> *[...], 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.*
>
> **barber2010bayesian**

Bajo la perspectiva de la cita de **barber2010bayesian**, escribir un algoritmo que aterrice la estructura matemática subyacente en una aplicación práctica, no resulta fácil. Sin embargo, con base en las ideas de **albert1993bayesian** la tarea se simplificó pues su definición induce el algoritmo (y viceversa). Al algoritmo también se le titula: *bayesian piece wise polinomial model (bpwpm)* y puede ser consultado en la página 8. Para facilitar la inferencia usando el 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`. En el apéndice **??** se dan más detalles sobre este proceso y se detalle el uso del paquete.

0.1. Aumentación de datos con variables latentes para respuestas binarias

En **albert1993bayesian**, los autores desarrollan métodos bayesianos para el análisis de respuestas binarias y policotómicas.[1] Para los objetivos del trabajo, en el caso bi-

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

nario su enfoque resultaba muy atractivo. Su modelo titulado *aumentación de datos para respuestas binarias*,[2] propone una definición del modelo probit como la presentada en (**??**) y (**??**), donde la introducción de las variables latentes es clave. Bajo esta definición, la derivación de las distribuciones marginales de los parámetros se vuelve una tarea relativamente fácil.[3] Asimismo, proponen usar distribuciones conjugadas normales para los parámetros $\boldsymbol{\beta}$ derivando en un algoritmo relativamente rápido pues la parte estocástica depende únicamente de simular distribuciones conocidas. Esto lleva a que los periodos de *burn-in* sean relativamente pequeños y que el adelgazamiento no sea fundamentalmente necesario.

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

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

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

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

donde $\widetilde{\boldsymbol{\psi}}_i(\mathbf{x}_i)$ es el renglón i de la matriz de transformación (**??**) presentada en la página **??**, lineal en el espacio transformado $\widetilde{\Psi}(\mathcal{X})$.[4] Sin embargo bajo el paradigma bayesiano, como el modelo recae en la definición de la variable latente $\mathbf{z}$ (descono-

2. *data augmentation for binary data*

3. **albert1993bayesian** también proponen un modelo con función liga t-student dando lugar a un modelo *tobit*.

4. Se enfatiza que visto de esta forma, el modelo es lineal tanto en parámetros como en estas nuevas covariables transformadas.

cidas pero modeladas con una distribución normal) éstas 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 (**??**) se tienen entonces dos grupos de parámetros: $\boldsymbol{\theta} = (\mathbf{z}, \boldsymbol{\beta}) \Rightarrow \pi(\boldsymbol{\theta} \,|\, \mathbf{y}, \mathbf{X}) = \pi(\mathbf{z}, \boldsymbol{\beta}|\mathbf{y}, \mathbf{X})$. Por lo tanto, la derivación de la densidad posterior resulta en:

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

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

Bajo los fundamentos del muestreador de Gibbs, dado que muestrear de (1) es complejo, se busca derivar entonces las distribuciones condicionales de $\mathbf{z}$ y $\boldsymbol{\beta}$. Para $\boldsymbol{\beta}$, la

densidad marginal condicional ésta entonces dada por:

$$\pi(\boldsymbol{\beta} \,|\, \mathbf{z}, \mathbf{y}, \mathbf{X}) = \frac{\pi(\mathbf{z}, \boldsymbol{\beta} \,|\, \mathbf{y}, \mathbf{X})}{\pi(\mathbf{z})} \tag{2}$$

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

$$= \frac{\pi(\mathbf{y} \,|\, \mathbf{z})}{\pi(\mathbf{y}, \mathbf{X})\ \pi(\mathbf{z})} \xrightarrow{C_1} \times\, \pi(\mathbf{z} \,|\, \boldsymbol{\beta}, \mathbf{X})\ \pi(\boldsymbol{\beta}) \tag{3}$$

$$= C_1\, \pi(\boldsymbol{\beta}) \prod_{i=1}^{n} \phi(z_i \,|\, \eta(\mathbf{x}_i), 1), \tag{4}$$

Se hace notar que (**??**) es la misma expresión que se derivaría si se tuviera una regresión lineal bayesiana teniendo a z de regresor, es decir, el modelo $z_i = \boldsymbol{\beta}^t \widetilde{\boldsymbol{\psi}}_i(\mathbf{x}_i) + e_i$ con $e_i \sim \mathcal{N}(0, 1)$ y z_i conocidas. De lo anterior, se observa la utilidad de la definición particular usada: convierte una clasificación probit a una regresión lineal haciendo uso de las variables latentes $\mathbf{z}$ como regresores, asociadas de manera unívoca con la respuesta binaria $\mathbf{y}$. Asimismo, la ecuación (**??**) se toma de la definición de probabilidad condicional, y el paso de (**??**) a (**??**) se puede hacer ya que, al definir $\mathbf{y}$ como en la ecuación (**??**), sus representaciones son análogas y el cociente se desvanece en una constante C_1 que no depende de $\boldsymbol{\beta}$.

