ejecución sea veloz.

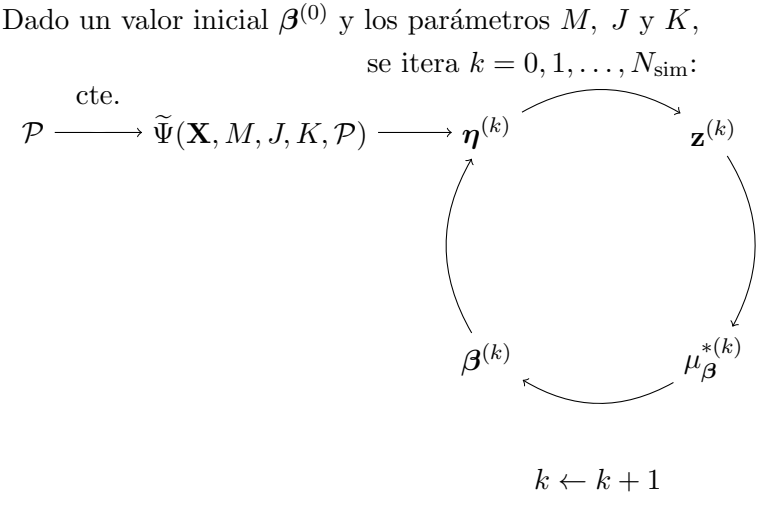


Figura 3.2: Esquema del algoritmo

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 arrojan, pueden ser utilizadas para hacer gráficas que evalúan el modelo y reflejan la intuición subyacente.

Para mostrar los resultados y las capacidades del modelo se presentan seis ejemplos. Los primeros cinco, corresponden a bases de datos simuladas en dos dimensiones, es decir, se tienen dos covariables $\mathbf{X} \in \mathbb{R}^2$, con diferentes patrones para la respuesta y

tanto lineales como no lineales. 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 reales de cáncer con múltiples variables. Al aumentar la dimensionalidad, el modelo ya no es visualmente intuitivo pero sigue obteniendo excelentes 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 4.3 de forma que no se saturara más la presentación.

4.1. Evaluación del modelo

No obstante, antes de poder presentar los ejemplos, se definen las dos métricas que se usarán para probar la efectividad (y precisión) de los modelos. Ya que se trabaja 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 la probabilidad 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 de*. 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 resultan 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. Asimismo, se puede definir 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. 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. Idealmente $ll = 0$ si se da una clasificación perfecta y conforme crezca, el modelo es peor. En la práctica y bajo un enfoque frecuentista, la función ll 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 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 las distribuciones normales bivariadas:

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

$$\underset{i=201,\ldots,350}{\text{Grupo 1:}} \quad \mathbf{x}_i \sim N_2\left(\left(\begin{array}{c} x_1 \\ x_2 \end{array}\right) \middle| \mu_1 = \left(\begin{array}{c} 4 \\ 4 \end{array}\right), \Sigma_1 = \left(\begin{array}{cc} 1 & -0.24 \\ -0.24 & 0.64 \end{array}\right)\right)$$

Las medias μ_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 en: $\mathcal{X}^2 \approx [0.3, 7.5] \times [-0.5, 5.9]$. 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, mostrando una clara separación entre los dos grupos sin sobre-ajustar. Para ello, se corren tres realizaciones del modelo. Para la primera, se escoge una frontera lineal con un solo nodo. La segunda realización, consta de parábolas continuas más no suaves sobre cuatro nodos. Finalmente la tercera realización consta de

2. Se recomienda visualizar la versión digital de este trabajo donde se aprecian con más claridad los colores. Disponible en ...

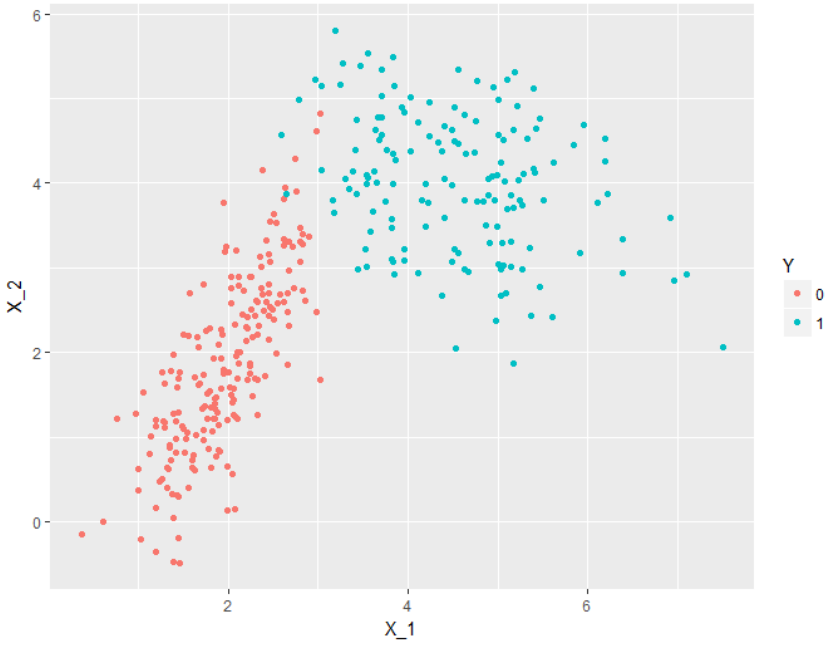


Figura 4.1: Ejemplo 1

splines cúbicos[3] en 3 nodos. 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^* el número de funciones base (por expansión de cada covariable) y λ 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 los modelos más comparables entre si y dar un vistazo rápido al modelo. La única modificación que se realiza entre las tres realizaciones es que, para la tercera, se estandarizan[5] los datos $\mathbf{X}$. 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 como en 4.2a como fron-

