UNIVERSIDAD DE SALAMANCA
DEPARTAMENTO DE ESTADÍSTICA
DOCTORADO EN ESTADÍSTICA MULTIVARIANTE APLICADA

CONTRIBUCIONES AL BIPLOT LOGÍSTICO BINARIO

TESIS DOCTORAL

JOSÉ GIOVANY BABATIVA MÁRQUEZ

DIRECTOR:
JOSE LUIS VICENTE-VILLARDÓN

SALAMANCA, ESPAÑA
2022

CONTRIBUCIONES AL BIPLOT LOGÍSTICO BINARIO

VNiVERSiDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

DEPARTAMENTO DE ESTADÍSTICA

Memoria para optar al Grado de Doctor
en Estadística Multivariante Aplicada
por el Departamento de Estadística de la
Universidad de Salamanca, presenta:

José Giovany Babativa Márquez

Salamanca

2022

VNiVERSiDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

DR. JOSE LUIS VICENTE VILLARDÓN

Profesor titular del Departamento de Estadística de la Universidad de Salamanca

---

CERTIFICA:

Que D. **JOSÉ GIOVANY BABATIVA MÁRQUEZ**, graduado en estadística, ha realizado en el Departamento de Estadística de la Universidad de Salamanca, bajo su dirección, el trabajo que para optar al Grado de Doctor en Estadística Multivariante Aplicada, que presenta con el título **CONTRIBUCIONES AL BIPLOT LOGÍSTICO BINARIO**, autorizando expresamente su lectura y defensa. Y para que conste, firma el presente certificado en Salamanca a 20 de junio de 2022.

Jose Luis Vicente Villardon

Firmado digitalmente por Jose Luis Vicente Villardon
Fecha: 2022.06.20 17:45:59 +02'00'

Jose Luis Vicente Villardón

"Nunca consideres el estudio como una obligación,
sino como una oportunidad para penetrar en el
bello y maravilloso mundo del saber"

Albert Einstein.

"La educación es el descubrimiento gradual
de nuestra propia ignorancia"

Will Durant

A mis hijas Salomé, Juanita y mi esposa Katerina.

Agradecimientos

Quiero expresar un profundo y sincero agradecimiento a la Dra. Purificación Galindo y al Dr. José Luis Vicente-Villardón por todo el apoyo que me dieron desde el momento en que llegué con mi familia a Salamanca, por todo el conocimiento que me brindaron, por sus sabios consejos, no sólo a nivel académico sino para la vida misma; los momentos que compartimos hicieron que este doctorado fuera una aventura maravillosa. Las palabras se quedan cortas para expresar todo mi sentimiento de gratitud, realmente disfruté cada instante de esta experiencia gracias a ustedes.

No puedo dejar de mencionar a quienes emprendieron esta travesía a mi lado y fueron el motor que me animó a seguir adelante. Ellas son, mis hijas, Salomé y Juanita, y nada de esto hubiera sido posible sin el apoyo incondicional de mi amada esposa Katerina, gracias por animarme, por tu paciencia y por todo el amor con el que siempre haces las cosas. Esta experiencia no hubiera sido lo mismo sin ustedes, que son la luz que guía mi camino y hacen que mi vida tenga un sentido especial.

A mi madre, Zaide Márquez, por impulsarme siempre a cumplir mis sueños. A mi segunda madre, la Sra. Ana, por su amor incondicional y porque siempre estuvo muy pendiente de nosotros en la distancia. A mi hermana Yesenia y mi cuñada Diana por encargarse de todos los chicharrones en nuestra ausencia, y a mis hermanos Andrés y Karen, que siempre estuvieron pendientes de lo que necesitáramos.

Por supuesto, quiero dar las gracias a Dios, porque sin él nada de esto hubiera sido posible.

Esta experiencia también nos permitió conocer personas estupendas. A Carmén y Raúl que fueron como nuestra familia en Salamanca, se cierra esta etapa, pero quedan grandes amigos.

Finalmente, quiero agradecer a todos aquellos que de alguna manera contribuyeron a que este ciclo de mi vida fuera una realidad. ¡Muchas gracias!

Resumen

Con los avances tecnológicos también se ha generado un crecimiento masivo en la cantidad y variedad de datos, esto nos brinda la oportunidad de tener una comprensión más profunda pero también introduce grandes desafíos estadísticos. Esto ha llevado a que se generen nuevas líneas de investigación que combinan los métodos estadísticos con los desarrollos en informática, y así implementar nuevas herramientas que permitan modelar y comprender conjuntos de datos complejos.

Los métodos de ordenación y reducción de la dimensionalidad son utilizados con frecuencia porque permiten simplificar los análisis con la mínima pérdida de información. En este contexto, los métodos biplot son una variedad de técnicas multivariantes que permiten reducir y visualizar de forma simultánea la información de un conjunto de datos, y han contribuido al avance de la ciencia por más de cinco décadas. Los aportes realizados en los métodos biplot han permitido que las técnicas puedan ser aplicadas en diferentes áreas del conocimiento, facilitando la toma de decisiones.

Inicialmente el biplot fue propuesto como una extensión del análisis de componentes principales basado en la descomposición en valores singulares y luego fue extendido para visualizar los resultados de otros métodos. Uno de estos se denomina biplot logístico, que es un tipo de biplot lineal para datos binarios que permite modelar la relación entre las variables observadas y las dimensiones del biplot a través de una curva de respuesta logística.

Este trabajo presenta contribuciones para los casos donde la matriz de información es binaria, proponiendo métodos que faciliten el análisis para grandes volúmenes de información, haciendo un aporte novedoso al combinar el biplot logístico con los métodos de optimización

aplicados en el contexto de machine learning y utilizando los desarrollos informáticos disponibles en la actualidad.

En este proyecto se investiga y se propone una metodología basada en validación cruzada que es adaptada para el biplot logístico, con el fin de contar con un método que permita identificar el número de dimensiones que son apropiadas para ajustar el modelo. De este procedimiento se obtiene un error de entrenamiento y un error de validación que pueden ser ilustrados en una gráfica y así visualizar el valor apropiado para el número de dimensiones que debe ser elegido.

De otra parte, con el fin de contribuir al proceso de análisis multivariante para matrices de datos binarias de tipo *big data*, se incorporan nuevas formulaciones que permiten obtener funciones de pérdida adecuadas para ajustar el biplot logístico cuando se tiene un alto volumen de datos. Para ello se realizan diferentes desarrollos teóricos que son postulados y demostrados en algunos teoremas. A partir de las funciones que permiten sustituir el problema de optimización por otro más simple, se realiza el desarrollo teórico para adaptar diferentes algoritmos que permiten estimar los parámetros del modelo. Asimismo, se explora un enfoque a partir de algoritmos basados en el gradiente conjugado. Para comparar el rendimiento de los algoritmos se usa un procedimiento de simulación que permite medir la capacidad que tienen los diferentes métodos para identificar el número de dimensiones del modelo y la habilidad que tienen para recuperar la matriz canónica de parámetros en escenarios con matrices balanceadas y en otros donde la matriz de datos está desequilibrada.

Partiendo de que la matriz de datos binaria puede estar incompleta, se incorpora una metodología que permite dar un tratamiento a los datos faltantes. Esta se desarrolla desde una nueva perspectiva que está basada en el método de proyección de datos propuesto por Pearson para un análisis de componentes principales. En este trabajo se realiza el desarrollo teórico que permite llegar a un problema de minimización y un algoritmo apropiado para obtener una solución al problema, con la ventaja de que las entradas faltantes en la matriz binaria también se van optimizando mientras se realiza el ajuste del modelo. Este enfoque además permite obtener la matriz de marcadores fila como una función de los marcadores columna, permitiendo la proyección de filas suplementarias sin tener que realizar nuevamente el proceso de optimización.

Con el fin de ilustrar su uso práctico y la interpretación de los resultados, los métodos propuestos son aplicados usando conjuntos de datos reales en diferentes contextos. Finalmente, para dar un soporte práctico a los investigadores de las diferentes áreas del conocimiento, los métodos propuestos y desarrollados teóricamente, son puestos a disposición en un paquete escrito en lenguaje R, denominado *BiplotML*, el cual cuenta con toda la documentación de ayuda y puede ser instalado desde el repositorio de CRAN.

Índice general

Índice de tablas

Índice de figuras

Lista de algoritmos

Notación

n	Número de filas de la matriz $\mathbf{X}$
p	Número de columnas de la matriz $\mathbf{X}$
$rank(\mathbf{X}) = r$	Rango de la matriz $\mathbf{X}$
k	Número de dimensiones
α	Velocidad de aprendizaje
δ_j	Umbral para aplicar la regla de clasificación para la variable j
C	Valor constante
D	Grado de desequilibrio de una matriz binaria
M	Número de pliegues o segmentos en el proceso de validación cruzada
R	Número de réplicas
d_l	Dirección de actualización del gradiente en el paso l
$G(\theta)$	Factor de normalización logaritmico para una distibución $p(\mathbf{x}_i, \theta)$
$\pi(\cdot)$	Inversa de la función de enlace logística
$\mathbf{x}_i \in \mathbb{R}^p$	Vector números reales de longitud p
$\mathbf{x}_i \in \{0, 1\}^p$	Vector binario de unos y ceros de longitud p
$\mathbf{b}_j \in \mathbb{R}^k$	marcadores para la i-ésima columna
$\mathbf{a}_i \in \mathbb{R}^p$	marcadores para la i-ésima fila
μ	Vector de desplazamiento en el modelo de biplot logístico
$\mathbf{1}_n$	Vector de unos de tamaño $n \times 1$
$\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$	Matriz de datos conformada por n vectores $\mathbf{x}_i$, $i = 1, \ldots, n$
$\mathbf{I}$	Matriz identidad
$\mathbf{U}$	Matriz formada por los vectores singulares izquierdos de $\mathbf{X}$

$\mathbf{V}$	Matriz formada por los vectores singulares derechos de $\mathbf{X}$
$\mathbf{\Lambda}$	matriz diagonal p-dimensional de los valores singulares de $\mathbf{X}$
$\mathbf{A}$	Matriz de marcadores fila
$\mathbf{B}$	Matriz de marcadores columna
$\mathbf{\Theta}$	Matriz canónica de parámetros
$\Pi = \pi(\mathbf{\Theta})$	Matriz de probabilidades esperada
$\mathbf{E}$	Matriz con los errores de aproximación
$\mathbf{H}$	Matriz Hessiana
$\mathbf{S}$	Matriz de similaridades entre filas
$\mathbf{\Delta}^2$	Matriz de cuadrados de las distancias
$\mathbf{W}$	Matriz binaria que codifica las entradas faltantes, donde $w_{ij} = 1$ si x_{ij} es conocido y $w_{ij} = 0$ en otro caso
$\|\mathbf{x}\| = \sqrt{\mathbf{x}^T\mathbf{x}}$	Norma de un vector $\mathbf{x}$
$\|\mathbf{X}\|_F^2 = \mathrm{tr}\left(\mathbf{X}^T\mathbf{X}\right)$	Norma de Frobenius de una matriz $\mathbf{X}$ al cuadrado
$\mathrm{tr}(\mathbf{X})$	Traza de una matriz $\mathbf{X}$
$\mathrm{Vec}(\mathbf{X})$	Operador de vectorización para una matriz $\mathbf{X}$
$\mathbf{D}_X$	Matriz diagonal que contiene los elementos de $\mathrm{Vec}(\mathbf{X})$
$\odot$	Producto de Hadamard

Abreviaturas

AF	Análisis Factorial
BCRA	Carcioma Invasivo de Mama
BCV	Validación Bi-Cruzada
CNA	Copia del Número de Alteraciones
cv error	Error de validación cruzada
ELB	Biplot Logístico Externo
FN	Cantidad de Falsos Negativos
FP	Cantidad de Falsos Positivos
IRT	Teoría de Respuesta al Ítem
LB	Biplot Logístico
LTA	Análisis de Rasgos Latentes
LUAD	Adenocarcinoma de Pulmón
MM	Mayorización y Minimización
PCA	Análisis de Componentes Principales
PCoA	Análisis de Coordenadas Principales
RMSE	Error Cuadrático Medio Relativo
SKCM	Melanoma Cutáneo de Piel
SVD	Descomposición de Valores Singulares
TEE	Tasa de Error Equilibrada
TPE	Tasa de Precisión Equilibrada
VN	Cantidad de Verdaderos Negativos
VP	Cantidad de verdaderos Positivos

Introducción

En los últimos años se ha producido un aumento de los datos de alta dimensión en diversas áreas del conocimiento. En estos casos, las técnicas multivariantes son especialmente útiles para capturar las estructuras subyacentes que permiten comprender las asociaciones presentes en los datos.

Los métodos biplot permiten visualizar una matriz de datos $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$, con $\mathbf{x}_i \in \mathbb{R}^p$, $i = 1, \ldots, n$, a partir de un sistema de coordenadas fila y columna Gower y col. (2011), el término biplot fue introducido por Gabriel (1971) en el contexto del análisis de componentes principales (PCA) para representar las variables a través de vectores dirigidos sobre el plano de coordenadas, desde entonces varias investigaciones han mostrado las ventajas del uso de la técnica y se ha implementado para visualizar los resultados de otras técnicas multivariantes como el escalamiento multidimensional, análisis de correspondencias, modelos lineales generalizados, HJ-Biplot, entre otras (Galindo Villardón, 1986; Gower y Hand, 1995; Gower y col., 2011; Greenacre y Blasius, 2006; Hernández-Sánchez y Vicente-Villardón, 2017).

Asumiendo que las variables están centradas y que $rank(\mathbf{X}) = r$, de acuerdo con (Eckart y Young, 1936) las coordenadas del biplot se pueden calcular usando la descomposición de valores singulares (SVD), $\mathbf{X} = \mathbf{U}\Lambda\mathbf{V}^T$ donde $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_r]$ y $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_r]$ son las matrices formadas por los vectores singulares por izquierda y por derecha de la matriz $\mathbf{X}$; Λ es la matriz diagonal p-dimensional formada por los valores singulares ordenados de forma decreciente $\lambda_1 \geq \ldots \geq \lambda_r > 0$. Con lo cual es posible aproximar la matriz $\mathbf{X} = \mathbf{AB^T} + \mathbf{E}$ donde $\mathbf{E}$ es la matriz que contiene los errores de la aproximación. Para un entero $k \leq r$, se obtiene la aproximación de rango k más cercana a $\mathbf{X}$ como $\hat{\mathbf{X}} = \mathbf{U}_{(k)}\Lambda_{(k)}\mathbf{V}_{(k)}^T = \mathbf{AB}^T$, donde $\mathbf{A} = \mathbf{U}_{(k)}\Lambda_{(k)}^{\gamma}$ y $\mathbf{B} = \mathbf{V}\Lambda_{(k)}^{(1-\gamma)}, 0 \leq \gamma \leq 1$; así, $\hat{\mathbf{X}}$ minimiza la norma de Frobenius

definida como

$$\left\|\mathbf{X}-\hat{\mathbf{X}}\right\|_F^2=\sum_{i=1}^{n}\left\|\mathbf{x}_i-(\mathbf{a}_{i1}\mathbf{b}_1+\ldots+\mathbf{a}_{ik}\mathbf{b}_k)\right\|^2. \tag{1}$$

De esta manera, la matriz $\mathbf{X}$ puede ser representada por marcadores $\mathbf{a_1},\ldots,\mathbf{a_n}$ para las filas y $\mathbf{b_1},\ldots,\mathbf{b_r}$ para las columnas, donde el ij-ésimo elemento de la matriz denotado x_{ij} es aproximado por el producto escalar $\mathbf{a_i^T b_j}$ y el espacio natural de parámetros está determinado por $\mathbf{\Theta}=\mathbf{AB}^T$.

Las elecciones más usuales para γ son los valores $0, 1$ y $\frac{1}{2}$. Cuando $\gamma=1$ los marcadores de las filas son las coordenadas sobre las componentes principales y el biplot se denomina JK-Biplot. Si $\gamma=0$ los marcadores de las filas son las coordenadas sobre las componentes principales estandarizadas, mientras que los marcadores de las columnas son las saturaciones de la matriz factorial en un Análisis Factorial cuando se usa una solución de Componentes Principales, a este biplot se denomina GH-Biplot. Si $\gamma=1/2$ se obtiene un biplot simétrico que no puede relacionarse específicamente con las técnicas conocidas. La elección de γ no afecta la aproximación de los elementos de la matriz inicial y su elección se basa en el interés que se tenga sobre los análisis de los individuos (filas) o de las variables (columnas). Una representación gráfica de la matriz $\mathbf{X}$, que permite que ambos marcadores puedan ser superpuestos en un mismo sistema de referencia con máxima calidad de representación fue propuesto en Galindo Villardón (1986) y, Galindo y Cuadras (1986) denominado HJ-Biplot.

Con el propósito de visualizar los productos escalares, que representan la aproximación de los elementos de la matriz de datos, $x_{ij}\approx\mathbf{a}_i^T\mathbf{b}_j$. Gower y Hand (1995) usan el hecho de que el producto escalar $\mathbf{a}_i^T\mathbf{b}_j$ es una constante para todos los puntos sobre la línea que proyecta $\mathbf{a}_i$ sobre $\mathbf{b}_j$ y proponen que las variables se representen usando ejes calibrados para que la proyección ortogonal de un punto fila sobre estos ejes aproxime el valor en la matriz de datos, y de esta forma la lectura sea similar a la que se realiza en los diagramas de dispersión.

La geometría de los biplot para subespacios lineales se describe en Gower y Hand (1995) y, Gower y col. (2011), mientras que Vicente-Villardon y col. (2006) presentan y desarrollan la geometría para el biplot de regresión. Suponga que en un biplot con $k=2$ dimensiones se quiere encontrar la dirección β_j en el espacio L generado por las dos columnas de $\mathbf{A}$, de manera que las proyecciones de los marcadores de $\mathbf{A}$ sobre esa dirección generen la mejor

predicción de los valores de la variable j, $\mathbf{x_j} \approx \mathbf{A}\beta_j$, donde la dirección está determinada por los marcadores de la j-ésima columna.

La Figura 1 ilustra la geometría para un biplot en un espacio de dos dimensiones, donde H es el plano que se obtiene al agregar una tercera dimensión para la variable j. El conjunto de puntos de este plano que predicen un valor fijo para $\mathbf{x}_j$ están dados por la línea recta que resulta de la intersección entre el plano H y el plano paralelo a L que pasa por el valor fijo. De esta forma, valores distintos se asocian con líneas paralelas diferentes en el plano H. Al considerar la recta ξ_j como aquella que es ortogonal a todas las líneas paralelas, esta define el eje de referencia que se usa para la predicción. Asimismo, se observa que los puntos de L que predicen distintos valores también están en líneas rectas paralelas; la proyección de ξ_j sobre L es perpendicular a dichas rectas, generando el eje biplot con la dirección del vector β_j (Vicente-Villardon y col., 2006).

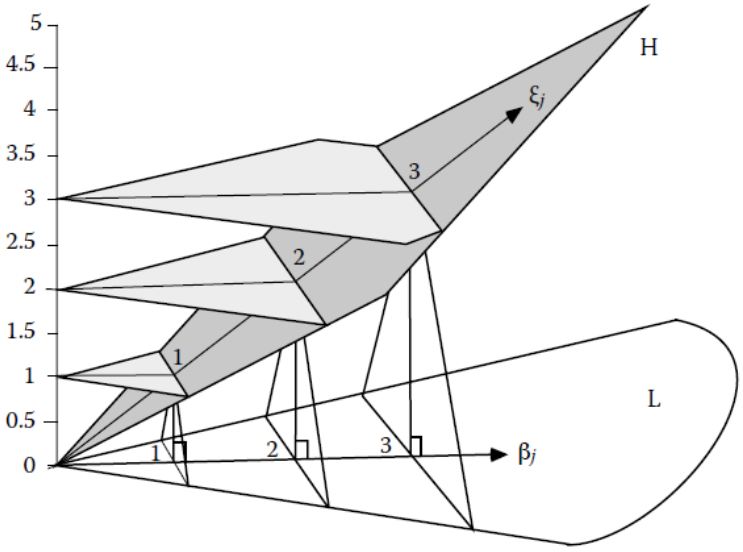


Figura 1: Geometría del biplot lineal. Tomado de Vicente-Villardon y col. (2006).

Para encontrar los marcadores sobre el eje biplot, $\beta_j = (b_{j1}, b_{j2})$, que predice un valor fijo ζ en la variable j, Vicente-Villardon y col. (2006) muestran que la solución para las coordenadas en las dos primeras dimensiones, (d_1, d_2), es:

$$d_1 = \frac{(\zeta - b_{j0})\, b_{j1}}{b_{j1}^2 + b_{j2}^2} \quad y \quad d_2 = \frac{(\zeta - b_{j0})\, b_{j2}}{b_{j1}^2 + b_{j2}^2}. \tag{2}$$

La bondad de ajuste se mide mediante los coeficientes de determinación calculados para las regresiones y pueden ser interpretados como la medida de calidad de la representación para

las variables. Estos métodos han sido muy utilizados para visualizar una matriz de datos o para encontrar las asociaciones presentes en dicha matriz y siguen siendo muy populares en diferentes contextos (Amor-Esteban y col., 2019; González-García y col., 2020; Groenen y col., 2015; Ijurko y col., 2022; Kendal, Sayar y col., 2016; Scrucca, 2014; Vicente-Villardon y Vicente-Gonzalez, 2021).

Con frecuencia se puede encontrar que la matriz de datos está compuesta por datos de una naturaleza que no son continuos, como recuentos, categóricos o valores binarios; en estos casos los métodos clásicos no suelen ser apropiados. Por ejemplo, para el caso binario en la medición de branding se parte de unos atributos que el consumidor considera que una marca puede o no tener y a partir de esto calcular el brand equity Keller (2008); en la evaluación de impacto de políticas públicas, algunas veces las respuestas suelen ser binarias para identificar si los beneficiarios tienen o no algunas características, o para identificar si algunas condiciones económicas o sociales cambiaron con respecto a una línea de base (Moerbeek y Maas, 2005; Moerbeek y col., 2001; Murray y col., 2004). Asimismo, en la investigación biológica y en particular en el análisis alteraciones genéticas y epigenéticas la cantidad de datos binarios es cada vez más grande Iorio y col. (2016). Por lo tanto, generalizar el biplot clásico para datos en escala real a datos de otros tipos es de gran interés.

Para el caso en que la matriz de datos es binaria se han estudiado diferentes técnicas y enfoques. El PCA logístico es la extensión del método clásico de componentes principales para datos binarios y fue estudiado por Schein y col. (2003), usando una distribución de probabilidad Bernoulli, donde se usa un método de mínimos cuadrados alternos para estimar los parámetros del modelo. De Leeuw (2006) propone un PCA para datos binarios y para estimar los parámetros del modelo utiliza un algoritmo en dos pasos, uno de mayorización y otro de minización, conocido como método MM, que itera una secuencia de descomposiciones de valores singulares ponderados o no ponderados. Posteriormente, Lee y col. (2010) introducen regularización en los vectores de carga y estiman los parámetros utilizando un algoritmo iterativo de mínimos cuadrados ponderados, pero computacionalmente, el algoritmo es demasiado exigente para ser útil cuando la dimensión de los datos es muy alta. Para resolver este problema, Lee y Huang (2013) proponen un algoritmo que combina un algoritmo de descenso coordinado con una mayorización y así reducir el esfuerzo

computacional. Recientemente, Landgraf y Lee (2020) proponen una formulación que no requiere la factorización de matrices y utilizan un método MM para estimar los parámetros del modelo de PCA logístico, pero su enfoque depende de un parámetro que representa la aproximación del infinito, y Song y col. (2020) ajustan el modelo PCA logístico utilizando un método MM donde incorpora umbrales basados en los valores singulares y así aliviar los posibles problemas de sobreajuste del modelo. Pero en ninguno de estos enfoques se proporciona una representación simultánea de filas y columnas para visualizar el conjunto de datos binarios.

Para lograr una representación simultánea en los casos donde las variables de la matriz de datos no son continuas, Gabriel (1998) describió un método denominado *regresión bilineal* para ajustar un biplot con datos cuya distribución pertenece a la familia exponencial, pero el algoritmo de estimación no se estableció claramente. Vicente-Villardon y col. (2006) proponen una representación basada sobre una escala de respuesta logística y lo denominan *biplot logístico* (LB), en este caso cada individuo es representado por un punto y las variables son representadas por vectores dirigidos a través del origen del sistema de coordenadas, de esta forma se obtiene una representación simultánea donde la proyección ortogonal de los marcadores de las filas sobre los vectores predicen la probabilidad de que la característica ocurra. A continuación se presentan algunos aspectos generales sobre el LB.

I.1. Aspectos generales del biplot logístico

Sea $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ una matriz binaria, con $\mathbf{x}_i \in \{0,1\}^p$, $i = 1, \ldots, n$ y $x_{ij} \sim Ber(\pi(\theta_{ij}))$, donde $\pi(\cdot)$ es la inversa de la función de enlace. Cuando la función de enlace logística es utilizada, $\pi(\theta_{ij}) = \{1 + exp(-\theta_{ij})\}^{-1}$, la cual representa la probabilidad de que la característica j esté presente en el individuo i, el log-odds de $\pi(\theta_{ij})$ es θ_{ij} con $\theta_{ij} = log\left\{\pi(\theta_{ij})/(1 - \pi(\theta_{ij}))\right\}$, el cual corresponde al parámetro natural de una distribución Bernoulli expresada en la forma de familia exponencial. Usando la distribución de probabilidad, se tiene que $P(X_{ij} = x_{ij}) = \pi(\theta_{ij})^{x_{ij}} \left(1 - \pi(\theta_{ij})\right)^{1-x_{ij}}$, y la función de pérdida es obtenida como el negativo del logaritmo de la función de verosimilitud

$$\mathcal{L}(\mathbf{\Theta}) = -\sum_{i=1}^{n}\sum_{j=1}^{p}\left[x_{ij} log(\pi(\theta_{ij})) + (1 - x_{ij}) log(1 - \pi(\theta_{ij}))\right]. \tag{3}$$

En este caso no es apropiado centrar las columnas porque la matriz centrada ya no estará formada por elementos iguales a cero o uno. Por lo tanto, se extiende la especificación del espacio de parámetros naturales, de la misma forma que en modelos lineales generalizados (McCullagh y Nelder, 1989), introduciendo los efectos principales de las variables, también conocido como término de sesgo o desplazamiento de las columnas $\boldsymbol{\mu}$ para obtener un centrado basado en el modelo. La matriz canónica de parámetros $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)^T$ puede ser representada en una estructura de baja dimensión por algún entero $k \leq r$ que satisface $\boldsymbol{\theta}_i = \boldsymbol{\mu} + \sum_{s=1}^{k} a_{is}\mathbf{b}_s$, $i = 1, \ldots, n$, que expresado en forma matricial se escribe como

$$\boldsymbol{\Theta} = logit\,(\boldsymbol{\Pi}) = \mathbf{1}_n\boldsymbol{\mu}^T + \mathbf{A}\mathbf{B}^T, \tag{4}$$

donde $\mathbf{1_n}$ es un vector n-dimensional de unos; $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)^T$; $\mathbf{A} = (\mathbf{a_1}, \ldots, \mathbf{a_n})^T$ con $\mathbf{a_i} \in \mathbb{R}^k, i = 1, \ldots, n$; $\mathbf{B} = (\mathbf{b_1}, \ldots, \mathbf{b_k})$ con $\mathbf{b_j} \in \mathbb{R}^p, j = 1, \ldots, k$; y $\boldsymbol{\Pi} = \pi\,(\boldsymbol{\Theta})$ es la matriz de probabilidades esperada cuyo ij-ésimo elemento es igual a $\pi(\theta_{ij})$. De esta manera, $\boldsymbol{\Theta} = logit(\boldsymbol{\Pi})$ es un biplot en escala logit y el log-odds es $\theta_{ij} = \mu_j + \mathbf{a}_i^T\mathbf{b}_j$.

El modelo del LB está relacionado con la regresión logística de la misma manera que el análisis biplot clásico está relacionado con la regresión lineal. De la misma manera, así como los biplots lineales están relacionados con PCA, LB está relacionado con el análisis de rasgos latentes (LTA) o la teoría de respuesta al ítem (IRT).

En Vicente-Villardon y col. (2006) se presenta la geometría para una solución con dos dimensiones. Al fijar $\mathbf{A}$ en el modelo del biplot logístico, se obtiene una superficie de respuesta logística H, como se observa en la Figura 2, para este caso el tercer eje presenta una escala para las probabilidades esperadas. Aunque la superficie de respuesta no es lineal, las intersecciones con los planos perpendiculares al eje de probabilidad son líneas rectas. Al igual que en el caso lineal, las rectas para las diferentes probabilidades son paralelas. Los puntos en L que predicen las diferentes probabilidades también son líneas paralelas; esto significa que la predicción en el LB se hace de la misma forma que en un biplot lineal; la principal diferencia es que marcadores fila igualmente espaciados no necesariamente corresponden a probabilidades igualmente espaciadas.

El procedimiento para encontrar las coordenadas de los marcadores para una probabilidad fija π cuando $k = 2$, se presenta en Vicente-Villardon y col. (2006). Para ello se busca el

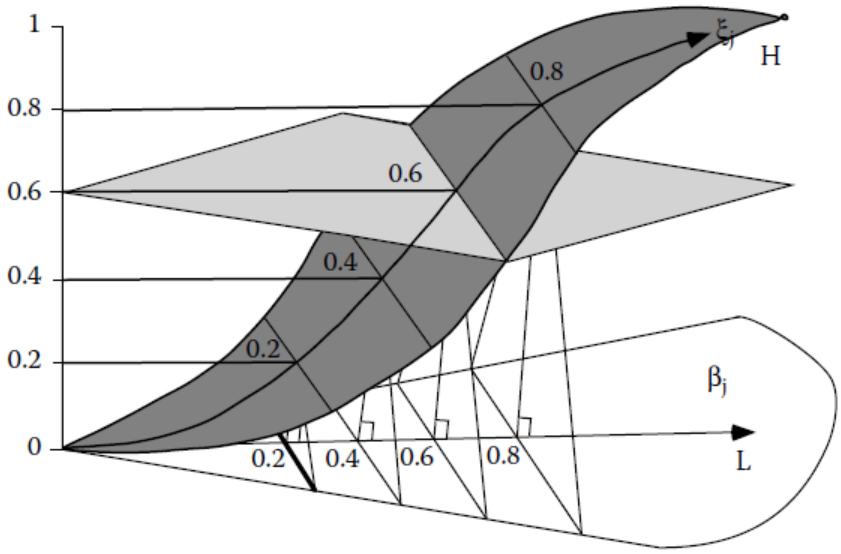


Figura 2: Geometría del biplot logístico. Tomado de Vicente-Villardon y col. (2006).

punto (d_{j1}, d_{j2}) que predice π y que está en el eje biplot, es decir, en la recta que une los puntos $(0,0)$ y (b_{j1}, b_{j2}), que se calcula como

$$d_{j1} = \frac{(logit(\pi) - \mu_j)b_{j1}}{\sum_{s=1}^{2} b_{js}^2}, \qquad d_{j2} = \frac{(logit(\pi) - \mu_j)b_{j2}}{\sum_{s=1}^{2} b_{js}^2}. \tag{5}$$

En el modelo LB, como en el biplot PCA, todas las direcciones pasan por el origen. En un biplot PCA para datos centrados, el origen representa la media de cada variable y la flecha muestra la dirección hacía donde crecen los valores. Como los datos binarios no se pueden centrar (mantenemos el término de desplazamiento de columna en el modelo), el origen no representa ningún valor particular de la probabilidad. En el modelo LB, las variables se representan con flechas (segmentos) y regularmente se comienza en el punto que predice una probabilidad de 0.5 y se termina en el punto que predice 0.75, estos valores son ajustables y dependerá del análisis que se pretenda en cada caso.

I.2. Proceso de estimación

A partir de la ecuación (3), Vicente-Villardon y col. (2006) proponen un esquema de estimación iterativo alternando la actualización de las matrices **A** y **B** hasta que se alcanza un nivel de precisión previamente definido. En cada iteración la función $\mathcal{L}$ se puede separar en una parte para cada fila o cada columna de la matriz de datos, minimizando cada

una por separado. Este proceso converge a un mínimo local, y puede considerarse una generalización del método de regresión de los biplots clásicos, puesto que en los casos donde los datos se ajusten a una distribución normal multivariante y se utilice la función de enlace identidad en lugar de la función logística, la solución coincidirá con la del biplot clásico.

Al fijar los marcadores fila, $\mathbf{A}$, la función de pérdida dada en (3) se puede separar en p partes, una por cada variable

$$\mathcal{L}(\mathbf{\Theta}) = -\sum_{j=1}^{p}\left[\sum_{i=1}^{n} x_{ij} log(\pi(\theta_{ij})) + (1-x_{ij}) log(1-\pi(\theta_{ij}))\right] \tag{6a}$$

$$= -\sum_{j=1}^{p} \mathcal{L}_j(\theta_{ij}). \tag{6b}$$

Minimizar cada $\mathcal{L}_j(\theta_{ij})$ es equivalente a realizar una regresión logística utilizando $\mathbf{x}_j$ como la variable dependiente y las columnas de $\mathbf{A}$ como las variables independientes. A este paso se le denomina *etapa de regresión*. Del mismo modo, la función de pérdida se puede separar en un sumando por cada fila de la matriz $\mathbf{X}$

$$\mathcal{L}(\mathbf{\Theta}) = -\sum_{i=1}^{n}\left[\sum_{j=1}^{p} x_{ij} log(\pi(\theta_{ij})) + (1-x_{ij}) log(1-\pi(\theta_{ij}))\right] \tag{7a}$$

$$= -\sum_{i=1}^{n} \mathcal{L}_i(\theta_{ij}). \tag{7b}$$

Las derivadas parciales con respecto a $a_{is}, s = 1, \ldots, S$, son

$$\frac{\partial \mathcal{L}_i(\theta_{ij})}{\partial a_{is}} = \sum_{j=1}^{p}\left[x_{ij}\frac{1}{\pi(\theta_{ij})}\frac{\partial \pi(\theta_{ij})}{\partial a_{is}} + (1-x_{ij})\frac{1}{1-\pi(\theta_{ij})} + \frac{\partial(1-\pi(\theta_{ij}))}{\partial a_{is}}\right]. \tag{8}$$

Teniendo en cuenta que $\pi(\theta_{ij}) = (1-\exp(-\theta_{ij}))^{-1}$ y $\theta_{ij} = \mu_j + \sum_{s=1}^{k} a_{is}b_{js} = \mu_j + \mathbf{a}_i^T\mathbf{b}_j$, se pueden obtener las derivadas parciales

$$\frac{\partial \pi(\theta_{ij})}{\partial a_{is}} = b_{js}\pi(\theta_{ij})(1-\pi(\theta_{ij})) \qquad \text{y} \qquad \frac{\partial(1-\pi(\theta_{ij}))}{\partial a_{is}} = -b_{js}\pi(\theta_{ij})(1-\pi(\theta_{ij})), \tag{9}$$

entonces el vector gradiente, $\mathbf{g} = (g_1, \ldots, g_k)$ está formado por los elementos

$$g_s = \frac{\partial \mathcal{L}_i(\theta_{ij})}{\partial a_{is}} = \sum_{j=1}^{p} b_{js}(x_{ij} - \pi(\theta_{ij})), \quad s = 1, \ldots, k. \tag{10}$$

En Vicente-Villardon y col. (2006) se encuentra la expresión para los elementos de la matriz hessiana $\mathbf{H}$, la cual tiene los elementos

$$h_{ss} = \frac{\partial^2 \mathcal{L}_i(\theta_{ij})}{\partial a_{is}^2} = -\sum_{j=1}^{p} b_{js}^2 \pi(\theta_{ij})(1 - \pi(\theta_{ij})) \text{ y } h_{ss'} = \frac{\partial^2 \mathcal{L}_i(\theta_{ij})}{\partial a_{is} \partial a_{is'}} = -\sum_{j=1}^{p} b_{js} b_{js'} \pi(\theta_{ij})(1 - \pi(\theta_{ij})). \tag{11}$$

Para encontrar las soluciones se puede asignar valores iniciales para $\mathbf{A}$, ortonormalizar la matriz ($\mathbf{A}^T\mathbf{A} = I$), seguir con un paso de regresión logística donde cada columna $\mathbf{x}_i$ son las variables dependientes y la matriz $\mathbf{A}$ son las variables independientes. Posteriormente, se realiza un paso de interpolación mediante el método de Newton-Raphson. Este procedimiento es iterativo y el algoritmo finaliza cuando se tenga un cambio pequeño sobre $\mathcal{L}(\mathbf{\Theta})$. El pseudocódigo se resume en el Algoritmo 1.

Algoritmo 1 Método alternante para estimar los parámetros de un modelo LB

Entrada X
Salida $\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}$

1: Inicializar $\mathbf{A}$ con el resultado de un PCA sobre $\mathbf{X}$.
2: **repeat**
3: **procedure** ORTONORMALIZAR
4: $\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{I}_k$
5: **end procedure**
6: **Etapa de regresión**. $logit(\pi(\theta_j)) = \mu_j + \mathbf{A}\mathbf{b}_j$, para $j = 1 \ldots, p$
7: Construir $\boldsymbol{\mu}$ y $\mathbf{B}$.
8: **Etapa de interpolación**. Actualizar $\mathbf{A} = (\mathbf{a_1}, \ldots, \mathbf{a_k})^T$ usando el método de Newton-Raphson
9: **until** $(\mathcal{L}(\mathbf{\Theta}_l) - \mathcal{L}(\mathbf{\Theta}_{l+1}))/\mathcal{L}(\mathbf{\Theta}_l) < \epsilon$

El paso de ortonormalización proporciona unicidad de las estimaciones de los parámetros de la misma manera que los vectores de longitud unitaria se toman en componentes principales. Esta restricción también se puede imponer sobre $\mathbf{B}$. El paso es opcional, y la ortonormalización se puede hacer a posteriori tomando la SVD de los valores esperados en θ (Vicente-Villardon y col., 2006).

I.3. Biplot logístico externo

El algoritmo 1 puede llegar a ser muy costoso computacionalmente, especialmente porque en la etapa de interpolación también se debe iterar. La propuesta de Demey y col. (2008), es que la regresión sobre las columnas de $\mathbf{X}$ en el procedimiento alternante para datos binarios es un modelo de regresión logística que se puede ajustar a la configuración obtenida de un análisis de coordenadas principales (PCoA), y aunque se podría utilizar todo el procedimiento de alternancia, PCoA es más simple. A esta combinación entre PCoA y regresión logística, se denomina biplot logístico externo (ELB).

Para la matriz binaria $\mathbf{X}$ se puede definir a_{ij} como el número de variables con respuesta igual a 1 en la fila i y en la fila j, b_{ij} el número de variables con respuesta igual a 0 en la fila i y 1 en la fila j, c_{ij} el número de variables con respuesta igual a 1 en la fila i y 0 en la fila j, y d_{ij} el número de variables con respuesta igual a 0 en la fila i y en la fila j. De modo que $a_{ij} + b_{ij} + c_{ij} + d_{ij} = p$. La similaridad entre las filas i, j es denotada por s_{ij}. Se han propuesto varios coeficientes de similitud que combinan las cantidades a_{ij}, b_{ij}, c_{ij} y d_{ij}. La Tabla 3 enumera algunos coeficientes, donde para mayor claridad se omiten los subíndices que representan a las filas i, j.

Si $\mathbf{S}$ es la matriz que contiene las similaridades entre las n filas de la matriz binaria $\mathbf{X}$, y $\mathbf{\Delta}$ es la matriz de disimilaridades o distancias entre las n filas. El algoritmo parte de un PCoA, como técnica de ordenación de las filas. PCoA se ocupa del problema de construir una configuración de n puntos en un espacio euclidiano de tal manera que la distancia entre dos puntos cualesquiera de la configuración se aproxime lo más posible a la disimilitud entre las filas representados por estos puntos. El objetivo es encontrar la configuración $\mathbf{A}$ en un espacio euclideano de dimensión k, cuya matriz de distancia entre puntos $\mathbf{D}$ esté lo más cercano posible a $\mathbf{\Delta}$ (Demey y col., 2008).

Tabla 3: Propiedades de algunos coeficientes de similaridad para variables binarias[1].

Variable	**Coeficiente Similaridad**[2]		**Rango**	**Métrica**[3]	**S** ≥ 0[4]
S_1	$\frac{a}{b+c}$	Kulczynski	$0, \infty$	Ind.	Si
S_2	$\frac{a}{a+b+c+d}$	Russel y Rao	$0, 1$	Si	Si
S_3	$\frac{a}{a+b+c}$	Jaccard	$0, 1$	Si	Si

Variable	Coeficiente Similaridad[2]		Rango	Métrica[3]	$\mathbf{S} \geq 0$[4]
S_4	$\frac{a+d}{a+b+c+d}$	Emparejamiento Simple	$0, 1$	Si	Si
S_5	$\frac{a}{a+2(b+c)}$	Anderberg	$0, 1$	Si	Si
S_6	$\frac{a+d}{a+2(b+c)+d}$	Rogers y Tanimoto	$0, 1$	Si	Si
S_7	$\frac{a}{a+\frac{1}{2}(b+c)}$	Sørensen, Dice, y Czekanowski	$0, 1$	No	Si
S_8	$\frac{a+d}{a+\frac{1}{2}(b+c)+d}$	Sneath y Sokal	$0, 1$	No	No
S_9	$\frac{a-(b+c)+d}{a+b+c+d}$	Hamman	$-1, 1$	Si	Si
S_{10}	$\frac{1}{2}\left(\frac{a}{a+b}+\frac{a}{a+c}\right)$	Kulczynski	$0, 1$	No	No
S_{11}	$\frac{1}{4}\left(\frac{a}{a+b}+\frac{a}{a+c}+\frac{c}{c+d}+\frac{d}{b+d}\right)$	Anderberg	$0, 1$	No	No
S_{12}	$\frac{a}{\sqrt{(a+b)(a+c)}}$	Ochiai	$0, 1$	No	Si
S_{13}	$\frac{ad}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$		$0, 1$	No	Si
S_{14}	$\frac{ad-bc}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$	φ de Pearson	$-1, 1$	No	Si
S_{15}	$\frac{ad-bc}{ad+bc}$	Yule	$-1, 1$	No	No

[1] Tomada de Gower y Warrens (2014).