Únicamente falta por definir a $\pi(\boldsymbol{\beta})$. En la práctica es común usar distribuciones *no informativas* sobre los parámetros de interés, cuando no se tiene experiencia sobre ellos. Sin embargo, para el modelo lineal bayesiano, existe una familia de distribuciones conjugadas, que son razonables para la aplicación que se busca, además, derivan

en resultados cerrados. En particular, se elige la distribución $\pi(\boldsymbol{\beta})$ como:

$$\boldsymbol{\beta} \sim \mathcal{N}_\lambda(\boldsymbol{\beta} \,|\, \boldsymbol{\mu}_{\boldsymbol{\beta}}, \Sigma_{\boldsymbol{\beta}}), \tag{5}$$

con el hiper-parámetro de media $\boldsymbol{\mu}_{\boldsymbol{\beta}} \in \mathbb{R}^\lambda$ y la matriz de covarianza $\Sigma_{\boldsymbol{\beta}} \in \mathbb{R}^{\lambda \times \lambda}$. Sustituyendo (2) en (**??**) y usando resultados estándar de modelos lineales (**banerjee2008gory**), se deriva que la densidad marginal conjugada para los parámetros es:

$$\boldsymbol{\beta} \,|\, \mathbf{y}, \mathbf{z}, \mathbf{X} \sim \mathcal{N}_\lambda(\boldsymbol{\beta} \,|\, \boldsymbol{\mu}^*_{\boldsymbol{\beta}}, \Sigma^*_{\boldsymbol{\beta}}), \tag{6}$$

donde,

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

Esta distribución es conjugada pues preserva la estructura normal de los parámetros, es decir, tanto la distribución inicial como la distribución posterior de $\boldsymbol{\beta}$ pertenecen a la familia de distribución normal, en donde únicamente se tienen que actualizar los hiperpámetros $\boldsymbol{\mu}_{\boldsymbol{\beta}}$ y $\Sigma_{\boldsymbol{\beta}}$. Sobre este proceso de actualización recae la idea de *aprendizaje bayesiano* pues se está actualizando la medida de incertidumbre en la presencia de nueva evidencia. Otra ventaja de usar distribuciones conjugadas es que son fáciles simular usando cualquier software estadístico, calculando previamente la media y covarianza y dando un valor (o iteración) para $\mathbf{z}$.[5] Con base en **banerjee2008gory**,

5. Se hace notar, que este estimador, es relativamente similar al estimador que se usa en una regresión *Ridge*, (**tibshirani1996regression**).

en el apéndice **??** se completan todos los pasos de esta derivación.

Ahora, condicionar sobre $\mathbf{z}$ es más sencillo y la derivación resulta similar. Comenzando con la expresión (1) y re-ordenando términos se tiene:

$$\begin{aligned}
\pi(\mathbf{z} \,|\, \boldsymbol{\beta}, \mathbf{y}, \mathbf{X}) &= \frac{\pi(\mathbf{z}, \boldsymbol{\beta} \,|\, \mathbf{y}, \mathbf{X})}{\pi(\boldsymbol{\beta})} \\
&= \frac{\pi(\mathbf{y} \,|\, \mathbf{z})\ \pi(\mathbf{z} \,|\, \boldsymbol{\beta}, \mathbf{X})\ \pi(\boldsymbol{\beta})}{\pi(\mathbf{y}, \mathbf{X})\ \pi(\boldsymbol{\beta})} \\
&= \cancel{\frac{1}{\pi(\mathbf{y}, \mathbf{X})}}^{C_1} \quad \pi(\mathbf{y} \,|\, \mathbf{z}) \times \pi(\mathbf{z} \,|\, \beta, \mathbf{X}) \\
&= C_1 \prod_{i=1}^{n} [I(y_i = 1) I(z_i > 0) + I(y_i = 0) I(z_i \leq 0)] \\
&\qquad \times \phi(z_i \,|\, \eta(\mathbf{x}_i), 1). \qquad (7)
\end{aligned}$$

De donde se observa que cada z_i es independiente (por el teorema de factorización) con distribución normal truncada en 0, es decir $\forall i = 1, \ldots, n$:

$$\begin{aligned}
z_i | y_i, \boldsymbol{\beta} &\sim \mathcal{N}(z_i | \boldsymbol{\beta}^t \widetilde{\boldsymbol{\psi}}_i(\mathbf{x}_i), 1) I_{(z_i > 0) I(y_i = 1)} \quad \text{truncamiento a la izquierda} \qquad (8) \\
z_i | y_i, \boldsymbol{\beta} &\sim \mathcal{N}(z_i | \boldsymbol{\beta}^t \widetilde{\boldsymbol{\psi}}_i(\mathbf{x}_i), 1) I_{(z_i \leq 0) I(y_i = 0)} \quad \text{truncamiento a la derecha.}
\end{aligned}$$

Resulta que estas distribuciones también son fáciles de simular usando los algoritmos de **devroye1986non**.