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

teras más suaves en las imágenes subsecuentes. Asimismo, para cada realización, se tiene la representación en 3D de cada función $\hat{\eta}$ que preserva la suavidad (o no) de sus componentes. Asimismo, 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 presenta $\hat{f}_2(x_2)$, 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

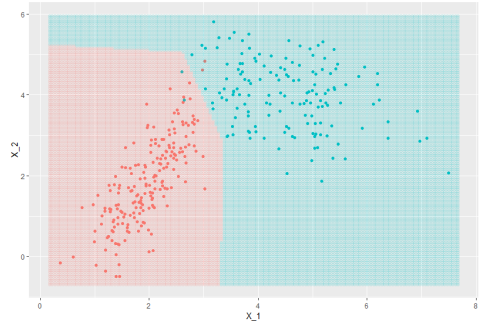
(a) Matriz de confusión

Realización	*ll*
1	0.04088
2	0.03464
3	0.03498

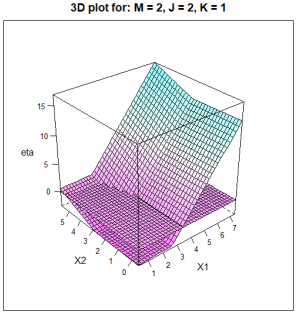
(b) *log-loss*

Tabla 4.3: Ejemplo 1 - resultados

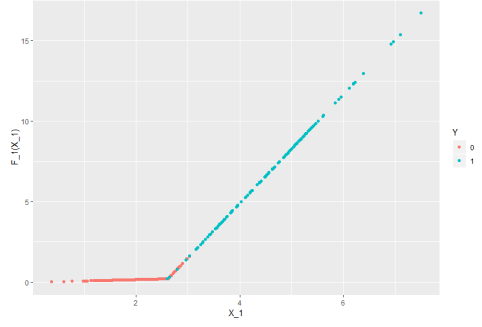
Al estar tratando con una base de datos tan sencilla, no es el enfoque comparar estas realizaciones entre si pues las tres logran exactamente la misma clasificación desglosada en la tabla 4.3a. Al compartir la matriz de confusión, por ende, las realizaciones también comparten una precisión de 98.9 %. 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, 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



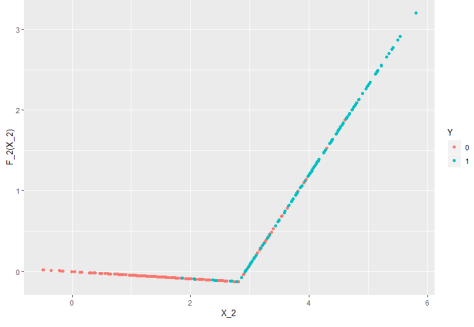
(a) Frontera de predicción



(b) Representación 3D de $\hat{\eta}$

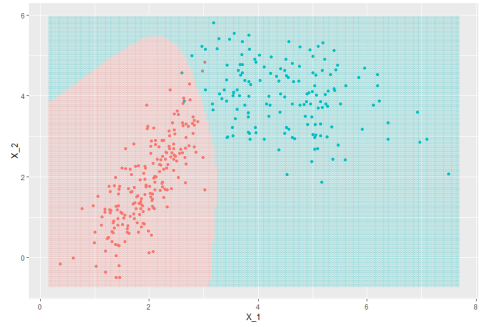


(c) $\hat{f}_1(x_1)$

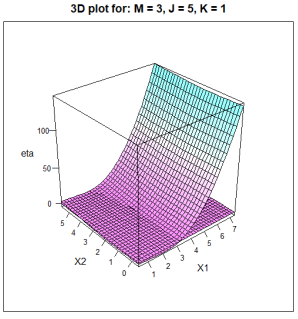


(d) $\hat{f}_2(x_2)$

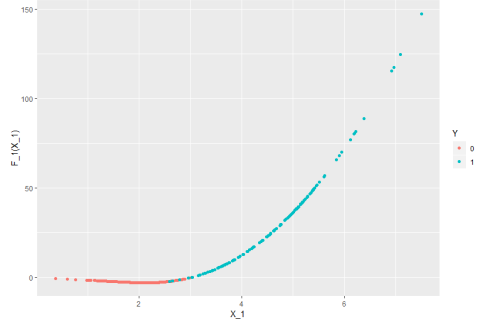
Figura 4.2: Realización 1 - fronteras lineales con un nodo (M = 2, J = 2 y K = 1)