[2] a,b,c y d son las frecuencias absolutas de los eventos $(1,1), (1,0), (0,1)$ y $(0,0)$ respectivamente.

[3] La propiedad métrica se verifica cuando $\Delta_{ij} + \Delta_{ik} \geq \Delta_{jk}$.

[4] $\mathbf{S} \geq 0$ indica si la matriz de similaridades es semidefinida positiva.

Para las métricas con rango entre cero y uno, la similaridad puede ser transformada a distancia de diferentes formas. Por ejemplo, $\Delta_{ij} = 1 - s_{ij}$, $\Delta_{ij} = \sqrt{1 - s_{ij}}$ o $\Delta^2_{ij} = s_{ii} + s_{jj} - 2s_{ij}$. Esta última es una de las recomendadas por Gower y Hand (1995) debido a sus propiedades, en notación matricial se puede escribir como

$$\mathbf{\Delta}^2 = 2\left(\mathbf{1}_n\mathbf{1}_n^T - \mathbf{S}\right), \tag{12}$$

donde $\mathbf{\Delta}^2$ denota la matriz de cuadrados de las distancias. Este procedimiento externo permite llegar a la matriz $\mathbf{A}$ y paso seguido, realizar una regresión tomando como variables dependientes a las columnas de $\mathbf{X}$. El pseudocódigo para el método propuesto por Demey y col. (2008) se resume en el Algoritmo 2.

Algoritmo 2 Estimación de los parámetros de un ELB

Entrada $\mathbf{X}$
Salida $\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}$

1: Calcular $\mathbf{S}$
2: $\boldsymbol{\Delta}^2 = 2\left(\mathbf{1}_n\mathbf{1}_n^T - \mathbf{S}\right)$
3: $\mathbf{Q} = -\frac{1}{2}\mathbf{H}\boldsymbol{\Delta}^2\mathbf{H}^T$, con $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$
4: Descomposición espectral: $\mathbf{Q} = \mathbf{U}\boldsymbol{\Lambda}^{1/2}\boldsymbol{\Lambda}^{1/2}\mathbf{U}^T$, con $\mathbf{U}^T\mathbf{U} = \mathbf{I}$
5: $\mathbf{A} = \mathbf{U}_{(k)}\boldsymbol{\Lambda}_{(k)}^{1/2}$
6: Regresión: $logit(\pi(\theta_j)) = \mu_j + \mathbf{A}\mathbf{b}_j$, para $j = 1\ldots, p$
7: Construir $\boldsymbol{\mu}$ y $\mathbf{B}$.

I.4. Problema de investigación

Actualmente estamos expuestos a un gran volumen de información y con ello se ha dado un gran valor a los datos. Esto no solo plantea la posibilidad de tener una mejor compresión de las relaciones subyacentes que se encuentran en la matriz de datos, sino que genera nuevos desafíos desde el punto de vista estadístico. Esto ha llevado a que se creen líneas de investigación de análisis multivariante que buscan desarrollar técnicas eficientes para el manejo de grandes volúmenes de datos, y así generar enfoques analíticos desde una perspectiva de *big data*.

En paralelo con los avances tecnológicos se han desarrollado alternativas que permiten obtener información de una manera masiva y de forma simple, bien sea a través del procesamiento de imágenes, por medio de redes sociales, registros financieros, datos de consumo a partir de campañas de fidelización, entre muchas otras. Por ejemplo, en campos como la bioinformática, la medición de las alteraciones genéticas y epigenéticas se presenta en matrices de alta dimensión donde algunas presentan características binarias (Iorio y col., 2016). Esto ya lo anticipaba Tukey (1962) cuando señalaba la importancia de los métodos de optimización en el campo de la aplicación estadística porque se vendrían muchos más datos en el futuro.

Asimismo, es habitual que toda esta información no esté completa, es decir, que la matriz de datos puede tener algunas entradas faltantes. Por ejemplo, en el *problema de Netflix* cada entrada ij de la matriz representa la calificación binaria que asigna el cliente i a la película j, pero este dato puede estar observado solo si el cliente ha visto la película, de lo contrario será un dato faltante. Por lo tanto, tratar de completar la matriz de datos

también se convierte en un reto y así poder generar nuevas recomendaciones en este caso.

El enfoque multivariante para matrices binarias se detalla en Vicente-Villardon y col. (2006), donde se introduce el modelo de biplot logístico (LB) y luego en Demey y col. (2008) se incorpora un procedimiento externo para calcular los parámetros del modelo. Sin embargo, cuando el volumen de datos es muy grande o la matriz es muy dispersa[1], los algoritmos actuales pueden llegar a ser muy exigentes desde un punto de vista computacional, dificultando llegar a una solución al problema.

De otra parte, la cantidad de parámetros a estimar va creciendo cuando se aumenta el número de filas o de columnas, y aún no se han explorado metodologías que permitan reducir la cantidad de parámetros a estimar. Adicionalmente, aún está por explorar las metodologías que pueden utilizarse para trabajar con matrices que tengan entradas faltantes, generando un proceso de imputación durante la etapa de estimación.

Otro aspecto que resulta de gran relevancia para ajustar un modelo LB es definir la cantidad de dimensiones que son necesarias para obtener una estimación más precisa del espacio de los parámetros del modelo. Para el biplot logístico se han usado algunas medidas basadas en la bondad del ajuste, pero hasta ahora no se ha investigado un procedimiento que permita elegir a priori el número de dimensiones que permita maximizar la precisión y evitar el sobreajuste.

Así que en esta investigación se proponen nuevas formas de realizar el ajuste para el modelo LB basado en un enfoque de optimización multivariante a partir de algoritmos utilizados en el contexto de aprendizaje automático *(machine learning)*. Para realizar la adaptación y proponer los algoritmos de estimación, es necesario demostrar algunas propiedades que permiten obtener funciones sustitutas que son más suaves, lo que permite que se puedan acelerar los procesos de convergencia. Asimismo, se explora y se desarrolla una metodología que permite tratar con los datos faltantes para llegar a una matriz completa a partir del modelo, que cuenta con la ventaja de estimar un número menor de parámetros y permite realizar de una forma simple la proyección de nuevas filas. Además se estudian, se adaptan y se implementan algunos métodos que permiten seleccionar la dimensionalidad del modelo.

Son siete los capítulos que dan cuerpo al presente documento, además de esta introducción,

[1]Las matrices dispersas son aquellas en las que la mayoría de los elementos son cero.

que buscó hacer un resumen muy general de los métodos biplot y se enfocó en describir el biplot logístico. En el primer capítulo se presenta el objetivo general y los objetivos específicos que se desarrollan en esta investigación.

El segundo capítulo se ocupa de realizar una generalización del biplot logístico, para ello se formulan los métodos desde un punto de vista probabilístico y la función de pérdida se expresa a partir de los factores de normalización para una distribución de familia exponencial, buscando de esta forma poder extender los métodos a datos no continuos. A partir de la formulación presentada, se usa el factor de normalización de una distribución Bernoulli y se llega a la función de pérdida. De esta forma, los métodos podrían ser extendidos para encontrar funciones de pérdida adecuadas dependiendo del tipo de datos. Debido a que la función de pérdida para el biplot logístico no es fácil de optimizar, se realiza un desarrollo teórico donde se postula y se demuestra un teorema que permite sustituir el problema de optimización por otro más simple. Asimismo, se desarrolla una metodología basada en validación cruzada que permite seleccionar de forma objetiva el número de dimensiones para el modelo de biplot logístico. Finalmente se realiza un resumen de las contribuciones más relevantes que son realizadas en el capítulo.

En el tercer capítulo se adaptan cuatro algoritmos basados en el gradiente conjugado y se desarrolla la metodología para un algoritmo de descenso coordinado por bloques basado en la función sustituta encontrada en el segundo capítulo, para ajustar el biplot logístico. Posteriormente se describe una metodología que permite simular matrices binarias con rango $k < p$ y se desarrolla un procedimiento de Monte Carlo que permite comparar el rendimiento de los diferentes algoritmos en cuanto a la capacidad que tienen para identificar el número de dimensiones del modelo y la habilidad que tienen para recuperar la matriz canónica de parámetros usando escenarios con matrices balanceadas y otros escenarios donde la matriz de datos está desequilibrada. Para ilustrar los métodos propuestos se usan datos reales sobre metilación del ADN. Finalmente, se recogen las contribuciones más importantes que se realizan en el capítulo.

El cuarto capítulo se encarga de desarrollar un nuevo enfoque que permita dar un tratamiento a las matrices binarias con datos faltantes, buscando llevar a cabo un proceso de imputación durante el procedimiento de optimización. Para esto se usa el enfoque de Pearson que busca la representación óptima de los datos multivariantes en el espacio de

baja dimensión al minimizar el error cuadrático medio de la proyección. Para llegar a una función que permita optimizar los parámetros del modelo se postula y se demuestra un teorema, y usando un algoritmo de descenso coordinado por bloques se realiza el proceso de optimización que permite realizar el proceso de imputación. Este enfoque presenta la ventaja de que la matriz de marcadores fila no hace parte del proceso y puede ser obtenida a partir de los marcadores columna, facilitando la proyección de nuevas filas como suplementarias, que hasta ahora no era posible sin tener que ejecutar un nuevo procedimiento de optimización. Esto tiene implícita otra ventaja, y es que al no tener que entrar la matriz de marcadores fila en el proceso entonces se reduce la cantidad de parámetros a estimar. Posteriormente, para ilustrar el método propuesto se realiza una aplicación utilizando datos reales sobre el conflicto armado en Colombia. Finalmente, se recogen las contribuciones más relevantes del capítulo.

En el quinto capítulo se presenta en detalle la librería *BiplotML*, que fue desarrollada como un producto que le permite a los usuarios tener un soporte para la aplicación del modelo de biplot logístico usando todos los métodos desarrollados en este trabajo. Se presentan los métodos que son implementados, la manera de llevar a cabo el procedimiento de validación cruzada, la forma de ajustar el modelo con base en los algoritmos presentados. Así como la descripción de los objetos de salida, entorno gráfico, y se incorpora un método para generar regiones de confianza. Finalmente, se realiza un resumen de las contribuciones realizadas.

El sexto capítulo hace la recopilación de las conclusiones sobre los hallazgos más relevantes, y finalmente en el séptimo capítulo se dan algunas recomendaciones para líneas futuras de investigación.

Capítulo 1

Objetivos

1.1. Objetivo general

El objetivo general de esta investigación es realizar contribuciones a los métodos biplot para el análisis de datos binarios multivariantes.

1.2. Objetivos específicos

1. Estudiar y proponer nuevas formulaciones que permitan obtener funciones de pérdida sustitutas donde se puedan implementar algoritmos eficientes para ajustar un biplot logístico cuando se tienen matrices binarias con un gran volumen de datos.

2. Investigar, adaptar y desarrollar un método que permita elegir de forma objetiva el número de dimensiones para ajustar el biplot logístico.

3. Adaptar e implementar algunos algoritmos de optimización multivariante usados en el contexto del aprendizaje automático para aplicarlos al caso de un biplot logístico.

4. Utilizar funciones sustitutas para proponer e implementar algoritmos de estimación que permitan ajustar los parámetros de un biplot logístico.

5. Proponer una metodología para ajustar el modelo de biplot logístico cuando hay presencia de datos faltantes y que impute las entradas faltantes en la matriz de datos durante el proceso de estimación.

6. Desarrollar un paquete en R donde se implementen las rutinas que faciliten el uso de los algoritmos propuestos.

Capítulo 2

Generalización del biplot logístico

2.1. Introducción

Desde este capítulo se empieza a tratar el problema de investigación. La sección 2.2 comienza presentando un enfoque desde una perspectiva probabilística para los métodos biplot y se presenta una formulación general que puede ser utilizada para encontrar funciones de pérdida adecuadas dependiendo del tipo de datos. A partir de la formulación general, en la sección 2.3 se aplican los factores de normalización basados en una distribución Bernoulli y se llega a la función de pérdida de un modelo LB. En la sección 2.4 se realiza un desarrollo teórico que permite sustituir la función de pérdida de un modelo LB por otra función que cuenta con unas propiedades que permiten resolver el problema de una manera eficiente. En la sección 2.5 y en la sección 2.6 se analizan algunas medidas de rendimiento, allí se investiga y se adapta un método de validación cruzada que permite elegir de forma objetiva el número de dimensiones para ajustar el modelo LB.

2.2. Enfoque probabilístico

El problema (1) también puede formularse como una estimación de máxima verosimilitud. Desde una perspectiva de modelamiento probabilístico, Tipping y Bishop (1999) mostraron para un PCA que la solución del problema maximiza la función de verosimilitud cuando se asume que cada observación es extraída de una distribución normal multivariante en un subespacio de baja dimensión.

La interpretación probabilística supone que cada $\mathbf{x}_i$ puede ser aproximado por una proyección lineal de variables latentes en el espacio de baja dimensión más un error con distribución normal, con lo cual $\mathbf{x}_i = \boldsymbol{\mu} + \mathbf{a}_i^T\mathbf{b} + \boldsymbol{\varepsilon}_i$, $i = 1, \ldots, n$, donde $\mathbf{a}$ y $\mathbf{b}$ son los marcadores de las filas y columnas respectivamente en el subespacio de baja dimensión; $\boldsymbol{\mu}$ es un vector de desplazamiento o de compensación y $\boldsymbol{\varepsilon}_i$ sigue una distribución $\mathcal{N}(0, \sigma^2\mathbf{I})$. Suponiendo un vector canónico de parámetros $\boldsymbol{\theta}_i = \boldsymbol{\mu} + \mathbf{a}_i^T\mathbf{b}$, entonces la probabilidad condicional de $\mathbf{x}_i$ dado $\boldsymbol{\theta}_i$, es representada por $p(\mathbf{x}_i; \boldsymbol{\theta}_i) \sim \mathcal{N}(\boldsymbol{\theta}_i, \sigma^2\mathbf{I})$. Por lo tanto, el problema de minimización del logaritmo de la función de verosimilitud de los datos con respecto al modelo paramétrico $\boldsymbol{\Theta} = \mathbf{1}_n\boldsymbol{\mu}^T + \mathbf{A}\mathbf{B}^T$ es equivalente a minimizar la siguiente función objetivo:

$$\sum_{i=1}^{n} \|\mathbf{x}_i - \boldsymbol{\theta}_i\|^2 = \|\mathbf{X} - \boldsymbol{\Theta}\|_F^2 \quad \text{sujeto a} \quad \mathbf{B}^T\mathbf{B} = \mathbf{I}. \tag{2.1}$$

Este problema es equivalente a minimizar el error cuadrático medio para encontrar una representación óptima de los datos multivariantes en el subespacio de dimensión reducida (Pearson, 1901). Por lo tanto, en el subespacio k-dimensional, cada $\mathbf{x}_i$ está representado por $\boldsymbol{\mu} + \mathbf{a}_i^T\mathbf{b}$. Aunque estos enfoques fueron propuestos para modelar datos continuos, pero no son apropiados cuando los datos están en una escala discreta o nominal.

Formular el problema desde una perspectiva probabilística es interesante porque permite extender la aplicación de los métodos clásicos. Por ejemplo, si los datos corresponden a valores binarios, enteros o son positivos entonces el supuesto de normalidad no es apropiado. De hecho, cuando los datos son recuentos es usual se modelen con una distribución de Poisson y si son binarios se opta por una distribución Bernoulli; este enfoque permite que otros tipos de datos puedan ser modelados por su correspondiente distribución de la familia exponencial. Con esta motivación, (Collins y col., 2002) proporcionan una generalización del PCA para datos con distribuciones de la familia exponencial utilizando el marco de modelos lineales generalizados. Mientras que (Landgraf y Lee, 2019; Lu y col., 2016; Schein y col., 2003; Singh y Gordon, 2008a; Udell y col., 2016) incluyen más funciones de pérdida, otros tipos de datos y también incorporan regularización al modelo.

Con el propósito de extender los métodos biplot al resto de distribuciones de la familia exponencial. Sea $p(\mathbf{x}_i; \boldsymbol{\theta}_i)$ cualquier conjunto parametrizado de distribuciones de la familia

exponencial, es decir, que

$$p(\mathbf{x}_i;\theta_i) = exp\left[\boldsymbol{\theta}_i\mathbf{x}_i - G(\boldsymbol{\theta}_i) + q(\mathbf{x}_i)\right], \tag{2.2}$$

donde $\boldsymbol{\theta}_i$ es el vector canónico de parámetros correspondiente a $\mathbf{x}_i$. Considerando que

$$\int p(\mathbf{x}_i;\theta_i)d\mathbf{x}_i = \int exp\left[\boldsymbol{\theta}_i\mathbf{x}_i - G(\boldsymbol{\theta}_i) + q(\mathbf{x}_i)\right]d\mathbf{x}_i = 1, \tag{2.3}$$

se deduce que

$$exp(-G(\boldsymbol{\theta}_i))\int exp\left[\boldsymbol{\theta}_i\mathbf{x}_i + q(\mathbf{x}_i)\right]d\mathbf{x}_i = 1, \tag{2.4}$$

por lo que

$$G(\boldsymbol{\theta}_i) = log\int exp\left[\boldsymbol{\theta}_i\mathbf{x}_i + q(\mathbf{x}_i)\right]d\mathbf{x}_i. \tag{2.5}$$

De modo que $G(\boldsymbol{\theta}_i)$ es el factor de normalización logaritmico que asegura que la integral (o la suma) de $p(\mathbf{x}_i;\theta_i)$ sobre el dominio de $\mathbf{x}_i$ es igual a 1. En las distribuciones de la familia exponencial se suelen tener diferentes funciones de $G(\cdot)$ (Bickel y Doksum, 2015). Por ejemplo, en el caso binario se puede asumir una distribución Bernoulli con probabilidad $\pi(\theta_{ij})$, por lo tanto

$$\begin{aligned}
p(x_{ij};\theta_{ij}) &= \pi(\theta_{ij})^x_{ij}(1-\pi(\theta_{ij}))^{1-x_{ij}} && \text{(2.6a)}\\
&= exp\left[x_{ij}log(\pi(\theta_{ij})) + (1-x_{ij})log(1-\pi(\theta_{ij}))\right] && \text{(2.6b)}\\
&= exp\left[x_{ij}log\left(\frac{\pi(\theta_{ij})}{1-\pi(\theta_{ij})}\right) + log(1-\pi(\theta_{ij}))\right] && \text{(2.6c)}\\
&= exp\left[x_{ij}\theta_{ij} - log(1+exp(\theta_{ij}))\right]. && \text{(2.6d)}
\end{aligned}$$

Así $\theta_{ij} = log\left\{\pi(\theta_{ij})/(1-\pi(\theta_{ij}))\right\}$ y $G(\theta_{ij}) = log(1+exp(\theta_{ij}))$. La primera derivada de la función $G(\theta_{ij})$, denotada como $g(\theta_{ij})$ es de gran relevancia; se puede demostrar que $g(\theta_{ij}) = E(x_{ij};\theta_{ij})$ (ver Bickel y Doksum, 2015, capítulo 1; Wasserman, 2013, pp. 52). En el caso de la distribución Bernoulli,

$$g(\theta_{ij}) = \frac{\partial G(\theta_{ij})}{\partial \theta_{ij}} \tag{2.7a}$$

$$= [1 + exp(-\theta_{ij})]^{-1} \tag{2.7b}$$

$$= \pi(\theta_{ij}), \tag{2.7c}$$

que corresponde a una función de enlace logística. En la Tabla 2.1 se presentan los factores de normalización para varias distribuciones de la familia exponencial.

Tabla 2.1: Factores de normalización para algunas distribuciones de la familia exponencial

Distribución	$G(\theta)$	$g(\theta)$
Normal	$\frac{\theta^2}{2}$	θ
Bernoulli	$log(1 + exp(\theta))$	$[1 + exp(-\theta)]^{-1}$
Poisson	$exp(\theta)$	$exp(\theta)$
Exponencial	$-log(-\theta)$	$-\frac{1}{\theta}$

Al tener en cuenta la forma general de la función de probabilidad para una distribución de la familia exponencial dada en (2.2), se puede maximizar la función de verosimilitud de los datos. Después de sustituir $\boldsymbol{\theta}_i = \boldsymbol{\mu} + \mathbf{a}^T\mathbf{b}$ en la función de verosimilitud y siguiendo lo establecido por Guo y Schuurmans (2008), se puede formular el problema de la siguiente manera

$$\min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} \mathcal{L}^* (\boldsymbol{\Theta}) = \min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} \sum_{i=1}^{n} G(\boldsymbol{\mu} + \mathbf{B}\mathbf{a}_i) - \text{tr}\left((\mathbf{1}_n\boldsymbol{\mu}^T + \mathbf{A}\mathbf{B}^T)\mathbf{X}^T\right). \tag{2.8}$$

Tomando como ejemplo $G(\boldsymbol{\theta}_i) = \boldsymbol{\theta}_i^T\boldsymbol{\theta}_i/2$ dado en la Tabla 2.1 para una función de distribución normal, entonces el logaritmo de la función de verosimilitud dado $\boldsymbol{\theta}$ es equivalente a $\sum_i - \|\mathbf{x}_i - \boldsymbol{\theta}_i\|^2$, así que el problema vuelve a ser el mismo que se presentó en la ecuación (2.1).

La formulación anterior permite extender los métodos biplot a otros tipos de datos, pero dependiendo de la función de distribución de la familia exponencial, el problema (2.8) puede conducir a una función de minimización que no se puede resolver de manera directa. Así que sería necesario llegar a una solución usando algoritmos iterativos. Para maximizar la eficiencia, en cada paso se puede involucrar un problema convexo o no convexo pero

que sea lo suficientemente simple para poder resolverlo. Una posibilidad es sustituir el problema de optimización complejo por un nuevo problema que sea equivalente pero que admita un procedimiento de optimización eficiente.

2.3. Biplot logístico obtenido desde la familia exponencial

Como se mencionó antes, la selección apropiada de $G(\theta)$ se da como resultado la distribución adecuada para modelar los datos dependiendo de su naturaleza, lo que conducirá intuitivamente al mejor rendimiento del modelo. En este sentido, la manera en que se generaliza de un biplot clásico a un biplot logístico binario es análogo a la forma en que se generaliza de una regresión lineal a una regresión logística (Vicente-Villardon y col., 2006).

Tomando en consideración que $G(\theta) = log(1 + exp(\theta))$ es adecuado para modelar datos binarios, se puede demostrar que minimizar la ecuación (2.8) es equivalente a minimizar la función de pérdida obtenida cuando se usa la función de verosimilitud, denotada por $\mathcal{L}(\mathbf{\Theta})$. Esto se demuestra con el siguiente Teorema.

Teorema 1. *Si* $G(\theta_{ij}) = log(1 + exp(\theta_{ij}))$ *y* $g(\theta_{ij}) = \frac{\partial G(\theta_{ij})}{\partial \theta_{ij}} = [1 + exp(-\theta_{ij})]^{-1}$ *entonces*

$$\begin{aligned}
\min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} \mathcal{L}^*(\mathbf{\Theta}) &= \min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} \sum_{i=1}^{n} G(\boldsymbol{\mu} + \mathbf{B}\mathbf{a}_i) - \text{tr}\left((\mathbf{1}_n\boldsymbol{\mu}^T + \mathbf{A}\mathbf{B}^T)\mathbf{X}^T\right) \\
&= \min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} -\sum_{i=1}^{n}\sum_{j=1}^{p} [x_{ij} log(\pi(\theta_{ij})) + (1 - x_{ij}) log(1 - \pi(\theta_{ij}))] \\
&= \min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} \mathcal{L}(\mathbf{\Theta})
\end{aligned}$$

Demostración.

$$\min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} \mathcal{L}^*(\mathbf{\Theta}) = \min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} \sum_{i=1}^{n} G(\boldsymbol{\mu} + \mathbf{B}\mathbf{a}_i) - \text{tr}\left((\mathbf{1}_n\boldsymbol{\mu}^T + \mathbf{A}\mathbf{B}^T)\mathbf{X}^T\right) \tag{2.9}$$

$$= \min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} \sum_{i=1}^{n}\sum_{j=1}^{p} \left[log(1 + exp(\mu_j + \mathbf{a}_i^T\mathbf{b}_j)) - (\mu_j + \mathbf{a}_i^T\mathbf{b}_j)x_{ij}\right] \tag{2.10}$$

$$= \min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} \sum_{i=1}^{n}\sum_{j=1}^{p} [log(1 + exp(\theta_{ij}) - log(exp(\theta_{ij}x_{ij}))] \tag{2.11}$$

$$= \min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} -\sum_{i=1}^{n}\sum_{j=1}^{p} log\left(\frac{exp(\theta_{ij}x_{ij})}{1 + exp(\theta_{ij})}\right) \tag{2.12}$$

Dado que $x_{ij} = \{0, 1\}$ y $g(\theta_{ij}) = [1 + exp(-\theta_{ij})]^{-1} = \pi(\theta_{ij})$, entonces

$$\min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} \mathcal{L}(\boldsymbol{\Theta}) = \min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} -\sum_{i=1}^{n}\sum_{j=1}^{p} log\left(\frac{\pi(\theta_{ij})^{x_{ij}}}{(1+\pi(\theta_{ij}))^{(1-x_{ij})}}\right) \tag{2.13}$$

$$= \min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} -\sum_{i=1}^{n}\sum_{j=1}^{p} [x_{ij} log(\pi(\theta_{ij})) + (1 - x_{ij}) log(1 - \pi(\theta_{ij}))] \tag{2.14}$$

$$= \min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} \mathcal{L}(\boldsymbol{\Theta}). \tag{2.15}$$

□

De esta forma se podría usar el factor de normalización de la familia exponencial según el tipo de datos y así llegar a una función de pérdida adecuada, incluso cuando se quiera incorporar una regularización dentro del modelo.

En contraste con el biplot clásico, en este caso no existe una solución explícita que permita minimizar (2.15), por lo que es necesario recurrir a algoritmos iterativos. Collins y col. (2002) proponen resolver el problema para el ACP a partir de una actualización secuencial similar a los métodos iterativos usados en los MLG (McCullagh y Nelder, 1989); mientras que en el caso de un biplot logístico Vicente-Villardon y col. (2006) proponen un método de Newton-Raphson y Demey y col. (2008) usan un procedimiento externo. Aunque la complejidad del problema se simplifica, estos métodos no garantizan una solución óptima especialmente cuando la matriz de datos es muy grande, dispersa o cuando hay datos faltantes, así que en este trabajo se exploran otros enfoques que utilizan algoritmos basados en el gradiente conjugado o a partir de una función sustituta.

2.4. Función sustituta para un biplot logístico

El problema (2.15) no se puede resolver de forma directa, una manera es usar un procedimiento que sustituya el problema de optimización por otro más simple y que conduzca a la misma solución. La idea es minimizar sucesivamente una función sustituta donde la secuencia de los minimizadores converge al óptimo de la función de pérdida.

Este es un método iterativo conocido como método MM, se denomina así porque funciona en dos pasos. En problemas de minimización, la primera M indica el paso de Mayorizar mientras que la segunda M representa el paso de Minimizar. Este método ha sido aplicado

en el análisis de escalamiento multidimensional (De Leeuw y Heiser, 1977), para el análisis de regresión robusta (Huber, 2011), en el análisis de correspondencias (Heiser, 1987), regresión logística (Böhning y Lindsay, 1988) y otros problemas aplicados en psicometría, imágenes médicas, procesamiento de señales o en problemas generales de optimización en machine learning (De Pierro, 1995; Kiers y Berge, 1992; Nguyen, 2017; Sun y col., 2016).

Formalmente, una función $g\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(l)}\right)$ es una función mayorizada o sustituta de $f(\boldsymbol{\theta})$ en el punto $\boldsymbol{\theta}^{(l)}$ si

$$f(\boldsymbol{\theta}^{(l)}) = g(\boldsymbol{\theta}^{(l)}|\boldsymbol{\theta}^{(l)}) \tag{2.16}$$

$$f(\boldsymbol{\theta}) \leq g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(l)}) \quad \text{para todo } \boldsymbol{\theta} \tag{2.17}$$

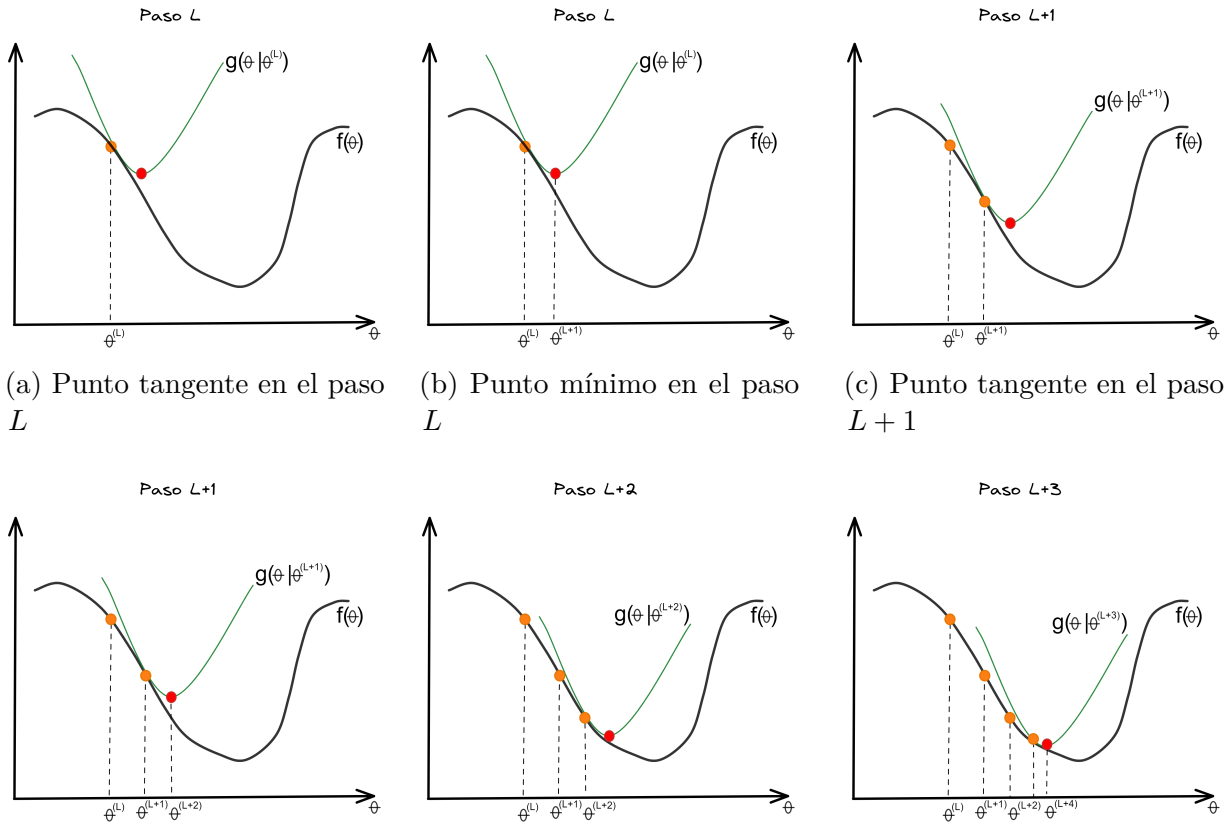
La primera es una condición de tangencia, mientras que la segunda es una condición de dominación (Lange, 2016). Esto significa que la superficie que toma a cada $\boldsymbol{\theta}$ y lo envía en $g\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(l)}\right)$ se encuentra sobre la superficie generada por $f(\boldsymbol{\theta})$ y es tangente a ésta en el punto $\boldsymbol{\theta} = \boldsymbol{\theta}^{(l)}$, donde $\boldsymbol{\theta}^{(l)}$ representa la l-ésima iteración en la búsqueda de la superficie $f(\boldsymbol{\theta})$ (Lange, 2013). La Figura 2.1 presenta una ilustración unidimensional del método, donde un algoritmo de minimización se aplica sobre la función mayorizada sustituta en lugar de la función objetivo inicial. Si $\boldsymbol{\theta}^{(l+1)}$ representa el mínimo de la función sustituta $g\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(l)}\right)$, entonces se puede demostrar que el método MM lleva a $f(\boldsymbol{\theta})$ en dirección descendente con cada iteración. De esta forma, las desigualdades

$$f\left(\boldsymbol{\theta}^{(l+1)}\right) \leq g\left(\boldsymbol{\theta}^{(l+1)}|\boldsymbol{\theta}^{(l)}\right) \leq g\left(\boldsymbol{\theta}^{(l)}|\boldsymbol{\theta}^{(l)}\right) = f\left(\boldsymbol{\theta}^{(l)}\right), \tag{2.18}$$

se obtienen directamente de la definición de $\boldsymbol{\theta}^{(l+1)}$ y de las condiciones de mayorización (2.16) y (2.17). Esta propiedad de descenso (2.18) le proporciona al método una valiosa propiedad que se puede aprovechar para estimar los parámetros de un modelo LB.

El rendimiento del algoritmo de minimización que se use en el segundo paso depende principalmente de la construcción de la función mayorizada sustituta. Existen varias formas de llegar a una función sustituta, por definición de convexidad, desigualdad de Jensen, desigualdad de Cauchy-Schwartz, expansión de Taylor, entre otros (Beck y Pan, 2018; Hunter y Lange, 2004).

Expresando la función de pérdida como $\mathcal{L}(\boldsymbol{\Theta}) = \sum_{i=1}^{n}\sum_{j=1}^{p} f(\theta_{ij})$, y si además se considera



(a) Punto tangente en el paso L

(b) Punto mínimo en el paso L

(c) Punto tangente en el paso $L+1$

(d) Punto mínimo en el paso $L+1$

(e) Punto tangente en el paso $L+2$

(f) Punto tangente en el paso $L+3$

Figura 2.1: Proceso iterativo de minimización usando una función sustituta.

que la función de enlace utilizada es logit, $\pi(\theta_{ij}) = (1 - \exp(-\theta_{ij}))^{-1}$, el gradiente es:

$$\begin{aligned}\nabla f(\theta_{ij}) &= -\left[x_{ij}\frac{1}{\pi(\theta_{ij})}\frac{\partial\pi(\theta_{ij})}{\partial\theta_{ij}} + (1-x_{ij})\frac{1}{1-\pi(\theta_{ij})}\frac{\partial(1-\pi(\theta_{ij}))}{\partial\theta_{ij}}\right] \\ &= -\left[x_{ij}(1-\pi(\theta_{ij})) - (1-x_{ij})\pi(\theta_{ij})\right] \\ &= \pi(\theta_{ij}) - x_{ij}.\end{aligned} \tag{2.19}$$

La segunda derivada, $\nabla^2 f(\theta_{ij}) = \pi(\theta_{ij})(1-\pi(\theta_{ij}))$ que es una función cuadrática que satisface que $0 \leq \nabla^2 f(\theta_{ij}) \leq 1/4$ debido a que x_{ij} se distribuye Bernoulli. A partir del resultado anterior, se puede usar la aproximación de Taylor para mayorizar $\mathcal{L}(\boldsymbol{\Theta})$ a una función cuadrática de $\boldsymbol{\Theta}$ usando el límite superior del gradiente de segundo orden, para lo cual se postula y se demuestra el siguiente Teorema.

Teorema 2. *Si* $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ *una matriz binaria, con* $\mathbf{x}_i \in \{0,1\}^p$, $i = 1, \ldots, n$ *y*

$x_{ij} \sim Ber(\pi(\theta_{ij}))$, *donde* $\pi(\theta_{ij}) = \{1 + exp(-\theta_{ij})\}^{-1}$ *con función de pérdida*

$$\mathcal{L}(\boldsymbol{\Theta}) = -\sum_{i=1}^{n}\sum_{j=1}^{p}\left[x_{ij}log(\pi(\theta_{ij})) + (1 - x_{ij})log(1 - \pi(\theta_{ij}))\right],$$

donde $\boldsymbol{\Theta} = \mathbf{1}_n\boldsymbol{\mu}^T + \mathbf{A}\mathbf{B}^T$, *es la matriz canónica de parámetros, entonces la función* $\mathcal{L}(\boldsymbol{\Theta})$ *puede ser mayorizada por*

$$\mathcal{G}\left(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(l)}\right) = \frac{1}{8}\left\|\mathbf{1}_n\boldsymbol{\mu}^T + \mathbf{A}\mathbf{B}^T - \mathbf{Z}_l\right\|_F^2,$$

con $\mathbf{Z}_l = \boldsymbol{\Theta}^{(l)} + 4(\mathbf{X} - \boldsymbol{\Pi}_l)$.

Demostración. Teniendo en cuenta que $\mathcal{L}(\boldsymbol{\Theta}) = \sum_{i=1}^{n}\sum_{j=1}^{p} f(\theta_{ij})$, se tiene que la función de pérdida con un parámetro, $f(\theta_{ij})$, puede ser aproximada de forma cuadrática en $\theta_{ij}^{(l)}$ usando la expansión de Taylor de segundo orden, así

$$\begin{aligned}
f(\theta_{ij}) &= -\left[x_{ij}log(\pi(\theta_{ij})) + (1 - x_{ij})log(1 - \pi(\theta_{ij}))\right] && (2.20)\\
&= f(\theta_{ij}^{(l)}) + \nabla f(\theta_{ij}^{(l)})(\theta_{ij} - \theta_{ij}^{(l)}) + \frac{1}{2}\nabla^2 f(\theta_{ij}^{(l)})(\theta_{ij} - \theta_{ij}^{(l)})^2 && (2.21)\\
&= f(\theta_{ij}^{(l)}) + (\pi(\theta_{ij}^{(l)}) - x_{ij})(\theta_{ij} - \theta_{ij}^{(l)}) + \frac{1}{2}\pi(\theta_{ij}^{(l)})(1 - \pi(\theta_{ij}^{(l)}))(\theta_{ij} - \theta_{ij}^{(l)})^2 && (2.22)
\end{aligned}$$

como $\nabla^2 f(\theta_{ij}^{(l)}) = \pi(\theta_{ij})(1 - \pi(\theta_{ij})) \leq 1/4$ entonces

$$\leq f(\theta_{ij}^{(l)}) + (\pi(\theta_{ij}^{(l)}) - x_{ij})(\theta_{ij} - \theta_{ij}^{(l)}) + \frac{1}{2}\frac{1}{4}(\theta_{ij} - \theta_{ij}^{(l)})^2 \quad (2.23)$$

completando el cuadrado se tiene que

$$\begin{aligned}
&= f(\theta_{ij}^{(l)}) + \frac{1}{8}\left[\theta_{ij} - \left(\theta_{ij}^{(l)} + \frac{\pi(\theta_{ij}^{(l)}) - x_{ij}}{1/4}\right)\right]^2 - \frac{1}{2}\frac{(\pi(\theta_{ij}^{(l)}) - x_{ij})^2}{1/4} && (2.24)\\
&= f(\theta_{ij}^{(l)}) + \frac{1}{8}\left(\theta_{ij} - \theta_{ij}^{(l)} + 4(\pi(\theta_{ij}^{(l)}) - x_{ij})\right)^2 - 2(\pi(\theta_{ij}^{(l)}) - x_{ij})^2 && (2.25)\\
&= \frac{1}{8}\left(\theta_{ij} - \theta_{ij}^{(l)} + 4(\pi(\theta_{ij}^{(l)}) - x_{ij})\right)^2 + C. && (2.26)
\end{aligned}$$

Por lo tanto, la función mayorizada para la función de pérdida usando toda la matriz canónica de parámetros se puede obtener desde la desigualdad (2.23), con lo cual

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\Theta}) &\leq \sum_{i=1}^{n}\sum_{j=1}^{p}\left[f(\theta_{ij}^{(l)}) + (\pi(\theta_{ij}^{(l)}) - x_{ij})(\theta_{ij} - \theta_{ij}^{(l)}) + \frac{1}{8}(\theta_{ij} - \theta_{ij}^{(l)})^2\right] && (2.27)\\
&= \frac{1}{8}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\theta_{ij} - z_{ij}^{(l)}\right)^2 + C, && (2.28)
\end{aligned}$$

donde C es una constante que no depende de θ_{ij}, $\theta_{ij}^{(l)}$ es la l-ésima aproximación de θ_{ij} y $z_{ij}^{(l)} = \theta_{ij}^{(l)} + 4\left(x_{ij} - \pi(\theta_{ij}^{(l)})\right)$. En términos matriciales, si $\mathbf{Z}_l$ es la matriz donde el ij-ésimo

elemento es igual a $z_{ij}^{(l)}$ entonces

$$\mathcal{L}(\boldsymbol{\Theta}) \leq \frac{1}{8} \|\boldsymbol{\Theta} - \mathbf{Z}_l\|_F^2 + C, \tag{2.29}$$

De esta manera, la función mayorizada o función sustituta de $\mathcal{L}(\boldsymbol{\Theta})$ es

$$\mathcal{G}\left(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(l)}\right) = \frac{1}{8} \left\|\mathbf{1}_n \boldsymbol{\mu}^T + \mathbf{A}\mathbf{B}^T - \mathbf{Z}_l\right\|_F^2. \tag{2.30}$$

□

2.5. Evaluación del modelo

En los siguientes capítulos se proponen diferentes algoritmos para estimar los parámetros del modelo LB. Con el propósito de evaluar el desempeño de los diferentes enfoques, se define el error de entrenamiento como la tasa de clasificación errónea al utilizar la estructura generada por el ajuste del modelo a partir de un algoritmo de minimización específico al utilizar un conjunto de datos incompleto.