0.2. Implementación algorítmica final

Finalmente se está en la posición de presentar el modelo en su versión final al haber definido todos sus componentes.[6]

Definición 0.1. El modelo *bpwpm* (final), aumentando sobre la definición **??**, $\forall i = 1, \ldots, n$:

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

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

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

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

$$= \sum_{\hat{\imath}=1}^{M-1} \beta_{j,\hat{\imath},0}\; x_{i,j}^{\hat{\imath}} + \sum_{\hat{\imath}=K}^{M-1} \sum_{\hat{\jmath}=1}^{J-1} \beta_{j,\hat{\imath},\hat{\jmath}}\; (x_{i,j} - \tau_{j,\hat{\jmath}})_+^{\hat{\imath}}. \tag{9}$$

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

$$N^* = JM - K(J-1) - 1 \tag{10}$$

$$\boldsymbol{\beta} \sim \mathcal{N}_\lambda(\boldsymbol{\beta} \,|\, \boldsymbol{\mu}_{\boldsymbol{\beta}}, \Sigma_{\boldsymbol{\beta}}) \qquad (\lambda = 1 + d \times N^*) \tag{2}$$

La ecuación (6) no es más que la expansión (**??**) presentada en la página **??** sobre

6. El modelo en su versión más completa puede resultar algo pesado en notación, sin embargo, se recuerda que existe un compendio de esta al inicio del trabajo.

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

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

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

$$\boldsymbol{\beta} \sim \mathcal{N}_\lambda(\boldsymbol{\beta} \,|\, \mu_{\boldsymbol{\beta}}, \Sigma_{\boldsymbol{\beta}}) \qquad (\lambda = 1 + d \times N^*) \tag{2}$$

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

con $\eta(\mathbf{x}_i) = \boldsymbol{\beta}^t \widetilde{\boldsymbol{\psi}}_i(\mathbf{x}_i)$ el i-ésimo renglón de $\boldsymbol{\eta}(\mathbf{X})$. De estas expresiones y juntándolo con el muestreador de Gibbs (**??**) definido por las distribuciones marginales de $\boldsymbol{\beta}$ y $\mathbf{z}$, (3) y (5) respectivamente, se presenta el algoritmo final en la página 8. El valor inicial $\mathbf{z}^{(0)}$, en realidad no se tiene que proporcionar pues se simula dependiendo de $\mathbf{y}$ y $\boldsymbol{\beta}^{(0)}$. Este valor inicial $\beta^{(0)}$ es arbitrario, pero se sugiere en **albert1993bayesian** que sea dado por el estimador de máxima verosimilitud o el de mínimos cuadrados para las respuestas binarias $\beta^{(0)} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{y}$. Sin embargo en la práctica, el algoritmo inizializa los parámetros en ceros por defecto. En la primera iteración, se esparcen por el espacio y van convergiendo a la distribución límite en relativamente poco tiempo como se observará en el capítulo **??**.

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 **??** se hace un compendio de las funciones y una breve descripción de su

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

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

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

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

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

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

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

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

Inicializar un vector de tamaño λ que contendrá las 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

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

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

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

fin

uso. Se exhorta al lector probarlo por si mismo.

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

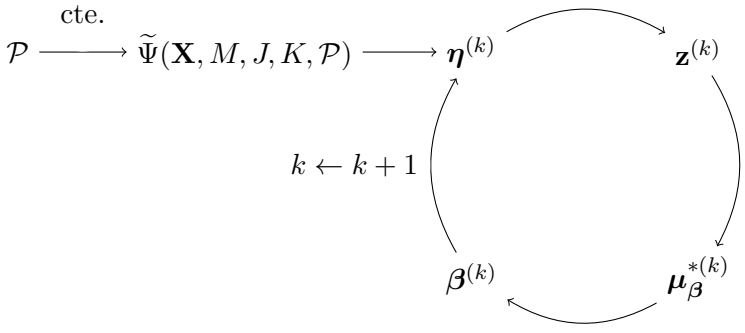

Figura 1: Esquema del proceso algorítmico

Ya que el modelo tiene muchos componentes y pasos intermedios, la figura 1 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.[7] 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. Definidos los componentes de la media actualizada, se calcula esta $\mu_{\boldsymbol{\beta}}^{*(k)}$ y se simulan los parámetros $\boldsymbol{\beta}$ que tienen distribución normal condicionada en $\mathbf{z}$. Finalmente se aumenta el contador en uno se repite el procedimiento. En cada iteración los parámetros se guardan en un objeto que la rutina regresa por completo

7. La implementación computacional de $\widetilde{\Psi}$, se basa en el diagrama **??** de la página **??** y la expresión (**??**). La sub-rutina que realiza la expansión tiene el nombre de `calculate_Psi` en el paquete y está vectorizada para que su ejecución sea veloz.

para su exploración posterior, visualización y validación. Se hace notar que $\Sigma^*_{\boldsymbol{\beta}}$ se tiene que calcular únicamente una vez pues no depende ni de $\mathbf{z}^{(k)}$ ni de $\boldsymbol{\beta}^{(k)}$.