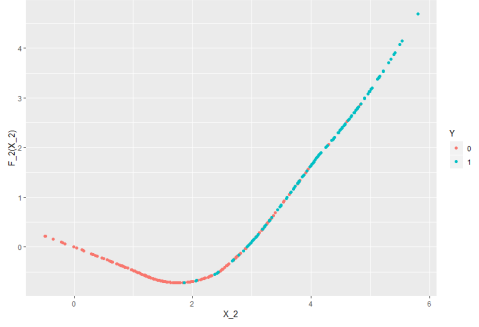
(a) Frontera de predicción



(b) Representación 3D de $\hat{\eta}$

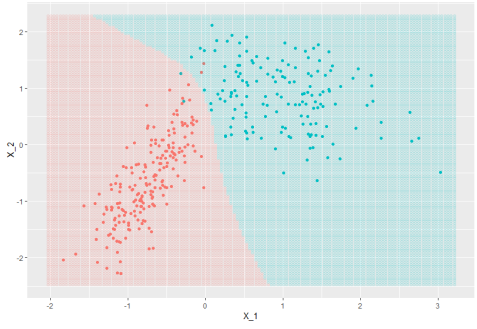


(c) $\hat{f}_1(x_1)$

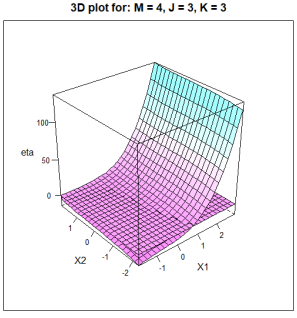


(d) $\hat{f}_2(x_2)$

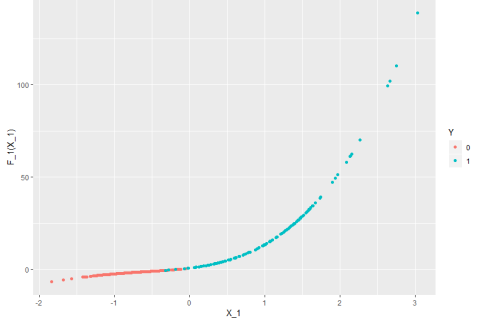
Figura 4.3: Realización 2 - parábolas continuas mas no suaves (M = 3, J = 5 y K = 1)



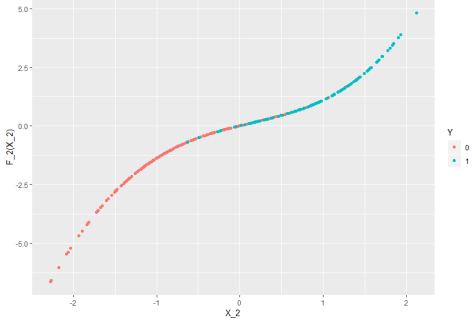
(a) Frontera de predicción



(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)

no se toma en cuenta el número de parámetros y la complejidad del modelo en si.

Cabe mencionar que para esta sencilla base de datos 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 las covariables. Sin embargo, se usa el modelo para ejemplificar las 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 prensentar. De igual forma, 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]

4.3. Ejemplo 2 - comparación contra un GLM

Aprovechando la familiaridad de la base de datos anterior, se decidió modificarala para que existieran 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 imágen 4.6a se presenta la base de datos modificada.

Con afán de comparar las predicciones del modelo *bpwpm* presentado en este trabajo, primeramente se corre un modelo probit lineal sobre esta base de datos. Es decir, se

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

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

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

De donde se obtien los resultados presentados en la tabla **??** y la figura 4.5.

Figura 4.5: frontera de predicción para modelo probit lineal

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:

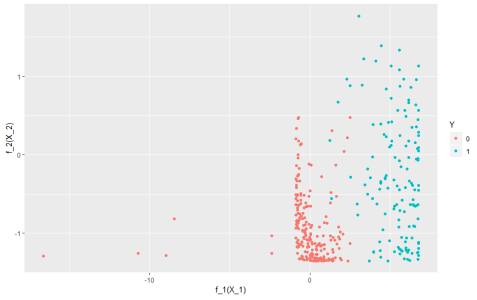
$$\phi^{-1}(\hat{p}_i) = \hat{\eta}(\mathbf{x}) = -4.67 + 0.45x_{i,1} + 0.91x_{i,2}, \tag{4.3}$$

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

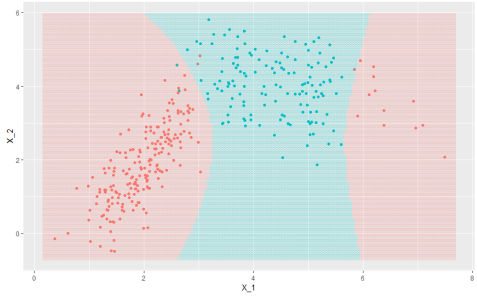
$$\begin{aligned} \phi(\hat{\eta}(\mathbf{x}_i)) \equiv 0.5 \qquad &\iff \\ \hat{\eta}(\mathbf{x}_i) = 0 \qquad &\iff \\ 0.45x_{i,1} + 0.91x_{i,2} = 4.67 \qquad & \end{aligned} \tag{4.4}$$

Ahora, se corre el modelo *bpwpm* con los parámetros escogidos resumidos en la tabla 4.5. Asimismo se presentan los resultados en la tabla 4.6 e imágenes en la figura 4.6.