Cada algoritmo permite llegar a una estimación del vector de desplazamiento del modelo $\hat{\boldsymbol{\mu}}$, los marcadores fila $\hat{\mathbf{A}}$ y de los marcadores columna $\hat{\mathbf{B}}$, con lo cual se puede calcular $\hat{\boldsymbol{\Theta}} = \mathbf{1}_n \hat{\boldsymbol{\mu}}^T + \hat{\mathbf{A}}\hat{\mathbf{B}}^T$ y, usando la matriz de probabilidades estimadas $\hat{\boldsymbol{\Pi}} = \pi\left(\hat{\boldsymbol{\Theta}}\right)$ se puede llegar a la matriz predicha $\hat{\mathbf{X}}$.

Para evaluar el rendimiento de los algoritmos, se propone seleccionar p umbrales, $0 < \delta_j < 1, j = 1, ..., p$, uno por cada variable de la matriz $\mathbf{X}$, y luego aplicar la regla de clasificación a partir de la matriz de probabilidades estimadas. Con regularidad ocurre que las clases de la matriz binaria $\mathbf{X}$ pueden estar desequilibradas, es decir, una categoría aparece con mayor frecuencia que la otra, así que el error de entrenamiento es calculado con la medida de tasa de error equilibrada (TEE), que se puede calcular como el complemento de la precisión equilibrada[1] (TPE), la cual es una medida que se considera más apropiada en estos casos (Velez y col., 2007; Wei y Dunbrack Jr, 2013). A continuación se presentan algunos elementos que permitirán realizar la definición formal.

En la Tabla 2.2, VP se refiere a la cantidad de *"verdaderos positivos"*, es decir donde el valor

[1]En el contexto del *Machine Learning* a la tasa de error equilibrada se le conoce como BER por sus siglas en inglés, *Balanced Error Rate*, mientras que la precisión equilibrada se acostumbra a denotar como BACC, *Balanced Accuracy*.

Tabla 2.2: Matriz de confusión.

	Valor real	
Valor predicho	1	0
1	VP	FP
0	FN	VN

real era 1 y el modelo lo predijo correctamente; FP es la cantidad de *"falsos positivos"*, en estos casos el valor real era 0 y el modelo realizó una predicción errada. FN es la cantidad de *"falsos negativos"*, que son los casos donde el valor real es 0 y el modelo predijo 1) y VN es la cantidad de *"verdaderos negativos"* que corresponde a los casos donde el valor real es 0 y el modelo lo predijo correctamente.

La precisión equilibrada se basa en la *sensibilidad*, también conocida como tasa verdaderos positivos o tasa de recuperación y, la *especificidad* también conocida como tasa de verdaderos positivos. La sensibilidad se define como:

$$Sensibilidad = \frac{VP}{VP + FN}. \tag{2.31}$$

La especificidad se define como:

$$Especificidad = \frac{VN}{VN + FP}. \tag{2.32}$$

La *precisión equilibrada* se calcula como el promedio entre la sensibilidad y la especificidad:

$$TPE = \frac{1}{2}\left(\frac{VP}{VP + FN} + \frac{VN}{VN + FP}\right). \tag{2.33}$$

Mientras que la tasa de error equilibra es $TEE = 1 - TPE$. De esta forma se evita que una de las clases tenga un mayor peso, lo que resulta muy útil para evaluar el desempeño de la clasificación, especialmente cuando el conjunto de datos está desequilibrado.

Para decidir cuando el valor predicho debe clasificarse como 0 o como 1, se propone seleccionar un umbral para cada variable. Este valor de umbral es seleccionado al minimizar la tasa de error equilibrada o lo que es equivalente, maximizar la tasa de precisión equilibrada, para cada variable sobre el conjunto de entrenamiento, y luego esta regla se aplica al conjunto de prueba, esto evita que los resultados puedan estar sesgados. El umbral para la variable j se define como el valor $0 < \delta_j < 1$ donde la tasa de error equilibrada, TEE, para dicha

variable es mínima:

$$\delta_j = \arg\min_{\delta} \{TEE(\mathbf{x}_j|\delta) : 0 < \delta < 1\}, j = 1, \ldots, p. \tag{2.34}$$

2.6. Cantidad de ejes a retener

En general, para los modelos de reducción de la dimensionalidad, la elección de k suele representar un problema (Owen, Perry y col., 2009). En los métodos clásicos, criterios basados en los valores propios o en el porcentaje de varianza explicada son regularmente utilizados como medida intuitiva para la selección de k. Cuando la matriz de datos en binaria, Vicente-Villardón y Hernández-Sánchez (2020) sugieren medir la capacidad de predicción de las variables utilizando, el pseudo R cuadrado de Nagelkerke, AIC, BIC o cualquier medida de ajuste utilizada tradicionalmente en la regresión logística. Sin embargo, encontrar un método que determine el valor apropiado de k antes de ajustar el modelo, aún es un problema por resolver en el modelo LB.

En este caso, la matriz canónica de parámetros $\mathbf{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)^T$ puede ser representada por marcadores fila y marcadores columna para algún entero $k \leq r$ que satisface que $\boldsymbol{\theta}_i = \boldsymbol{\mu} + \sum_{s=1}^{k} a_{is}\mathbf{b}_s$, $i = 1, \ldots, n$. De modo que, estimar un valor apropiado para k es un aspecto clave que influye en la especificación del modelo y, que podría considerarse como un hiperparámetro del mismo.

En el caso de los modelos supervisados de regresión o de clasificación, se puede usar la validación cruzada para calcular la precisión del modelo o para optimizar los hiperparámetros (Fu, 1998; Sirimongkolkasem y Drikvandi, 2019). El concepto de validación cruzada fue inicialmente propuesto por Mosier (1951) como una forma de evaluar la efectividad de los pesos de un modelo usando una segunda muestra extraída de manera similar. El procedimiento consiste en eliminar algunas filas del conjunto de datos, por ejemplo el 30 % y dejarlo como conjunto de prueba, luego se ajusta el modelo con los datos restantes, comúnmente denominado conjunto de entrenamiento. Finalmente, el modelo ajustado se aplica al conjunto de prueba para medir su precisión, la Figura 2.2 presenta una ilustración de los pasos necesarios para realizar este proceso.

Un aspecto relevante a destacar es que al omitir algunas filas del conjunto de datos, en

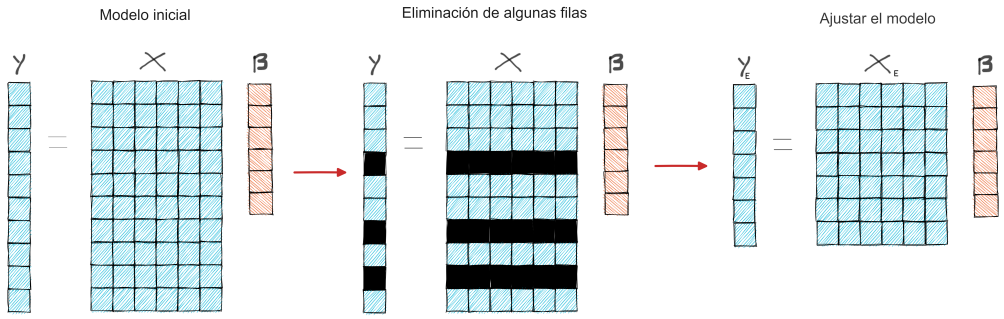


Figura 2.2: Procedimiento de validación en modelos de regresión.

los modelos supervisados no se afecta el espacio de parámetros del modelo (vector $\boldsymbol{\beta}$). Sin embargo, este procedimiento de validación cruzada, no se puede adaptar con facilidad a los métodos biplot.

Para ilustar el problema de aplicar la validación cruzada de la forma tradicional en los métodos biplot, se considera un biplot clásico bajo las condiciones presentadas en la ecuación (1) de la introducción, en donde, sin pérdida de generalidad, se supone que las variables están centradas, por lo que el problema de minimización es

$$\min_{\mathbf{A},\mathbf{B}} \left\| \mathbf{X} - \mathbf{A}\mathbf{B}^T \right\|_F^2 . \tag{2.35}$$

La matriz $\mathbf{X}$ es aproximada por el espacio de parámetros dado por $\boldsymbol{\Theta} = \mathbf{A}\mathbf{B}^T$, de forma ilustrativa se vería como en la Figura 2.3.

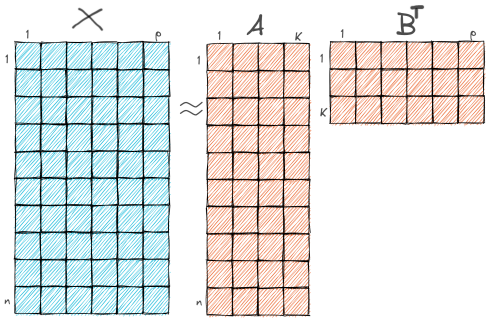


Figura 2.3: Aproximación de la matriz de datos.

Al aplicar un método de validación cruzada de la misma forma que se realiza en modelos supervisados, se eliminarían filas o columnas de la matriz de datos, pero como se observa en la Figura 2.4, esto implicaría que no se pueden estimar todos los elementos del espacio de parámetros, es decir, se omitiría una fila de $\mathbf{A}$ o una columna de $\mathbf{B}^T$. Eastment y Krzanowski (1982) sugieren algunos ajustes que combinan el resultado de la SVD cuando se omite la

fila i y el resultado de la SVD cuando se omite la columna j, pero la eliminación separada de filas y columnas generalmente produce errores que se elevan al cuadrado y disminuyen monótonamente con k (S. Dias y Krzanowski, 2003). Como resultado, en la práctica se utilizan algunos ajustes incómodos basados en grados de libertad (Owen, Perry y col., 2009). Este y otros problemas de eliminar una fila o columna de la matriz de datos son revisados con mayor detalle por Bro y col. (2008) y, Owen, Perry y col. (2009).

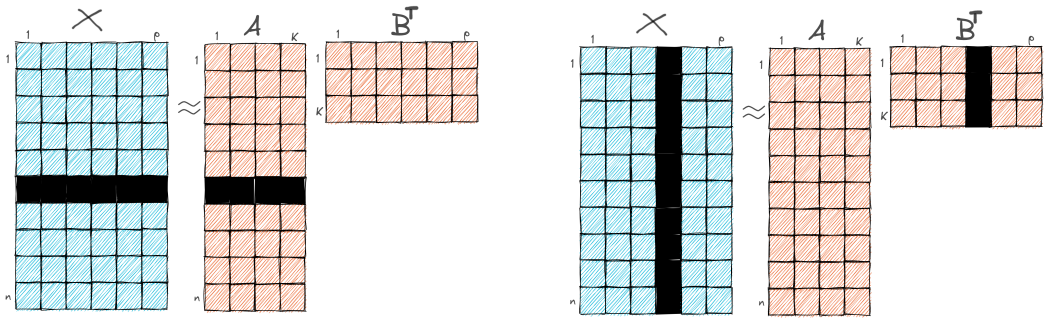


Figura 2.4: SVD al omitir una fila o columna.

El problema anterior se puede evitar en el análisis multivariante no supervisado usando un patrón de eliminación disperso o diagonal sugerido por Wold en el contexto de PCA[2], como un método utilizado para identificar las dimensiones que mejor describen las variaciones sistemáticas en los datos (Wold, 1978, 1976). Por su parte Gabriel (2002) propone un método de validación bi-cruzada (BCV), omitiendo simultáneamente una fila y columna, este método luego es extendido por Owen, Perry y col. (2009) a modelos de factorización de matrices no negativas (NMF) y, por Fu y Perry (2020) para la elección del número de clústers en K-medias. En cualquiera de los casos el enfoque usa el principio básico de la validación cruzada, que consiste en omitir parte de los datos, y estimar los valores omitidos usando un modelo para luego comparar las estimaciones obtenidas con los valores reales.

Aunque el procedimiento que se presenta para la elección del valor de k en el modelo LB podría usar cualquiera de los dos enfoques de eliminación, acá se usa el patrón diagonal propuesto por Wold (1978). Para esto se eligen M segmentos o pliegues[3], para el segmento m ($m = 1, \ldots, M$) los elementos $m, m + M, m + 2M$, etc., enumerados por fila, pueden ser tomados como valores faltantes (véase la Figura 2.5). De esta manera, al completar los

[2]Este método de eliminación también se conoce como *"moteo"* de Wold.

[3]En algoritmos de machine learning se conoce como validación cruzada K-fold, acá se usará la letra m para referirse a los segmentos y M para el total de segmentos.

M segmentos, todos los elementos han sido omitidos una vez. Wold (1978) sugiere que la matriz original puede ser dividida con M entre 4 y 7 segmentos, mientras que Bro y col. (2008) utilizan $M = 7$.

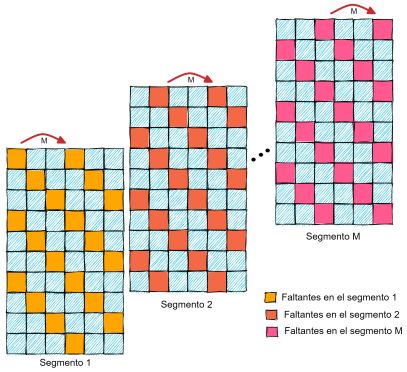


Figura 2.5: Patrón de eliminación diagonal de Wold (1978).

El procedimiento propuesto para encontrar el valor apropiado de k en el modelo LB consiste en utilizar un patrón de eliminación diagonal y tratar esos datos como faltantes en la generación de los M segmentos, evitando así la eliminación de filas o columnas completas. En el segmento m, se divide la matriz $\mathbf{X}$ en $\mathbf{X}^{(-m)}$ y $\mathbf{X}^{(m)}$, donde $\mathbf{X}^{(-m)}$ contiene todas las observaciones excepto las omitidas por el patrón de eliminación, mientras que $\mathbf{X}^{(m)}$ contiene solo las observaciones omitidas. Sea $\mathbf{W}$ una matriz binaria donde $w_{ij} = 0$ si x_{ij} es excluido y $w_{ij} = 1$ en otro caso. Para un valor fijo k, el modelo LB se ajusta teniendo en cuenta solo las entradas conocidas, de esa forma la función de pérdida es:

$$\min_{\boldsymbol{\mu},\mathbf{A},\mathbf{B}} - log(p(\mathbf{X};\boldsymbol{\Theta},\mathbf{W})) \tag{2.36a}$$

$$= -log\left(\prod_{i=1}^{n}\prod_{j=1}^{p}\left[p(x_{ij};\theta_{ij})\right]^{w_{ij}}\right) \tag{2.36b}$$

$$= -\sum_{i=1}^{n}\sum_{j=1}^{p} w_{ij}\left[x_{ij}log(\pi(\theta_{ij})) + (1-x_{ij})log(1-\pi(\theta_{ij}))\right]. \tag{2.36c}$$

Una vez estimado el espacio de parámetros se pueden predecir los valores eliminados de la matriz de datos a partir de los umbrales δ_j determinados para cada variable según se señaló en la sección 2.5. Con la matriz predicha se calcula la tasa de error equilibrada, $TEE = 1 - TPE$. Cuando la medida es calculada teniendo en cuenta toda la matriz predicha, se denominará *error de entrenamiento*, pero este indicador puede ser muy flexible

para medir la tasa de clasificación errada, así que también se realiza el cálculo considerando solo los datos eliminados inicialmente, este se denominará *error de generalización*. La Figura 2.6 presenta una ilustración del procedimiento, mientras que el pseudocódigo se escribe formalmente en Algoritmo 3.

Algoritmo 3 Algoritmo de validación cruzada para determinar el valor de k en el modelo LB

Entrada X
Salida k

1: $\text{mín}_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} - \sum_{i=1}^{n}\sum_{j=1}^{p}\left[x_{ij}log(\pi(\theta_{ij})) + (1-x_{ij})log(1-\pi(\theta_{ij}))\right]$
2: Calcular $\delta_j = \arg\min_{\delta}\left\{TEE(\mathbf{x}_j|\delta) : 0 < \delta < 1\right\}, j = 1, \ldots, p$
3: **for** $k = 1$ `hasta` K **do**
4: **for** $m = 1$ `hasta` M **do**
5: Separar $\mathbf{X}$ en $\mathbf{X}^{(-m)}$ y $\mathbf{X}^{(m)}$ usando un procedimiento de eliminación diagonal.
6: $\text{mín}_{\boldsymbol{\mu},\mathbf{A},\mathbf{B}} - \sum_{i=1}^{n}\sum_{j=1}^{p} w_{ij}\left[x_{ij}log(\pi(\theta_{ij})) + (1-x_{ij})log(1-\pi(\theta_{ij}))\right]$
7: Estimar $\hat{\boldsymbol{\Theta}} = \mathbf{1}_n\hat{\boldsymbol{\mu}}^T + \hat{\mathbf{A}}\hat{\mathbf{B}}^T$
8: Calcular $\hat{\boldsymbol{\Pi}} = \pi(\hat{\boldsymbol{\Theta}})$
9: Calcular $\hat{\mathbf{X}}^{(m)}$, $\hat{x}_{ij}^{(m)} = 1$ si $\pi(\boldsymbol{\Theta}) > \delta_j$, y $\hat{x}_{ij}^{(m)} = 0$ en otro caso.
10: Calcular $TEE_m^k = 1 - \frac{1}{2}\left(\frac{VP}{VP+FN} + \frac{VN}{VN+FP}\right)$
11: **end for**
12: Calcular $EG^{(k)} = \frac{1}{M}\sum_{m=1}^{M} TEE_m^k$
13: **end for**
14: $k = \arg\min_{k} EG^{(k)}$

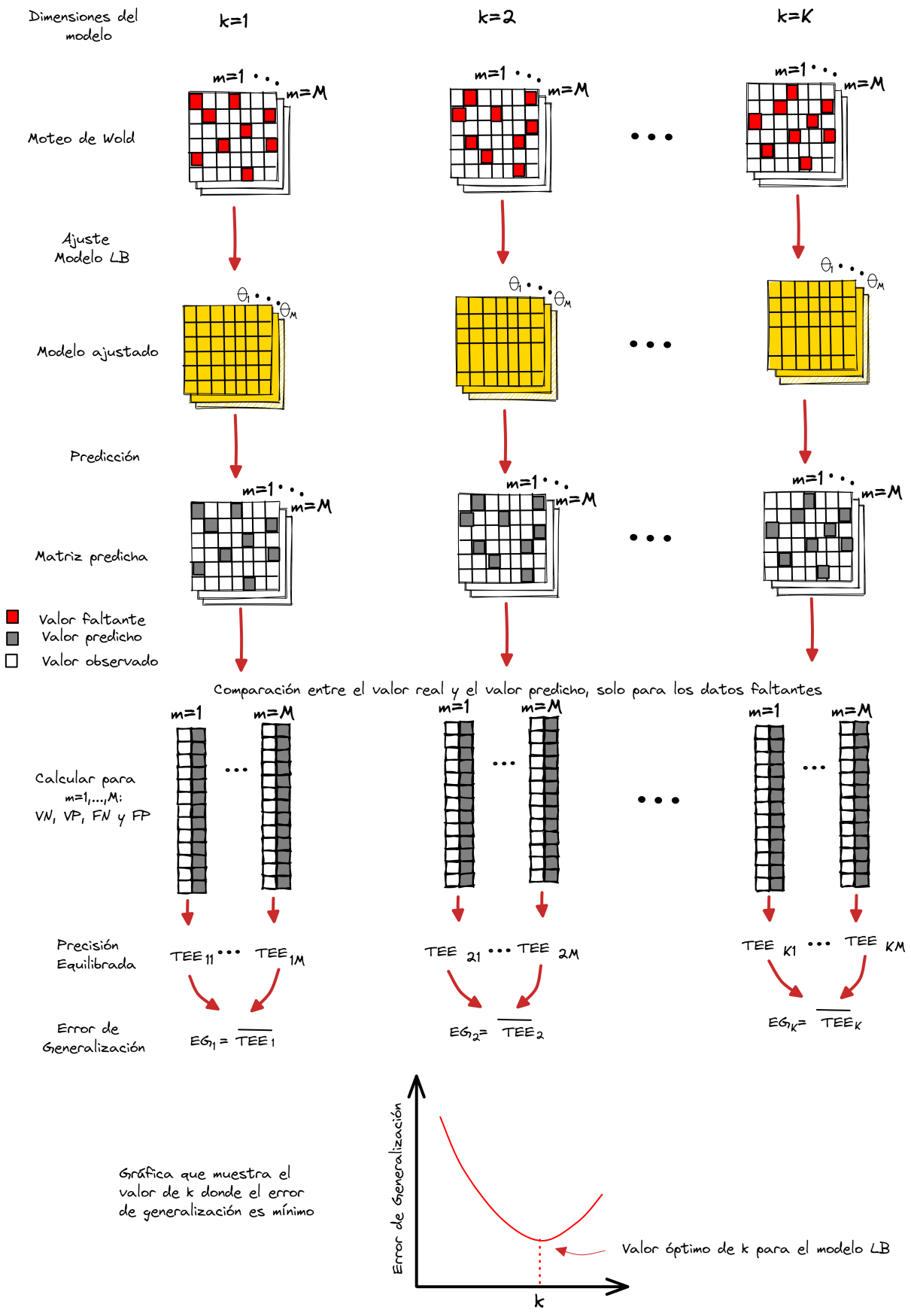


Figura 2.6: Proceso de validación cruzada para elegir el valor óptimo de k en el modelo LB.

2.7. Contribuciones realizadas en este capítulo

A partir de la formulación general, los métodos biplot pueden ser extendidos para encontrar funciones de pérdida adecuadas a partir de los factores de normalización de las distribuciones de la familia exponencial. Esto permitió llegar a la función de pérdida de un biplot logístico desde otra perspectiva, y que puede ser utilizada para formular el biplot para otros tipos de datos.

El problema de minimización para un modelo LB fue sustituido por otro que resulta más fácil de resolver, lo que permite acelerar la convergencia de los algoritmos que se usen para estimar los parámetros del modelo.

Se presenta un procedimiento que permite la elección del número de dimensiones, que hasta ahora no había sido investigado para un modelo LB. Este método se basa en un algoritmo de validación cruzada usando un proceso de eliminación que evita perder una columna o fila de forma completa, de ese modo, se usan sólo los valores conocidos para ajustar el modelo LB y tomar los datos faltantes generados por el procedimiento para calcular los indicadores de rendimiento, permitiendo encontrar el punto de inflexión que representa el valor apropiado para k.

Un resumen de las contribuciones presentadas en este capítulo fueron presentadas en:

- ***I Foro Internacional de Charlas Multidisciplinarias de Análisis de Datos***, evento organizado por el Centro de Investigación de Estadística Multivariante Aplicada (CIEMA) de la Universidad de Colima, México. Celebrado del 08 y 19 de junio del 2020, con el trabajo *"El impacto de la estadística para la toma de decisiones. Un análisis de las comorbilidades de fallecidos por el SARS-CoV-2 usando biplot logístico"*.
- ***Celebración de los 13 años de la Facultad de Ciencias Exactas y Naturales***, evento organizado por la Universidad de Cartagena, el día 05 de junio del 2020, con la conferencia *"La estadística: Métodos que generan valor en la toma de decisiones"*.

Capítulo 3

Biplot logístico usando algoritmos de aprendizaje automático

3.1. Introducción

En la actualidad se cuenta con acceso a un gran volumen de datos, esto ha llevado a que los métodos estadísticos también se tengan que adaptar a las nuevas necesidades para el análisis de la información a partir de nuevas herramientas que permiten modelar y comprender conjuntos de datos complejos. James y col. (2013) lo denominan aprendizaje estadístico, un área recientemente desarrollada en estadística que se combina con desarrollos paralelos en informática y, en particular de aprendizaje automático más conocido como *machine learning*.

El campo abarca muchos métodos, en los cuales se usan diferentes modelos de optimización matemática para determinar la solución más eficiente a un problema de maximización o de minimización de una función objetivo, que ha sido definida para medir el rendimiento del ajuste de un modelo. En general, puede ser computacionalmente difícil encontrar el óptimo global de un modelo generalizado para una matriz de bajo rango, pero hay varias formas hacerlo (Udell y col., 2016). En la literatura se presenta una amplia variedad de algoritmos para la factorización de matrices. Por ejemplo, existen métodos de minimización alternante (De Leeuw, 1984; Vicente-Villardon y col., 2006), métodos de Newton alternantes (Singh y Gordon, 2008b), descenso de gradiente (Recht y col., 2012; Recht y col., 2011), gradientes conjugados (Rennie y Srebro, 2005; Srebro y Jaakkola, 2003), métodos EM (Mazumder y col., 2010; Tipping y Bishop, 1999) y, métodos de Mayorización y Minimización (MM)

(De Leeuw, 2006; Landgraf y Lee, 2020; Song y col., 2020).

Los métodos de optimización multivariante se pueden clasificar en algoritmos que no usan el gradiente, los que usan el gradiente y los que usan la matriz hessiana. Los algoritmos que no se basan en el gradiente no resultan muy eficientes porque suelen ser costosos computacionalmente, aunque son útiles en escenarios donde resulta complejo diferenciar la función objetivo (Powell, 2008), así que no son considerados en este trabajo. En el caso de la función de pérdida de un modelo LB, se puede proporcionar el gradiente y de esta forma indicar la dirección de búsqueda que debe seguir el algoritmo, así que es posible adaptar algoritmos como el descenso del gradiente o métodos no lineales que usan el gradiente conjugado para estimar los parámetros del modelo.

Como se presentó previamente, el algoritmo alternante propuesto por (Vicente-Villardon y col., 2006) usa la matriz hessiana para llegar a una solución del problema, pero el algoritmo requiere de una doble iteración. También se presentó el procedimiento externo propuesto por (Demey y col., 2008). Sin embargo, estos métodos pueden llegar ser muy exigentes desde un punto de vista computacional cuando se tiene grandes volúmenes de datos o cuando la matriz de datos es dispersa.

Para resolver este problema, en este capítulo se explora un enfoque a partir del gradiente conjugado y otro a partir de la función sustituta con el fin de usar un método MM. Estas formulaciones permiten llegar a soluciones que se basan en algoritmos que requieren especificar un punto de arranque por el usuario, denotado por $\mathbf{\Theta_0}$, $\boldsymbol{\mu}_0$, $\mathbf{A}_0$ o $\mathbf{B}_0$. En general, asignar un punto de partida adecuado puede contribuir en una reducción del tiempo necesario para encontrar el minimizador. Por ejemplo, un buen punto de arranque podría estar dado por las coordenadas principales o componentes principales, pero solo este paso podría ser costoso computacionalmente para grandes volúmenes de datos, por esto se ha decidido utilizar entradas no informativas a partir de valores aleatorios con distribución uniforme y así adaptar los algoritmos desde una perspectiva más general.

Este capítulo está organizado de la siguiente manera: la sección 3.2 presenta la propuesta para la adaptación de los algoritmos basados en el gradiente conjugado para estimar los parámetros del modelo LB; en la sección 3.3 se presenta un enfoque que usa un procedimiento basado en un método MM para usar la función sustituta y aplicar un algoritmo de descenso

coordinado por bloques para estimar los parámetros del modelo LB; en la sección 3.4 se hace un resumen de las medidas que se usan para medir el rendimiento de los algoritmos que se van a comparar. Para generar escenarios que permitan la comparación de los diferentes modelos, se simulan matrices binarias, el procedimiento utilizado se presenta en la sección 3.5 y en la sección 3.6 se realiza la descripción del proceso de Monte Carlo y se presentan los resultados obtenidos. Finalmente, en la sección 3.7 se realiza una aplicación utilizando datos reales sobre metilaciones del ADN.

3.2. Algoritmo del gradiente conjugado

Una de las técnicas más conocidas es el algoritmo del descenso del gradiente, el cual permite resolver el problema de minimización desde un enfoque basado en el gradiente de la función. El algoritmo se basa en actualizar cada parámetro $\boldsymbol{\Theta} = (\theta_1, \ldots, \theta_n)^T$ usando una velocidad o tasa de aprendizaje α. Sin embargo, una de las limitaciones del algoritmo del descenso del gradiente es que la velocidad de aprendizaje, α, no se actualiza automáticamente a partir de los resultados que se van obteniendo en cada iteración. Por el contrario, los métodos basados en el gradiente conjugado permiten resolver este inconveniente, proporcionando una dirección de búsqueda.

Los métodos de gradiente conjugado se atribuyen a Hestenes y Stiefel (1952), quienes propusieron un algoritmo para conjuntos de ecuaciones lineales. En la década de 1960, los métodos de gradiente conjugado se extendieron para resolver problemas de otro tipo de funciones. El primer algoritmo de gradiente conjugado no lineal fue propuesto por Fletcher y Reeves (1964), y luego en Polak y Ribiere (1969) y Polyak (1969). Con el pasar de los años se han propuesto muchas variantes; algunas de éstas son adaptadas al modelo LB y son presentadas en esta sección.

Una de las características clave de estos algoritmos es que no requieren del almacenamiento de una matriz, por lo que el desempeño suele ser bastante eficiente. Dentro de las propiedades más destacadas del algoritmo es su capacidad de generar, de forma automática y sin mayor costo computacional, un conjunto de vectores con una propiedad conocida como conjugada, de modo que calcula el nuevo vector d_l utilizando solo el vector anterior d_{l-1}, así que no necesita del conocimiento (ni almacenamiento) de todos los demás vectores conjugados

$d_{l-2}, \ldots, d_1$, por eso el método es considerado de poca memoria, lo que representa que sea de bajo coste computacional y puede llegar a ser eficiente en contextos de big data (Nocedal y Wright, 2006).

El procedimiento general de optimización parte de un punto θ_l, a partir del gradiente se identifica la dirección de descenso con mayor pendiente, d_l^T, y para calcular la velocidad de aprendizaje α_l se lleva a cabo una estrategia de búsqueda en línea que identifique el mínimo aproximado de $\mathcal{L}(\boldsymbol{\Theta})$ a lo largo de d_l, posteriormente un parámetro β_l define la regla para la actualización de la dirección basada en el gradiente, que permite actualizar de forma simultánea el espacio de parámetros. De acuerdo con lo anterior, el algoritmo está incompleto sin especificar un método de búsqueda de línea y la regla de actualización, por lo que se describen a continuación.

3.2.1. Métodos de búsqueda en línea

En los métodos de búsqueda en línea, el algoritmo elige una dirección d_l y busca a lo largo de esta dirección desde la iteración actual $\boldsymbol{\Theta}_l$ para lograr una nueva iteración con un valor más bajo de la función de pérdida. La velocidad de aprendizaje α en el paso l, se puede encontrar aproximando la solución al siguiente problema de minimización univariante a lo largo de una dirección d_l

$$\min_{\alpha>0} \mathcal{L}(\boldsymbol{\Theta} + \alpha d_l). \tag{3.1}$$

Resolver (3.1) de forma exacta puede ser costoso e innecesario (Nocedal y Wright, 2006), para ello se han propuesto varias reglas para encontrar el valor de α_l, que garantizan la convergencia global del algoritmo de optimización, estas son conocidas como reglas de minimización direccional.

Los algoritmos de búsqueda de línea, en general, prueban una secuencia de valores candidatos para la velocidad de aprendizaje, y se detienen cuando uno de estos valores cumple con ciertas condiciones. Para el modelo LB, una condición simple que debe ser impuesta sobre α_l es requerir una reducción en la función de pérdida $\mathcal{L}$, de modo que $\mathcal{L}(\boldsymbol{\Theta}_l + \alpha_l d_l) < \mathcal{L}(\boldsymbol{\Theta}_l)$, pero este requisito no es suficiente para obtener una convergencia. Un ejemplo sencillo se puede observar con una función convexa, la Figura 3.10 muestra que el mínimo está por debajo de cero, y aunque la secuencia de iteraciones de x_l para la cual la función

$f(x_l) = a/l$ donde a es un valor constante, produce una disminución en cada iteración, su límite tiende a cero. La reducción insuficiente de la función en cada paso hace que el algoritmo no logre una convergencia al punto mínimo (Nocedal y Wright, 2006). Para evitar el problema anterior, es necesario que se cumpla una condición de disminución suficiente.

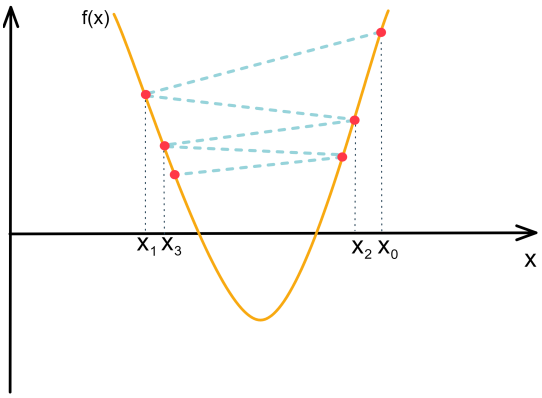


Figura 3.1: Condición de reducción en la función de pérdida.

Un requerimiento en los algoritmos de gradiente conjugado para lograr una convergencia en la implementación es que se cumplan las *condiciones de Wolfe*, que están dadas por una condición de disminución suficiente, conocida como *condición de Armijo* y una *condición de curvatura* (Pytlak, 2008). La condición impuesta sobre α_l es denominada *condición de Armijo*, que establece la siguiente desigualdad:

$$\mathcal{L}\left(\boldsymbol{\Theta}_l + \alpha d_l\right) \leq \mathcal{L}\left(\boldsymbol{\Theta}_l\right) + c_1 \alpha \nabla \mathcal{L}_l^T d_l, \tag{3.2}$$

para una constante $c_1 \in (0, 1)$. Esto significa que la reducción en la función de pérdida debe ser proporcional a la velocidad de aprendizaje α_l y a la dirección $\nabla \mathcal{L}_l^T d_l$.

Aunque la *condición de Armijo* impone cotas sobre el tamaño de la velocidad de aprendizaje en la dirección del descenso, puede ocurrir que la velocidad de aprendizaje sea tan pequeña que no haya una convergencia, así que Wolfe incluye una segunda *condición de curvatura*, donde se requiere que dado el valor de α se cumpla:

$$\nabla \mathcal{L}\left(\boldsymbol{\Theta}_l + \alpha_l d_l\right)^T d_l \leq c_2 \nabla \mathcal{L}_l^T d_l, \tag{3.3}$$

con $c_2 \in (c_1, 1)$ y c_1 la constante establecida en la *condición de Armijo*. La parte de la izquierda corresponde a la derivada de $\mathcal{L}\left(\boldsymbol{\Theta} + \alpha d_l\right)$ con respecto a α_l, que regularmente se denota por $\phi'(\alpha_l)$. Así que esta condición impone que la pendiente de ϕ en α_l sea mayor

que c_2 veces la pendiente inicial. Esto tiene mucho sentido porque si la pendiente es muy negativa entonces esto es un indicador de que se puede decrecer mucho en esa dirección, en cambio si la pendiente no es tan negativa o es positiva entonces es una señal de que la búsqueda debe terminar porque no se puede esperar una mayor disminución. En ocasiones se requiere modificar la condición de curvatura para forzar que α_l se ubique en al menos una vecindad de un minimizador local o punto estacionario de ϕ. Las *condiciones fuertes de Wolf* requieren que dado α_l se cumpla

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\Theta}_l + \alpha d_l\right) &\leq \mathcal{L}\left(\boldsymbol{\Theta}_l\right) + c_1 \alpha \nabla \mathcal{L}_l^T d_l, \\ \left\| \nabla \mathcal{L}\left(\boldsymbol{\Theta}_l + \alpha_l d_l\right)^T d_l \right\| &\leq c_2 \left\| \nabla \mathcal{L}_l^T d_l \right\|, \end{aligned} \tag{3.4}$$

con $0 < c_1 < c_2 < 1$. En la práctica es usual tomar valores de c_1 cercanos a cero y $c_2 = 0.9$ cuando las direcciones de búsqueda d_l se eligen con métodos Newton o de cuasi-Newton; mientras que $c_2 = 0.1$ es usual cuando d_l es obtenido por un método de gradiente conjugado no lineal (Nocedal y Wright, 2006).

3.2.2. Adaptación del algoritmo del gradiente conjugado a un biplot logístico

Sea $\nabla \mathcal{L}\left(\boldsymbol{\Theta}\right)$ una matriz donde el ij-ésimo elemento es igual a $\pi(\theta_{ij}) - x_{ij}$, que puede ser expresado en forma matricial como

$$\nabla \mathcal{L}\left(\boldsymbol{\Theta}\right) = \boldsymbol{\Pi} - \mathbf{X}. \tag{3.5}$$

Puesto que $\theta_{ij} = \mu_j + \mathbf{a}_i^T \mathbf{b}_j$, entonces $\mathcal{L}\left(\boldsymbol{\Theta}\right)$ es una función que involucra las matrices $\mathbf{A}$ y $\mathbf{B}$ a través de $\pi(\theta_{ij}) = \pi(\mu_j + \mathbf{a}_i^T \mathbf{b}_j)$. De esta forma, también es posible calcular las derivadas parciales con respecto a $\boldsymbol{\mu}$, $\mathbf{A}$, y $\mathbf{B}$, así:

$$\frac{\partial f(\theta_{ij})}{\partial a_{is}} = -\left[x_{ij} \frac{1}{\pi(\theta_{ij})} \frac{\partial \pi(\theta_{ij})}{\partial a_{is}} + (1 - x_{ij}) \frac{1}{1 - \pi(\theta_{ij})} + \frac{\partial (1 - \pi(\theta_{ij}))}{\partial a_{is}} \right]. \tag{3.6}$$

Usando el resultado obtenido en la ecuación (9) de la sección I.2, donde

$$\frac{\partial \pi(\theta_{ij})}{\partial a_{is}} = b_{js} \pi(\theta_{ij})(1 - \pi(\theta_{ij})), \tag{3.7}$$

y

$$\frac{\partial(1-\pi(\theta_{ij}))}{\partial a_{is}} = -b_{js}\pi(\theta_{ij})(1-\pi(\theta_{ij})), \tag{3.8}$$

entonces

$$\frac{\partial f(\theta_{ij})}{\partial a_{is}} = b_{js}(\pi(\theta_{ij}) - x_{ij}), \quad s = 1, \ldots, k. \tag{3.9}$$

De forma análoga se puede obtener las derivadas parciales con respecto a b_{js}, así:

$$\frac{\partial f(\theta_{ij})}{\partial b_{js}} = -\left[x_{ij}\frac{1}{\pi(\theta_{ij})}\frac{\partial \pi(\theta_{ij})}{\partial b_{js}} + (1-x_{ij})\frac{1}{1-\pi(\theta_{ij})}\frac{\partial(1-\pi(\theta_{ij}))}{\partial b_{js}}\right] \tag{3.10}$$

$$= a_{is}(\pi(\theta_{ij}) - x_{ij}), \quad s = 1, \ldots, k, \tag{3.11}$$

donde $\frac{\partial \pi(\theta_{ij})}{\partial b_{js}} = a_{is}\pi(\theta_{ij})(1-\pi(\theta_{ij}))$.

Para el término de desplazamiento del modelo, las derivadas parciales con respecto a μ_j son:

$$\frac{\partial f(\theta_{ij})}{\partial \mu_j} = -\left[x_{ij}\frac{1}{\pi(\theta_{ij})}\frac{\partial \pi(\theta_{ij})}{\partial \mu_j} + (1-x_{ij})\frac{1}{1-\pi(\theta_{ij})}\frac{\partial(1-\pi(\theta_{ij})}{\partial \mu_j}\right] \tag{3.12}$$

$$= (\pi_{ij} - x_{ij}). \tag{3.13}$$

Estas derivadas se pueden expresar de forma matricial de la siguiente manera:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = (\boldsymbol{\Pi} - \mathbf{X})^T \mathbf{1}_n \tag{3.14}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{A}} = (\boldsymbol{\Pi} - \mathbf{X})\,\mathbf{B} \tag{3.15}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{B}} = (\boldsymbol{\Pi} - \mathbf{X})^T \mathbf{A} \tag{3.16}$$

La inicialización del algoritmo comienza en el punto $\boldsymbol{\Theta}_0 = \mathbf{1}_n\boldsymbol{\mu}_0^T + \mathbf{A}_0\mathbf{B}_0^T$, que puede ser configurado por el usuario o mediante una inicialización aleatoria. Para una inicialización aleatoria, todos los elementos de $\mathbf{A}_0 = (\mathbf{a_1^0}, \ldots, \mathbf{a_n^0})^T$, $\mathbf{B}_0 = (\mathbf{b_1^0}, \ldots, \mathbf{b_k^0})$ y $\boldsymbol{\mu}_0$ se pueden generar a partir de una distribución uniforme. La primera dirección de búsqueda d_l^T es elegida como la dirección de descenso más empinada desde el punto inicial $\boldsymbol{\Theta}_0$; luego, se usa un método de búsqueda de línea para calcular el parámetro de longitud de paso α_l, y usando un escalar β_l, se define la regla para actualizar la dirección basada en el gradiente en cada iteración, para de esta forma actualizar simultáneamente $\boldsymbol{\mu}$, $\mathbf{A}$ y $\mathbf{B}$, y así estimar el

espacio natural de los parámetros $\boldsymbol{\Theta}$. El pseudocódigo se escribe formalmente en Algoritmo 4.

Algoritmo 4 Algoritmo basado en el Gradiente Conjugado para ajustar el modelo biplot logístico

Entrada $\mathbf{X}$
Salida $\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}$

Inicializar $\boldsymbol{\mu}_0, \mathbf{A}_0, \mathbf{B}_0$
$\boldsymbol{\Theta}_0 = \mathbf{1}_n \boldsymbol{\mu}_0^T + \mathbf{A}_0 \mathbf{B}_0^T$
$\nabla \mathcal{L}_0 = \nabla \mathcal{L}(\boldsymbol{\Theta}_0)$
$d_0 = -\nabla \mathcal{L}_0$
$l = 0$
repeat
    $\boldsymbol{\Pi}_l = \pi(\boldsymbol{\Theta}_l)$
    $\alpha_l = \arg\min_{\alpha > 0} \mathcal{L}(\boldsymbol{\Theta}_l + \alpha d_l)$
    $\mathbf{A}_{l+1} = \mathbf{A}_l + \alpha_l (\boldsymbol{\Pi}_l - \mathbf{X}) \mathbf{B}_l$
    $\mathbf{B}_{l+1} = \mathbf{B}_l + \alpha_l (\boldsymbol{\Pi}_l - \mathbf{X})^T \mathbf{A}_l$
    $\boldsymbol{\mu}_{l+1} = \boldsymbol{\mu}_l + \alpha_l (\boldsymbol{\Pi}_l - \mathbf{X})^T \mathbf{1}_n$
    $\boldsymbol{\Theta}_{l+1} = \mathbf{1}_n \boldsymbol{\mu}_{l+1}^T + \mathbf{A}_{l+1} \mathbf{B}_{l+1}^T$
    $\nabla \mathcal{L}_{l+1} = \pi(\boldsymbol{\Theta}_{l+1}) - \mathbf{X}$
    Calcular β_{l+1} de acuerdo con alguna de las formulas dadas en (3.17)
    $d_{l+1} = -\nabla \mathcal{L}_{l+1} + \beta_{l+1} d_l$
until $(\mathcal{L}(\boldsymbol{\Theta}_l) - \mathcal{L}(\boldsymbol{\Theta}_{l+1})) / \mathcal{L}(\boldsymbol{\Theta}_l) < \epsilon$

Como se ha señalado hasta ahora, este es un método que se considera de bajo costo computacional debido a que en cada iteración solo requiere la evaluación del gradiente y la función de pérdida, convirtiéndolo en un algoritmo eficiente para grandes conjuntos de datos (Nocedal y Wright, 2006; Pytlak, 2008). Para actualizar la dirección basada en el gradiente, se utilizan cuatro fórmulas: Fletcher–Reeves (FR) (Fletcher y Powell, 1963), Polak–Ribière–Polyak (PRP) (Polyak, 1969), Hestenes–Stiefel (HS) (Hestenes y Stiefel, 1952) y Dai–Yuan (DY) (Dai y Yuan, 1999), que para el modelo del biplot logístico se pueden escribir como

$$\beta_{l+1}^{FR} = \frac{\|\nabla \mathcal{L}_{l+1}\|^2}{\|\nabla \mathcal{L}_l\|^2}; \quad \beta_{l+1}^{PRP} = \frac{\nabla \mathcal{L}_{l+1}^T \Delta_l}{\|\nabla \mathcal{L}_l\|^2}; \quad \beta_{l+1}^{HS} = \frac{\nabla \mathcal{L}_{l+1}^T \Delta_l}{d_l^T \Delta_l}; \quad \beta_{l+1}^{DY} = \frac{\|\nabla \mathcal{L}_{l+1}\|^2}{d_l^T \Delta_l}, \tag{3.17}$$

donde $\Delta_l = \nabla \mathcal{L}_{l+1} - \nabla \mathcal{L}_l$, y $\|\cdot\|$ es la norma euclidiana. Las formulas para actualizar la dirección de búsqueda han sido muy investigadas en la literatura y en general corresponden a combinaciones de las cuatro formulas mencionadas previamente (Andrei, 2008; Dai y Yuan, 2001; Dong y col., 2015; Liu y Li, 2014; Yuan y Zhang, 2013; Yuan y col., 2019;

Zhang y col., 2006), de modo que cualquiera de estas podría utilizarse en el Algoritmo 4 para ajustar el modelo de LB.

Para lograr la convergencia en la implementación del algoritmo del gradiente conjugado, a menudo se requiere que la longitud de paso obtenida con una búsqueda de línea sea exacta o satisfaga las condiciones fuertes de Wolfe. Se ha demostrado que el método Fletcher-Reeves converge globalmente bajo una búsqueda en línea que cumpla las condiciones fuertes de Wolfe con $c_2 < 1/2$ (Al-Baali, 1985; Dai y Yuan, 1996), por lo que esta condición asegura que todas las direcciones d_l son direcciones descendentes desde $\mathcal{L}$.

3.3. Algoritmo de descenso coordinado por bloques

Los métodos de *descenso coordinado por bloques* (BCD), también conocidos como algoritmo no lineal de Gauss-Seidel, son un conjunto de algoritmos de optimización que realizan una actualización de parámetros por bloques de forma secuencial en cada iteración. Una forma de realizar la optimización consiste en utilizar métodos alternados en cada iteración para actualizar un bloque de parámetros mientras los demás permanecen fijos.

Entre las ventajas que se le atribuyen a este método se encuentran (Sun y col., 2016):

1. Cada subproblema puede ser más sencillo de resolver e incluso puede tener una solución exacta.
2. La función de pérdida decrece con cada iteración.
3. Es posible implementar procesos de computación en paralelo o distribuidos.

En la sección 2.4 se presentó la función sustituta para el biplot logístico, de modo que el espacio de parámetros $\boldsymbol{\Theta}$ puede ser divido en bloques y sobre la función sustituta aplicar el algoritmo de descenso coordinado por bloques. Como se señaló en el capítulo anterior, esto se conoce como un método MM, que tiene un primer paso *"M"* que consiste en mayorizar la función de pérdida y luego se realiza un segundo paso *"M"* donde se aplica un algoritmo que permite minimizar la función mayorizada.

De acuerdo con el Teorema 2 que fue demostrado en la sección 2.4, la función de pérdida

de un modelo LB, $\mathcal{L}(\boldsymbol{\Theta})$, puede ser mayorizada por

$$\mathcal{G}\left(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(l)}\right) = \frac{1}{8}\left\|\boldsymbol{\Theta} - \mathbf{Z}_l\right\|_F^2, \tag{3.18}$$

donde $\mathbf{Z}_l = \boldsymbol{\Theta}^{(l)} + 4\left(\mathbf{X} - \boldsymbol{\Pi}_l\right)$, con $\boldsymbol{\Pi}_l = \pi(\boldsymbol{\Theta}^{(l)})$. Reemplazando el espacio de parámetros por $\boldsymbol{\Theta} = \mathbf{1}_n\boldsymbol{\mu}^T + \mathbf{A}\mathbf{B}^T$, el problema de minimización se puede escribir como

$$\underset{\boldsymbol{\mu},\mathbf{A},\mathbf{B}}{\text{mín}} \left\|\mathbf{1}_n\boldsymbol{\mu}^T + \mathbf{A}\mathbf{B}^T - \mathbf{Z}_l\right\|_F^2. \tag{3.19}$$

Los parámetros del modelo LB pueden ser estimados de forma secuencial mediante el algoritmo BCD. Teniendo en cuenta que $\theta_{ij} = \mu_j + \mathbf{a}_i^T\mathbf{b}_j$ entonces el problema de minimización para la l-ésima iteración se puede escribir como

$$\mathcal{G}\left(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(l)}\right) = \frac{1}{8}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\mu_j + \mathbf{a}_i^T\mathbf{b}_j - z_{ij}^{(l)}\right)^2. \tag{3.20}$$

Al fijar $\mathbf{A}$ y $\mathbf{B}$ se define $z_{ij}^{*(l)} = z_{ij}^{(l)} - \mathbf{a}_i^T\mathbf{b}_j$, de modo que el estimador de μ_j se obtiene como

$$\hat{\mu}_j = \arg\underset{\mu_j}{\text{mín}} \sum_{i=1}^{n}\left(\mu_j - z_{ij}^{*(l)}\right)^2 \tag{3.21}$$

$$= \arg\underset{\mu_j}{\text{mín}} \sum_{i=1}^{n}\left(z_{ij}^{*(l)} - \mu_j\right)^2 \tag{3.22}$$

Que corresponde a un problema de mínimos cuadrados, así que:

$$\hat{\mu}_j = \frac{1}{n}\sum_{i=1}^{n} z_{ij}^{*(l)} \tag{3.23}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(z_{ij}^{(l)} - \mathbf{a}_i^T\mathbf{b}_j\right) \tag{3.24}$$

Al expresarlo en términos matriciales, la solución para el vector de desplazamiento en la iteración l es

$$\boldsymbol{\mu} = \frac{1}{n}\left(\mathbf{Z}_l - \mathbf{A}\mathbf{B}^T\right)^T\mathbf{1}_n. \tag{3.25}$$

En este punto se requiere una restricción adicional para tener una solución única al problema de encontrar un subespacio afín para los datos. Entonces, para lograr que el modelo sea identificable se considera que $\mathbf{1}_n^T\mathbf{A}\mathbf{B}^T = 0$. De esta manera $\boldsymbol{\mu}$ se puede estimar con el vector

de medias de cada columna de la matriz $\mathbf{Z}_l$,

$$\boldsymbol{\mu} = \frac{1}{n}\mathbf{Z}_l^T\mathbf{1}_n. \tag{3.26}$$

Para actualizar en la l-ésima iteración los bloques de parámetros definidos por $\mathbf{A}$ y $\mathbf{B}$ después de haber estimado $\boldsymbol{\mu}$, se puede definir $z_{ij,(c)}^{(l)} = z_{ij}^{(l)} - \mu_j$, que en forma matricial es

$$\begin{aligned}
\mathbf{Z}_l^c &= \mathbf{Z}_l - \mathbf{1}_n\boldsymbol{\mu}^T && \text{(3.27a)}\\
&= \mathbf{Z}_l - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\mathbf{Z}_l && \text{(3.27b)}\\
&= \mathbf{Z}_l\left(\mathbf{I} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right) && \text{(3.27c)}\\
&= \mathbf{P}\mathbf{Z}_l && \text{(3.27d)}
\end{aligned}$$

De esta manera, el problema de optimización se reduce a

$$\min_{\mathbf{A},\mathbf{B}} \left\|\mathbf{A}\mathbf{B}^T - \mathbf{P}\mathbf{Z}_l\right\|_F^2, \tag{3.28}$$

donde $\mathbf{P} = \mathbf{I} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$ es la matriz que centra las columnas. De modo que la mejor aproximación de $\mathbf{A}$ y $\mathbf{B}$ en la l-ésima iteración se obtiene de $\mathbf{Z}_l^c = \mathbf{U}\Lambda\mathbf{V}^T$, donde se tomará $\mathbf{A} = \mathbf{U}\Lambda$ y $\mathbf{B} = \mathbf{V}^T$. Nótese que por defecto se obtiene que $\mathbf{B}^T\mathbf{B} = \mathbf{I}$.

Para inicializar el algoritmo basado en el método MM, los elementos de $\boldsymbol{\Theta}_0 = \mathbf{1}_n\boldsymbol{\mu}_0^T + \mathbf{A}_0\mathbf{B}_0^T$, pueden ser tomados de una distribución uniforme. El pseudocódigo se describe en el Algoritmo 5.

3.4. Medidas de desempeño para la evaluación del modelo

Para evaluar el rendimiento de los algoritmos basados en el grandiente conjugado y el algoritmo basado en el método MM, se utiliza el error de entrenamiento presentado en la sección 2.5, que se define como el promedio de la tasa de clasificación errónea cuando se usa la estructura con k dimensiones generada por el modelo LB con el conjunto de entrenamiento.

Algoritmo 5 Algoritmo de descenso coordinado por bloques para ajustar el modelo biplot logístico

Entrada $\mathbf{X}$
Salida $\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}$

Inicializar $\boldsymbol{\mu}_0, \mathbf{A}_0, \mathbf{B}_0$
$\boldsymbol{\Theta}_0 = \mathbf{1}_n \boldsymbol{\mu}_0^T + \mathbf{A}_0 \mathbf{B}_0^T$
$l = 0$
repeat
    $\boldsymbol{\Pi}_l = \pi(\boldsymbol{\Theta}_l)$
    $\mathbf{Z}_l = \boldsymbol{\Theta}_l + 4(\mathbf{X} - \boldsymbol{\Pi}_l)$
    $\boldsymbol{\mu}_{(l+1)} = \frac{1}{n} \mathbf{Z}_l \mathbf{1}_n$
    $\mathbf{Z}_{l+1}^c = \mathbf{P}\mathbf{Z}_l$, con $\mathbf{P} = \mathbf{I} - \frac{1}{n}\mathbf{1}_n \mathbf{1}_n^T$
    $\mathbf{Z}_{l+1}^c = \mathbf{U}\Lambda\mathbf{V}^T$
    $\mathbf{A}_{l+1} = \mathbf{U}\Lambda$
    $\mathbf{B}_{l+1} = \mathbf{V}$
    $\boldsymbol{\Theta}_{l+1} = \mathbf{1}_n \boldsymbol{\mu}_{l+1}^T + \mathbf{A}_{l+1}\mathbf{B}_{l+1}^T$
until $(\mathcal{L}(\boldsymbol{\Theta}_l) - \mathcal{L}(\boldsymbol{\Theta}_{l+1}))/\mathcal{L}(\boldsymbol{\Theta}_l) < \epsilon$

Considerando que la tasa de error equilibrada, TEE, calculada sobre todo el conjunto de datos de entrenamiento puede llegar a ser muy optimista al momento de calcular la tasa de clasificación errónea, también se tendrá en cuenta el procedimiento de validación cruzada propuesto para un modelo LB que fue presentado en la sección 2.6, donde se usa un conjunto de datos de prueba que es independiente del conjunto de datos de entrenamiento.

Debido a que el modelo LB es un método multivariante no supervisado, el procedimiento de validación cruzada permite encontrar el número de dimensiones que se deben retener en el modelo (Bro y col., 2008; Fu y Perry, 2020; Gabriel, 2002; Owen, Perry y col., 2009; Wold, 1978). Por lo tanto, constituye un valor estimado del hiperparámetro k que permite evaluar la capacidad de los diferentes algoritmos de estimación para identificar el espacio de baja dimensión.

También se mide la capacidad que tienen los modelos basados en cada algoritmo para recuperar la matriz canónica de parámetros dada por el log-odds, $\boldsymbol{\Theta}$; para esto, se usa el error cuadrático medio relativo (RMSE), definido como

$$RMSE(\boldsymbol{\Theta}) = \frac{\left\|\boldsymbol{\Theta} - \hat{\boldsymbol{\Theta}}\right\|_F^2}{\|\boldsymbol{\Theta}\|_F^2}. \tag{3.29}$$

donde $\boldsymbol{\Theta}$ la verdadera matriz de parámetros y $\hat{\boldsymbol{\Theta}}$ es la matriz estimada por el modelo LB usando alguno de los algoritmos propuestos.

3.5. Proceso de simulación de la matriz de datos

Los conjuntos de datos se simularon a partir de un modelo de variables latentes con diferentes niveles de dispersión y con una estructura de baja dimensión según el modelo LB, $\mathbf{\Theta} = \mathbf{1}_n\boldsymbol{\mu}^T + \mathbf{A}\mathbf{B}^T$.

Cada columna de la matriz de marcadores columna $\mathbf{B} = (\mathbf{b_1}, \ldots, \mathbf{b_k})$ es obtenida desde una distribución normal estándar. Posteriormente la matriz $\mathbf{B}$ es sometida a un proceso de ortogonalización de Gram-Schmidt para asegurar $\mathbf{B}^T\mathbf{B} = \mathbf{I}_k$. La matriz de marcadores fila $\mathbf{A}$ se obtiene de una distribución normal multivariante con vector de medias cero, y matriz de varianzas y covarianzas igual a la matriz identidad. El término de compensación o desplazamiento $\boldsymbol{\mu}$ se utiliza como control del nivel de dispersión en la matriz, determinado por una constante $0 < D < 1$. De esta forma el log-odds $\mathbf{\Theta}$ permite tener la estructura en un subespacio de dimensión $k < p$.

Una vez simulado $\mathbf{\Theta}$ se puede calcular la matriz de probabilidades usando la función logística, $\mathbf{\Pi} = \pi(\mathbf{\Theta})$, con lo cual la matriz $\mathbf{X}$ se puede obtener como realizaciones de variables aleatorias x_{ij} con distribución Bernoulli con parámetro $\pi(\theta_{ij})$, que es el ij-ésimo elemento de la matriz $\mathbf{\Pi}$. El Algoritmo 6 presenta el pseudocódigo para generar la matriz binaria $\mathbf{X}$.

Algoritmo 6 Algoritmo para simular matrices de datos binarias

Entrada n, p, k, D.
Salida $\mathbf{X}, \mathbf{\Theta}, \boldsymbol{\mu}, \mathbf{A}, \mathbf{B}$

1: $\mathbf{B} = (\mathbf{b_1}, \ldots, \mathbf{b_k})$ con $\mathbf{b_j} \sim \mathcal{N}(0, 1), j = 1, \ldots, k$.
2: **procedure** Ortogonalización de Gram–Schmidt
3: $\quad \mathbf{B}^T\mathbf{B} = \mathbf{I}_k$
4: **end procedure**
5: $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)^T$ con $\mu_i = ln\left(\dfrac{D}{1-D}\right)$
6: $\mathbf{A} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_k)$
7: $\mathbf{\Theta} = \mathbf{1}_n\boldsymbol{\mu}^T + \mathbf{A}\mathbf{B}^T$
8: $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$, con $x_{ij} \sim Ber(\pi(\theta_{ij})), \pi(\theta_{ij}) = \{1 + exp(-\theta_{ij})\}^{-1}$.

3.6. Estudio de Monte Carlo

Se simularon matrices binarias con $n = 100, 300, 500$; $p = 50, 100$; $k = 3$; y $D = 0.5, 0.3, 0.2, 0.1$, donde el parámetro D representa la proporción de unos en la matriz $\mathbf{X}$. El propósito de generar diferentes niveles de dispersión en la matriz de datos se hace para verificar si esto afecta el rendimiento de los algoritmos para encontrar la estructura de baja dimensión del modelo LB. Las combinaciones de n, p, k y D generan los diferentes escenarios; en cada escenario se simulan R matrices de forma independiente y se calculan las medidas basadas en el error de entrenamiento, el error de generalización determinado por la validación cruzada y RMSE para evaluar el desempeño de los algoritmos. A continuación se enuncian los pasos del proceso de Monte Carlo:

1. Usando el algoritmo 6, generar R matrices binarias $\mathbf{X}_1, \ldots, \mathbf{X}_R$ de n filas y p columnas, con desequilibrio D y estructura de baja dimensión $k = 3$.

2. Conservar las matrices de los marcadores fila $\mathbf{A}_1, \ldots, \mathbf{A}_R$, así como las matrices de marcadores columna $\mathbf{B}_1, \ldots, \mathbf{B}_R$, el espacio de parámetros $\mathbf{\Theta}_1, \ldots, \mathbf{\Theta}_R$ y la matriz de probabilidades $\mathbf{\Pi}_1, \ldots, \mathbf{\Pi}_R$.

3. Hacer $k = 1$.

4. Para cada matriz $\mathbf{X}_i, i = 1, \ldots, R$, ajustar el modelo LB usando los algoritmos basados en el gradiente conjugado: FR, PRP, HS y DY, y el algoritmo del descenso coordinado por bloques (denominado acá MM).

5. Cada algoritmo proporciona una matriz de marcadores fila y de marcadores columna que permiten estimar el espacio de parámetros: $\left\{\hat{\mathbf{\Theta}}_i^{FR}, \hat{\mathbf{\Theta}}_i^{PRP}, \hat{\mathbf{\Theta}}_i^{HS}, \hat{\mathbf{\Theta}}_i^{DY}, \hat{\mathbf{\Theta}}_i^{MM}\right\}$, para $i = 1 \ldots, R$.

6. Calular la habilidad que cada algoritmo tiene para recuperar el log-odds, $\mathbf{\Theta}$, usando el error cuadrático medio relativo de la estimación del espacio de parámetros, así:

$$RMSE(\mathbf{\Theta}_i) = \frac{\left\|\mathbf{\Theta}_i - \hat{\mathbf{\Theta}}_i\right\|_F^2}{\|\mathbf{\Theta}_i\|_F^2}, i = 1, \ldots, R. \tag{3.30}$$

7. Calcular las matrices predichas con cada algoritmo: $\left\{\hat{\mathbf{\Pi}}_i^{FR}, \hat{\mathbf{\Pi}}_i^{PRP}, \hat{\mathbf{\Pi}}_i^{HS}, \hat{\mathbf{\Pi}}_i^{DY}, \hat{\mathbf{\Pi}}_i^{MM}\right\}$, para $i = 1 \ldots, R$.

8. Calcular el umbral, δ_j, que minimiza la tasa de error equilibrada para la variable j, para $j = 1 \ldots, p$ en cada matriz $\mathbf{X}_i, i = 1, \ldots, R$.

9. Estimar las matrices de datos con el modelo generado por cada algoritmo $\left\{\hat{\mathbf{X}}_i^{FR}, \hat{\mathbf{X}}_i^{PRP}, \hat{\mathbf{X}}_i^{HS}, \hat{\mathbf{X}}_i^{DY}, \hat{\mathbf{X}}_i^{MM}\right\}$, para $i = 1 \ldots, R$.

10. Para cada matriz $\hat{\mathbf{X}}_i^a$ con $i = 1 \ldots, R$ y $a = \{FR, PRP, HS, DY, MM\}$, aplicar el algoritmo 3, de validación cruzada y obtener el error de entrenamiento y el error de generalización para cada matriz con cada algoritmo.

11. Calcular el error de validación cruzada *(cv error)* y el error cuadrático medio relativo *(RMSE)*, como la media de los errores de generalización y la desviación estándar de la estimación usando un estimador bootstrap, así

$$cv\_error^a = \frac{1}{R}\sum_{i=1}^{R} EG_i^a \tag{3.31}$$

$$RMSE(\mathbf{\Theta}^a) = \frac{1}{R}\sum_{i=1}^{R} RMSE(\mathbf{\Theta}_i^a) \tag{3.32}$$

$$ee^a = \sqrt{\frac{1}{R}\sum_{i=1}^{R}(EG_i^a - cv\_error^a)^2} \tag{3.33}$$

12. Si el valor de k es menor que 5 entonces hacer $k = k + 1$ y volver al paso 4.

La ilustración del proceso se presenta en la Figura 3.2. En el paso 10, la medida del error es calculada sobre todos los valores de la matriz predicha y también se hace sobre los valores faltantes, en el primer caso se denomina *error de entrenamiento* y en el segundo caso se denomina *error de validación* o *cv_error*. Se usó un valor de $R = 30$ que generó errores estándar menores a 1 % en la estimación de las dos métricas del error.

Para la simulación de Monte Carlo se escribió un código en R y se utilizó un enfoque de programación funcional y un procesamiento paralelo. El código computacional se encuentra en el Anexo A y en el repositorio de `GitHub` jgbabativam.

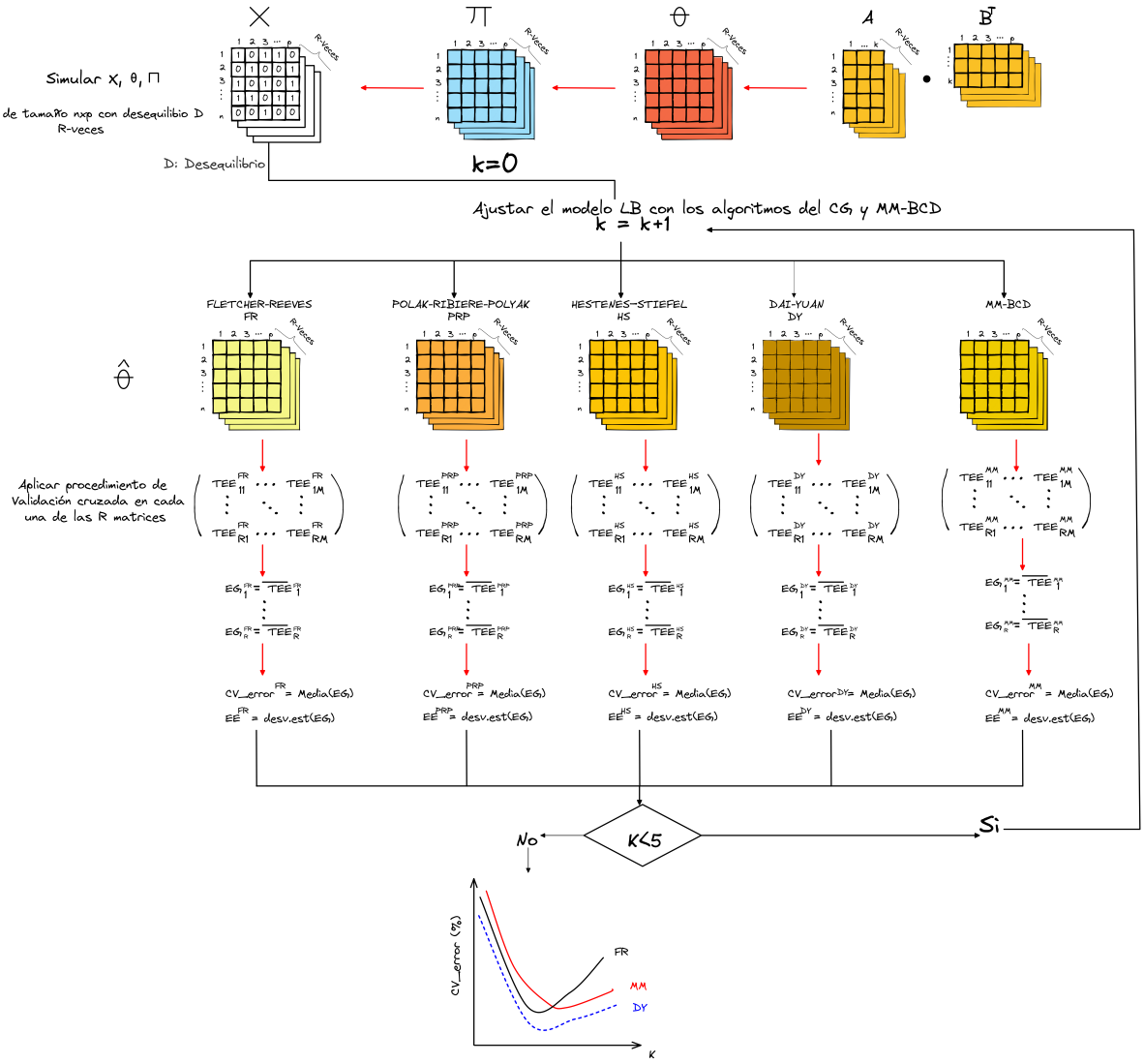


Figura 3.2: Proceso de Monte Carlo para comparar el rendimiento de los algoritmos.

3.6.1. Resultados para matrices balanceadas

La Figura 3.3 presenta el error de validación cruzada de los algoritmos basados en el gradiente conjugado y el algoritmo de descenso coordinado por bloques usando la función mayorizada (MM) cuando la matriz $\mathbf{X}$ está balanceada. Se destaca que en todos los tamaños de muestra estudiados ($n = 100, 300, 500$) y cantidad de columnas ($p = 50, 100$), los cinco algoritmos permiten llegar a una selección adecuada del modelo LB debido a que el error se minimiza en $k = 3$, que corresponde a la dimensionalidad real. Para $k > 3$, los cinco

métodos de estimación comenzaron a presentar sobreajuste.

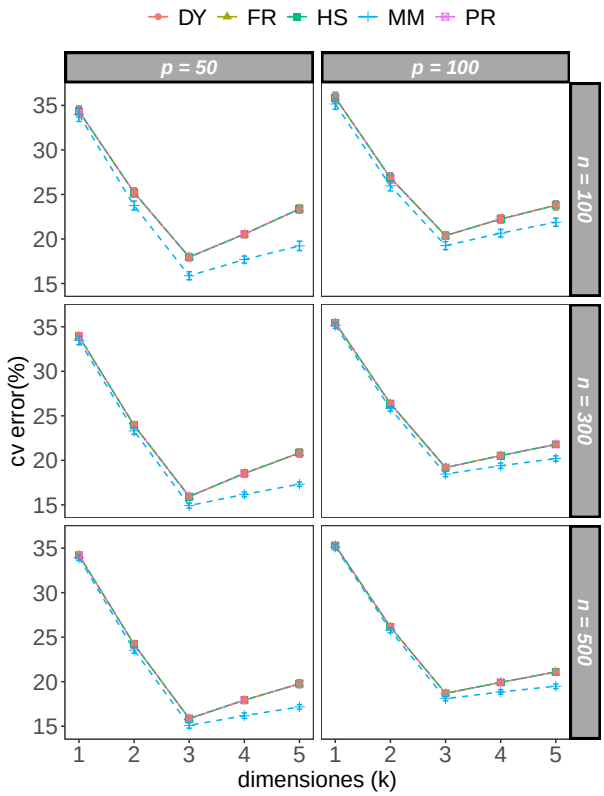
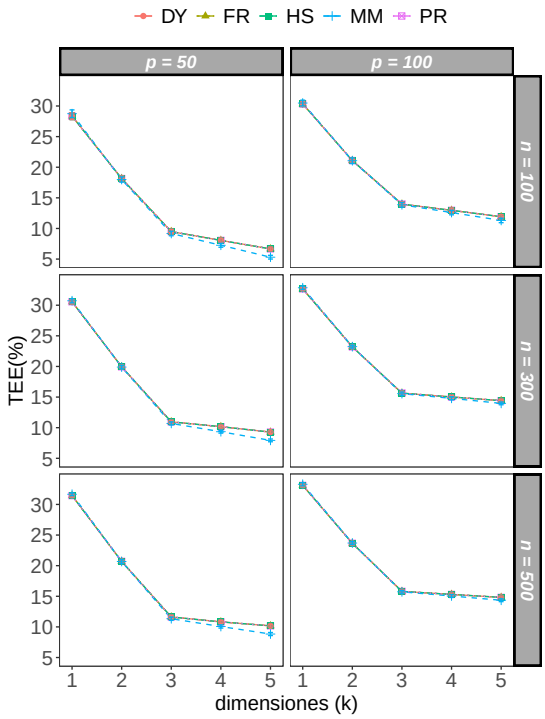


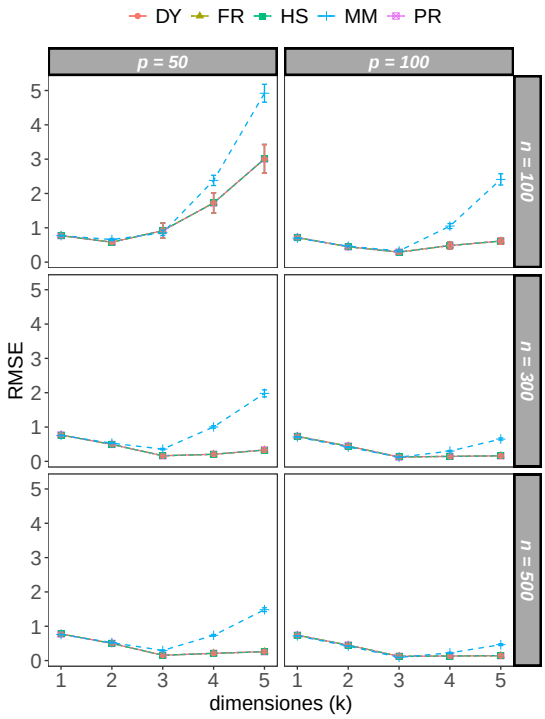
Figura 3.3: Error de validación cruzada con datos balanceados.

Como se esperaba, el error de entrenamiento tiene un comportamiento descendente cuando el valor de k se incrementa. La Figura 3.4a presenta la tasa de error equilibrada, TEE, cuando se tiene en cuenta toda la matriz $\mathbf{X}$ en el cálculo del error. Como se puede observar, la pendiente se desacelera cuando se alcanzan las tres dimensiones subyacentes en la matriz, así que un criterio de codo podría ser utilizado como señal del número de dimensiones a elegir.

En la Figura 3.4b se presenta la estimación del error cuadrático medio relativo (RMSE) para la matriz $\mathbf{\Theta}$; vemos que los algoritmos basados en el gradiente conjugado y el algoritmo MM muestran resultados similares cuando el número de dimensiones elegidas para el modelo LB es menor o igual a 3. Mientras que, cuando el modelo tiene más de las tres dimensiones, los algoritmos basados en el gradiente conjugado presentan un RMSE más bajo que el algoritmo MM, aunque, para un valor fijo de p, estas brechas se cierran a medida que n se aumenta.



(a) Error de entrenamiento.

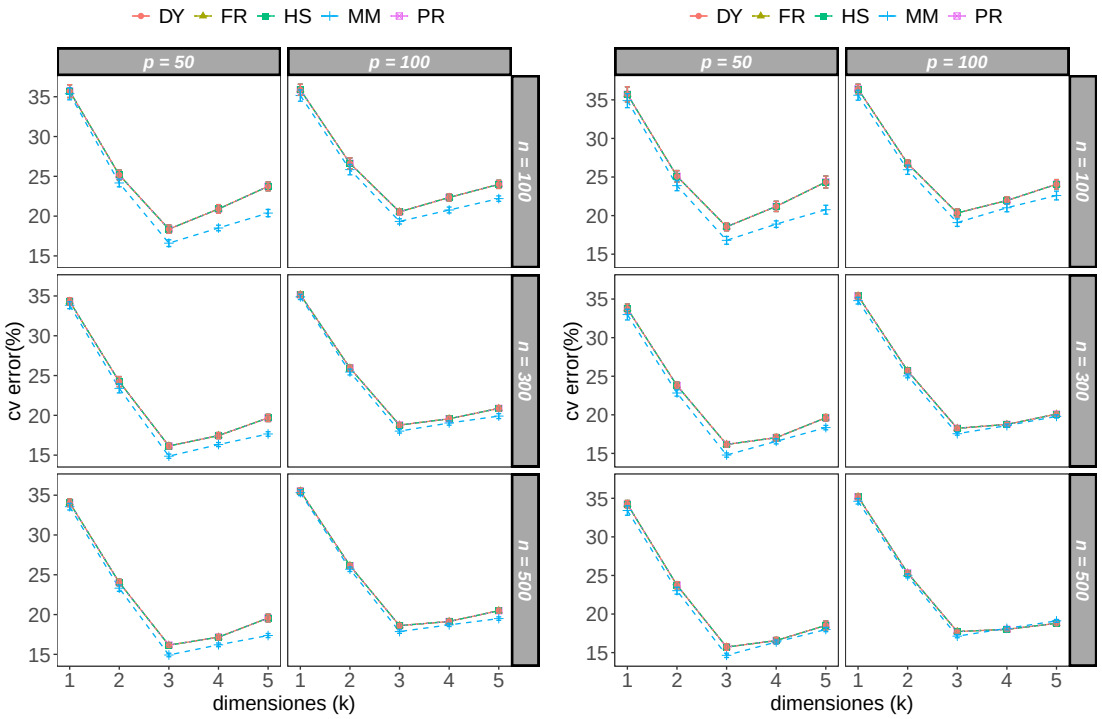


(b) Error cuadrático medio relativo

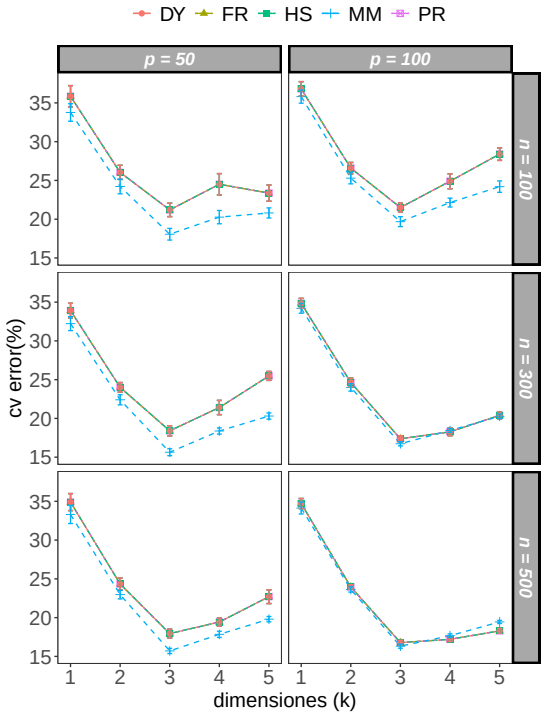
Figura 3.4: Error de entrenamiento y RMSE con datos balanceados.

3.6.2. Resultados para matrices desbalanceadas

En el proceso de simulación el grado de desequilibrio se controla con el parámetro D, que representa la proporción de unos en la matriz. La Figura 3.5a–c muestra los errores de validación cruzada cuando los datos están desequilibrados con $D = 0.3, 0.2$ y 0.1, respectivamente. En todos los escenarios estudiados se observa que el error se minimiza cuando se alcanzó el número de dimensiones subyacentes en el espacio de las variables. De modo que el grado de desequilibrio no afecta el rendimiento de los algoritmos en términos de encontrar correctamente el número de dimensiones.

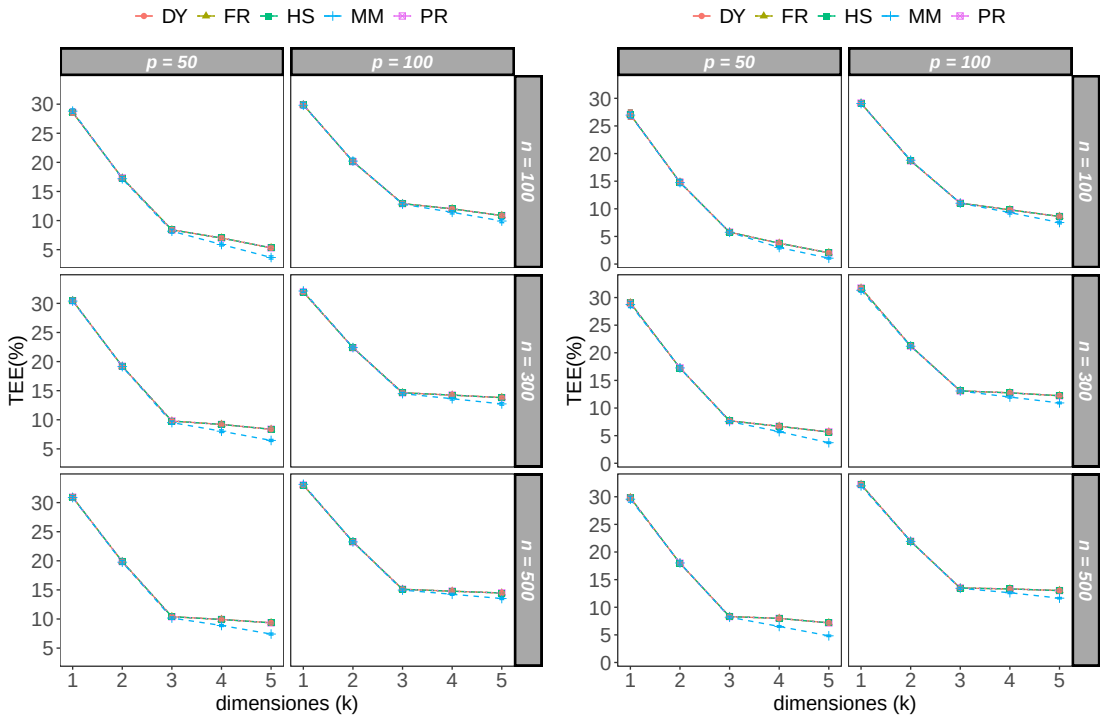


(a) $D = 0.3$ (b) $D = 0.2$



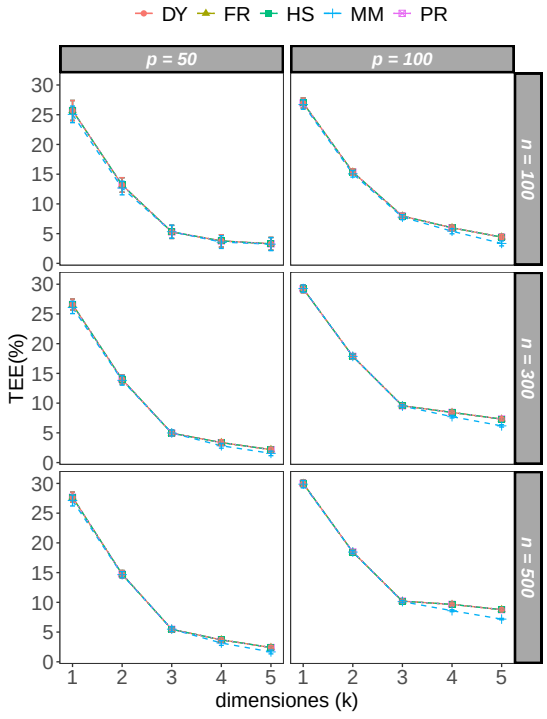
(c) $D = 0.1$

Figura 3.5: Error de validación cruzada para conjuntos de datos desequilibrados



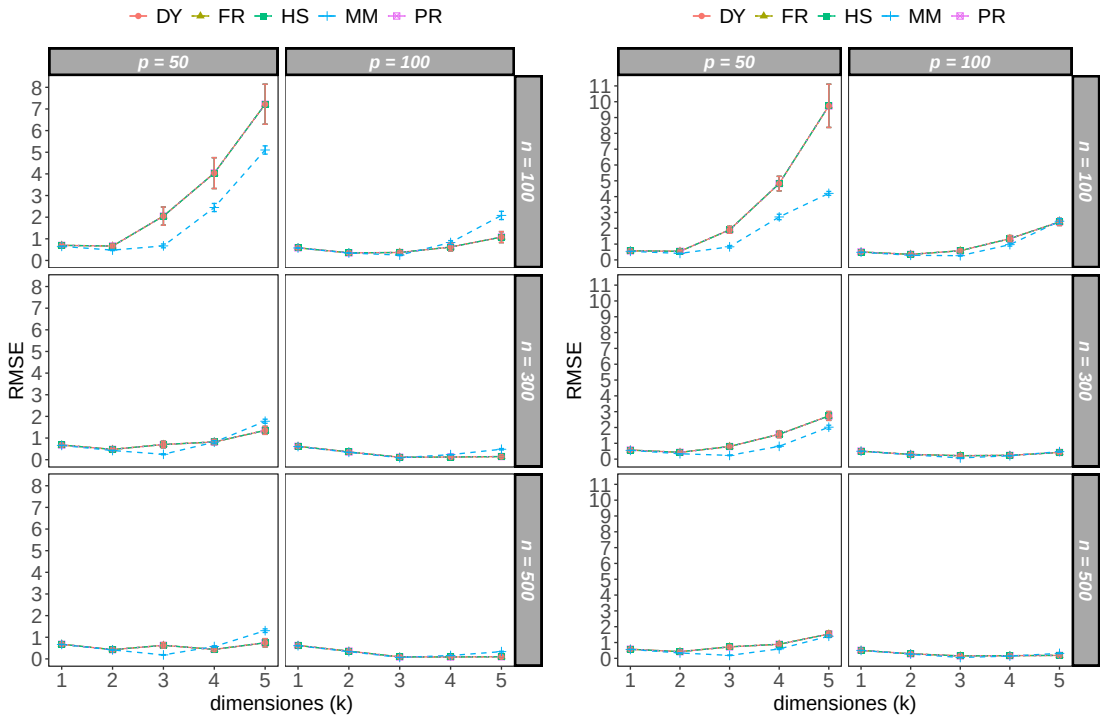
(a) $D = 0.3$

(b) $D = 0.2$



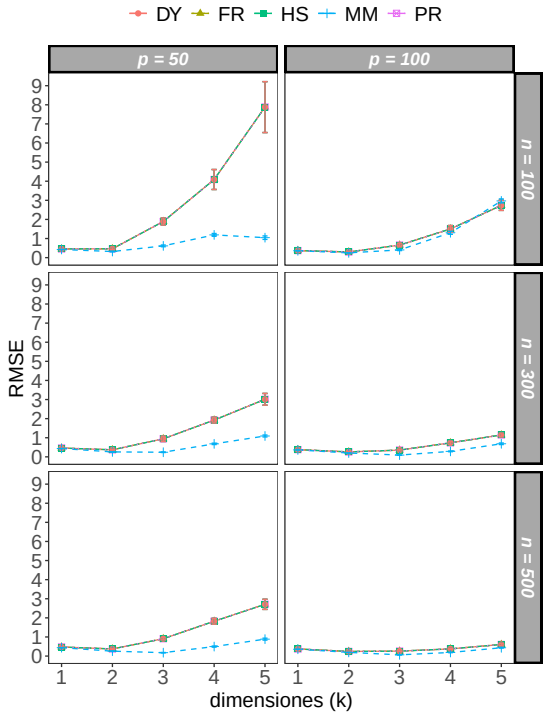
(c) $D = 0.1$

Figura 3.6: Error de entrenamiento para conjuntos de datos desequilibrados.