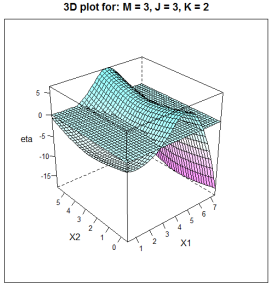
(a) Base del ejemplo 1 modificada



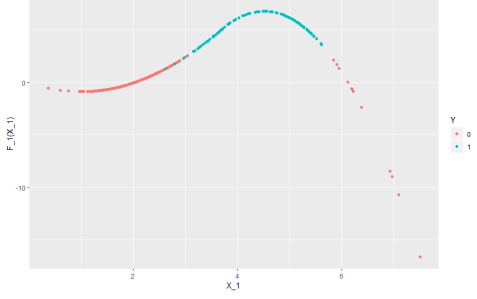
(b) Transformación no lineal



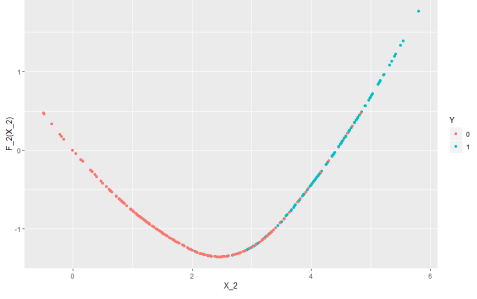
(c) Frontera de predicción



(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 $K = 2$)

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 3 - región parabólica

Juntando todo, el modelo final tiene la forma:

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

Contrastando los resultados del modelo probit lineal 4.4 contra el modelo *bpwpm* 4.6, se observa que se tiene una mejora en precisión sustancial. Claramente esto se debe a la no linealidad de la frontera de clasificación, imagen 4.6c, contra la frontera perfectamente lineal del modelo tradicional, imagen 4.5. Sin embargo, uno de los beneficios es que para el modelo probit lineal, esta frontera se puede derivar de forma explicita (4.4), mientras que para el modelo *bpwpm* implicaría resolver numéricamente la ecuación no lineal (4.5). Asimismo, al compara la métrica *log-loss*, se observa que se tiene una mejora sustancial. No obstante, uno de los pormenores

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 3 - resultados

grandes del modelo *bpwpm* es que el número de parámetros aumenta considerable para realizar la estimación. En cuanto a tiempo computacional sin embargo, no aumenta significativamente.

La ecuación (4.5) también permite observar la expansión en bases final de $\eta(\cdot)$ para esta realización y elección de M, J y K. Asimismo, posteriormente de cortar la cadena, se tienene las estimaciones puntuales de $\boldsymbol{\beta}$ y los nodos $\mathcal{P}$ en la tabla 4.6. Es necesario remarcar, la trasnformación que está realizando las funciones no lineales f_j y se observa en las imágenes 4.6a hacia **??**.

4.3.1. Análisis de convergencia

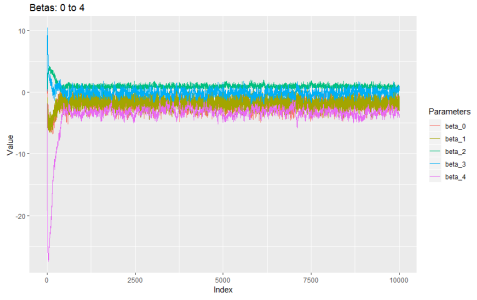
- Hablar de que se está buscando hacer

- Mejorar Este ejemplo es interesante pues, como se ve en la imágen **??**, la *sabana* que antes era creciente a medida que x_1 crecía, ahora se vuelve a curvar, volviéndose negativa otra vez y clasificando bien la segunda sección roja. Una vez más, se tienen esos pocos puntos que no quedan bien clasificados, incluyendo uno nuevo cerca de las coordenadas cartesianas $(5.8, 2.3)$. Para estos datos, se debe usar un nodo adicional cerca de la segunda región, ya que la curvatura, deriva de él. El parámetro K en este ejemplo no es muy relevante como se ve en la imagen **??**, nuevamente porque $\hat{f}_1(x_1)$ pareciera ser suficientemente suave sin tener que restringir el modelo. Finalmente, se enfatiza que vuelve a suceder lo mismo que pasó con el ejemplo 6, donde la información se podía resumir únicamente con las primeras dos dimensiones.

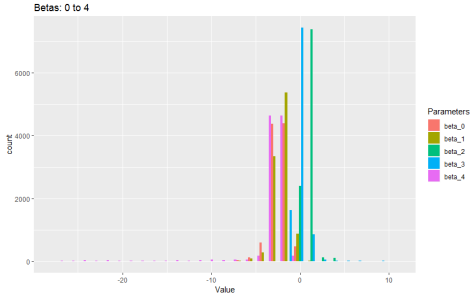
Métrica	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$
Mínimo	-6.79	-6.63	-0.39	-2.11	-27.40
Primer Cuartíl	-2.54	-2.22	0.66	-0.48	-3.72
Media	-2.03	-1.73	0.90	-0.07	-3.68
Mediana	-1.98	-1.69	0.85	-0.09	-3.28
Tercer Cuartíl	-1.44	-1.17	1.05	0.27	-2.86
Máximo	0.85	1.56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

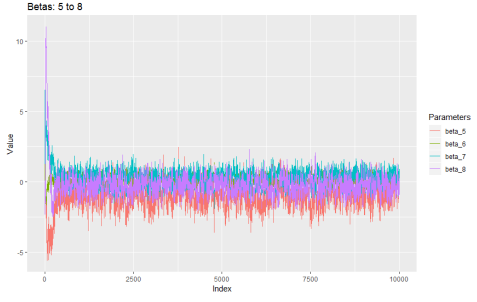
Tabla 4.7: resúmenes numéricos para las cadenas de $\boldsymbol{\beta}$



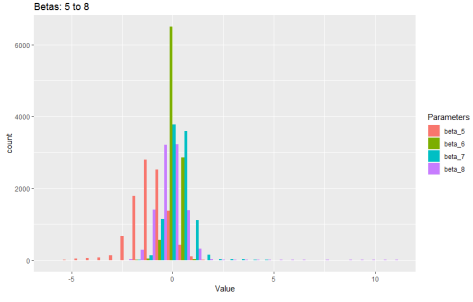
(a) Trazas de $\hat{\beta}_0$ a $\hat{\beta}_4$