(a) $D = 0.3$

(b) $D = 0.2$



(c) $D = 0.1$

Figura 3.7: Estimación del error cuadrático medio de $\boldsymbol{\Theta}$ para conjuntos de datos desbalanceados.

El error de entrenamiento para conjuntos de datos desequilibrados con diferentes niveles de dispersión se presenta en la Figura 3.6a–c. En todos los escenarios estudiados se observa que el porcentaje de pérdida en el error de entrenamiento se estabiliza a partir de la tercera dimensión. De esta forma, los diferentes algoritmos permiten seleccionar adecuadamente el número de dimensiones utilizando el método del codo y no se ven afectados por el grado de desequilibrio de la matriz.

La Figura 3.7a–c presenta la habilidad que tienen los diferentes algoritmos para recuperar la matriz de parámetros. El RMSE de la estimación del log-odds, $\mathbf{\Theta}$, para los diferentes niveles de dispersión permite observar que los algoritmos presentan rendimientos similares. Para los escenarios con $p = 50$, se observa que el RMSE se aumenta de forma importante cuando el número de dimensiones elegido para el modelo es mayor que 3, en especial cuando el número de observaciones (n) tiende a disminuir y se tiene un alto grado de dispersión ($D = 0.1$), originando unas brechas importantes en el rendimiento de los algoritmos basados en el gradiente conjugado y el método MM con el algoritmo de descenso coordinado por bloques, aunque estas diferencias disminuyen cuando el valor de p o el valor de n se aumenta. En conclusión, para mantener un control del error cuadrático medio, resulta de gran relevancia una apropiada elección del hiperparámetro k en el modelo LB, donde la metodología basada en la validación cruzada propuesta en este trabajo mostró ser exitosa.

3.6.3. Desempeño computacional

El rendimiento computacional de los algoritmos se midió en una computadora con un procesador Intel Core i7-3517U con 6 GB de RAM. La Tabla 3.1 muestra el tiempo de ejecución en segundos para 100 réplicas con $k = 3$ y un criterio de parada de $\epsilon = 10^{-4}$. Los cinco algoritmos presentan tiempos de ejecución competitivos y convergen relativamente rápido cuando el máximo de los cambios absolutos de los parámetros estimados en dos iteraciones consecutivas fue inferior a 10^{-4}.

Los tiempos de CPU de los algoritmos basados en el gradiente conjugado, en general, tuvieron tiempos similares y presentaron un mejor desempeño que el algoritmo MM cuando $p = 50$ y el nivel de dispersión comienza a ser alto ($D \leq 0.2$), o cuando $n = 100$, $p = 100$ y $D = 0.1$. En los demás casos, el algoritmo basado en el método MM de descenso coordinado

por bloques obtenido desde la función sustituta funcionó mejor, especialmente cuando n y p aumentaron, lo que resultó en un rendimiento que fue hasta seis veces más rápido que los algoritmos basados en el gradiente conjugado en escenarios equilibrados.

Tabla 3.1: Tiempo de ejecución en segundos para ajustar el modelo LB con $k = 3$, $\epsilon = 10^{-4}$ y 100 repeticiones.

n	p	D	DY	FR	HS	PR	MM
100	50	0.1	29.9	30.1	29.9	29.9	166.1
300	50	0.1	84.6	85.4	85.0	85.3	276.3
500	50	0.1	141.2	141.5	141.9	141.1	356.3
100	100	0.1	58.2	57.8	57.9	58.6	115.1
300	100	0.1	168.6	167.4	168.9	169.4	155.3
500	100	0.1	302.4	279.6	330.8	322.3	215.5
100	50	0.2	30.1	30.1	30.3	30.1	60.8
300	50	0.2	102.8	103.1	102.5	100.8	133.3
500	50	0.2	141.7	159.3	142.4	141.9	140.1
100	100	0.2	62.4	58.4	63.6	58.6	35.7
300	100	0.2	169.8	169.0	168.4	168.9	67.3
500	100	0.2	283.0	288.0	281.9	283.1	99.5
100	50	0.3	30.1	30.6	30.2	30.4	36.6
300	50	0.3	86.2	86.6	86.2	86.2	60.1
500	50	0.3	143.1	143.2	143.0	143.9	99.3
100	100	0.3	58.7	59.0	58.8	58.8	26.8
300	100	0.3	170.5	170.5	170.3	169.7	50.1
500	100	0.3	284.2	284.5	284.4	284.6	77.3
100	50	0.5	30.2	31.1	32.6	30.9	27.6
300	50	0.5	87.1	98.7	87.5	86.9	55.8
500	50	0.5	145.6	144.6	145.4	144.8	103.0
100	100	0.5	58.7	58.9	59.2	59.2	18.8
300	100	0.5	170.8	171.2	171.0	171.1	30.2
500	100	0.5	286.4	327.9	290.6	341.6	56.4

3.7. Aplicación

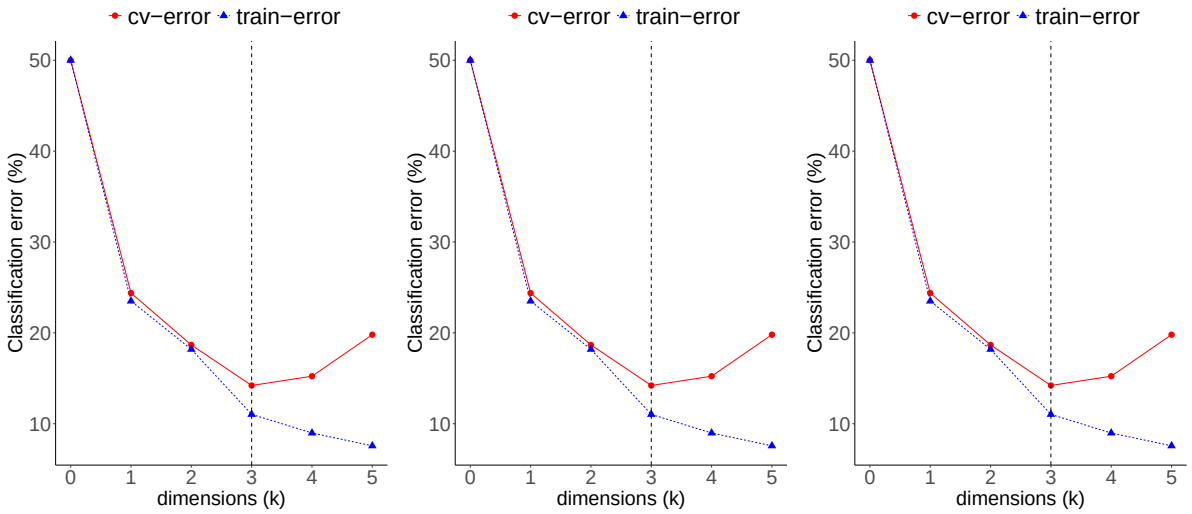
Para aplicar la metodología propuesta, se utilizan los datos del Genomic Determinants of Sensitivity in Cancer 1000 (GDSC1000) de la investigación de Iorio y col. (2016), de donde se pueden extraer diferentes tipos de información sobre líneas celulares de cáncer provenientes de más de 11 mil tumores para 30 tipos de cáncer que integran mutaciones somáticas, copia del número de alteraciones (CNA), metilaciones del ADN y cambios de expresión de genética. Las primeras tres son obtenidas como datos binarios mientras que

la expresión genética está medida con variables cuantitativas que son continuas.

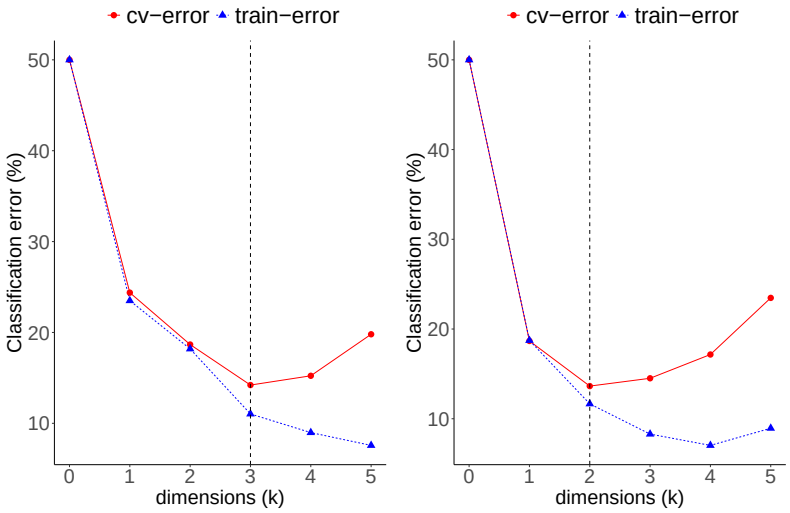
Para ilustrar los métodos, se utilizan los datos de metilación, y para facilitar la interpretación de los resultados, se incluyeron tres tipos de cáncer: carcinoma invasivo de mama (BRCA), adenocarcinoma de pulmón (LUAD) y melanoma cutáneo de piel (SKCM). Los conjuntos de datos publicados en la página de GDSC1000 no se encuentran estructurados en dos modos, filas y columnas, de manera que fue necesario realizar un preprocesamiento para ordenar los conjuntos de datos y separar la información de metilación en una sola matriz de información.

Luego del procesamiento previo, el conjunto de datos de metilación cuenta con 160 filas y 38 variables, cada variable es una isla CpG ubicada en la región promotora de genes. En este caso, el código 1 indica un alto nivel de metilación y el 0 indica un nivel bajo; aproximadamente 27 % de la matriz de datos binarios son unos.

La Figura 3.8 muestra el error de validación cruzada y el error de entrenamiento usando los algoritmos de gradiente conjugado y el algoritmo de descenso coordinado por bloques basado en el método MM (MM-BCD). Si $k = 0$, el modelo (4) solo consideró el término $\boldsymbol{\mu}$, lo que significa que $\boldsymbol{\Theta} = \mathbf{1}_n \boldsymbol{\mu}^T$, donde μ_j es la proporción de unos en la columna j y se usó como referencia para observar el rendimiento de los algoritmos cuando se incluyen más dimensiones al incorporar los marcadores de fila y columna, $\boldsymbol{\Theta} = \mathbf{1}_n \boldsymbol{\mu}^T + \mathbf{A}\mathbf{B}^T$.



(a) Fletcher–Reeves (b) Hestenes–Stiefel (c) Polak–Ribière–Polyak



(d) Dai–Yuan (e) MM-BCD

Figura 3.8: Validación cruzada para los algoritmos basados en el gradiente conjugado y de descenso coordinado por bloques (MM-BCD) para los datos de metilación.

La Figura 3.9 presenta el biplot obtenido para los datos de metilación usando el algoritmo de gradiente conjugado con la formula de actulización de Fletcher–Reeves; las variables están representadas por vectores dirigidos (segmentos) que comienzan en el punto que predice una probabilidad de 0.5 y terminan en el punto que predice una probabilidad de 0.75. Por lo tanto, los vectores cortos indican un aumento rápido de la probabilidad y la proyección ortogonal de los marcadores fila sobre el vector, aproximan la probabilidad de encontrar altos niveles de metilación en la línea celular.

La interpretación es la misma que se establece en Vicente-Villardon y col. (2006). De modo que la posición del segmento, que corresponde al punto que predice una probabilidad de 0.5, puede comenzar en cualquier lado alrededor del origen. Por ejemplo, en la Figura 3.9, la variable *DUSP22* apunta hacía el origen, al hacer la proyección ortogonal de los puntos en la dirección del vector, la mayoría de ellos quedan proyectados después del punto de referencia donde se inicia el segmento; esto significa que casi todas las líneas celulares de los tres grupos tienen altas probabilidades ajustadas de tener altos niveles de metilación en esa variable.

El algoritmo de estimación permite separar las líneas celulares en tres grupos claramente identificados. En el tipo de cáncer *BRCA*, variables como *NAPRT1*, *THY1* o *ADCY4* se dirigen hacia la parte positiva de la dimensión 1 y por tanto tienen mayor probabilidad de presentar niveles elevados de metilación. Las líneas celulares *LUAD* se ubican en la parte negativa de la dimensión 2, por lo que tienen una alta propensión a presentar altos niveles de metilación en variables como *HIST1H2BH*, *ZNF382* y *XKR6*. Finalmente, las líneas celulares para el tipo de cáncer *SKCM* se ubican en la parte negativa de la dimensión 1 y tienen mayor probabilidad de presentar altos niveles de metilación en variables como *LOC100130522*, *CHRFAM7A* o *DHRS4L2*.

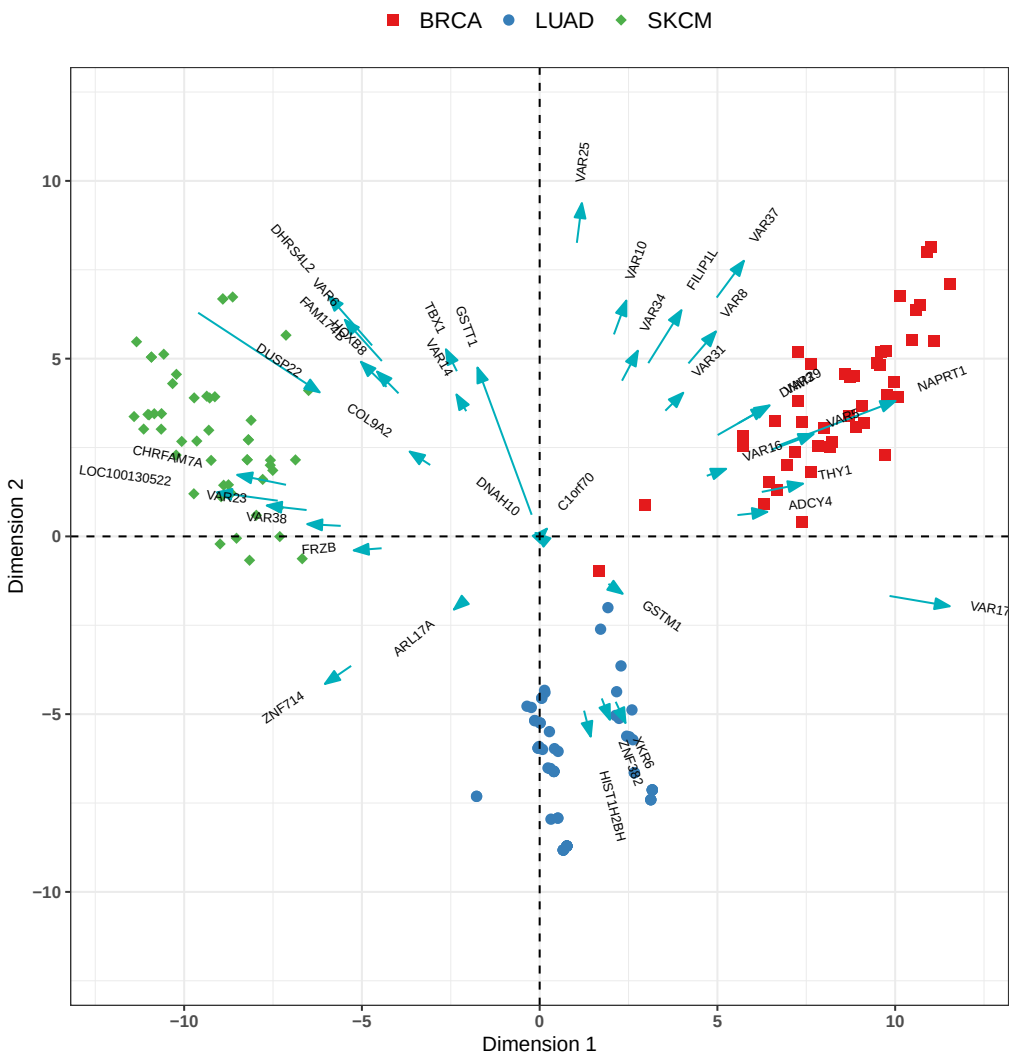


Figura 3.9: Biplot logístico usando a el algoritmo del gradiente conjugado de Fletcher–Reeves para el conjunto de datos de metilación.

La Tabla 3.2 muestra la tasa de clasificaciones correctas para cada variable usando las medidas de sensibilidad y especificidad; estas medidas permiten determinar si el modelo presenta una buena clasificación para los dos tipos de datos en cada variable. La sensibilidad mide la tasa de verdaderos positivos, mientras que la especificidad mide la tasa de verdaderos negativos y la medida global corresponde a la tasa total de clasificaciones correctas para cada variable. En general, el modelo con tres dimensiones y utilizando el algoritmo basado en el gradiente conjugado con la fórmula Fletcher–Reeves generó valores altos para la sensibilidad; sólo el gen *GSTT1* presentó una sensibilidad relativamente baja, con un 72 % de verdaderos positivos. En cuanto a la especificidad, el gen *LOC391322* obtuvo la tasa de verdaderos negativos más baja, con un 80.9 %. Por lo tanto, el modelo presenta resultados

satisfactorios.

Tabla 3.2: Sensibilidad y especificidad para cada variable cuando se ajusta el modelo LB con el algoritmo de gradiente conjugado de Fletcher–Reeves y $k = 3$.

Variable	**Sensibilidad (%)**	**Especificidad (%)**	**Global (%)**
GSTM1	97.1	94.8	96.2
C1orf70	100.0	100.0	100.0
DNM3	100.0	94.9	96.2
COL9A2	100.0	100.0	100.0
VAR5	100.0	94.3	95.6
VAR6	100.0	88.6	91.2
THY1	100.0	95.8	96.9
VAR8	100.0	93.6	95.0
DNAH10	100.0	100.0	100.0
VAR10	100.0	90.8	92.5
DHRS4L2	100.0	88.8	91.2
ADCY4	100.0	98.3	98.8
CHRFAM7A	100.0	92.4	94.4
VAR14	100.0	99.1	99.4
FAM174B	100.0	94.8	96.2
VAR16	100.0	99.1	99.4
VAR17	100.0	86.3	87.5
ARL17A	100.0	98.6	99.4
HOXB8	100.0	98.2	98.8
LOC100130522	100.0	89.3	91.9
ZNF714	100.0	100.0	100.0
ZNF382	97.9	94.6	95.6
VAR23	100.0	96.5	97.5
FRZB	100.0	99.1	99.4
VAR25	100.0	94.7	95.0
TBX1	100.0	98.3	98.8
LOC391322	100.0	80.9	81.2
GSTT1	72.0	87.1	80.0
VAR29	100.0	96.6	97.5
FILIP1L	97.1	93.6	94.4
VAR31	100.0	97.5	98.1
HIST1H2BH	100.0	89.7	92.5
DUSP22	100.0	98.4	99.4
VAR34	95.0	97.5	96.9
XKR6	100.0	93.2	95.0
NAPRT1	96.9	87.5	89.4
VAR37	95.8	95.6	95.6
VAR38	97.7	98.3	98.1

3.8. Contribuciones realizadas en este capítulo

En este capítulo se propuso y se desarrolló una metodología para estimar los parámetros del modelo LB utilizando algoritmos de gradiente conjugado no lineal y otra metodología usando un algoritmo de descenso coordinado por bloques a partir de la función sustituta demostrada en el Teorema 2.

Se incorporó el procedimiento de validación cruzada adaptado para el biplot logístico que fue descrito en el algoritmo 3, para realizar la selección del hiperparámetro k, lo que permite elegir el número de dimensiones para el modelo LB.

Los métodos basados en los algoritmos presentados son una contribución importante porque brindan alternativas que permiten resolver algunos problemas que se pueden presentar cuando se tiene un volumen alto de datos o cuando la matriz de datos está desequilibrada.

El desarrollo teórico permitió proponer cinco algoritmos iterativos que cuentan con la propiedad de que la función de pérdida decrece con cada iteración. Las propiedades de los algoritmos propuestos para ajustarse a un modelo LB fueron estudiadas generando conjuntos de datos de rango $k = 3$ y diferentes niveles de dispersión para $n = 100, 300, 500$ filas y $p = 50, 100$ columnas. La precisión de los algoritmos se midió utilizando el error de entrenamiento, el error de validación cruzada (error cv) y el error cuadrático medio relativo (RMSE) de las probabilidades logarítmicas. De acuerdo con el estudio de Monte Carlo se pudo establecer que el criterio de validación cruzada es exitoso en la estimación del hiperparámetro que representa el número de dimensiones, lo que permite especificar de manera precisa el modelo LB, obteniendo así el mejor rendimiento de los algoritmos propuestos en términos de recuperación de la estructura de bajo rango.

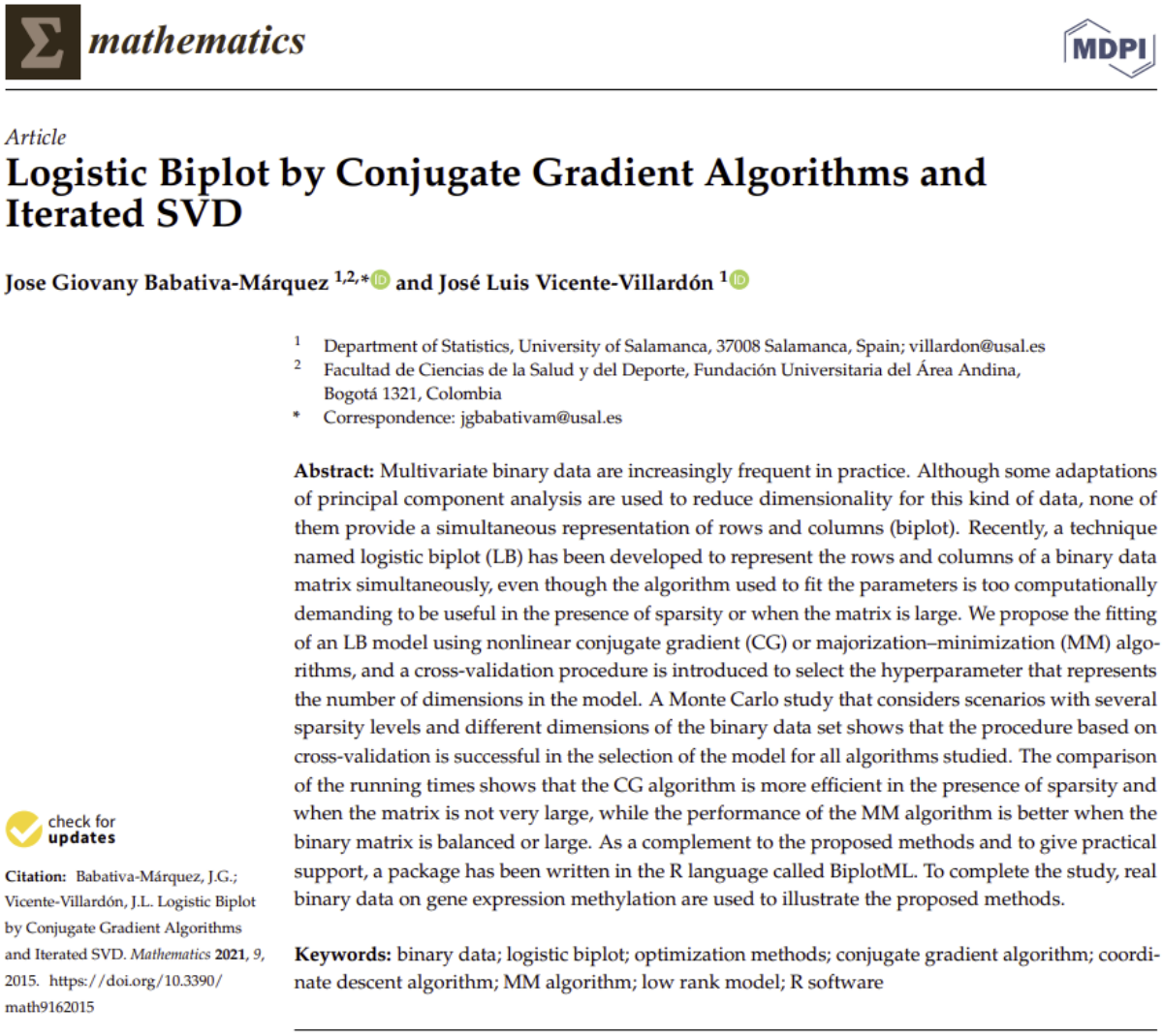
La comparación de los tiempos de ejecución mostró que los algoritmos convergen rápido. Los algoritmos basados en el gradiente conjugado son más eficientes cuando las matrices tienden a ser muy desequilibradas y no muy grandes, mientras que el rendimiento del algoritmo de descenso coordinado por bloques, basado en el método MM, es mejor cuando el número de filas y columnas tiende a aumentar; por lo tanto, es preferible para matrices grandes.

Un resumen de las contribuciones presentadas en este capítulo fueron presentadas en las ***IV***

Jornadas de Estadística como Herramienta Científica, evento organizado por el Departamento de Estadística e Investigación Operativa de la Universidad de Jaén, realizado entre el 24 y 26 de marzo del 2021, con el trabajo *"Biplot logístico usando algoritmos de machine learning"*.

Asimismo, un resumen de los resultados y hallagos más relevantes de este capítulo fueron publicados en un artículo en la revista *Mathematics* (JCR 2020: 2.258 Q1; Scopus 2021: 75/378 Q1):

Babativa-Márquez, J. G., & Vicente-Villardón, J. L. (2021). *Logistic Biplot by Conjugate Gradient Algorithms and Iterated SVD*. Mathematics, 9(16), 2015. https://doi.org/10.3390/math9162015.



mathematics

MDPI

Article

Logistic Biplot by Conjugate Gradient Algorithms and Iterated SVD

Jose Giovany Babativa-Márquez [1,2,*] and José Luis Vicente-Villardón [1]

[1] Department of Statistics, University of Salamanca, 37008 Salamanca, Spain; villardon@usal.es
[2] Facultad de Ciencias de la Salud y del Deporte, Fundación Universitaria del Área Andina, Bogotá 1321, Colombia
* Correspondence: jgbabativam@usal.es

Abstract: Multivariate binary data are increasingly frequent in practice. Although some adaptations of principal component analysis are used to reduce dimensionality for this kind of data, none of them provide a simultaneous representation of rows and columns (biplot). Recently, a technique named logistic biplot (LB) has been developed to represent the rows and columns of a binary data matrix simultaneously, even though the algorithm used to fit the parameters is too computationally demanding to be useful in the presence of sparsity or when the matrix is large. We propose the fitting of an LB model using nonlinear conjugate gradient (CG) or majorization–minimization (MM) algorithms, and a cross-validation procedure is introduced to select the hyperparameter that represents the number of dimensions in the model. A Monte Carlo study that considers scenarios with several sparsity levels and different dimensions of the binary data set shows that the procedure based on cross-validation is successful in the selection of the model for all algorithms studied. The comparison of the running times shows that the CG algorithm is more efficient in the presence of sparsity and when the matrix is not very large, while the performance of the MM algorithm is better when the binary matrix is balanced or large. As a complement to the proposed methods and to give practical support, a package has been written in the R language called BiplotML. To complete the study, real binary data on gene expression methylation are used to illustrate the proposed methods.

Keywords: binary data; logistic biplot; optimization methods; conjugate gradient algorithm; coordinate descent algorithm; MM algorithm; low rank model; R software

check for updates

Citation: Babativa-Márquez, J.G.; Vicente-Villardón, J.L. Logistic Biplot by Conjugate Gradient Algorithms and Iterated SVD. *Mathematics* **2021**, *9*, 2015. https://doi.org/10.3390/math9162015

Figura 3.10: Publicación en la revista *Mathematics* - JCR 2020: 2.258 Q1; Scopus 2021: 75/378 Q1.

Capítulo 4

Biplot logístico con información faltante usando proyección de datos

4.1. Introducción

El acceso a las redes sociales, plataformas de streaming y en general todos los medios digitales que permiten recolectar información han puesto de manifiesto la importancia de la analítica de los datos, ahora es más frecuente que se deba lidiar con matrices de datos que son cada vez más grandes. Muchas de estas se codifican en información binaria, donde resulta de gran interés realizar análisis que permitan descubrir patrones subyacentes en los datos. En estos casos los métodos biplot pueden ser utilizados para explotar la información y encontrar relaciones multivariantes que no son fáciles de identificar.

Los métodos ilustrados hasta el momento para ajustar el modelo LB se basan en una estimación del espacio de parámetros $\mathbf{\Theta}$ donde $\boldsymbol{\mu}$, $\mathbf{A}$ y $\mathbf{B}$ son estimados en cada iteración. Por lo tanto, bajo este enfoque cada fila tiene sus propios parámetros, $\boldsymbol{\theta}_i = \boldsymbol{\mu}_i + \sum_{s=1}^{k} a_{is}\mathbf{b}_s$, $i = 1, \ldots, n$, así que el número de parámetros a estimar se incrementa con el número de filas o tamaño de la muestra, lo que puede representar un problema cuando se tienen grandes volúmenes de datos.

El enfoque manejado hasta ahora también dificulta la proyección de nuevos individuos como filas suplementarias, ya que sería necesario ajustar nuevamente el modelo para encontrar las coordenadas de las nuevas filas, lo que podría generar un posible problema de sobreajuste del modelo. El reto consiste en encontrar un método que permita estimar los parámetros de la matriz de alguno de los marcadores, y obtener la otra como una función de los

marcadores estimados. Esto conlleva a una reducción en la cantidad de parámetros a estimar y disminuye el esfuerzo computacional, con la ventaja adicional de que se podrían proyectar nuevas filas sin tener que ajustar nuevamente el modelo.

Con la motivación anterior, en este capítulo se desarrolla un nuevo enfoque para ajustar el modelo LB que permita resolver las problemáticas enunciadas previamente y se incorpore el hecho de que la matriz $\mathbf{X}$ puede tener datos faltantes.

En este capítulo está organizado de la siguiente forma. En la sección 4.2 se presenta el método de proyección de datos para un biplot clásico. En la sección 4.3 se realiza una adaptación del método de proyección de datos para un modelo LB. En la sección 4.4 se presenta el desarrollo teórico que permite llegar a un problema de optimización que considera los datos faltantes. En la sección 4.5 se realiza el desarrollo para formular un algoritmo que permite resolver el problema de minimización y logra una imputación de los valores faltantes durante el proceso de optmización. Con el fin de aplicar la metodología popuesta, en la sección 4.6 se presenta una aplicación usando datos reales sobre el conflicto armado en Colombia. Finalmente, la sección 4.7 se encarga de realizar un resumen de algunas de las contribuciones más importantes de este capítulo.

4.2. Biplot para datos continuos usando proyección de datos

El método se desarrolla usando el enfoque de Pearson (1901) para un PCA, que consiste en encontrar la representación óptima de los datos multivariantes en el espacio de baja dimensión al minimizar el error cuadrático medio de la proyección. Dado un conjunto de datos $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$, con $\mathbf{x}_i \in \mathbb{R}^p$, $i = 1, \ldots, n$, con vector de medias $\boldsymbol{\mu} \in \mathbb{R}^p$. Si $rank(\mathbf{X}) = r$, entonces para un entero positivo $k \leq r$, el problema de optimización consiste en encontrar $\mathbf{A}$ y $\mathbf{B}$ que minimice la norma de Frobenius dada en la ecuación (1), que para los datos sin centrar puede ser escrito como

$$\sum_{i=1}^{n} \|\mathbf{x}_i - \boldsymbol{\mu} - (\mathbf{a}_{i1}\mathbf{b}_1 + \ldots + \mathbf{a}_{ik}\mathbf{b}_k)\|^2 = \left\|\mathbf{X} - \mathbf{1}_n\boldsymbol{\mu}^T - \mathbf{A}\mathbf{B}^T\right\|_F^2. \tag{4.1}$$

Como se mencionó en los capítulos anteriores, de acuerdo con Eckart y Young (1936),

la mejor aproximación de bajo rango para $\mathbf{X}$ es obtenida como $\mathbf{AB}^T = \mathbf{U}\Lambda\mathbf{V}^T$, donde $\mathbf{A} = \mathbf{U}\Lambda^\gamma$ y $\mathbf{B} = \mathbf{V}\Lambda^{(1-\gamma)}, 0 \leq \gamma \leq 1$, es decir que $\mathbf{A}$ corresponde a los k vectores singulares por izquierda escalados por los k valores singulares a la potencia γ, mientras que $\mathbf{B}$ corresponde a los k vectores singulares por derecha ponderados por los k valores singulares a la potencia $1-\gamma$. De esta manera, la matriz $\mathbf{X}$ puede ser representada por los marcadores $\mathbf{a_1}, \ldots, \mathbf{a_n}$ para las filas y $\mathbf{b_1}, \ldots, \mathbf{b_k}$ para las columnas, donde el ij-ésimo elemento de la matriz denonato por x_{ij} es aproximado por el producto $\mathbf{a_i^T b_j}$ y el espacio natural de parámetros está determinado por $\boldsymbol{\Theta} = \mathbf{1}_n\boldsymbol{\mu}^T + \mathbf{AB}^T$.

La solución del problema también puede ser expresada como una proyección de los datos sobre un subespacio de baja dimensión que minimice la suma de los cuadrados de las distancias desde $\mathbf{x}_i$ a su proyección $\boldsymbol{\theta_i}$. Pearson (1901) muestra que el mínimo error cuadrático medio de la representación k-dimensional

$$\left\|\mathbf{X} - \mathbf{1}_n\boldsymbol{\mu}^T - (\mathbf{X} - \mathbf{1}_n\boldsymbol{\mu}^T)\mathbf{V}\mathbf{V}^T\right\|_F^2, \tag{4.2}$$

se obtiene cuando $\boldsymbol{\mu}$ es el vector de medias y $\mathbf{V}$ son los vectores singulares por derecha de la matriz $\mathbf{X}$. De esta manera,

$$\begin{aligned}\mathbf{AB}^T &= \mathbf{U}\Lambda\mathbf{V}^T \\ &= \left(\mathbf{X} - \mathbf{1}_n\boldsymbol{\mu}^T\right)\mathbf{V}\mathbf{V}^T,\end{aligned} \tag{4.3}$$

donde $\mathbf{V}^T\mathbf{V} = \mathbf{I}$. Al suponer que cada $\mathbf{x}_i$ es una extracción aleatoria con $\mathbf{x}_i \sim N(\boldsymbol{\theta}_i, \mathbf{I}_p)$ y $\boldsymbol{\theta}_i$ es restingido al subespacio de k dimensiones, este problema puede formularse desde un enfoque probabilístico como una estimación de máxima verosimilitud de la misma forma que se mostró en la sección 2.2. Lo que permite incorporar funciones de probabilidad para datos binarios.

4.3. Adaptación del método de proyección de datos para el biplot logístico

Debido a la naturaleza binaria de los datos, podemos asumir que $\mathbf{x}_i$ proviene de una distribución de familia exponencial con parámetro natural $\boldsymbol{\theta}_i$, $i = 1, \ldots, n$, entonces el biplot se encontraría en un subespacio de k dimensiones para los parámetros canónicos

al minimizar la función de pérdida. Como se ha señalado antes, a diferencia de un biplot clásico, cuando la matrix $\mathbf{X}$ es binaria, no se puede llevar a cabo el procedimiento de centrado porque los datos centrados ya no son ceros y unos, así que el vector $\boldsymbol{\mu}$ se introduce en el biplot logístico como término de compensación para hacer un centrado basado en el modelo y la función de pérdida se construye partiendo de que $\mathbf{x}_i$ proviene de una distribución Bernoulli (Schein y col., 2003; Vicente-Villardon y col., 2006).

Hasta ahora ninguno de los métodos propuestos para ajustar un modelo LB ha considerado la presencia de información faltante en la matriz de datos, pero este es un problema usual en la práctica. Con la motivación de que la matriz binaria $\mathbf{X}$ puede tener valores faltantes, se aborda el problema de estimación usando un enfoque que permita encontrar $\boldsymbol{\mu}$, $\mathbf{A}$ y $\mathbf{B}$ al minimizar el error $\|\mathbf{X}-\boldsymbol{\Theta}\|_F^2$ considerando solo las entradas conocidas de $\mathbf{X}$ y que permita estimar los datos faltantes, para ello se define la matriz de pesos $\mathbf{W} \in \mathbb{R}^{n\times p}$ con entradas w_{ij} que permite codificar la localización de los datos faltantes en la martriz $\mathbf{X}$, así

$$w_{ij} = \begin{cases} 1 & \text{si } x_{ij} \text{ es conocido,} \\ 0 & \text{si } x_{ij} \text{ es es un dato faltante.} \end{cases} \tag{4.4}$$

Expresando $\mathbf{AB}^T$ como en (4.3), el problema de optimización consiste en

$$\begin{aligned} \min_{\mathbf{V}:\mathbf{V}^T\mathbf{V}=\mathbf{I}} \|\mathbf{W}\odot(\mathbf{X}-\boldsymbol{\Theta})\|_F^2 &= \min_{\mathbf{V}:\mathbf{V}^T\mathbf{V}=\mathbf{I}} \left\|\mathbf{W}\odot(\mathbf{X}-\mathbf{1}_n\boldsymbol{\mu}^T-\mathbf{AB}^T)\right\|_F^2 \\ &= \min_{\mathbf{V}:\mathbf{V}^T\mathbf{V}=\mathbf{I}} \left\|\mathbf{W}\odot(\mathbf{X}-\mathbf{1}_n\boldsymbol{\mu}^T-(\mathbf{X}-\mathbf{1}_n\boldsymbol{\mu}^T)\mathbf{VV}^T)\right\|_F^2, \end{aligned} \tag{4.5}$$

donde $\odot$ denota el producto de Hadamard. De esta manera, la solución para los marcadores fila de un biplot logístico usando la proyección del espacio de los parámetros es $\mathbf{A} = (\mathbf{X}-\mathbf{1}_n\boldsymbol{\mu}^T)\mathbf{V}$, que cuenta con menos parámetros a estimar en comparación con las metodologías la formulación clásica del biplot logístico (Babativa-Márquez y Vicente-Villardón, 2021; Demey y col., 2008; Vicente-Villardon y col., 2006). Además, este nuevo enfoque permite la proyección de filas suplementarias de una forma sencilla, sin tener que llevar a cabo otro proceso de optimización, dado que los marcadores fila se obtienen a partir de los marcadores columna $\mathbf{B} = \mathbf{V}$, evitando así un posible sobreajuste del modelo.