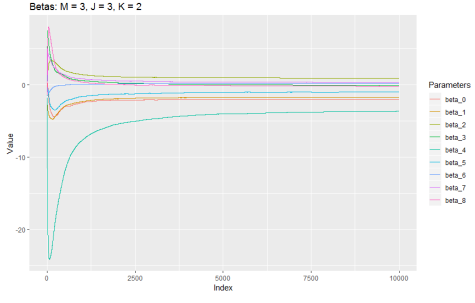
(b) Histogramas de $\hat{\beta}_0$ a $\hat{\beta}_4$



(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

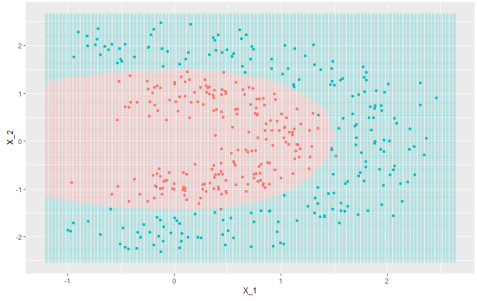
4.4. Ejemplos 3 a 5 - otros resultados interesantes

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 sería imposible para un GLM.

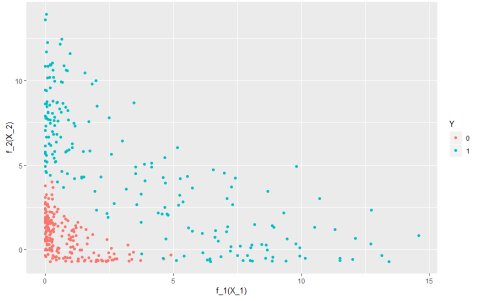
Ejemplo 3 - región parabólicos

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, inclusive 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.

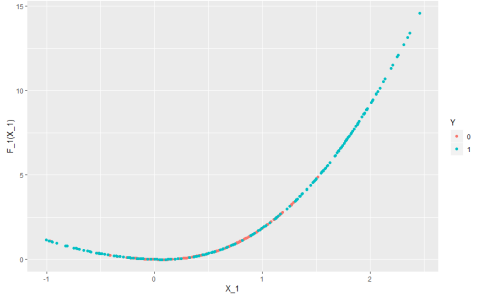
Esta es una realización particularmente interesante pues con un total $\lambda = 11$ parámetros se logra una precisión alta además de obtener convergencia relativamente rápido



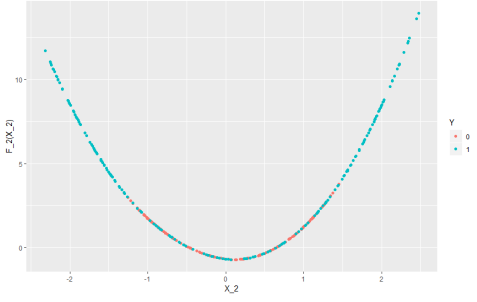
(a) Frontera



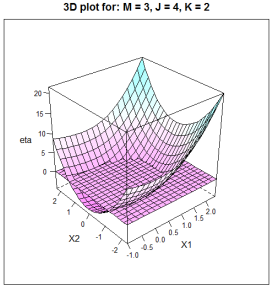
(b) Transformación no lineal



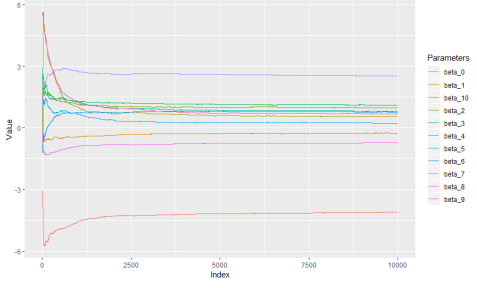
(c) $\hat{f}_1(x_1)$



(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. Avanzando con las regiones no lineales, se obtuvo una base de datos pequeña del curso en linea de ML de NG (2018) que presenta una frontera de clasificación ovalada.[8] 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 las interacciones entre las 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 ($J = 2$ y $\lambda = 7$) y cadenas

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

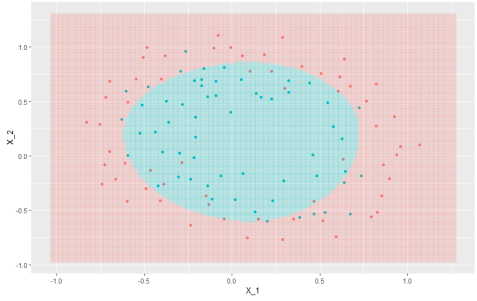
Tabla 4.11: Ejemplo 4 - resultados

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, sin embargo, se logra una reducción significativa en el número de parámetros pues el modelo saurado de NG (2018) inicia con 28 parámetros.[9]. 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 muy pocas observaciones y parámetros.