En este caso, fuera de estimar $\boldsymbol{\mu}$ y $\mathbf{V}$, el objetivo del modelo LB también es estimar los valores faltantes $(\mathbf{11}^T-\mathbf{W})\odot\mathbf{X}$, a partir de los datos conocidos. Aunque, es claro que no

siempre es posible encontrar la solución a este problema y que la matriz completa $\mathbf{X}$ pueda ser recuperada depende de cuáles y cuántas entradas faltan. Por ejemplo, si faltan todos los datos de una fila o una columna, entonces no habrá forma de recuperar esa información. Asimismo, si el porcentaje de datos faltantes resulta ser relativamente grande, como en la mayoría de algoritmos usados para asignar valores plausibles, se espera que el desempeño del método se vea afectado. En Candès y Recht (2009) estudian algunas condiciones que se deben tener en cuenta para recuperar perfectamente la mayoría de las entradas faltantes.

Por lo tanto, para evitar soluciones ambiguas debido a las situaciones anteriores, en adelante se supondrá que las ubicaciones de las entradas que faltan son lo suficientemente aleatorias para que la probabilidad de que formen un patrón evidente sea muy baja. Además, se supone que la proporción de datos observados es suficiente para obtener una buena recuperación de la matriz de datos.

4.4. Función sustituta para el biplot logístico con datos faltantes

Recordando, se tiene la matriz $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ con $\mathbf{x}_i \in \{0,1\}^p$, $i = 1, \ldots, n$, $rank(\mathbf{X}) = r$ y $x_{ij} \sim Ber(\pi(\theta_{ij}))$, donde $\pi(\cdot)$ es la inversa de la función de enlace, en este trabajo se usa la función logística, $\pi(\theta_{ij}) = \{1 + exp(-\theta_{ij})\}^{-1}$, que representa la probabilidad esperada de que la característica j se presente en el individuo i, el log-odds $\pi(\theta_{ij})$ es θ_{ij} con $\theta_{ij} = log\{\pi(\theta_{ij})/(1 - \pi(\theta_{ij}))\}$, que corresponde al espacio de parámetros naturales de una distribución Bernoulli expresada en forma de familia exponencial. Teniendo en cuenta que $w_{ij} = 1$ si x_{ij} es conocido y $w_{ij} = 0$ cuando x_{ij} es un dato faltante. La función de pérdida es obtenida del negativo del logaritmo de la función de verosimilitud, así:

$$\mathcal{L}(\boldsymbol{\Theta}) = -log(p(\mathbf{X}; \boldsymbol{\Theta}, \mathbf{W})) \tag{4.6a}$$

$$= -log\left(\prod_{i=1}^{n}\prod_{j=1}^{p}\left[p(x_{ij};\theta_{ij})\right]^{w_{ij}}\right) \tag{4.6b}$$

$$= -\sum_{i=1}^{n}\sum_{j=1}^{p} w_{ij}\left[x_{ij}log(\pi(\theta_{ij})) + (1-x_{ij})log(1-\pi(\theta_{ij}))\right] \tag{4.6c}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{p} w_{ij} f(\theta_{ij}). \tag{4.6d}$$

A partir del Teorema 2 la función la función sustituta sin datos faltantes es:

$$f(\theta_{ij}) \leq \frac{1}{8}\left(\theta_{ij} - \theta_{ij}^{(l)} + 4(\pi(\theta_{ij}^{(l)}) - x_{ij})\right)^2 + C, \tag{4.7}$$

donde C es una constante que no depende de $\boldsymbol{\Theta}$ y $\theta_{ij}^{(l)}$ es la aproximación de θ_{ij} en la l-ésima iteración. Haciendo $z_{ij}^{(l)} = \theta_{ij}^{(l)} + 4(x_{ij} - \pi(\theta_{ij}^{(l)}))$ y $\mathbf{Z}_l$ la matriz con el ij-ésimo elemento igual a $z_{ij}^{(l)}$. Por lo tanto, la función de pérdida en presencia de datos faltantes puede ser mayorizada como

$$\mathcal{L}(\boldsymbol{\Theta}) \leq \frac{1}{8}\sum_{i=1}^{n}\sum_{j=1}^{p} w_{ij}\left(\theta_{ij} - z_{ij}^{(l)}\right)^2 + C \tag{4.8a}$$

$$= \frac{1}{8}\left\|(\boldsymbol{\Theta} - \mathbf{Z}_l)\odot\mathbf{W}\right\|_F^2 + C. \tag{4.8b}$$

La función anterior no es fácil de trabajar, pero también puede ser mayorizada. Para ello se define y se demuestra el siguiente teorema.

Teorema 3. *Si* $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ *es una matriz binaria, con* $\mathbf{x}_i \in \{0,1\}^p$, $i = 1, \ldots, n$ *y* $x_{ij} \sim Ber(\pi(\theta_{ij}))$, *donde* $\pi(\theta_{ij}) = \{1 + exp(-\theta_{ij})\}^{-1}$, *con* $\boldsymbol{\Theta} = \mathbf{1}_n\boldsymbol{\mu}^T + \mathbf{A}\mathbf{B}^T$, *la matriz canónica de parámetros y* $\mathbf{W} \in \mathbb{R}^{n\times p}$ *es una matriz binaria con entradas* w_{ij} *que permite codificar con cero la localización de los datos faltantes en* $\mathbf{X}$ *y* $\mathbf{Z}_l = \boldsymbol{\Theta}_l + 4(\mathbf{X} - \boldsymbol{\Pi}_l)$ *entonces*

$$\left\|(\boldsymbol{\Theta} - \mathbf{Z}_l)\odot\mathbf{W}\right\|_F^2 \leq \left\|\boldsymbol{\Theta} - \mathbf{M}_l\right\|_F^2,$$

con $\mathbf{M}_l = \boldsymbol{\Theta}_l + 4\left[\mathbf{W}\odot(\mathbf{X} - \boldsymbol{\Pi}_l)\right]$.

Demostración. De acuerdo con Kiers (1997), se puede escribir

$$\left\|(\boldsymbol{\Theta} - \mathbf{Z}_l)\odot\mathbf{W}\right\|_F^2 = \left\|\mathbf{D}_W\left(\mathrm{Vec}(\mathbf{Z}_l) - \mathrm{Vec}(\boldsymbol{\Theta})\right)\right\|_F^2, \tag{4.9}$$

con $\mathbf{D}_W$ es la matriz diagonal que contiene los elementos de Vec($\mathbf{W}$), donde Vec($\cdot$) denota el operador de vectorización de una matriz. Que al usar la función de Heiser (1987), se puede llegar a la función mayorizada

$$\mathcal{G}\left(\boldsymbol{\Theta}|\boldsymbol{\Theta}_l, \mathbf{Z}_l, \mathbf{W}\right) = w_m^2 \left\|\text{Vec}(\boldsymbol{\Theta}) - \left(\text{Vec}(\boldsymbol{\Theta}_l) + w_m^{-2}\mathbf{D}_W^2(\text{Vec}(\mathbf{Z}_l) - \text{Vec}(\boldsymbol{\Theta}_l))\right)\right\|_F^2 + C \tag{4.10}$$

$$= w_m^2 \left\|\boldsymbol{\Theta} - \left(\boldsymbol{\Theta}_l + w_m^{-2}\mathbf{W}^2 \odot \mathbf{Z}_l - w_m^{-2}\mathbf{W}^2 \odot \boldsymbol{\Theta}_l\right)\right\|_F^2 + C, \tag{4.11}$$

donde $\mathbf{W}^2 = \mathbf{W} \odot \mathbf{W}$, C es una constante, w_m^2 es el valor propio más grande de $\mathbf{D}_W^2$, el cual es el máximo de los elementos al cuadrado de $\mathbf{W}$.

Para el caso del modelo LB con datos faltantes se tiene que $\mathbf{W}$ es una matriz binaria donde el elemento $w_{ij} = 1$ si el dato es conocido y $w_{ij} = 0$ si es faltante. De modo que $\mathbf{W}^2 = \mathbf{W} \odot \mathbf{W} = \mathbf{W}$ y dado que w_m^2 es el máximo de los elementos al cuadrado de $\mathbf{W}$, entonces $w_m^2 = 1$. De esta forma, se tiene que

$$\|(\boldsymbol{\Theta} - \mathbf{Z}_l) \odot \mathbf{W}\|_F^2 \leq \|\boldsymbol{\Theta} - (\mathbf{W} \odot \mathbf{Z_l} + \boldsymbol{\Theta}_l - \mathbf{W} \odot \boldsymbol{\Theta}_l)\|_F^2 \tag{4.12}$$

$$= \|\boldsymbol{\Theta} - (\mathbf{W} \odot [\boldsymbol{\Theta}_l + 4\,(\mathbf{X} - \boldsymbol{\Pi}_l)] + \boldsymbol{\Theta}_l - \mathbf{W} \odot \boldsymbol{\Theta}_l)\|_F^2 \tag{4.13}$$

$$= \|\boldsymbol{\Theta} - (\mathbf{W} \odot \boldsymbol{\Theta}_l + 4\,[\mathbf{W} \odot (\mathbf{X} - \boldsymbol{\Pi}_l)] + \boldsymbol{\Theta}_l - \mathbf{W} \odot \boldsymbol{\Theta}_l)\|_F^2 \tag{4.14}$$

$$= \|\boldsymbol{\Theta} - (\boldsymbol{\Theta}_l + 4\,[\mathbf{W} \odot (\mathbf{X} - \boldsymbol{\Pi}_l)])\|_F^2 \tag{4.15}$$

$$= \|\boldsymbol{\Theta} - \mathbf{M}_l\|_F^2, \tag{4.16}$$

con $\mathbf{M}_l = \boldsymbol{\Theta}_l + 4\,[\mathbf{W} \odot (\mathbf{X} - \boldsymbol{\Pi}_l)]$. □

Teniendo en cuenta el resultado del Teorema 3, el problema de optimización para un modelo LB con datos faltantes usando la función sustituta obtenida por la mayorización, se puede escribir como

$$\min_{\mathbf{V},\boldsymbol{\mu}} \quad \|\boldsymbol{\Theta} - \mathbf{M}_l\|_F^2 \tag{4.17a}$$

$$\text{sujeto a} \quad \boldsymbol{\Theta} = \mathbf{1}_n\boldsymbol{\mu}^T + (\mathbf{X} - \mathbf{1}_n\boldsymbol{\mu}^T)\mathbf{V}\mathbf{V}^T, \tag{4.17b}$$

$$\mathbf{M}_l = \boldsymbol{\Theta}_l + 4\,[\mathbf{W} \odot (\mathbf{X} - \boldsymbol{\Pi}_l)]$$

$$\mathbf{V}^T\mathbf{V} = \mathbf{I}$$

4.5. Algoritmo de estimación MM-BCD

Al igual que en el capítulo 3, el problema de mayorización dado en la ecuación (4.17a), se puede resolver durante la l-ésima iteración usando un algoritmo de descenso coordinado

por bloques, para lo cual se asignan valores iniciales para $\mathbf{V}$ y $\boldsymbol{\mu}$. De esta manera, al fijar $\boldsymbol{\mu}$ el minimizador de la función sustituta de la función objetivo puede ser encontrado.

Denotando $\mathbf{X}_c = \mathbf{X} - \mathbf{1}_n\boldsymbol{\mu}^T$ y $\mathbf{M}_{l,c} = \mathbf{M}_l - \mathbf{1}_n\boldsymbol{\mu}^T$, entonces

$$\begin{aligned}
\min_{\mathbf{V}:\mathbf{V}^T\mathbf{V}=\mathbf{I}} \|\boldsymbol{\Theta} - \mathbf{M}_l\|_F^2 &= \min_{\mathbf{V}:\mathbf{V}^T\mathbf{V}=\mathbf{I}} \left\|\mathbf{1}_n\boldsymbol{\mu}^T + (\mathbf{X} - \mathbf{1}_n\boldsymbol{\mu}^T)\mathbf{V}\mathbf{V}^T - \mathbf{M}_l\right\|_F^2 && (4.18)\\
&= \text{mín}\left\{\text{tr}\left(\mathbf{V}\mathbf{V}^T\mathbf{X}_c^T\mathbf{X}_c\mathbf{V}\mathbf{V}^T\right) - \text{tr}\left(\mathbf{V}\mathbf{V}^T\mathbf{X}_c^T\mathbf{M}_{l,c}\right) - \text{tr}\left(\mathbf{M}_{l,c}^T\mathbf{X}_c\mathbf{V}\mathbf{V}^T\right)\right\} && (4.19)\\
&= \text{mín}\,\text{tr}\left(\mathbf{V}^T\left(\mathbf{X}_c^T\mathbf{X}_c - \mathbf{X}_c^T\mathbf{M}_{l,c} - \mathbf{M}_{l,c}^T\mathbf{X}_c\right)\mathbf{V}\right) && (4.20)\\
&= \text{máx}\,\text{tr}\left(\mathbf{V}^T\left(\mathbf{M}_{l,c}^T\mathbf{X}_c + \mathbf{X}_c^T\mathbf{M}_{l,c} - \mathbf{X}_c^T\mathbf{X}_c\right)\mathbf{V}\right). && (4.21)
\end{aligned}$$

La traza se maximiza cuando $\mathbf{V}$ está compuesta por los k primeros vectores propios de la matriz simétrica $\mathbf{Y}_l = \mathbf{M}_{l,c}^T\mathbf{X}_c + \mathbf{X}_c^T\mathbf{M}_{l,c} - \mathbf{X}_c^T\mathbf{X}_c$.

La actualización de $\boldsymbol{\mu}$ se obtiene al fijar $\mathbf{V}$ en la ecuación (4.17a), el cual es un problema de mínimos cuadrados, de modo que la solución para la l-ésima iteración es $\boldsymbol{\mu}_l = \frac{1}{n}\left(\mathbf{M}_l - \mathbf{X}\mathbf{V}\mathbf{V}^T\right)^T\mathbf{1}_n$. Después de obtener $\boldsymbol{\mu}$ se puede calcular $\mathbf{Y}_l$ para obtener la solución de la SVD, así $\mathbf{Y}_l = \boldsymbol{\Gamma}\boldsymbol{\Lambda}\boldsymbol{\Gamma}^T$ y se toma $\mathbf{V}$ como los primeros k vectores propios de $\boldsymbol{\Gamma}$.

Para dar un tratamiento a los datos faltantes en $\mathbf{X}$ y lograr una imputación durante el proceso de minimización de la función sustituta dada en (4.17a); los valores faltantes son reemplazados por un valor inicial y se ajusta el modelo para los datos completos (incluyendo las estimaciones de los datos faltantes). En la l-ésima iteración, la matriz con los datos imputados se calcula como $\mathbf{X}_l = \mathbf{W} \odot \mathbf{X} + (\mathbf{1}\mathbf{1}^T - \mathbf{W}) \odot \hat{\mathbf{X}}_l$, donde el ij-ésimo elemento de $\hat{\mathbf{X}}_l$ es igual a 1 cuando la inversa de la función de enlace logit supera el umbral δ_j y cero en otro caso. De esta manera los valores observados son los mismos y los valores faltantes son reemplazados por el valor ajustado. La forma en que se calculan los umbrales δ_j es la misma que se usó en los algoritmos anteriores, donde se elige el valor que minimiza la tasa de error equilibrada (TEE) para la variable j, $j = 1\ldots, p$.

El problema de optimización se resuelve de forma iterativa asignando valores iniciales para $\mathbf{V}$ y $\boldsymbol{\mu}$, esta inicialización puede hacerse usando una estrategia aleatoria o con valores definidos por el usuario. El pseudocódigo se describe en el Algoritmo 7.

Algoritmo 7 Algoritmo para ajustar el modelo LB con datos faltantes usando proyección de datos

Entrada $\mathbf{X}$
Salida $\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}$

Inicializar $\boldsymbol{\mu}_0, \mathbf{V}_0$
Calcular $\mathbf{W}$, con $w_{ij} = 1$ si x_{ij} es conocido y $w_{ij} = 0$ si x_{ij} es un valor faltante.
$\mathbf{X}_0 = \mathbf{W} \odot \mathbf{X} + \frac{1}{2}(\mathbf{1} - \mathbf{W})$
Calcular $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_p)$ donde $\delta_j = \arg\min_{\delta} \{TER(\mathbf{x}_j|\delta) : 0 < \delta < 1\}, j = 1, \ldots, p.$
$l = 0$
repeat
    $\boldsymbol{\Theta}_l = \mathbf{1}_n \boldsymbol{\mu}_l^T + (\mathbf{X}_l - \mathbf{1}_n \boldsymbol{\mu}_l^T)\mathbf{V}_l \mathbf{V}_l^T$
    $\mathbf{Z}_l = \boldsymbol{\Theta}_l + 4\left(\mathbf{X}_l - \pi\left(\boldsymbol{\Theta}_l\right)\right)$
    $\mathbf{M}_l = \mathbf{W} \odot \mathbf{Z}_l + (\mathbf{1}\mathbf{1}^T - \mathbf{W}) \odot \boldsymbol{\Theta}_l$
    $\mathbf{X}_{l,c} = \mathbf{X}_l - \mathbf{1}_n \boldsymbol{\mu}_l^T$ y $\mathbf{M}_{l,c} = \mathbf{M}_l - \mathbf{1}_n \boldsymbol{\mu}_l^T$
    $\mathbf{Y}_l = \mathbf{M}_{l,c}^T \mathbf{X}_c + \mathbf{X}_c^T \mathbf{M}_{l,c} - \mathbf{X}_c^T \mathbf{X}_c$
    $\mathbf{Y}_l = \boldsymbol{\Gamma} \boldsymbol{\Lambda} \boldsymbol{\Gamma}^T$
    $\mathbf{V}_{l+1} = \boldsymbol{\Gamma}_k$, los primeros k vectores propios en $\boldsymbol{\Gamma}$
    $\boldsymbol{\mu}_{l+1} = \frac{1}{n}\left(\mathbf{M}_l - \mathbf{X}_l \mathbf{V}_{l+1} \mathbf{V}_{l+1}^T\right)^T \mathbf{1}_n$
    $\boldsymbol{\Theta}_{l+1} = \mathbf{1}_n \boldsymbol{\mu}_{l+1}^T + (\mathbf{X}_l - \mathbf{1}_n \boldsymbol{\mu}_{l+1}^T)\mathbf{V}_{l+1} \mathbf{V}_{l+1}^T$
    Los valores faltantes son imputados con los valores ajustados $\hat{\mathbf{X}}_{l+1}$, donde $\hat{x}_{ij} = 1$ si $\pi(\theta_{ij}^{l+1}) > \alpha_j$ y $\hat{x}_{ij}^l = 0$, en otro caso.
    $\mathbf{X}_{l+1} = \mathbf{W} \odot \mathbf{X} + (\mathbf{1} - \mathbf{W}) \odot \hat{\mathbf{X}}_{l+1}$
    $l = l + 1$
until $\left(\mathcal{L}\left(\boldsymbol{\Theta}_{l\text{-}1}\right) - \mathcal{L}\left(\boldsymbol{\Theta}_l\right)\right) / \mathcal{L}\left(\boldsymbol{\Theta}_{l\text{-}1}\right) < \epsilon$
$\mathbf{A} = (\mathbf{X}_l - \mathbf{1}_n \boldsymbol{\mu}_l^T)\mathbf{V}_l$
$\mathbf{B} = \mathbf{V}_l$

4.6. Aplicación

El conflicto armado interno en Colombia lleva más de 5 décadas, de acuerdo con el informe *¡Basta ya!*, publicado por el grupo Memoria Histórica (2013), del Centro Nacional de Memoria Histórica (CNMH). Para el año 2012 ya se tenían más de 220.000 muertes a causa del conflicto, donde los principales responsables fueron los grupos paramilitares y los grupos guerrilleros. El Registro Único de Víctimas reportaba en el año 2021 un total de 9.134.347 víctimas entre desaparición forzada, amenazas, secuestros, homicidios, reclutamiento de menores, desplazamiento, entre otras modalidades de violencia que han ocurrido en más de 10 millones de eventos[1]. Con el propósito de responder a la pregunta de ¿Quién le hizo qué a quién, cuándo, dónde y cómo?, el Observatorio de Memoria y Conflicto (OMC) ha

[1] https://www.unidadvictimas.gov.co

realizado la documentación de más de 268.000 muertes que se han presentado en el marco del conflicto[2], en cada caso se pueden identificar las modalidades de violencia que sufrieron las víctimas, el responsable, los hechos simultáneos, así como la fecha, lugar de los hechos, entre otras características. Para los investigadores resulta de gran relevancia poder analizar el entramado de la violencia e identificar los patrones que permiten explicar las dinámicas del conflicto.

Debido a que en ocasiones las víctimas se pueden identifiar con facilidad o la fuente donde se registró el hecho no contiene toda la información, entonces la matriz de datos es incompleta. De modo que un enfoque a partir un biplot logístico para datos faltantes puede ser apropiado para describir algunos patrones de los responsables mediante una aproximación de la matriz de datos binaria en un subespacio de dimensión reducida, lo que permite observar las asociaciones presentes en los datos. De esta manera, la representación mediante el biplot podría revelar la estructura subyacente que usan los grupos armados en el marco del conflicto.

A partir de los datos publicados por el OMC, se analiza una muestra de 7.165 eventos ocurridos entre los años 1980 y 2010 para 14 tipos de violencia. Estos datos se organizaron en una matriz binaria, $\mathbf{X}$, donde x_{ij} se codifica con 1 cuando en el evento i se usó la violencia j y 0 en otro caso. La matriz está compuesta por un 15.4 % de 1's y las proporciones de uso de las violencias estuvieron en un rango entre 0.05 y 0.56, mientras que la proporción de datos faltantes fue del 4.7 %.

El enfoque mediante un biplot logístico para datos faltantes permite la representación simultánea de filas y columnas. Para especificar el modelo se realiza el procedimiento de validación cruzada y así determinar el número de dimensiones. En la Figura 4.1 se observa que $k = 4$ minimiza el error de validación cruzadas, así que se considera como un valor apropiado para ajustar el modelo.

[2] https://micrositios.centrodememoriahistorica.gov.co/observatorio/

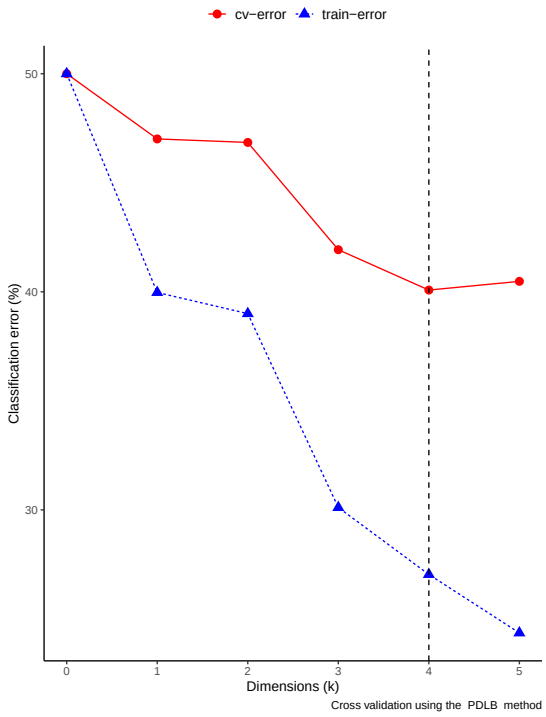


Figura 4.1: Procedimiento de validación cruzada para los datos del conflicto.

La Figura 4.2 muestra el biplot obtenido con el conjunto de datos del conflicto armado. Para ajustar el modelo se usó el algoritmo 7, de descenso de coordenadas en bloque con datos faltantes. Los segmentos con flechas representan a las variables, iniciando en el punto que predice una probabilidad de 0.5 y finaliza en el punto que predice una probabilidad de 0.75. Por lo tanto, los vectores cortos indican un rápido aumento de la probabilidad de que se presente el tipo de violencia, mientras que la proyección ortogonal de los marcadores fila sobre el vector aproximan a la probabilidad de que se presente el tipo de violencia.

En la Figura 4.2 se observa una separación de los marcadores fila en tres clúster, caracterizados principalmente por el grupo responsable. Los hechos como extorsión, desaparición forzada o pillaje tienen una mayor probabilidad de ser cometidos por grupos paramilitares. Los ataques terroristas, ataques contra la población civil o el reclutamiento de niños, niñas y adolescentes presentan una mayor probabilidad de que sean cometidos principalmente por grupos guerrilleros. Finalmente, la violencia sexual, el secuestro o la detención arbitraria son hechos que se asocian con una alta probabilidad a los agentes del estado.

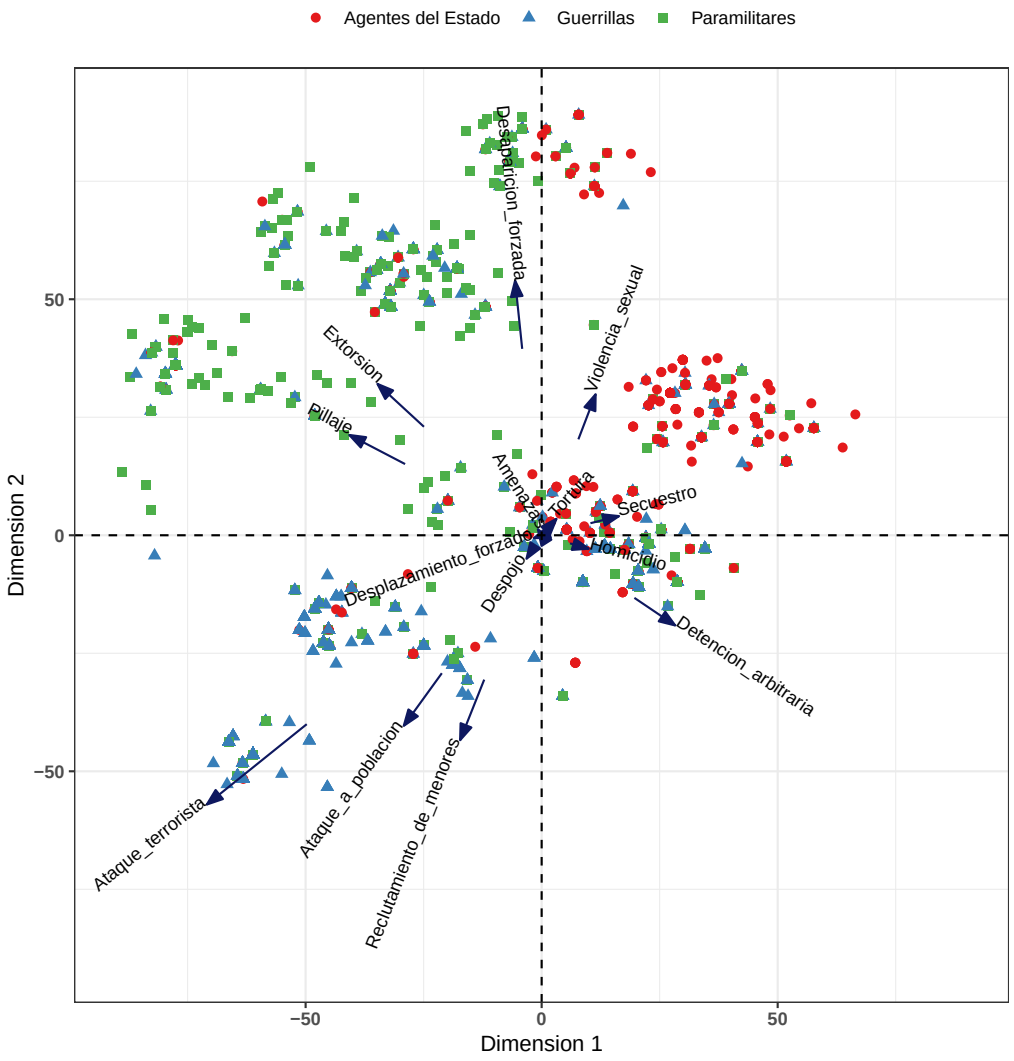


Figura 4.2: Biplot logístico del conflicto armado en Colombia usando el método de proyección de datos.

La Tabla 4.1 muestra la sensibilidad y especificidad para cada tipo de violencia. La sensibilidad mide la tasa de verdaderos positivos, la especificidad mide la tasa de verdaderos negativos y la medida global corresponde a la tasa total de clasificaciones correctas para cada variable. Estas medidas permiten determinar que el modelo, en general, exhibe una buena tasa de clasificación para los dos tipos de datos en cada variable.

Tabla 4.1: Sensibilidad y especificidad para cada variable en el ajuste del modelo LB usando el método de proyección de datos.

Variable	**Sensibilidad**	**Especificidad**	**Global**
Homicidio	99.8	100.0	100.0
Amenazas	100.0	100.0	100.0
Tortura	100.0	100.0	100.0
Despojo	100.0	100.0	100.0
Desplazamiento forzado	99.9	100.0	100.0
Secuestro	99.3	97.3	97.5
Reclutamiento de menores	76.9	91.5	90.5
Pillaje	61.0	88.4	86.6
Violencia Sexual	93.6	73.8	75.6
Detención Arbitraria	91.3	70.0	71.4
Ataque terrorista	80.3	58.0	59.5
Extorsión	72.5	52.5	53.7
Ataque a población	94.9	50.6	53.9
Desaparición Forzada	68.9	51.1	52.0

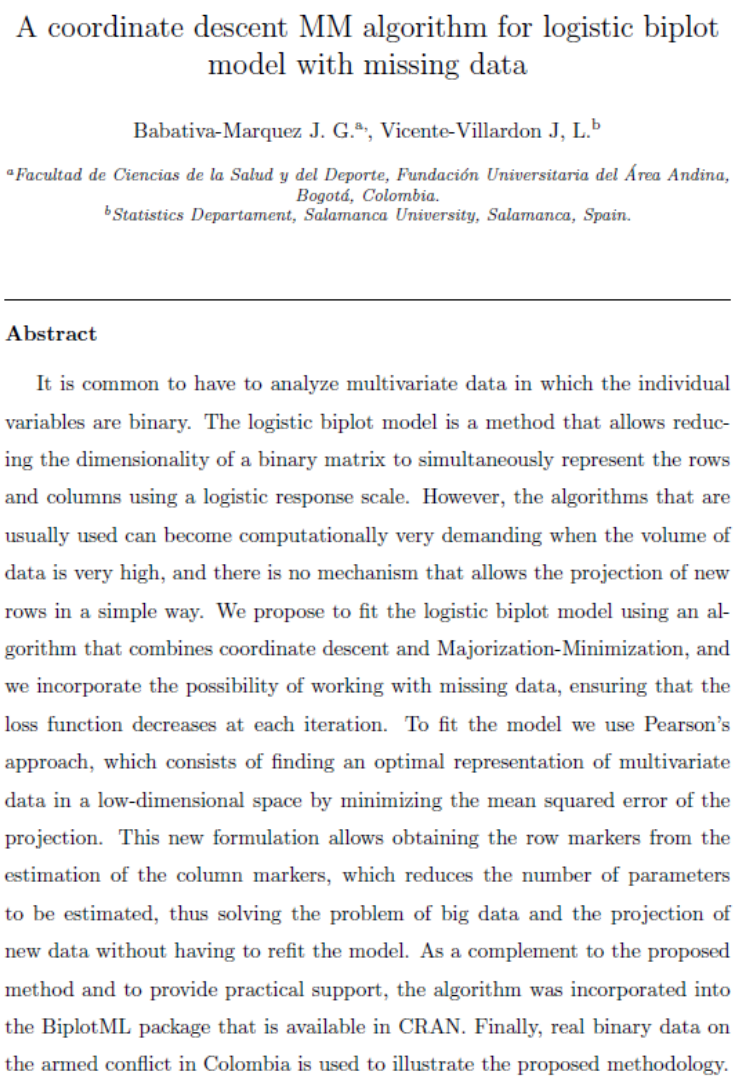
4.7. Contribuciones realizadas en este capítulo

El modelo biplot logístico es un método muy útil para describir las relaciones subyacentes en una matriz binaria. Sin embargo, con los algoritmos actuales, el número de parámetros a estimar se incrementa cuando se aumenta el número de filas de la matriz binaria, lo que representa un problema para grandes volúmenes de datos. Adicionalmente, debido a que cada fila cuenta con sus propios parámetros, no es posible realizar una proyección de nuevas filas como suplementarias, ya que sería necesario aplicar nuevamente el procedimiento de optimización para encontrar sus marcadores, generando así un posible problema de sobreajuste del modelo.

En este capítulo se desarrolló un algoritmo, que además de dar una solución a los problemas anteriores, permite el tratamiento de datos faltantes. Para estimar los parámetros del modelo de biplot logístico se inicia con un procedimiento de mayorización que permite obtener una función objetivo sustituta que acelera la convergencia debido está basada en una función cuadrática del espacio de parámetros, por lo que encontrar el mínimo resulta más simple; en el proceso de optimización se usa un algoritmo de descenso coordinado por bloques donde se introduce la posibilidad de usar matrices con datos faltantes y así minimizar la función mayorizada.

Este nuevo enfoque presenta varias ventajas. Una de ellas es que se evita que el número de parámetros aumente con el número de observaciones, debido a que es posible obtener los marcadores de fila de forma sencilla a partir de la estimación de los parámetros de los marcadores de columna, facilitando también la predicción de los marcadores para nuevas filas, tal y como ocurre en los biplots clásicos. Además, permite trabajar con matrices que tienen datos faltantes, generando una salida con la matriz imputada. Esto convierte a esta metodología en una poderosa herramienta para trabajar con grandes volúmenes de datos. Para ilustrar la metodología se usaron datos reales sobre el conflicto armado en Colombia.

un resumen de los resultados y hallagos más relevantes de este capítulo han sido sometidos a la revista *Annual Review of Statistics and Its Application* (JCR 5.81 Q1)



A coordinate descent MM algorithm for logistic biplot model with missing data

Babativa-Marquez J. G.[a], Vicente-Villardon J, L.[b]

[a]*Facultad de Ciencias de la Salud y del Deporte, Fundación Universitaria del Área Andina, Bogotá, Colombia.*
[b]*Statistics Departament, Salamanca University, Salamanca, Spain.*

Abstract

It is common to have to analyze multivariate data in which the individual variables are binary. The logistic biplot model is a method that allows reducing the dimensionality of a binary matrix to simultaneously represent the rows and columns using a logistic response scale. However, the algorithms that are usually used can become computationally very demanding when the volume of data is very high, and there is no mechanism that allows the projection of new rows in a simple way. We propose to fit the logistic biplot model using an algorithm that combines coordinate descent and Majorization-Minimization, and we incorporate the possibility of working with missing data, ensuring that the loss function decreases at each iteration. To fit the model we use Pearson's approach, which consists of finding an optimal representation of multivariate data in a low-dimensional space by minimizing the mean squared error of the projection. This new formulation allows obtaining the row markers from the estimation of the column markers, which reduces the number of parameters to be estimated, thus solving the problem of big data and the projection of new data without having to refit the model. As a complement to the proposed method and to provide practical support, the algorithm was incorporated into the BiplotML package that is available in CRAN. Finally, real binary data on the armed conflict in Colombia is used to illustrate the proposed methodology.

Figura 4.3: Artículo sometido a la revista *Annual Review of Statistics and Its Application* - JCR 2020: 5.81 Q1; SJR 2021: 3.1 Q1.

Capítulo 5

Paquete BiplotML

5.1. Introducción

En este capítulo se presenta una introducción al uso del paquete `BiplotML` J.G (2022), que es un paquete escrito en lenguaje R Core Team (2022) y se encuentra disponible en el repositorio *Comprehensive R Archive Network* (CRAN) en el enlace https://cran.r-project.org/web/packages/BiplotML/index.html. El objetivo de este desarrollo es dar un soporte práctico a los métodos propuestos en esta tesis y algunas extensiones que podrán ser implementadas posteriormente.

El primer paso consiste en instalar el paquete, para ello se puede usar el comando `install.packages("BiplotML")` o desde las ventanas se puede buscar el paquete, tal como se presenta en la Figura 5.1.

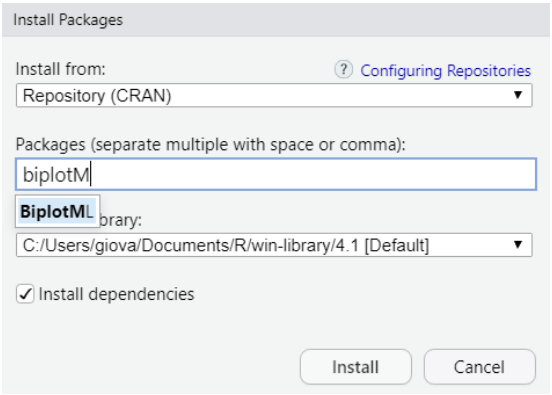


Figura 5.1: Instalación del paquete BiplotML desde el repositorio de CRAN.

Una vez instalado, el paquete se carga con `library(BiplotML)`. Las funciones cuentan con

la documentación requerida por CRAN, así que para acceder a la ayuda se usa el comando tradicional `help()` o `?`, donde se encontrará la descripción de la función, la definición de los argumentos, referencias y un ejemplo de uso. Por ejemplo, el comando `?cv_LogBip` devuelve:

```
## Cross-Validation for logistic biplot
##
## Description:
##
##      This function run cross-validation for logistic biplot
##
## Usage:
##
##      cv_LogBip(
##        data,
##        k = 0:5,
##        K = 7,
##        method = "MM",
##        type = NULL,
##        plot = TRUE,
##        maxit = NULL
##      )
##
## Arguments:
##
##     data: Binary matrix.
##
##        k: Dimensions to analyze. By default 'k = 1:3'.
##
##        K: folds. By default 'K = 7'.
##
##   method: Method to be used to estimate the parameters. By default '
##           method="MM"'
##
##     type: For the conjugate-gradients method. Takes value 1 for the
##           Fletcher-Reeves update, 2 for Polak-Ribiere and 3 for
##           Beale-Sorenson.
##
##     plot: draw the graph. By default 'plot=TRUE'
##
##    maxit: The maximum number of iterations. Defaults to 100 for the
##           gradient methods, and 2000 for MM algorithm.
##
## Value:
##
```

```
##      Training error and generalization error for a logistic biplot
##      model.
##
## Author(s):
##
##      Giovany Babativa <gbabativam@gmail.com>
##
## References:
##
##      Bro R and Kjeldahl K and Smilde AK. (2008). Cross-validation of
##      component models: a critical look at current methods. Analytical
##      and bioanalytical chemistry. 390(5):1241-1251
##
##      Wold S. (1978). Cross-validatory estimation of the number of
##      components in factor and principal components models.
##      Technometrics. 20(4):397-405.
##
## See Also:
##
##      'LogBip, pred_LB, fitted_LB, simBin'
##
## Examples:
##
##      set.seed(1234)
##      x <- simBin(n = 100, p = 50, k = 3, D = 0.5, C = 20)
##      # cross-validation with coordinate descendent MM algorithm
##      cv_MM <- cv_LogBip(data = x$X, k=0:5, method = "MM", maxit = 1000)
##
##      # cross-validation with CG Fletcher-Reeves algorithm
##      cv_CG <- cv_LogBip(data = x$X, k=0:5, method = "CG", type = 1)
##
##      # cross-validation with projection data and block coordinate descending algorithm
##      cv_PB <- cv_LogBip(data = x$X, k=0:5, method = "PDLB", maxit = 1000)
```

5.2. Métodos implementados

El paquete permite mostrar resultados tanto numéricos como gráficos para todos los métodos descritos en este trabajo y está compuesto por 11 funciones principales que le permiten al usuario obtener los resultados para todos los métodos propuestos. Los detalles sobre el uso de las funciones descritas en la Tabla 5.1 se pueden obtener desde la ayuda siguiendo el procedimiento enunciado previamente. Las funciones principales también se apoyan en 8 funciones auxiliares que optimizan el código de programación.

Tabla 5.1: Resumen de las funciones del paquete **BiplotML**.

Función	**Descripción**
cv_logBip	Calcula y grafica los errores de validación cruzada para el modelo LB para un algoritmo predeterminado.
simBin	Simula una matriz binaria de orden $n \times p$ de rango k con un desequilibrio específico.
sdv_MM	Ajusta el modelo LB usando el algoritmo MM-BCD.
proj_LogBip	Ajusta el modelo LB en presencia de datos faltantes usando el método de proyección de datos y un algoritmo MM-BCD.
LogBip	Función que ajusta y genera el biplot del modelo LB basado en un algortimo predefinido.
pred_LB	Predice la matriz binaria y calcula los umbrales óptimos por variable que minimizan la tasa de error equilibrada (TEE).
fitted_LB	Calcula la matriz estimada para el log-odds o la matriz de las probabilidades esperada para un modelo LB.
plotBLB	Gráfica el biplot para un objeto de clase BiplotML.
bootBLB	Aplica una metodología bootstrap para dibujar las elipses de confianza en un modelo LB.
gradientDesc	Ajusta un modelo LB usando el algoritmo del descenso del gradiente.

Para ilustrar el uso del paquete se aplicarán los métodos a los conjuntos de datos del Genomic Determinants of Sensitivity in Cancer 1000 (GDSC1000) de la investigación de Iorio y col. (2016), de donde se pueden extraer diferentes tipos de información sobre líneas celulares de cáncer provenientes de más de 11 mil tumores para 30 tipos de cáncer que integran mutaciones somáticas, copia del número de alteraciones (CNA), metilaciones del ADN y cambios de expresión de génes. Las primeras tres son obtenidas como datos binarios mientras que la expresión genética está medida con variables cuantitativas que son continuas.