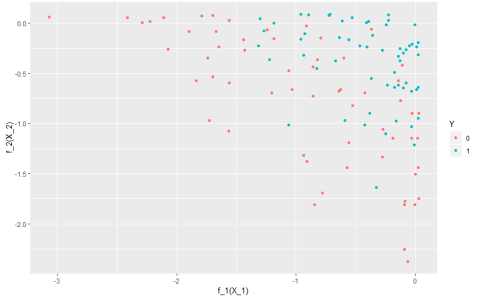
Ejemplo 5 - *yin-yang*, limitaciones del modelo

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.[10] 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

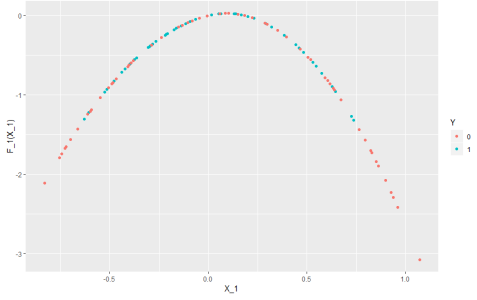
9. Dada la regularización, muchos de estos se desvanecían.
10. O al menos el presentado en este trabajo



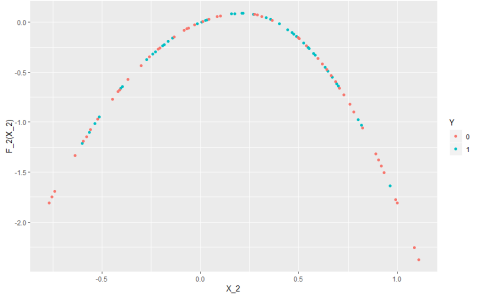
(a) Frontera



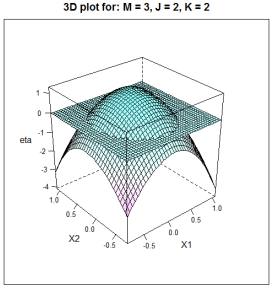
(b) Transformación no lineal



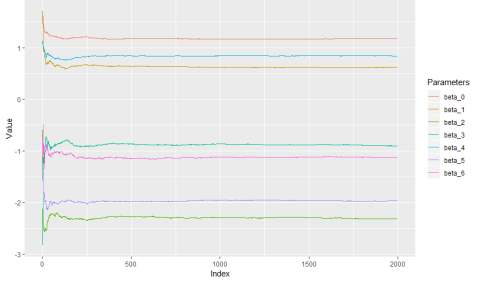
(c) $\hat{f}_1(x_1)$



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



(e) Representación 3D de $\hat{\eta}$

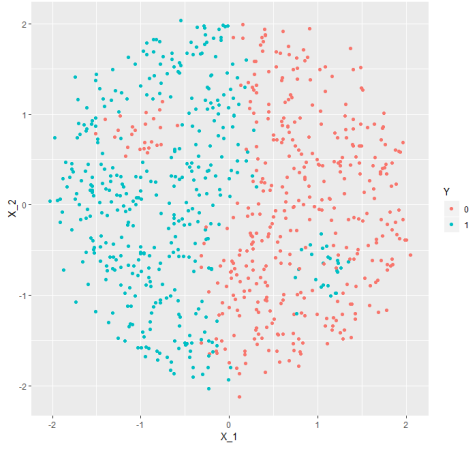


(f) Medias ergódicas

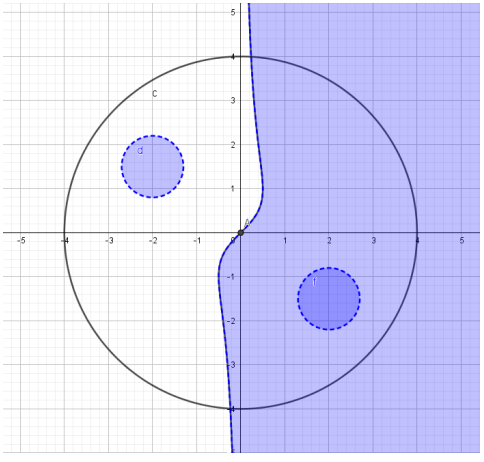
Figura 4.9: ejemplo 4 - parábolas suaves en un nodos ($M = 3$, $J = 2$ y $K = 2$)

reto donde se conjuntaron varias áreas de la matemática aplicada. En el software `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



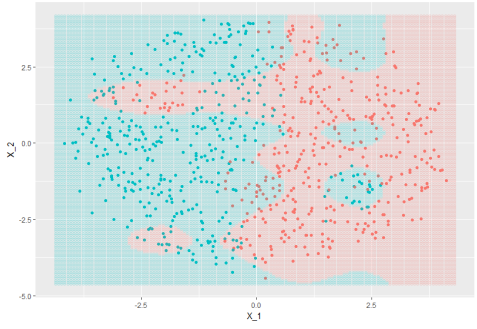
(b) Salida del software `GeoGebra`

Figura 4.10: Patrón yin-yang

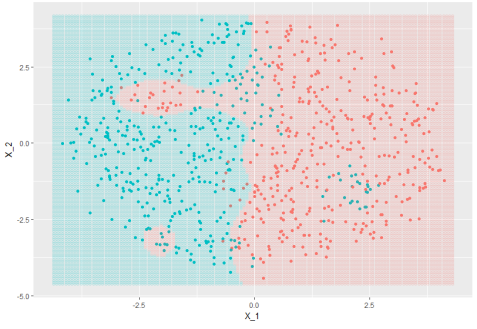
Una vez dibujadas las ecuaciones, se generó una base de datos de aproximadamente $n \approx 800$ observaciones con una distribución uniforme dentro del círculo usando coordenadas polares. A estos puntos se les asignó la categoría cero, posteriormente, se cambió la categoría a los puntos que cumplieran con las desigualdades. Poste-

riormente, se le añadió algo de ruido normal a cada punto para darle aleatoriedad a la base de datos pero manteniendo el patrón. Finalmente, se escala la base para centrarla en cero.