El archivo contiene todos los datos unidos en una sola matriz en un formato diferente al requerido y por esta razón fue necesario realizar un preprocesamiento que permitió organizar los datos y adecuarlos para aplicar los métodos. Para facilitar los análisis obtenidos se incluyeron solo tres tipos de cáncer: carcinoma invasivo de mama (BRCA), adenocarcinoma de pulmón (LUAD) y melanoma cutáneo de piel (SKCM). Los datos ordenados pueden ser descargados del repositorio de `GitHub` jgbabativam. A continuación se cargan los conjuntos de datos:

```
load(here::here("data/xMethy.rda"))
load(here::here("data/xMuts.rda"))
load(here::here("data/xCNA.rda"))
```

Luego del preprocesamiento los conjuntos de datos quedan con 160 filas. Los datos de mutación tienen 197 variables, cada variable es un probable gen impulsor o supresor del cáncer, en este caso un gen tiene el código 1 cuando se clasifica como mutado y 0 cuando se clasifica como de tipo salvaje, en este caso, la matriz de datos está muy dispersa y solo el 2.1 % de los genes están mutados. Para los datos de CNA se tienen 412 variables, cada variable es un CNR (copy number region) en un cromosoma, que se etiqueta con 1 cuando la aberración está presente en una muestra y 0 en caso contrario, en este caso el 6.6 % de los datos presentaron aberraciones en la muestra. El conjunto de datos de metilación del ADN tiene 38 variables, cada variable es una isla CpG ubicada en la región promotora de genes, en este caso el valor 1 representa un alto nivel de metilación y 0 un nivel bajo, el 27.1 % de los datos es 1.

5.3. Validación cruzada

En el modelo LB, el vector de parámetros para la fila i está dado por $\boldsymbol{\theta}_i = \boldsymbol{\mu} + \sum_{s=1}^{k} a_{is}\mathbf{b}_s$, $i = 1, \ldots, n$, de modo que el primer paso para especificar el modelo es establecer un valor apropiado para el hiperparámetro k que representa el número de dimensiones. La función `cv_LogBip()` realiza el procedimiento de validación cruzada presentado en el Algoritmo 3. El procedimiento permite elegir el método de estimación, con el argumento `method`, así:

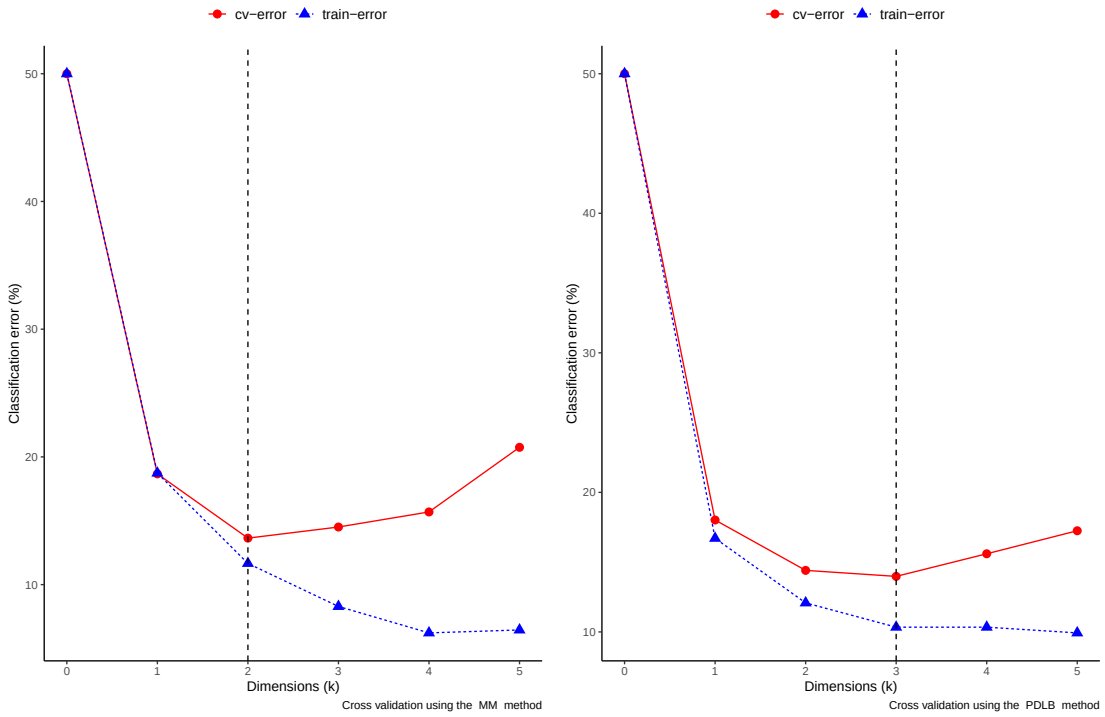
- `method = "MM"` para usar el método MM basado en el algoritmo de descenso coordinado por bloques presentado en el Algoritmo 5.
- `method = "PDLB"` para usar el método de proyección de datos con datos faltantes presentado en el Algoritmo 7.
- `method = 'CG'` con `type = 1` para usar el método basado en gradiente conjugado presentado en el Algoritmo 4 con la formula de actualización de Fletcher-Reeves.
- `method = 'CG'` con `type = 2` para usar el método basado en gradiente conjugado presentado en el Algoritmo 4 con la formula de actualización de Polak–Ribière–Polyak.

- `method = 'CG'` con `type = 3` para usar el método basado en gradiente conjugado presentado en el Algoritmo 4 con la formula de actualización de Hestenes–Stiefel.
- `method = 'CG'` con `type = 4` para usar el método basado en gradiente conjugado presentado en el Algoritmo 4 con la formula de actualización de Dai–Yuan.

Para aplicar el método de validación cruzada sobre el conjunto de datos de metilación se puede usar la función según el modelo que se vaya a aplicar, en este caso se elimina la variable correspondiente al tipo de cáncer para que solo entre la matriz binaria al procedimiento, el ejemplo se hace para el método MM, PDLB y CG con la fórmula de FR, respectivamente:

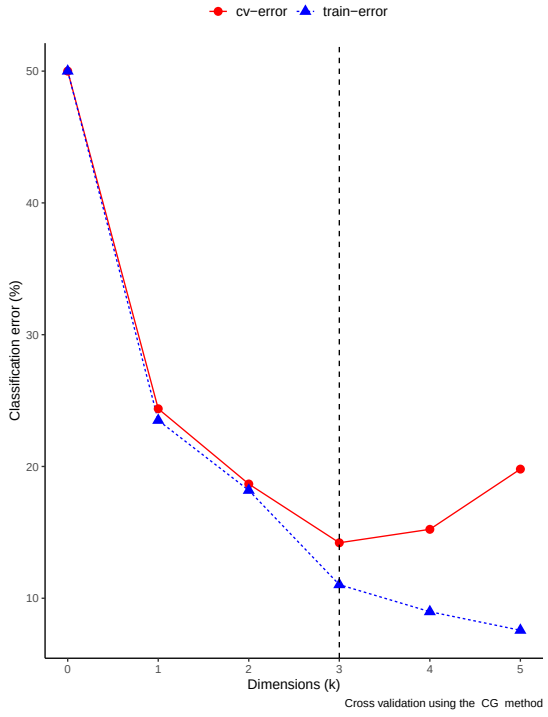
```
binMethy <- xMethy |> select(-`Cancer Type`)

cvMet_MM <- cv_LogBip(data = binMethy, k=0:5, method = "MM")
cvMet_PDLB <- cv_LogBip(data = binMethy, k=0:5, method = "PDLB")
cvMet_CG <- cv_LogBip(data = binMethy, k=0:5, method = "CG", type = 1)
```

Los comandos anteriores generan la gráfica con los errores de entrenamiento y de validación cruzada con cada algoritmo, así mismo contienen un objeto con los resultados numéricos en una tabla. La Figura 5.2 presenta los resultados obtenidos.



(a) MM-BCD

(b) Proyección de datos



(c) Gradiente conjugado – FR

Figura 5.2: Resultados de la validación cruzada para el modelo LB con los datos de metilación usando diferentes algoritmos.

Cuando la matriz de datos es muy dispersa, el procedimiento de *"moteo"* de Wold puede ocasionar que algunas columnas queden con varianza nula debido a que todos los elementos retenidos para una columna son del mismo tipo. Así que es recomendable usar un procedimiento que elimine un porcentaje de datos de cada tipo por cada columna de forma aleatoria, y así evitar este problema. Actualmente se está implementando este procedimiento en el paquete, el cual será agregado con un nuevo argumento `elim = c("Wold", "Prop")`, donde la segunda opción se refiere a la eliminación aleatoria y proporcional por columna.

5.4. Ajuste del modelo de biplot logístico

La función `LogBip` devuelve la estimación de los parámetros del modelo: $\boldsymbol{\mu}$, $\mathbf{A}$ y $\mathbf{B}$ con los algoritmos presentados en este trabajo. Los argumentos de la función son:

- `x`: Matriz de datos binaria.
- `k`: Dimensiones para ajustar el modelo.
- `method`: Es el método de estimación, se puede especificar con `MM`, `CG` o `PDLB`. Para usar el algoritmo de descenso de coordenadas por bloques presentado en el capítulo 3, se usa `MM`, mientras que `CG` es para métodos basados en el gradiente conjugado y `PDLB` es para usar un enfoque por proyección de datos[1]
- `type`: Este argumento es necesario cuando el método de estimación es basado en el gradiente conjugado. Se debe asignar el valor 1 para usar la fórmula de actualización de Fletcher–Reeves, 2 para Polak–Ribière–Polyak, 3 para la fórmula de Hestenes–Stiefel y 4 para Dai–Yuan.
- `plot`: Es un valor lógico, por defecto se realiza la grafica del biplot. Pero el usuario puede especificar `plot = F` para no obtener el gráfico del biplot en la salida.
- `maxit`: Permite especificar el número máximo de iteraciones de los algoritmos. Si la convergencia se logra antes de llegar al máximo, el programa se detiene, en caso de que no se haya logrado la convergencia en el máximo, envía un mensaje de advertencia.
- `endsegm`: Permite especificar el punto en el que termina el segmento para cada variable. Por defecto `endsegm = 0.75`.

[1]Cuando se dectan valores faltantes en la matriz, la función cambia automáticamente el método a *PDLB* y envía un mensaje de advertencia.

- `label.ind`: Permite colocar las etiquetas de las filas en el biplot, por defecto `label.ind = F`.
- `col.ind`: Permite especificar el color para graficar los marcadores fila, este puede ser un valor constante o basado en un factor.
- `draw`: Tipo de gráfico. El usuario puede elegir graficar el biplot, los marcadores de las filas o los marcadores de las columna. `draw = c("biplot", ïnd", "var")`, por defecto `draw = "biplot"`.

El siguiente comando toma como entrada la matriz binaria de metilación, ajusta el modelo con el valor mínimo de k obtenido en el proceso de validación cruzada para el método MM y le asigna un color[2] a los marcadores fila dependiendo del tipo de cáncer. Se ha pedido que no se realice la gráfica del biplot para ilustrar más adelante otras ventajas que pueden ser obtenidas desde el objeto de salida.

```
bipMethy_MM <- LogBip(x = binMethy,
                      k = which.min(cvMet_MM[,2]) - 1,
                      method = "MM",
                      col.ind = xMethy$`Cancer Type`, plot = FALSE)
```

Asimismo, el siguiente comando permite ajustar un modelo LB con el algoritmo de descenso coordinado por bloques para un enfoque por proyección de datos, donde el parámetro k también lo elegimos como el valor mínimo obtenido en el proceso de validación cruzada.

```
bipMethy_PDLB <- LogBip(x = binMethy,
                        k = which.min(cvMet_PDLB[,2]) - 1,
                        method = "PDLB",
                        plot = FALSE)
```

Para los algortimos basados en el gradiente conjugado es necesario especificar la fórmula de actualización

[2]Esto sería útil en caso de que el argumento plot no sea falso, pero se deja acá como ejemplo para el caso en que el gráfico se pida de forma automática

```
bipMethy_CG <- LogBip(x = binMethy,
                      k = which.min(cvMet_CG[,2]) - 1,
                      method = "CG",
                      type = 1,
                      plot = FALSE)
```

Todos los algoritmos generan la estimación de los parámetros del modelo. Por ejemplo, en la Tabla 5.2 se presentan los marcadores para las 10 primeras filas cuando se usa el método basado en la proyección de datos, `head(bipMethy_PDLB$Ahat, n = 10)`

Tabla 5.2: Primeros 10 marcadores fila obtenidos con el método de proyección de datos.

	Dim1	Dim2	Dim3
AU565	-1.7995379	-1.6279459	0.5279644
BT-20	-2.4924546	-2.2425934	0.4657982
BT-474	-2.2667013	-1.9301392	-0.7934985
BT-483	-2.6243481	-2.1437502	-0.7180910
BT-549	-1.9469951	-1.6717388	-0.8741191
CAL-120	-1.9196080	-1.8088814	0.3879724
CAL-148	-2.5981613	-2.0943091	-0.8328592
CAL-51	-1.3917055	-1.1537283	-0.7695153
CAL-85-1	-0.5027182	-0.7602844	-0.8828173
CAMA-1	-2.6243481	-2.1437502	-0.7180910

En el caso de los marcadores de las columnas, la matriz $\mathbf{B}$ contiene en la primera columna el vector de desplazamiento $\boldsymbol{\mu}$ que permite un centrado basado en el modelo, y es denotado con `bb0`. En la Tabla 5.3 se presenta el resultado para los 10 primeros marcadores de las columnas, `head(bipMethy_PDLB$Bhat, n = 10)`.

5.5. Objetos de salida y entorno gráfico

Los objetos de salida de la función `LogBip` son de clase `BiplotML` y de tipo `list`, esta lista contiene la estimación de los parámetros, el método utilizado, la cantidad de iteraciones realizadas. En el caso del método de proyección de datos para un biplot logístico con datos faltantes, contiene la matriz de datos con los valores imputados. A continuación se presenta la estructura del objeto `bipMethy_PDLB`

```
## List of 5
```

Tabla 5.3: Primeros 10 marcadores columna obtenidos con el método de proyección de datos.

	bb0	bb1	bb2	bb3
GSTM1	0.6748397	-0.2131377	0.1600189	-0.0228226
C1orf70	-0.9575073	-0.1959325	-0.1778403	0.0141776
DNM3	-1.1042414	-0.1772024	-0.1538489	0.0030585
COL9A2	-0.9021373	0.2195890	-0.1363315	0.0041814
VAR5	-1.2142879	-0.1675442	-0.1419435	0.0474629
VAR6	-1.2837310	0.1687035	-0.0889283	-0.0094566
THY1	-1.1398131	-0.1774763	-0.1551845	-0.0034279
VAR8	-1.3480539	-0.1575316	-0.1294746	-0.0508979
DNAH10	-0.8809594	0.2229483	-0.1390669	-0.0128365
VAR10	-1.5388415	-0.1356559	-0.0982396	0.0138307

```
##  $ Ahat          :'data.frame':   160 obs. of  2 variables:
##   ..$ Dim1: num [1:160] 37.3 53.9 55.5 86.6 31.3 ...
##   ..$ Dim2: num [1:160] 16.67 21.89 1.08 -0.28 26.17 ...
##  $ Bhat          :'data.frame':   38 obs. of  3 variables:
##   ..$ bb0: num [1:38] -0.265 -5.059 -3.789 -5.301 -5.455 ...
##   ..$ bb1: num [1:38] 0.264 0.221 0.119 -0.168 0.138 ...
##   ..$ bb2: num [1:38] -0.2258 0.1295 0.0545 0.21 0.0722 ...
##  $ method        : chr "coordinate descendent MM"
##  $ loss_function: num [1:482] 0.862 0.383 0.284 0.23 0.202 ...
##  $ iterations    : int 482
##  - attr(*, "class")= chr [1:2] "BiplotML" "list"
```

Para graficar el biplot a partir de la estimación de los marcadores fila y columna, se usa la función `plotBLB`, que ha sido programada en entorno `ggplot2` (Wickham, 2016). De modo que se pueden agregar capas al objeto usando la sintaxis del paquete. Esto tiene la ventaja de que le permite al usuario realizar algunos ajustes de diseño según sus necesidades.

Los siguientes comandos producen el biplot presentado en la Figura 5.3, en este caso se agregó una capa con el título, un subtitulo y una nota al pie. Además, se usa un tema vacío, `theme_void()`, pero hubiera podido ser cualquiera de los que vienen para ser usados en un entorno de `ggplot2`; también se usó sintaxis de `ggplot2` para colocar la leyenda en la parte baja del biplot y asignar un tamaño de letra 10.

```
PlotMet_MM <-  plotBLB(bipMethy_MM, xylim = c(-100,100),
                       col.ind = xMuts$`Cancer Type`)+
               labs(title = "Biplot Logístico",
                    subtitle = "Estimación con el algoritmo MM-BCD",
                    caption = "Gráfico elaborado por Giovany Babativa") +
               theme_void() +
               theme(legend.position = "bottom",
                     legend.title=element_blank(),
                     legend.text = element_text(size = 10))
```

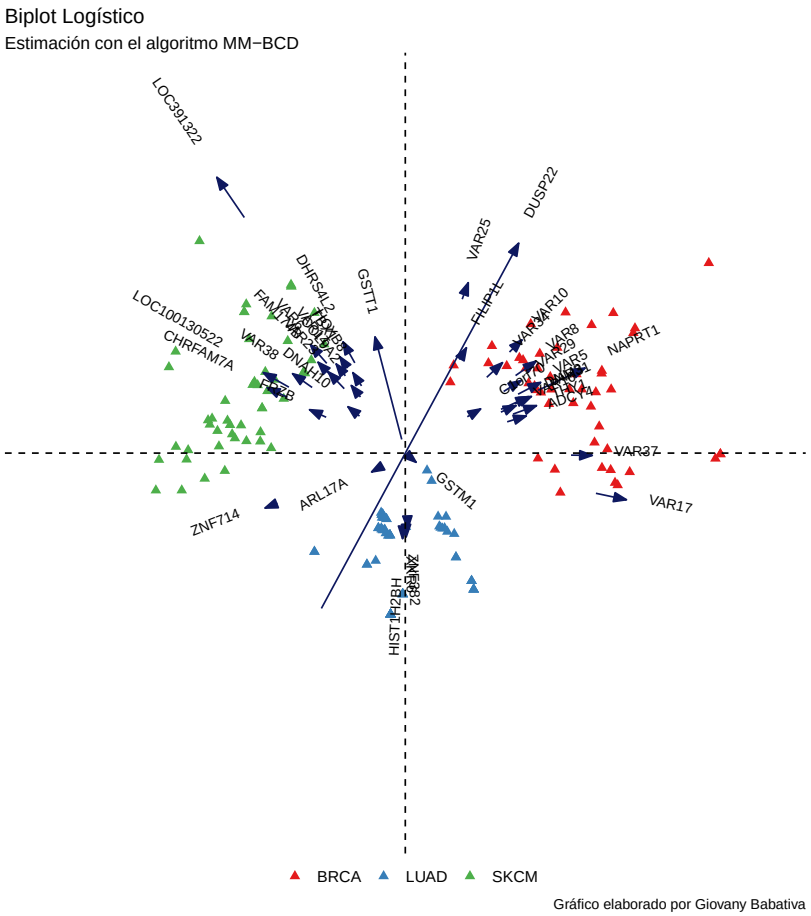


Figura 5.3: Biplot logístico para los datos de metilación usando el método MM-BCD.

El objeto con los parámetros estimados usando el algoritmo de descenso coordinado por bloques con un enfoque de proyección de datos, `bipMethy_PDLB`, es utilizado para ilustrar el uso del entorno gráfico. En este caso se ha decido trazar los marcadores de las filas, así que se usa el argumento `draw = ïnd"` y se han agregado algunas capas para lograr que la leyenda quede en la parte de arriba del gráfico y los nombres de los ejes se han modificado para que queden en español. La Figura 5.4 presenta el resultado obtenido,

una característica relevante a destacar es la forma como el algoritmo logra separar los marcadores en 3 grupos claramente identificados. Es importante señalar que en los casos donde las etiquetas se sobreponen, es posible agregar el argumento `repel = TRUE` en la función `plotBLB()`.

```
marcCol_PDLB <- plotBLB(bipMethy_PDLB, col.ind = xMuts$`Cancer Type`,
                        xylim = c(-3,3), draw = "ind") +
                theme(legend.position = "top") +
                labs(x = "Dimensión 1", y = "Dimensión 2",
                     caption = "")
```

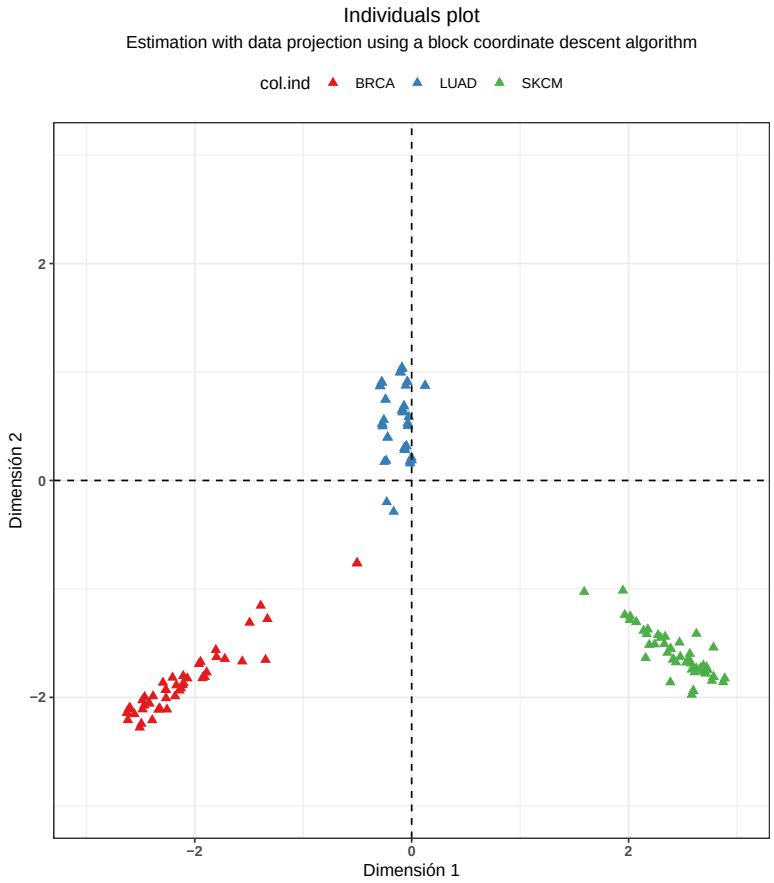


Figura 5.4: Gráfico de los marcadores fila para los datos de metilación usando el método basado en la proyección de datos.

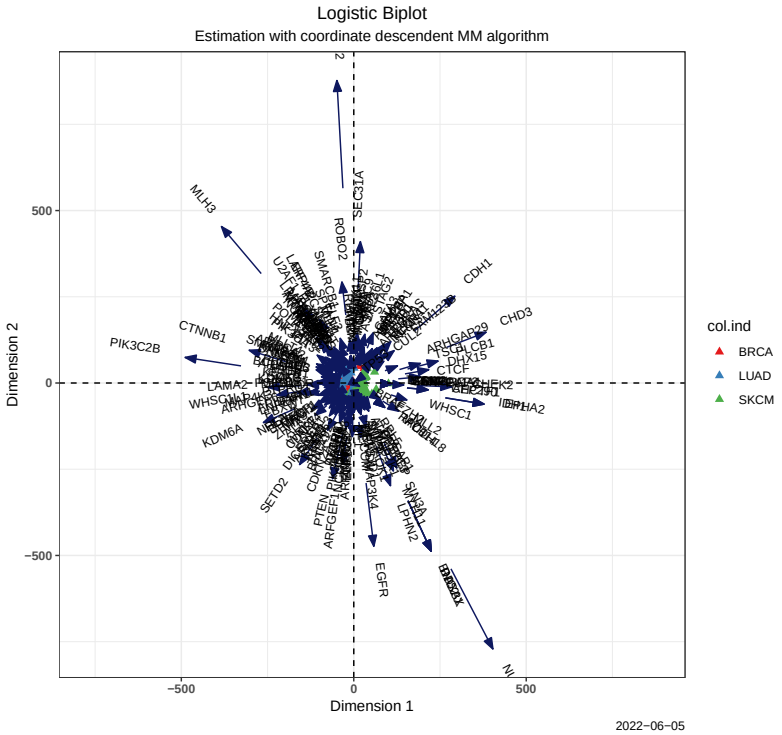
Para ilustar una salida inmediata, se considera el conjunto de datos de mutación. En la siguiente sintaxis se ingresa la matriz de datos binaria, para ello se usa el operador de tubería, `|>`, para eliminar la variable del tipo de cáncer que no hace parte de la matriz de entrada y se especifica que a los marcadores por fila se les asigne un color según el tipo de

cáncer. Los valores estimados se encuentran en el objeto `bipMuts_MM` y el gráfico del biplot se imprime automáticamente. Por defecto se usa el algoritmo MM.

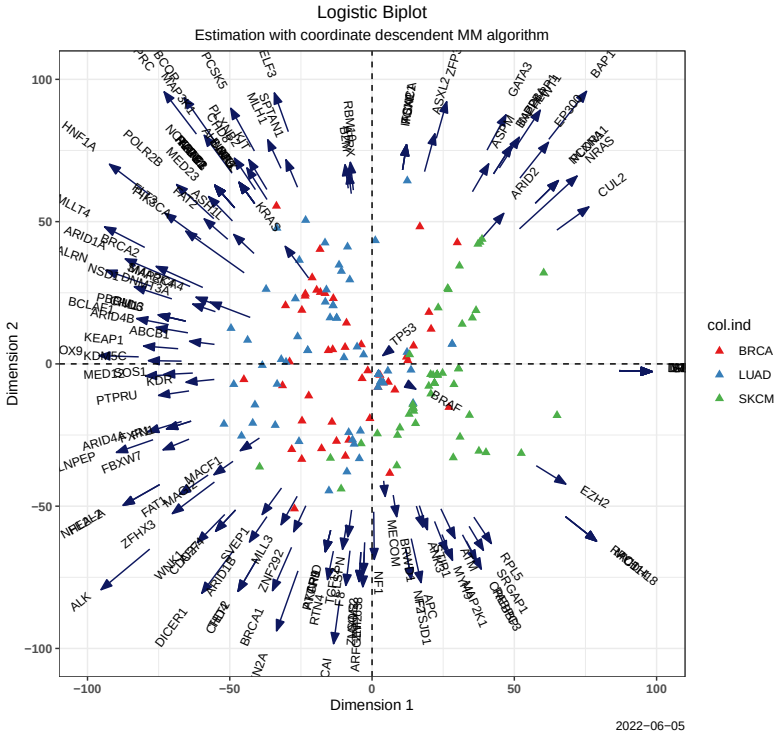
```
bipMuts_MM <- LogBip(x = xMuts |> select(-`Cancer Type`),
                     col.ind = xMuts$`Cancer Type`)
```

El gráfico toma como límites al valor más extremo por encima y por debajo entre los dos ejes. Pero el usuario puede cambiar la escala, acá se usa la función `plotBLB()` con el argumento `xylim = c(-100,100)` para hacerlo. la Figura 5.5 presenta el resultado antes y después de cambiar la escala.

```
plotBLB(bipMuts_MM, xylim = c(-100,100), col.ind = xMuts$`Cancer Type`)
```



(a) biplot obtenido por defecto



(b) biplot cambiando la escala

Figura 5.5: Biplot logístico para los datos de mutación usando el algoritmo MM-BCD.

5.6. Estimación y predicción

Para obtener la estimación del espacio de parámetros: $\boldsymbol{\Theta} = logit\left(\boldsymbol{\Pi}\right) = \mathbf{1}_n\boldsymbol{\mu}^T + \mathbf{AB}^T$, y la matriz de probalidades estimada: $\boldsymbol{\Pi} = \pi\left(\boldsymbol{\Theta}\right)$, se usa la función `fitted_LB()` que tiene dos argumentos, el objeto de entrada que debe ser de clase `BiplotML` y el tipo de retorno que se desea.

Con el argumento *type = c("link", response")* se define la matriz que se desea, *type = "link"* devuelve el espacio de parámetros estimados para el log-odds, $\boldsymbol{\Theta}$. La Tabla 5.4 presenta los resultados para las 10 primeras filas y 7 primeras columnas para los datos de metilación cuando el algoritmo de descenso coordinado por bloques basado en el método MM es utilizado. El código que permite llegar al resultado es:

```
Theta <- fitted_LB(bipMethy_MM, type = "link")
```

Tabla 5.4: Primeras 10 filas y 7 columnas para la matriz del log-odds estimado con el conjunto de datos de metilación usando el algoritmo MM-BCD.

	GSTM1	C1orf70	DNM3	COL9A2	VAR5	VAR6	THY1
AU565	5.819	5.334	1.560	-8.050	0.903	-7.191	1.472
BT-20	9.051	9.696	3.833	-9.753	3.587	-8.326	4.256
BT-474	14.164	7.350	2.887	-14.386	2.302	-11.009	3.363
BT-483	22.681	14.037	6.514	-19.881	6.500	-14.460	7.946
BT-549	2.111	5.257	1.373	-5.063	0.770	-5.417	1.100
CAL-120	7.112	5.412	1.651	-9.081	0.979	-7.805	1.630
CAL-148	6.147	12.556	5.121	-6.802	5.231	-6.657	5.639
CAL-51	13.440	2.952	0.697	-14.721	-0.345	-11.082	0.779
CAL-85-1	-2.198	1.002	-0.890	-2.465	-1.874	-3.754	-1.718
CAMA-1	22.681	14.037	6.514	-19.881	6.500	-14.460	7.946

Para obtener la matriz de probabilidades estimadas: $\boldsymbol{\Pi} = \pi\left(\boldsymbol{\Theta}\right)$, se usa el argumento *type = response"*. La Tabla 5.5 presenta los resultados para las 10 primeras filas y 7 primeras columnas de la matriz de probabilidades estimadas para los datos de metilación cuando se usa el algoritmo de descenso coordinado por bloques basado en el método MM. El siguiente código permite obtener la matriz de probabilidades:

```
Pi <- fitted_LB(bipMethy_MM, type = "response")
```

Tabla 5.5: Primeras 10 filas y 7 columnas para la matriz de probabilidades estimadas con el conjunto de datos de metilación usando el algoritmo MM-BCD.

	GSTM1	C1orf70	DNM3	COL9A2	VAR5	VAR6	THY1
AU565	0.997	0.995	0.826	0.000	0.711	0.001	0.813
BT-20	1.000	1.000	0.979	0.000	0.973	0.000	0.986
BT-474	1.000	0.999	0.947	0.000	0.909	0.000	0.967
BT-483	1.000	1.000	0.999	0.000	0.998	0.000	1.000
BT-549	0.892	0.995	0.798	0.006	0.684	0.004	0.750
CAL-120	0.999	0.996	0.839	0.000	0.727	0.000	0.836
CAL-148	0.998	1.000	0.994	0.001	0.995	0.001	0.996
CAL-51	1.000	0.950	0.668	0.000	0.415	0.000	0.686
CAL-85-1	0.100	0.732	0.291	0.078	0.133	0.023	0.152
CAMA-1	1.000	1.000	0.999	0.000	0.998	0.000	1.000

Para predecir la matriz de datos se usa la función `pred_LB()`, que tiene 3 argumentos. El objeto de entrada que debe ser de clase `BiplotML`, la matriz binaria y el número de cortes para calcular los umbrales δ_j.

Para obtener las matrices predichas para el conjunto de datos de metilación con cada uno de los algoritmos, se puede usar los siguientes comandos:

```
Xpred_Methy_MM <- pred_LB(bipMethy_MM, binMethy, ncuts = 100)
Xpred_Methy_PDLB <- pred_LB(bipMethy_PDLB, binMethy, ncuts = 100)
Xpred_Methy_CG <- pred_LB(bipMethy_CG, binMethy, ncuts = 100)
```

El argumento `ncuts` toma 100 cortes equiespaciados entre 0 y 1 para evaluar el rendimiento que tiene el algoritmo utilizado para hacer la predicción en cada columna, y tomará el valor de $\delta_j, j = 1, \ldots, p$, como el punto donde el valor de la tasa de error equilibrada TEE sea mínima. De esta forma, para la columna j se clasifica a un valor predicho como 1 si la probabilidad estimada es mayor que δ_j, de lo contrario se clasifica en cero. Cuando se usa el método de proyección de datos basado en que la matriz binaria tiene datos faltantes, `PDLB`, la matriz imputada también se obtiene como un objeto de la salida de la función `LogBip()`.

El objeto de salida cuenta con la información del umbral utilizado en la regla de clasificación para cada variable, la matriz predicha, y la tabla con los valores de las medidas de sensibilidad y especificidad para cada variable.

```
## List of 4
##  $ thresholds: tibble [38 x 3] (S3: tbl_df/tbl/data.frame)
##   ..$ variable : chr [1:38] "GSTM1" "C1orf70" "DNM3" "COL9A2" ...
##   ..$ threshold: num [1:38] 0.5556 0.5556 0.0404 0.0808 0.1414 ...
##   ..$ BACC     : num [1:38] 0.862 0 2.542 0.455 3.279 ...
##  $ predictX  : num [1:160, 1:38] 1 1 1 1 1 1 1 1 0 1 ...
##  $ fitted    :'data.frame':  38 obs. of  3 variables:
##   ..$ Sensitivy  : num [1:38] 100 100 100 100 100 100 100 100 100 100 ...
##   ..$ Specificity: num [1:38] 1 1.9 1.9 1.8 1.9 1.8 1.9 1.9 1.8 2 ...
##   ..$ Global     : num [1:38] 99.4 100 96.2 99.4 95 91.2 96.9 94.4 100 95.6 ...
##  $ BACC      : num 4.65
##  - attr(*, "class")= chr [1:2] "BiplotML" "list"
```

5.7. Simulación de matrices binarias

Para evaluar el rendimiento de los algoritmos en este trabajo fue necesario simular matrices binarias y así tener un conocimiento previo de la estructura de la matriz de datos, su dimensionalidad y matriz canónica de parámetros. El algoritmo 6 presentado en el capítulo 3 permitió generar matrices binarias para hacer estas comparaciones. De modo que se consideró relevante incluirlo en el paquete con el fin de facilitar a nuevos usuarios la simulación de matrices binarias de bajo rango.

El algoritmo que permite simular matrices binarias se incorpora en la función `simBin()`. El uso de es relativamente simple, el usuario debe especificar la cantidad de filas de la matriz en el argumento `n`, la cantidad de columnas en el argumento `p`, la dimensionalidad en el argumento `k` y el grado de desequilibrio con el argumento `D`. El argumento `C` permite controlar la escala, por defecto se deja en el valor 1, pero es posible que el usuario deba modificar este valor para obtener un grado de desequilibrio más preciso.

Los siguientes comandos permiten generar una matriz binaria de 100 filas por 50 columnas de rango 3, donde la cantidad de unos y de ceros está equilibrada.

```r
x <- simBin(n = 100, p = 50, k = 3, D = 0.5)
```

El objeto de salida es de clase `list`, que corresponde a una lista de objetos. El usuario podrá encontrar la matriz $\mathbf{X}$ de $n \times p$, la matriz de probabilidades real $\mathbf{\Pi} = \pi(\mathbf{\Theta})$ cuando la función de enlace es de tipo logística y la matriz de parámetros $\mathbf{\Theta}$. También encuentra los valores correspondientes para $\boldsymbol{\mu}$, $\mathbf{A}$ y $\mathbf{B}$, así como los valores para el desequilibrio de la matriz simulada, la cantidad de filas y de columnas.

```
## List of 9
##  $ X    : int [1:100, 1:50] 1 1 1 0 1 1 0 0 1 1 ...
##  $ P    : num [1:100, 1:50] 0.465 0.603 0.457 0.52 0.485 ...
##  $ Theta: num [1:100, 1:50] -0.1384 0.4163 -0.1712 0.0819 -0.0607 ...
##  $ A    : num [1:100, 1:3] -0.1097 -0.667 -1.1954 0.0321 -0.0514 ...
##  $ B    : num [1:50, 1:3] -0.0439 -0.0302 0.0122 -0.127 -0.0675 ...
##  $ mu   : num [1:50] 0 0 0 0 0 0 0 0 0 0 ...
##  $ D    : num 0.496
##  $ n    : num 100
##  $ p    : num 50
```

5.8. Regiones de confianza

Aunque no hace parte de los objetivos de esta investigación. Se ha estado trabajando en la inferencia del modelo LB. En particular, cuando se desea conocer si los elementos que se representan en las filas, que pueden ser sujetos, lugares de muestreo u otro, se diferencian en las características medidas en la matriz $\mathbf{X}$.

Dado que los marcadores están definidos por parámetros que son estimados, tiene sentido pensar en la varianza de la estimación, la cual es necesaria para realizar una inferencia estadística sobre el biplot.

El propósito consiste en poder visualizar la variabilidad en el gráfico del biplot usando elipses de confianza. Se ha trabajado desde un enfoque de bootstrap no paramétrico que ha sido aplicado en el contexto de PCA (Chateau y Lebart, 1996; Chatterjee, 1984; Milan y Whittaker, 1995; Tibshirani y Efron, 1993; Timmerman y col., 2007) y también ha sido utilizado para estimar la variablidad de los estimadores de los parámetros en los métodos biplot (Nieto y col., 2014).

El paquete ya cuenta con la implementación de la metodología para los algoritmos basados en el gradiente conjugado mediante la función `bootBLB()`. Los siguientes comandos pueden ser utilizados para obtener las elipses de confianza de los marcadores fila a partir del conjunto de datos de metilación:

```
out <- bootBLB(x = binMethy, sup = FALSE, ellipses = TRUE, plot = FALSE)
```

El argumento `sup` permite definir el tratamiento para las filas que no son seleccionadas en cada réplica del proceso de remuestreo. Cuando `sup=TRUE`, los marcadores de las filas que no están en cada réplica son proyectadas como filas suplementarias. En este caso no se ha solicitado el gráfico del biplot automáticamente con el fin de modificar la escala de los ejes, así como cambiar los nombres al español.

```
plotBLB(x = out, ellipses = TRUE, xylim = c(-4, 4)) +
   labs(x = "Dimensión 1", y = "Dimensión 2",
        caption = "", title = "", subtitle = "")
```

La Figura 5.6 presenta el resultado obtenido. La interpretación puede llegar a resultar compleja cuando el número de filas es muy alto, pero en este caso permite observar las diferencias entre varias filas.

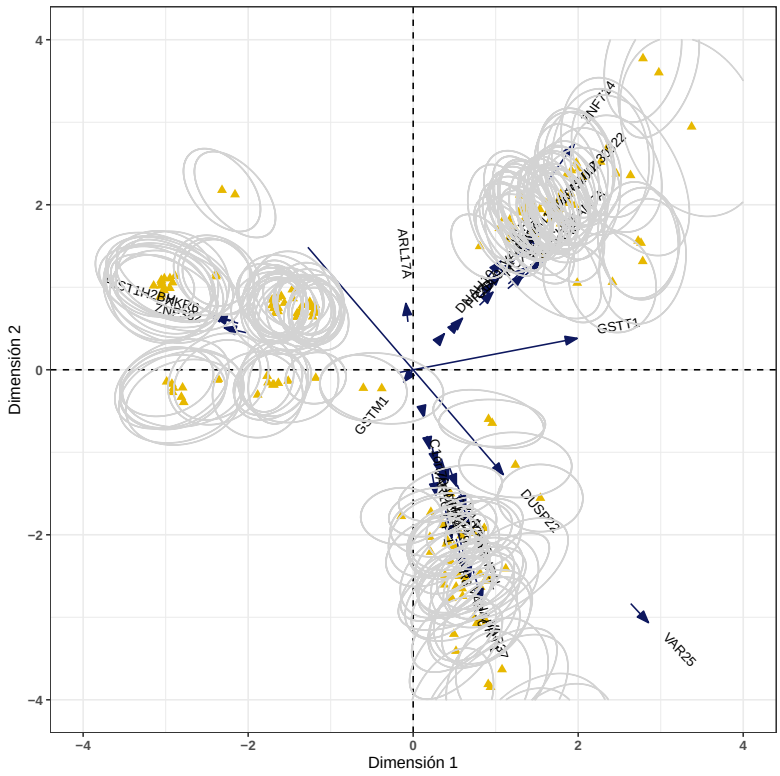


Figura 5.6: Elipses de confianza usando el algoritmo basado en el gradiente conjugado.

5.9. Contribuciones realizadas en este capítulo

El paquete `BiplotML` es producto de la investigación realizada en este trabajo y le permite a cualquier usuario aplcar los diferentes métodos y algoritmos desarrollados desde un entorno y flujo de trabajo de una forma simple.

El paquete en versión estable se encuentra en el repositorio de CRAN para que pueda ser instalado en cualquier computadora. Además, la versión en desarrollo puede ser encontrada en el repositorio de GitHub: https://github.com/jgbabativam/BiplotML, el cual va contando con nuevas funciones de prueba que se encuentran en desarrollo, y que puede ser instalado de la siguiente forma:

```
install.packages("devtools")
devtools::install_github("jgbabativam/BiplotML")
```

Actualmente se está finalizando el artículo: *"BiplotML: An R package for Logistic Biplot*

Model using Machine Learning Algorithms.", que será sometido a *"The R Journal":* https://journal.r-project.org/.