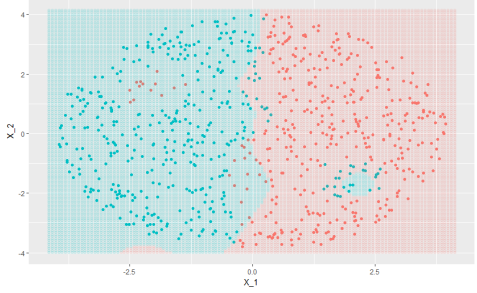
Se corrieron un sinfín de realizaciones del modelo, tratando de calibrar los parámetros y captar de la mejor manera posible el patrón. Sin embargo y aunque el modelo casi siempre lograba una precisión de cerca de 85 %, no se logra la clasificación esperada identificando los puntos de color dentro de 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. Para las dos primeras 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. En la imagen 4.11a, el modelo detecta relativamente bien la curva que separa las regiones, sino que detecta de forma aislada, el circulo azul de la esquina inferior derecha. Finalmente 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 cuadrícula donde las interacciones son difíciles de discernir. Conforme aumenta el número de nodos, más complejo se vuelve el modelo. Es por ello, que los picos y valles se repiten en un patrón uniforme. Asimismo, dada la naturaleza global de los polinomios y esta interacción, el modelo tiene esta estructura decreciente siempre, derivando que los picos y los valles nunca alcancen las regiones extremas en polos opuestos. De igual forma, la uniformidad y simetría impar, inherente a esta base de



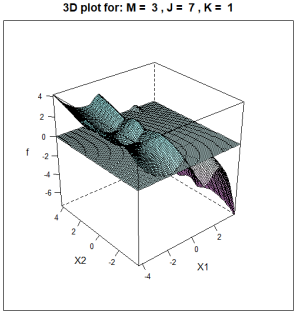
(a) Sobre-ajuste



(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

datos, llevó a que la estimación de los parámetros fuera óptima dentro de las capacidades del modelo. Otra desventaja de esta base, es que estos modelos se tuvieron que correr con un número grande de nodos $J \approx 20$, derivando en un número de parámetros aún más alto.

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 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.[11] En la figura 4.12, se muestran los gráficos de puntos pareados para todas las posibles combinaciones de covariables además de cierta información adicional. Este proceso se lleva a cabo para tratar de

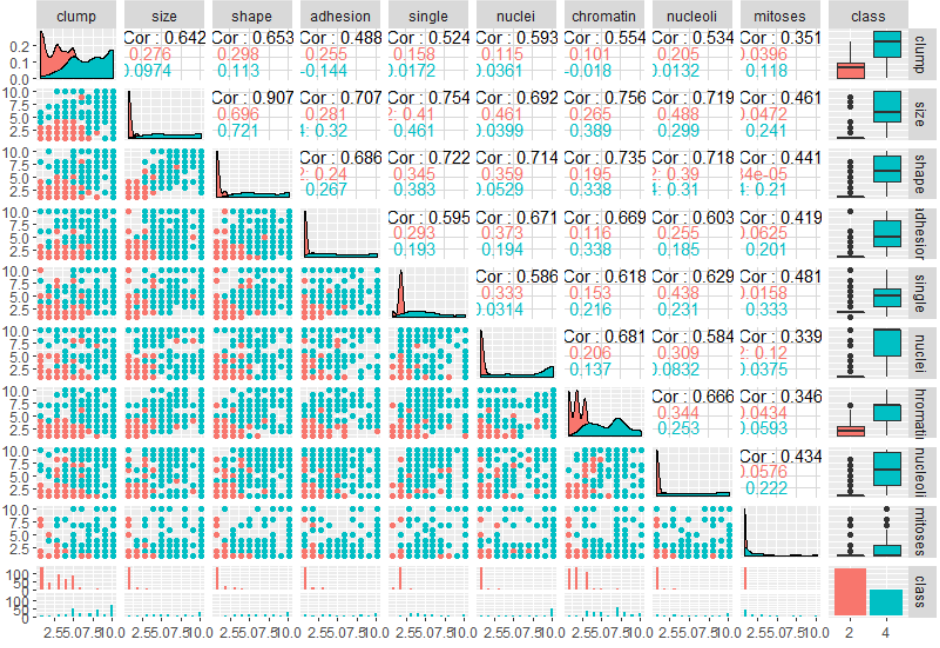
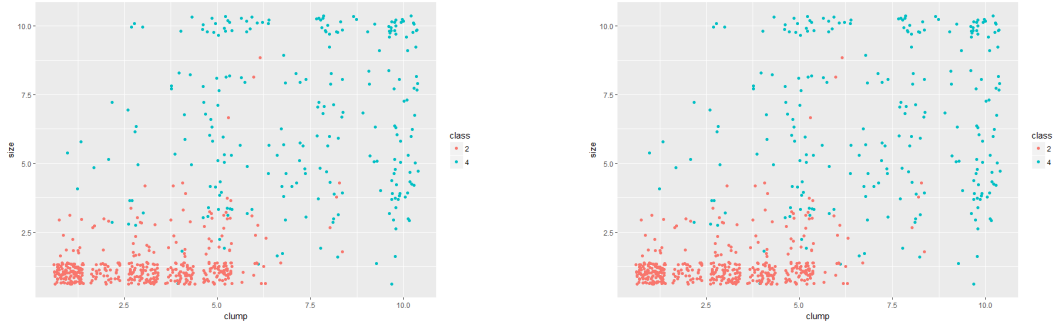


Figura 4.12: análisis exploratorio para selección de variables

seleccionar las covariables relevantes y estudiar un poco los datos. Asimismo, 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*[12] debido a que parecieran ser las que mejor separan el espacio.