Capítulo 6

Conclusiones y discusión

Los avances tecnológicos han permitido que las formas de recolectar información en la actualidad sean más simples y rápidas, y cada vez nos vemos más enfrentados a diferentes problemas de análisis de datos a partir de grandes volúmenes información. Esto no solo nos brinda la oportunidad de tener una comprensión más profunda de los patrones o relaciones que pueden revelar los datos, sino que también introduce nuevos desafíos estadísticos. Algunos de estos están vinculados con el rendimiento que tienen los algoritmos de estimación para llegar a una solución práctica en un método específico.

En el capítulo 2 planteó una formulación general para los métodos biplot, esto permitió plantear el problema desde un enfoque probabilístico para llegar a funciones de pérdida adecuadas basado en los factores de normalización de la familia exponencial. Este trabajo se enfocó en el análisis para matrices binarias, así que se usó el factor de normalización de una distribución Bernoulli, pero la formulación puede ser utilizada para extender los métodos biplot con el fin de encontrar funciones de pérdida adecuadas para otros tipos de datos dependiendo de la función de distribución de la familia exponencial.

En el mismo capítulo se postuló y se demostró el Teorema 2, que permite sustituir la función de pérdida de un biplot logístico por otra que tiene la propiedad de estar basada en una función cuadrática. Esta nueva formulación está basada en una función más sencilla que facilita la aplicación de algoritmos más eficientes para ajustar el modelo, lo que representa una gran ventaja especialmente para grandes volúmenes de datos.

Adicionalmente, en el capítulo 2 se propuso una metodología para evaluar el modelo y poder estimar el número de dimensiones que son necesarias para realizar el ajuste. Para ello

se desarrolló un algoritmo que permite identificar de una forma objetiva el número de ejes a retener y que se basa en un procedimiento de validación cruzada para datos multivariantes.

En el capítulo 3 se adaptó el algoritmo del gradiente conjugado para ajustar un biplot logístico, para ello se pueden usar 4 formulas diferentes que permiten actualizar la dirección basada en el gradiente en cada iteración. Asimismo, se desarrolló una metodología para usar un algoritmo de descenso coordinado por bloques a partir de la función sustituta, basado en la metodología MM[1], con lo cual se propuso y se implementó el algoritmo para la estimación del espacio de parámetros de un modelo de biplot logístico. El rendimiento para los 5 algoritmos fue estudiado usando métodos de Monte Carlo, donde se consideraron matrices de diferentes dimensiones con una estructura de rango 3 y diferentes grados de desequilibrio, midiendo la capacidad que tienen para encontrar el número correcto de dimensiones y la habilidad para recuperar la estructura real de la matriz de parámetros. Finalmente, las metodologías propuestas fueron aplicadas a un conjunto de datos real sobre metilaciones del ADN para tres tipos de cáncer: carcinoma invasivo de mama (BRCA), adenocarcinoma de pulmón (LUAD) y melanoma cutáneo de piel (SKCM). A partir del desarrollo de este capítulo se puede concluir que:

1. El desarrollo teórico permitió proponer cinco algoritmos iterativos para ajustar un biplot logístico, que cuentan con la propiedad de que la función de pérdida decrece con cada iteración.

2. El rendimiento de los algoritmos basados en el gradiente conjugado, así como el algoritmo basado en el descenso coordinado por bloques a partir de la función sustituta no se ven afectados por el grado de desequilibrio de la matriz de datos.

3. Elegir un número de dimensiones superior al óptimo, genera un incremento en el error cuadrático medio relativo en la estimación de la matriz de parámetros, especialmente para el algoritmo basado en el método MM, aunque, para un valor fijo de p, las brechas se cierran en la medida que el número de filas se aumenta.

4. Para mantener un control del error cuadrático medio, resulta de gran relevancia una

[1]Se le conoce así porque es un procedimiento en dos pasos. La primera M es el paso de mayorización para identificar una función sustituta, y la segunda M es por el paso donde se aplica algún algoritmo que permita minimizar la función

apropiada elección del hiperparámetro k, que representa el número de dimensiones en el modelo LB, donde la metodología basada en la validación cruzada propuesta en este trabajo para el modelo LB mostró ser exitosa.

5. Los métodos basados en los algoritmos presentados son una contribución importante porque brindan alternativas que permiten resolver algunos problemas que se pueden presentar cuando se tiene un volumen alto de datos o cuando la matriz de datos está desequilibrada.

6. El desempeño computacional de los algoritmos fue satisfactorio. Se pudo observar que el método MM de descenso coordinado por bloques obtenido desde la función sustituta funcionó mejor, especialmente cuando n y p aumentaron, lo que resultó en un rendimiento que fue hasta seis veces más rápido que los algoritmos basados en el gradiente conjugado en escenarios equilibrados.

En el capitulo 4.1, se propuso y se desarrolló la teoría para un modelo de biplot logístico cuando se tienen datos faltantes. La propuesta parte de cambiar de formular el problema de minimización en un biplot logístico desde otra perspectiva que está basada en el enfoque propuesto por Pearson para un PCA, que consiste en buscar la solución como una proyección de los datos sobre un subespacio de baja dimensión minimizando el error cuadrático medio de la representación de dimensión k. El problema de minimización se formula a partir de la función de pérdida solo para los valores conocidos en la matriz de datos y se lleva a cabo un proceso de doble mayorización, que es soportado por la postulación del Teorema 2, mencionado previamente, y del Teorema 3, postulado y demostrado en este capítulo. Para la nueva función sustituta se desarrolla y se implementa un algoritmo basado en el descenso coordinado por bloques, que a partir de una matriz simétrica permite usar la descomposición espectral y llegar a una solución para ajustar los parámetros del modelo. La metodología propuesta fue aplicada a un conjunto real de datos incompleto sobre el conflicto armado en Colombia. A partir del desarrollo de este capítulo se puede concluir que:

1. El método propuesto permite que la matriz de datos de entrada tenga valores faltantes y como resultado del proceso de ajuste del modelo de biplot logístico se puede obtener la matriz con los valores imputados.

2. Como resultado de la metodología propuesta, la matriz de marcadores fila puede ser obtenida a partir de la matriz de marcadores columna, tal y como ocurre en el biplot clásico. Esto reduce la cantidad de parámetros a estimar en comparación con otras metodologías. Lo que representa una gran ventaja cuando se manejan grandes volúmenes de información a nivel de individuos o filas.

3. Esta formulación permite hacer la proyección de nuevas filas como suplementarias sin tener que volver a ajustar todo el modelo. Esto es posible gracias al resultado mencionado en el punto anterior.

Las implementaciones informáticas son parte fundamental del desarrollo académico, especialmente en contextos como el de este trabajo. Estas le permiten al usuario poder utilizar los diferentes desarrollos sin tener que entrar a la complejidad de los algoritmos. El problema es que, en general, se requiere de un tiempo importante para su desarrollo. En este trabajo se buscó dar un soporte práctico a los métodos propuestos, no solo desarrollando los códigos de programación, sino buscando la manera más eficiente de escribir los algoritmos, tratando de implementar metodologías de programación que garanticen un buen desempeño computacional. Al ponerlo a disposición de los usuarios, se espera que los métodos propuestos puedan ser utilizados en diferentes áreas del conocimiento, proporcionando funciones donde se pueden cambiar los argumentos según las necesidades de cada usuario y generando objetos gráficos de salida en entornos que puedan ser complementados por otros paquetes cuando se quieran realizar modificaciones. Todos los procedimientos, metodologías y algoritmos propuestos en el desarrollo de este trabajo fueron organizados en un paquete de R denominado *BiplotML*, y de esta forma facilitar el uso de los métodos en la práctica. En el capítulo 5 se presenta una introducción al manejo, así como las salidas del paquete. Algunos elementos a destacar son:

1. El paquete se encuentra publicado en el repositorio de CRAN, por lo que puede ser instalado en cualquier ordenador.

2. Todas las funciones del paquete fueron documentadas siguiendo las normas de CRAN, las cuales pueden ser consultadas en la ayuda de cada función o en el repositorio https://cran.r-project.org/web/packages/BiplotML/BiplotML.pdf.

3. La implementación de los algoritmos basados en el gradiente conjugado, el procedimiento MM y el método de ajuste por proyección de datos presentados en este trabajo están implementados en el paquete dentro de una función, donde el usuario solo debe especificar su matriz binaria y el método que desea implementar.

4. Los métodos para llevar a cabo el proceso de validación cruzada y el cálculo de errores de entrenamiento son implementados con una salida gráfica que le indica al usuario el número de dimensiones que debe utilizar para ajustar el modelo.

5. Se agregaron diferentes argumentos en las funciones para generar una mejor experiencia. El usuario puede elegir el color para los marcadores fila y columna, además tiene la opción de que los marcadores fila se representen con un color diferente dependiendo de una variable cualitativa, entre otras. Los objetos gráficos son elaborados en entorno *ggplot2*, de modo que los usuarios pueden agregar capas que le permiten tener un diseño del gráfico según su gusto y necesidad.

6. En el paquete se encuentra una metodología de bootstrap no paramétrico para realizar la inferencia de los resultados obtenidos con un biplot logístico binario, incluyendo un paso que usa un procedimiento procrustes para minimizar el efecto de las reflexiones o rotaciones generadas por las diferentes configuraciones y que puede distorsionar la varianza de los estimadores.

Capítulo 7

Líneas futuras de investigación

A partir de los métodos investigados y de los resultados obtenidos en este trabajo se pueden continuar o abrir otras líneas que pueden ser investigadas en el marco de los métodos biplot.

1. Aunque el patrón de eliminación de Wold resultó muy eficiente para realizar el procedimiento de validación cruzada, existen otros enfoques que podrían llegar a ser ensayados. Un método de validación bi-cruzada (BCV), que consiste en omitir simultáneamente una fila y columna fue propuesto por Gabriel (2002), y luego fue generalizado por Owen, Perry y col. (2009) a modelos de factorización de matrices no negativas (NMF) y, por Fu y Perry (2020) para la elección del número de clústers en K-medias. El proceso de validación bi-cruzada consiste en tener un patrón que, al reorganizar las filas y columnas, divide la matriz de datos en cuatro bloques, donde un bloque de la diagonal es observado y el otro corresponde a los datos eliminados. La metodología consiste en usar el bloque diagonal observado como conjunto de entrenamiento para ajustar el modelo y luego usar el modelo para predecir el bloque no observado (conjunto de prueba) basado en los bloques fuera de la diagonal.

2. Considerar penalizaciones para construir un modelo sparse dependiendo del tipo de datos, el problema puede ser fomulado como:

$$\min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} \mathcal{L}^*\left(\boldsymbol{\Theta}\right) = \min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} \sum_{i=1}^{n} G(\boldsymbol{\mu} + \mathbf{B}\mathbf{a}_i) - tr\left((\mathbf{1}_n\boldsymbol{\mu}^T + \mathbf{A}\mathbf{B}^T)\mathbf{X}^T\right) + P(\lambda). \tag{7.1}$$

En el caso del biplot logístico la función a minimizar podría ser expresada como

$$\min_{\boldsymbol{\mu};\mathbf{A};\mathbf{B}} \mathcal{L}\left(\boldsymbol{\Theta}\right) + \lambda h(\boldsymbol{\Theta}) \tag{7.2}$$

donde λ es un parámetro de regularización. Basado en el método MM, se puede mayorizar la función de pérdida y luego usar un algoritmo de minimización. El problema puede ser abordado mayorizando de forma separada $\mathcal{L}\left(\boldsymbol{\Theta}\right)$ y $h(\boldsymbol{\Theta})$. De esta forma el Teorema 2 o el Teorema 3 pueden ser utilizados como la función de mayorización de $\mathcal{L}\left(\boldsymbol{\Theta}\right)$, y elegir una función de penalización que se pueda mayorizar de una forma simple.

3. Basado en la metodología desarrollada en el capítulo 4, es posible formular un método análogo al biplot dinámico propuesto por Egido y Galindo (2015), para un arreglo binario de tres vías. De esta forma se ajustaría el modelo de biplot logístico para una de las ocasiones y las demás se proyectarían utilizando la estimación obtenida para los marcadores de las columnas. La interpretación de las distancias en la escala del biplot logístico, así como las demás propiedades obtenidas en el biplot dinámico serían parte de la investigación.

4. Utilizando la metodología propuesta y desarrollada en el capítulo 4, se podría adaptar el modelo de X-STATIS para el caso en que las matrices sean binarias. De este modo se construye $\mathbf{Z}$ como la matriz que en cada columna contiene el $\text{Vec}(\mathbf{X}_t)$ para $t = 1, \ldots, T$. La matriz $\mathbf{Z}$ puede ser tratada con el método basado en la proyección de datos y se puede dar tratamiento a los valores faltantes. De esta forma se podría construir $\boldsymbol{\Theta}_Z$ y llegar a una matriz de compromiso. Dado que el método de proyección de datos permite proyectar las filas de una matriz como suplementarias, entonces podría ser factible representar los marcadores fila para las matrices iniciales en el espacio del compromiso.

Bibliografía

[1] Mehiddin Al-Baali. «Descent property and global convergence of the Fletcher—Reeves method with inexact line search». En: *IMA Journal of Numerical Analysis* 5.1 (1985), págs. 121-124.

[2] Víctor Amor-Esteban y col. «A multivariate proposal for a national corporate social responsibility practices index (NCSRPI) for international settings». En: *Social Indicators Research* 143.2 (2019), págs. 525-560.

[3] Neculai Andrei. «A hybrid conjugate gradient algorithm for unconstrained optimization as a convex combination of Hestenes-Stiefel and Dai-Yuan». En: *Studies in Informatics and Control* 17.1 (2008), pág. 57.

[4] Jose Giovany Babativa-Márquez y José Luis Vicente-Villardón. «Logistic Biplot by Conjugate Gradient Algorithms and Iterated SVD». En: *Mathematics* 9.16 (2021). ISSN: 2227-7390. DOI: 10.3390/math9162015.

[5] Amir Beck y Dror Pan. «Convergence of an inexact majorization-minimization method for solving a class of composite optimization problems». En: *Large-Scale and Distributed Optimization*. Springer, 2018, págs. 375-410.

[6] Peter J Bickel y Kjell A Doksum. *Mathematical statistics: basic ideas and selected topics, volume I*. Vol. 117. CRC Press, 2015.

[7] Dankmar Böhning y Bruce G Lindsay. «Monotonicity of quadratic-approximation algorithms». En: *Annals of the Institute of Statistical Mathematics* 40.4 (1988), págs. 641-663.

[8] Rasmus Bro y col. «Cross-validation of component models: a critical look at current methods». En: *Analytical and bioanalytical chemistry* 390.5 (2008), págs. 1241-1251.

[9] Emmanuel J Candès y Benjamin Recht. «Exact matrix completion via convex optimization». En: *Foundations of Computational mathematics* 9.6 (2009), págs. 717-772.

[10] Frederic Chateau y Ludovic Lebart. «Assessing sample variability in the visualization techniques related to principal component analysis: bootstrap and alternative simulation methods». En: *COMPSTAT*. Springer. 1996, págs. 205-210.

[11] Sangit Chatterjee. «Variance estimation in factor analysis: An application of the bootstrap». En: *British Journal of Mathematical and Statistical Psychology* 37.2 (1984), págs. 252-262.

[12] Michael Collins y col. «A generalization of principal components analysis to the exponential family». En: *Advances in neural information processing systems*. 2002, págs. 617-624.

[13] Yu-Hong Dai y Yaxiang Yuan. «A nonlinear conjugate gradient method with a strong global convergence property». En: *SIAM Journal on optimization* 10.1 (1999), págs. 177-182.

[14] Yu-Hong Dai y Yaxiang Yuan. «An efficient hybrid conjugate gradient method for unconstrained optimization». En: *Annals of Operations Research* 103.1-4 (2001), págs. 33-47.

[15] Yu-Hong Dai y Yaxiang Yuan. «Convergence properties of the Fletcher-Reeves method». En: *IMA Journal of Numerical Analysis* 16.2 (1996), págs. 155-164.

[16] Jan De Leeuw. «Principal component analysis of binary data by iterated singular value decomposition». En: *Computational Statistics & Data Analysis* 50 (ene. de 2006), págs. 21-39. DOI: 10.1016/j.csda.2004.07.010.

[17] Jan De Leeuw. «The Gifi system of nonlinear multivariate analysis». En: *Data analysis and informatics III* (1984), págs. 415-424.

[18] Jan De Leeuw y Willem J Heiser. «Convergence of correction matrix algorithms for multidimensional scaling». En: *Geometric representations of relational data* 36 (1977), págs. 735-752.

[19] Alvaro R De Pierro. «A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography». En: *IEEE transactions on medical imaging* 14.1 (1995), págs. 132-137.

[20] Jhonny Demey y col. «Identifying molecular markers associated with classification of genotypes by External Logistic Biplots». En: *Bioinformatics (Oxford, England)* 24 (nov. de 2008), págs. 2832-8. DOI: 10.1093/bioinformatics/btn552.

[21] Xiao Liang Dong y col. «A modified Hestenes–Stiefel conjugate gradient method with sufficient descent condition and conjugacy condition». En: *Journal of Computational and Applied Mathematics* 281 (2015), págs. 239-249.

[22] HT Eastment y WJ Krzanowski. «Cross-validatory choice of the number of components from a principal component analysis». En: *Technometrics* 24.1 (1982), págs. 73-77.

[23] Carl Eckart y Gale Young. «The approximation of one matrix by another of lower rank». En: *Psychometrika* 1.3 (1936), págs. 211-218.

[24] Jaime Egido y Purificación Galindo. «Dynamic Biplot. Evolution of the economic freedom in the European Union». En: *British Journal of Applied Science and Technology* 11.3 (2015), págs. 1-13.

[25] Reeves Fletcher y Colin M Reeves. «Function minimization by conjugate gradients». En: *The computer journal* 7.2 (1964), págs. 149-154.

[26] Roger Fletcher y Michael JD Powell. «A rapidly convergent descent method for minimization». En: *The computer journal* 6.2 (1963), págs. 163-168.

[27] Wei Fu y Patrick O Perry. «Estimating the number of clusters using cross-validation». En: *Journal of Computational and Graphical Statistics* 29.1 (2020), págs. 162-173.

[28] Wenjiang J Fu. «Penalized regressions: the bridge versus the lasso». En: *Journal of computational and graphical statistics* 7.3 (1998), págs. 397-416.

[29] K. Ruben Gabriel. «Generalised Bilinear Regression». En: *Biometrika* 85.3 (1998), págs. 689-700. ISSN: 00063444.

[30] K Ruben Gabriel. «Le biplot-outil d'exploration de données multidimensionnelles». En: *Journal de la société française de statistique* 143.3-4 (2002), págs. 5-55.

[31] Karl Ruben Gabriel. «The biplot graphic display of matrices with application to principal component analysis». En: *Biometrika* 58.3 (1971), págs. 453-467.

[32] M Purificación Galindo y Carles M Cuadras. «Una extensión del método Biplot y su relación con otras técnicas». En: *Publicaciones de Bioestadística y Biomatemática* 17 (1986).

[33] Mª Purificación Galindo Villardón. «Una alternativa de representación simultánea: HJ-Biplot». En: *Questiio* 10 (abr. de 1986), págs. 13-23.

[34] Nerea González-García y col. «Multivariate analysis reveals differentially expressed genes among distinct subtypes of diffuse astrocytic gliomas: diagnostic implications». En: *Scientific Reports* 10.1 (2020), págs. 1-12.

[35] John C Gower y David J Hand. *Biplots*. Vol. 54. CRC Press, 1995.

[36] John C Gower y Matthijs J Warrens. «Similarity, dissimilarity, and distance, measures of». En: *Wiley StatsRef: Statistics Reference Online* (2014), págs. 1-11.

[37] John C Gower y col. *Understanding biplots*. John Wiley y Sons, 2011.

[38] Michael Greenacre y Jörg Blasius. *Multiple Correspondence Analysis and related Methods*. Ene. de 2006.

[39] Patrick JF Groenen y col. «Spline-based nonlinear biplots». En: *Advances in Data Analysis and Classification* 9.2 (2015), págs. 219-238.

[40] Yuhong Guo y Dale Schuurmans. «Efficient global optimization for exponential family PCA and low-rank matrix factorization». En: *2008 46th Annual Allerton Conference on Communication, Control, and Computing*. IEEE. 2008, págs. 1100-1107.

[41] Willem J Heiser. «Correspondence analysis with least absolute residuals». En: *Computational Statistics & Data Analysis* 5.4 (1987), págs. 337-356.

[42] Julio César Hernández-Sánchez y José Luis Vicente-Villardón. «Logistic biplot for nominal data». En: *Advances in Data Analysis and Classification* 11.2 (2017), págs. 307-326. DOI: 10.1007/s11634-016-0249-7.

[43] Magnus R Hestenes y Eduard Stiefel. «Methods of Conjugate Gradients for Solving». En: *Journal of research of the National Bureau of Standards* 49.6 (1952), pág. 409.

[44] Peter J Huber. «Robust statistics». En: *International encyclopedia of statistical science*. Springer, 2011, págs. 1248-1251.

[45] David R Hunter y Kenneth Lange. «A tutorial on MM algorithms». En: *The American Statistician* 58.1 (2004), págs. 30-37.

[46] Carla Ijurko y col. «A 29-gene signature associated with NOX2 discriminates acute myeloid leukemia prognosis and survival». En: *American Journal of Hematology* 97.4 (2022), págs. 448-457.

[47] Francesco Iorio y col. «A landscape of pharmacogenomic interactions in cancer». En: *Cell* 166.3 (2016), págs. 740-754.

[48] Gareth James y col. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[49] Babativa J.G. *BiplotML: Biplots Estimation with Machine Learning Algorithms*. 2022.

[50] K.L. Keller. *Strategic Brand Management: Building, Measuring, and Managing Brand Equity*. Pearson/Prentice Hall, 2008. ISBN: 9780132336222.

[51] E Kendal, MS Sayar y col. «The stability of some spring triticale genotypes using biplot analysis». En: *J. Anim. Plant Sci* 26.3 (2016), págs. 754-765.

[52] Henk AL Kiers. «Weighted least squares fitting using ordinary least squares algorithms». En: *Psychometrika* 62.2 (1997), págs. 251-266.

[53] Henk AL Kiers y Jos MF ten Berge. «Minimization of a class of matrix trace functions by means of refined majorization». En: *Psychometrika* 57.3 (1992), págs. 371-382.

[54] Andrew J. Landgraf y Yoonkyung Lee. «Dimensionality reduction for binary data through the projection of natural parameters». En: *Journal of Multivariate Analysis* 180 (2020), pág. 104668. ISSN: 0047-259X. DOI: https://doi.org/10.1016/j.jmva.2020.104668.

[55] Andrew J Landgraf y Yoonkyung Lee. «Generalized principal component analysis: Projection of saturated model parameters». En: *Technometrics* (2019), págs. 1-14.

[56] Kenneth Lange. *MM optimization algorithms*. SIAM, 2016.

[57] Kenneth Lange. *Optimization*. Vol. 95. Springer Science & Business Media, 2013.

[58] Seokho Lee y Jianhua Huang. «A coordinate descent MM algorithm for fast computation of sparse logistic PCA». En: *Computational Statistics & Data Analysis* 62 (jun. de 2013), 26–38. DOI: 10.1016/j.csda.2013.01.001.

[59] Seokho Lee y col. «Sparse logistic principal components analysis for binary data». En: *Ann. Appl. Stat.* 4.3 (sep. de 2010), págs. 1579-1601. DOI: 10.1214/10-AOAS327.

[60] JK Liu y SJ Li. «New hybrid conjugate gradient method for unconstrained optimization». En: *Applied Mathematics and Computation* 245 (2014), págs. 36-43.

[61] Meng Lu y col. «Sparse exponential family Principal Component Analysis». En: *Pattern Recognition* 60 (2016), págs. 681 -691. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2016.05.024.

[62] Rahul Mazumder y col. «Spectral regularization algorithms for learning large incomplete matrices». En: *The Journal of Machine Learning Research* 11 (2010), págs. 2287-2322.

[63] P. McCullagh y J.A. Nelder. *Generalized Linear Models, Second Edition*. Chapman and Hall/CRC Monographs on Statistics and Applied Probability Series. Chapman & Hall, 1989. ISBN: 9780412317606.

[64] Grupo de Memoria Histórica. *¡Basta ya! Colombia: Memorias de guerra y dignidad*. Centro Nacional de Memoria Histórica. Bogotá: Imprenta Nacional, 2013.

[65] L. Milan y J. C. Whittaker. «Application of the parametric bootstrap to models that incorporate a singular value decomposition.» English. En: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 44.1 (1995), págs. 31-49. ISSN: 0035-9254.

[66] Mirjam Moerbeek y Cora Maas. «Optimal Experimental Designs for Multilevel Logistic Models with Two Binary Predictors». En: *Communications in Statistics - Theory and Methods* 34 (mayo de 2005). DOI: 10.1081/STA-200056839.

[67] Mirjam Moerbeek y col. «Optimal Experimental Designs for Multilevel Logistic Models». En: *Journal of the Royal Statistical Society: Series D (The Statistician)* 50 (dic. de 2001), págs. 17 -30. DOI: 10.1111/1467-9884.00257.

[68] Charles I Mosier. «I. Problems and designs of cross-validation 1». En: *Educational and Psychological Measurement* 11.1 (1951), págs. 5-11.

[69] David Murray y col. «Design and Analysis of Group-Randomized Trials: A Review of Recent Methodological Developments». En: *American journal of public health* 94 (abr. de 2004), págs. 423-32. DOI: 10.2105/AJPH.94.3.423.

[70] Hien D Nguyen. «An introduction to Majorization-Minimization algorithms for machine learning and statistical estimation». En: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7.2 (2017), e1198.

[71] Ana B Nieto y col. «A Methodology for Biplots based on bootstrapping with R». En: *Revista colombiana de estadística* 37.2 (2014), págs. 367-397.

[72] Jorge Nocedal y Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[73] Art B Owen, Patrick O Perry y col. «Bi-cross-validation of the SVD and the nonnegative matrix factorization». En: *The annals of applied statistics* 3.2 (2009), págs. 564-594.

[74] Karl Pearson. «LIII. On lines and planes of closest fit to systems of points in space». En: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), págs. 559-572.

[75] Elijah Polak y Gerard Ribiere. «Note sur la convergence de méthodes de directions conjuguées». En: *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique* 3.R1 (1969), págs. 35-43.

[76] Boris T Polyak. «The conjugate gradient method in extremal problems». En: *USSR Computational Mathematics and Mathematical Physics* 9.4 (1969), págs. 94-112.

[77] Michael JD Powell. «Developments of NEWUOA for minimization without derivatives». En: *IMA journal of numerical analysis* 28.4 (2008), págs. 649-664.

[78] Radoslaw Pytlak. *Conjugate gradient algorithms in nonconvex optimization*. Vol. 89. Springer Science & Business Media, 2008.

[79] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2022.

[80] Ben Recht y col. «Factoring nonnegative matrices with linear programs». En: *Advances in neural information processing systems* 25 (2012).

[81] Benjamin Recht y col. «Hogwild!: A lock-free approach to parallelizing stochastic gradient descent». En: *Advances in neural information processing systems* 24 (2011).

[82] Jasson DM Rennie y Nathan Srebro. «Fast maximum margin matrix factorization for collaborative prediction». En: *Proceedings of the 22nd international conference on Machine learning*. 2005, págs. 713-719.

[83] Carlos T dos S. Dias y Wojtek J Krzanowski. «Model selection and cross validation in additive main effect and multiplicative interaction models». En: *Crop Science* 43.3 (2003), págs. 865-873.

[84] Andrew I. Schein y col. «A Generalized Linear Model for Principal Component Analysis of Binary Data». En: *In Proceedings of the 9 th International Workshop on Artificial Intelligence and Statistics*. 2003, pág. 546431.

[85] Luca Scrucca. «Graphical tools for model-based mixture discriminant analysis». En: *Advances in Data Analysis and Classification* 8.2 (2014), págs. 147-165.

[86] Ajit P Singh y Geoffrey J Gordon. «A unified view of matrix factorization models». En: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2008, págs. 358-373.

[87] Ajit P Singh y Geoffrey J Gordon. «A unified view of matrix factorization models». En: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2008, págs. 358-373.

[88] Tanin Sirimongkolkasem y Reza Drikvandi. «On regularisation methods for analysis of high dimensional data». En: *Annals of Data Science* 6.4 (2019), págs. 737-763.

[89] Yipeng Song y col. «Logistic principal component analysis via non-convex singular value thresholding». En: *Chemometrics and Intelligent Laboratory Systems* (2020), pág. 104089.

[90] Nathan Srebro y Tommi Jaakkola. «Weighted low-rank approximations». En: *Proceedings of the 20th international conference on machine learning (ICML-03)*. 2003, págs. 720-727.

[91] Ying Sun y col. «Majorization-minimization algorithms in signal processing, communications, and machine learning». En: *IEEE Transactions on Signal Processing* 65.3 (2016), págs. 794-816.

[92] Robert J Tibshirani y Bradley Efron. «An introduction to the bootstrap». En: *Monographs on statistics and applied probability* 57 (1993), págs. 1-436.

[93] Marieke E Timmerman y col. «Estimating confidence intervals for principal component loadings: a comparison between the bootstrap and asymptotic results». En: *British Journal of Mathematical and Statistical Psychology* 60.2 (2007), págs. 295-314.

[94] Michael E Tipping y Christopher M Bishop. «Probabilistic principal component analysis». En: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999), págs. 611-622.

[95] John W Tukey. «The future of data analysis». En: *The annals of mathematical statistics* 33.1 (1962), págs. 1-67.

[96] Madeleine Udell y col. «Generalized Low Rank Models». En: *Foundations and Trends in Machine Learning* 9 (2016), págs. 1-118. DOI: 10.1561/2200000055.

[97] Digna R Velez y col. «A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction». En: *Genetic Epidemiology: the Official Publication of the International Genetic Epidemiology Society* 31.4 (2007), págs. 306-315.

[98] J.L. Vicente-Villardon y col. «Logistic Biplots». En: *Multiple Correspondence Analysis and related Methods*. Chapman-Hall, 2006. Cap. 23, págs. 503-521. ISBN: 9780470973196.

[99] José L Vicente-Villardón y Julio C Hernández-Sánchez. «External Logistic Biplots for Mixed Types of Data». En: *Advanced Studies in Classification and Data Science*. Springer, 2020, págs. 169-183.

[100] Jose L Vicente-Villardon y Laura Vicente-Gonzalez. «Redundancy Analysis for Binary Data Based on Logistic Responses». En: *Data Analysis and Rationality in a Complex World 16*. Springer International Publishing. 2021, págs. 331-339.

[101] Larry Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

[102] Qiong Wei y Roland L Dunbrack Jr. «The role of balanced training and testing data sets for binary classifiers in bioinformatics». En: *PloS one* 8.7 (2013), e67863.

[103] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN: 978-3-319-24277-4.

[104] Svante Wold. «Cross-validatory estimation of the number of components in factor and principal components models». En: *Technometrics* 20.4 (1978), págs. 397-405.

[105] Svante Wold. «Pattern recognition by means of disjoint principal components models». En: *Pattern recognition* 8.3 (1976), págs. 127-139.

[106] Gonglin Yuan y Maojun Zhang. «A modified Hestenes-Stiefel conjugate gradient algorithm for large-scale optimization». En: *Numerical Functional Analysis and Optimization* 34.8 (2013), págs. 914-937.

[107] Gonglin Yuan y col. «The global convergence of the Polak–Ribiere–Polyak conjugate gradient algorithm under inexact line search for nonconvex functions». En: *Journal of Computational and Applied Mathematics* 362 (2019), págs. 262-275.

[108] Li Zhang y col. «A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence». En: *IMA Journal of Numerical Analysis* 26.4 (2006), págs. 629-640.

Apéndice A

Código para el proceso de Monte Carlo

El proceso de Monte Carlo consideró múltiples escenarios con el fin de evaluar el rendimiento de los algoritmos y calcular el error estándar de las estimaciones. Así que se implementó el siguiente código que utiliza un esquema basado en programación funcional y procesamiento paralelo.

```
library(pacman)
p_load(BiplotML, here, data.table, tidyverse,
       bcv, pracma, MASS, beepr,
       future, furrr, future.apply, parallel)

source(here::here("./R/0. functions.R"))

params <- tribble(
  ~n,  ~p, ~k,   ~D,  ~C,
 100,  50,  3,  0.5,  20,
 300,  50,  3,  0.5,  20,
 500,  50,  3,  0.5,  20,
1000,  50,  3,  0.5,  20,
 100, 100,  3,  0.5,  20,
 300, 100,  3,  0.5,  20,
 500, 100,  3,  0.5,  20,
1000, 100,  3,  0.5,  20,
 100, 200,  3,  0.5,  20,
 300, 200,  3,  0.5,  20,
 500, 200,  3,  0.5,  20,
 100,  50,  3,  0.1,  20,  #Desbalance 0.3
 300,  50,  3,  0.1,  20,  #Desbalance 0.3
 500,  50,  3,  0.1,  20,  #Desbalance 0.3
1000,  50,  3,  0.1,  20,  #Desbalance 0.3
 100, 100,  3, 0.15,  20,  #Desbalance 0.3
```

```
 300, 100,  3, 0.15,  20,  #Desbalance 0.3
 500, 100,  3, 0.15,  20,  #Desbalance 0.3
1000, 100,  3, 0.15,  20,  #Desbalance 0.3
 100, 200,  3,  0.2,  20,
 300, 200,  3,  0.2,  20,
 500, 200,  3,  0.2,  20,
 100,  50,  3, 0.02,  20,  #Desbalance 0.2
 300,  50,  3, 0.02,  20,  #Desbalance 0.2
 500,  50,  3, 0.02,  20,  #Desbalance 0.2
1000,  50,  3, 0.02,  20,  #Desbalance 0.2
 100, 100,  3, 0.055, 20,  #Desbalance 0.2
 300, 100,  3, 0.055, 20,  #Desbalance 0.2
 500, 100,  3, 0.055, 20,  #Desbalance 0.2
1000, 100,  3, 0.055, 20,  #Desbalance 0.2
 100, 200,  3, 0.085, 20,
 300, 200,  3, 0.085, 20,
 500, 200,  3, 0.085, 20,
 100,  50,  3, 0.002, 20,  #Desbalance 0.1
 300,  50,  3, 0.002, 20,  #Desbalance 0.1
 500,  50,  3, 0.002, 20,  #Desbalance 0.1
1000,  50,  3, 0.002, 20,  #Desbalance 0.1
 100, 100,  3, 0.009, 20,  #Desbalance 0.1
 300, 100,  3, 0.009, 20,  #Desbalance 0.1
 500, 100,  3, 0.009, 20,  #Desbalance 0.1
1000, 100,  3, 0.009, 20,  #Desbalance 0.1
 100, 200,  3, 0.025, 20,
 300, 200,  3, 0.025, 20,
 500, 200,  3, 0.025, 20,
)

###-----------------Simular X, \Theta, \Pi.

samples <- function(n, p, k, D, C){
  xs <- simBin(n = n, p = p, k = k, D = D, C = C)
  return(xs)
}

###----------------------------------.
### Monte Carlo

plan(multiprocess)

inicio <- Sys.time()
R <- 30
### Estimo los errores para R remuestras usando programación funcional
sale <- furrr::future_map(1:R, function(x){
```

```
lsamples = params %>% purrr::pmap(samples)
out <- future_mapply(function(x, dimen){
  #--- BFGS Algorithm
  BFGS <- BiplotML::LogBip(x$X, k = dimen, method = "BFGS",
                           plot = FALSE, random_start = FALSE)
  ThetaBFGS <- BiplotML::fitted_LB(BFGS, type = "link")
  predBFGS <- BiplotML::pred_LB(BFGS, x = x$X, ncuts = 50)
  rBFGS <- predBFGS$BACC; rmse_BFGS <- rmse(x$Theta, ThetaBFGS)
  cvBFGS <- crossval(x$X, k = dimen, thres = predBFGS$thresholds,
                     method = "BFGS")
  #--- Fletcher-Reeves Algorithm
  FR <- BiplotML::LogBip(x$X, k = dimen, method = "CG", type = 1,
                         plot = FALSE, random_start = FALSE)
  ThetaFR <- BiplotML::fitted_LB(FR, type = "link")
  predFR <- BiplotML::pred_LB(FR, x = x$X, ncuts = 50)
  fr_CG <- predFR$BACC; rmse_FR <- rmse(x$Theta, ThetaFR)
  cvFR <- crossval(x$X, k = dimen, thres = predFR$thresholds,
                   method = "CG", type = 1)
  #--- Polak--Ribiere Algorithm
  PR <- BiplotML::LogBip(x$X, k = dimen, method = "CG", type = 2,
                         plot = FALSE, random_start = FALSE)
  ThetaPR <- BiplotML::fitted_LB(PR, type = "link")
  predPR <- BiplotML::pred_LB(PR, x = x$X, ncuts = 50)
  pr_CG <- predPR$BACC; rmse_PR <- rmse(x$Theta, ThetaPR)
  cvPR <- crossval(x$X, k = dimen, thres = predPR$thresholds,
                   method = "CG", type = 2)
  #--- Beale--Sorenson Algorithm
  BS <- BiplotML::LogBip(x$X, k = dimen, method = "CG", type = 3,
                         plot = FALSE, random_start = FALSE)
  ThetaBS <- BiplotML::fitted_LB(BS, type = "link")
  predBS <- BiplotML::pred_LB(BS, x = x$X, ncuts = 50)
  bs_CG <- predBS$BACC; rmse_BS <- rmse(x$Theta, ThetaBS)
  cvBS <- crossval(x$X, k = dimen, thres = predBS$thresholds,
                   method = "CG", type = 3)
  #--- Dai--Yuan Algorithm
  DY <- BiplotML::LogBip(x$X, k = dimen, method = "CG", type = 4,
                         plot = FALSE, random_start = FALSE)
  ThetaDY <- BiplotML::fitted_LB(DY, type = "link")
  predDY <- BiplotML::pred_LB(DY, x = x$X, ncuts = 50)
  DY_CG <- predDY$BACC; rmse_DY <- rmse(x$Theta, ThetaDY)
  cvDY <- crossval(x$X, k = dimen, thres = predDY$thresholds,
                   method = "CG", type = 4)
  #--- MM - BCD algorithm
  tMM <- BiplotML::LogBip(x$X, k = dimen, method = "MM",
                          plot = FALSE)
  ThetaMM <- BiplotML::fitted_LB(tMM, type = "link")
```

```
            predMM <- BiplotML::pred_LB(tMM, x = x$X, ncuts = 50)
            CD_MM <- predMM$BACC; rmse_MM <- rmse(x$Theta, ThetaMM)
            cvMM <- crossval(x$X, k = dimen, thres = predMM$thresholds,
                             method = "MM")
            #----- Salida.
            outr <- list(n = x$n, p = x$p, k = dimen,
                         D_t = x$D_t, D_r = round(x$D_r, 2),
                         BFGS = rBFGS, fr_CG = fr_CG, pr_CG = pr_CG,
                         bs_CG=bs_CG, DY_CG = DY_CG, MM = CD_MM,
                         rmse_BFGS = rmse_BFGS, rmse_FR = rmse_FR,
                         rmse_PR = rmse_PR, rmse_BS = rmse_BS,
                         rmse_DY = rmse_DY, rmse_MM = rmse_MM,
                         cvTBFGS = cvBFGS$cvT, cvTFR = cvFR$cvT,
                         cvTPR = cvPR$cvT, cvTBS = cvBS$cvT,
                         cvTDY = cvDY$cvT, cvTMM = cvMM$cvT,
                         cvDBFGS = cvBFGS$cvD, cvDFR = cvFR$cvD,
                         cvDPR = cvPR$cvD, cvDBS = cvBS$cvD,
                         cvDDY = cvDY$cvD, cvDMM = cvMM$cvD)
            return(outr)
                 }, lsamples, 1:5)
        }, .progress = TRUE)



resulta <- as.data.frame(matrix(unlist(sale), ncol=29, byrow=TRUE))
colnames(resulta) <-
        c("n", "p", "k", "desb.teorico", "desb.real",
          "BFGS", "CG_Fletcher", "CG_Polak", "CG_Beale", "CG_DY", "MM",
          "RMSE_BFGS", "RMSE_FR", "RMSE_PR", "RMSE_BS", "RMSE_DY", "RMSE_MM",
          "cvT.BFGS", "cvT.FR", "cvT.PR", "cvT.BS", "cvT.DY", "cvT.MM",
          "cvD.BFGS", "cvD.FR", "cvD.PR", "cvD.BS", "cvD.DY", "cvD.MM")

#save(resulta, file=here::here("data/results.rda"))
fin <- Sys.time()
fin - inicio
beepr::beep(8)

###---- Medidas finales de precisión y error
resumen <- resulta %>% group_by(n, p, k, desb.teorico) %>%
           summarise_all(mean) %>% ungroup()

resumen %>%
  dplyr::select(n, p, k, desb.teorico, desb.real, starts_with("cvD")) %>%
  pivot_longer(-c("n", "p", "k", "desb.teorico", "desb.real"),
                 names_to = "Algoritmo", values_to = "Error") %>%
  ggplot(aes(x = k, y = Error, group = Algoritmo)) +
```

```
  geom_line(aes(linetype=Algoritmo, color=Algoritmo)) +
  geom_point(aes(color=Algoritmo))+
  scale_y_continuous(breaks = seq(0,100,5)) +
  scale_x_continuous(breaks = seq(0, 5, 1)) +
  labs(y = "BACC (%)", x = "k") + theme_bw() +
  facet_grid( n ~ p)

####---- Fin
```