11. Forma en la que se presenta la cadena de ADN en el núcleo celular.
12. Estas covariables corresponden a el espesor de los tumores, su tamaño y la textura de la cromatina en las células respectivamente.

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 a simple vista, además de que se tienen puntos idénticos con clasificaciones contrarias. Sin embargo, a simple vista se detecta cierto patrón en los datos. Para poder hablar de



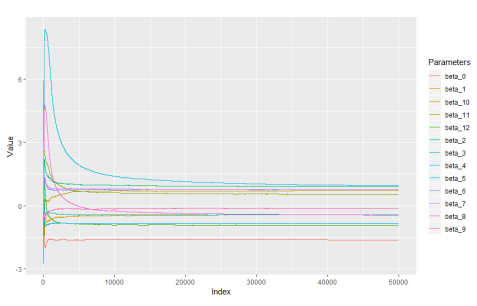
(a) Variables *clump* y *chromatin*

(b) Variables *clump* y *size*

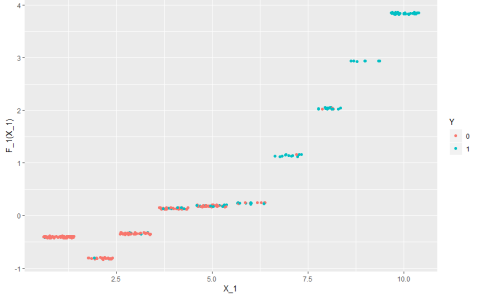
Figura 4.13: gráficos de puntos con ruido para separar las observaciones

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.[13] 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.

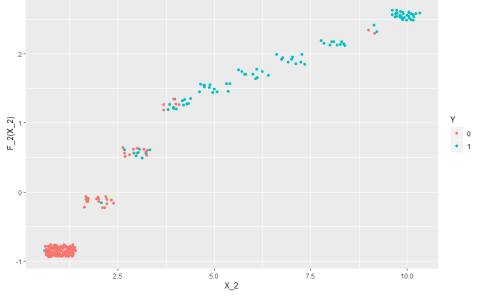
13. La diferencia de 16 observaciones entre la suma de entrenamiento y prueba, contra las 699 originales, se debe a que estas estaban incompletas y por lo tanto se descartan.



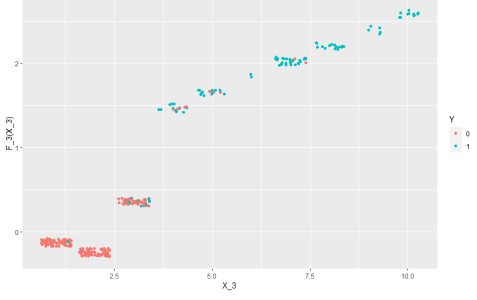
(a) Medias ergódica



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



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



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

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: datos médicos - resultados

Inclusive, 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 correctos M, J y K, el número total de parámetros ($\lambda = 13$) no necesita aumentar demasiado para lograr una buena separación del espacio.[14] 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. e 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.

14. Se corrieron otras realizaciones del modelo aumentando el número de covariables d. Sin embargo, no logró aumentar significativamente la precisión del modelo.

Capítulo 5

Conclusiones

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

5.1. Consideraciones finales del modelo

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

Convergencia y sus implicaciones

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

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

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

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

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

Calibración de los parámetros y velocidad del algoritmo

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

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

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

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

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

5.2. Posibles mejoras y actualizaciones

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

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

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

3. Usualmente el criterio de Akaike

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

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

Otra modificación considerada es volver el algoritmo de muestreo Gibbs en algo menos rígido. Como se menciona en el Capítulo 3, se toman distribuciones conjugadas para el proceso de aprendizaje bayesiano pues simplifica mucho la derivación de la ecuación (**??**), conviriendola en (3.7) lo cual permite que el muestreo sea sencillo, requiriendo únicamente álgebra lineal y simulaciones de distribuciones normales multivariadas. Aunque el supuesto no es malo, sería bueno poder incorporar distribuciones a priori arbitrarias, para poder reflejar conocimiento previo de la base de

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

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

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

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

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

5.3. El aprendizaje de una máquina

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

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

4. Usualmente de manera frecuentista.

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

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

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

Apéndice A

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

Apéndice C

Paquete en R: desarrollo y lista de funciones

Bibliografía

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

Alpaydin, Ethem. 2014. *Introduction to machine learning.* MIT press.

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

Barber, D. 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, A. 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 FM Smith. 2001. *Bayesian Theory.* John Wiley & Sons.

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

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

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

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

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

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

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

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

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

Hastie, T.J., y R.J. Tibshirani. 1990. *Generalized additive models.* Chapman & Hall, London.

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

Hoerl, Arthur E, y Robert W Kennard. 1970. «Ridge regression: Biased estimation for nonorthogonal problems». *Technometrics* 12 (1): 55-67.

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

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

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

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

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

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

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

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

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

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

Tibshirani, Robert. 1996. «Regression shrinkage and selection via the lasso». *Journal of the Royal Statistical Society. Series B (.*

Tierney, L. 1994. «Markov chains for exploring posterior distributions». *the Annals of Statistics:* 1701-1728.

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

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