

Aprendizaje en modelos lineales multi-respuesta en altas dimensiones con búsqueda cuántica no oracular (cuando el rango de la matriz de coeficientes es conocido)

Learning in High-Dimensional Multi-response Linear Models with Quantum non-Oracular Search (when the range of the coefficient matrix is known)

Víctor Hugo Hernández Castillo

Universidad Nacional Autónoma de México, Ciudad de México, MÉXICO
ORCID: 0009-0004-5107-6918 | vichernancs@ciencias.unam.mx

Recibido 25/03/2025, aceptado 01/06/2025.

Resumen

El presente trabajo estudia *el modelo lineal multivariado de dimensión alta y rango bajo*, este nos proporciona información relevante de una muestra dada, con la ayuda de un *algoritmo de la computación cuántica híbrido* (combina la computación cuántica y clásica, para resolver un problema). El método estima cada una de las entradas de la matriz de coeficientes identificando las columnas linealmente independientes, además de estimar la matriz de errores. Daremos un ejemplo concreto cuando *el rango de la matriz de coeficientes es conocido*. Cuando el rango no es conocido, tenemos un problema NP-hard con solución. La obtención de la matriz de coeficientes se obtiene a través de un algoritmo de búsqueda cuántica no oracular (NQS, non oracular quantum search algorithm), vea [1], [2], en su etapa inicial se debe de realizar una búsqueda combinatoria sobre todos los posibles rangos, después de ello sobre todas las posibles distribuciones de las columnas linealmente independientes en la matriz.

En nuestra investigación realizamos parcialmente la ejecución del mencionado algoritmo cuántico, sin embargo, mostramos a detalle la parte crítica del mismo, que consiste en reducir a la función de pérdida L_n definida en [1], [2], a un *problema de optimización cuadrática multivariada*, además de optimizarla, esta etapa es un punto de partida para aplicar completamente el algoritmo. Mostramos una sola optimización de todas las que se deberán realizar a través de una búsqueda combinatoria especificada en [1], [2]. La última etapa del algoritmo NQS, consiste en elaborar una sucesión creciente a partir de los parámetros de cada optimización mencionadas sustituidos en la función de pérdida L_n , trabajo que está no incluido en este artículo y que corresponde a un problema NP-hard (cuando el rango de la matriz no es conocido). La comprensión de nuestro trabajo, permitirá a distintas áreas de estudio que posean datos multivariados reales poder realizar predicciones como la demografía, medicina, etc.

Palabras clave: modelo lineal multi-respuesta, análisis multivariado, algoritmo de la computación cuántica.

Abstract

This paper studies *the high-dimensional, low-rank multivariate linear model*, which provides us with relevant information given a sample, with the help of a *hybrid quantum computing algorithm* (combining quantum and classical computing to solve a problem). The method estimates each of the entries of the coefficient matrix by identifying the linearly independent columns, in addition to estimating the error matrix. We will give a concrete example when *the rank of the coefficient matrix is known*. When the rank is unknown, we have a NP-hard problem with solution. The coefficient matrix is obtained through of a non-oracular quantum search algorithm (NQS), see [1],[2], in its initial stage, a combinatorial search must be performed on all possible ranks, then on all possible distributions of the linearly independent columns in the matrix.

In our research we partially perform the execution of the mentioned quantum algorithm, however, we show in detail the critical part of it, which consists in reducing the loss function L_n defined in [1], [2], to a *multivariate quadratic optimization problem*, furthermore optimizing it, this stage is a starting point for the full implementation of the algorithm. We show only one optimization of all those to be performed through a combinatorial search specified in [1], [2]. The last stage of the NQS algorithm consists in elaborating an increasing sequence from the parameters of each optimization substituted in the loss function L_n , work which is not included in this article and which corresponds to an NP-hard problem (when the rank of the matrix is unknown). The understanding of our work will allow different areas of study with real multivariate data to make predictions such as demography, medicine, etc.

Keywords: linear multi-response model, multivariate analysis, quantum computing algorithm.

I. INTRODUCCIÓN

Una aproximación usada comúnmente para el tratado en los modelos lineales multivariado de dimension alta y rango bajo, se basa en la descomposición de la matriz de coeficientes \mathbf{A} , la cual supongamos que tiene tamaño $p \times q$ con rango r , por ejemplo, de acuerdo al método de descomposición en valores singulares [5], \mathbf{A} se puede descomponer como la multiplicación \mathbf{CDQ} , donde \mathbf{C} es una matriz de tamaño $p \times \hat{r}$, \mathbf{Q} es una matriz de tamaño $\hat{r} \times q$ y \mathbf{D} es una matriz diagonal de tamaño \hat{r} , donde \hat{r} es un estimador de r . De acuerdo al enfoque utilizado para poder estimar a r , se pueden producir distintos estimadores \hat{r} , vea [2], [11]. Estos poseen buenas propiedades asintóticas, sin embargo, cuando no se cumple el supuesto de rango bajo, uno se encuentra con el dilema de crear parámetros desconocidos, para estimar a su vez parámetros desconocidos, lo que implica que la estimación de \mathbf{C} , \mathbf{Q} , y \mathbf{D} envuelve al menos $(p + q)r$ parámetros desconocidos, que son más que los parámetros desconocidos en \mathbf{A} , es decir, $pr + rq = (p + q)r > pq$, vea [1],[2]. Una nueva línea de estudio en [1],[2] proponen estimar tanto a r como a la matriz de coeficientes \mathbf{A} sin el método de descomposición. La intuición es representar a la matriz de coeficiente \mathbf{A} a través de dos tipos de columnas, el primer tipo son las columnas linealmente independientes cuyo número es igual al rango r y el segundo tipo de columnas, si es que existen, se forman como una combinación lineal de las columnas linealmente independientes mencionadas. Estadísticamente este nuevo proceso de aprendizaje es más eficiente que el enfoque de mínimos cuadrados y los enfoques basados en la descomposición, ya que solo se necesitan estimar $(p + q)r - r^2$ parámetros que son menores o iguales a pq , es decir, $(p + q)r - r^2 \leq pq$, desigualdad que se obtiene con mayor claridad a partir de una de las hipótesis del método propuesto en [1],[2], a saber $r \leq p < q$.

Nuestra contribución en el presente artículo recide en haber modelado un ejemplo concreto para el cálculo de la matriz de coeficientes $\hat{\mathbf{A}}$, un estimador de la matriz \mathbf{A} , el cual muestra una de las predicciones citadas en [1], [2], a saber que $\hat{\mathbf{A}}$ es una aproximación muy cercana a \mathbf{A} . Con este cálculo comprendimos que $\hat{\mathbf{A}}$ representa una optimización de una sola instancia de la función de pérdida L_n presente en [1],[2], y que definimos abajo. Este cálculo a su vez nos hizo comprender que L_n modela un problema NP-hard (una clase de problemas computacionales que son al menos tan difíciles, como los problemas más difíciles en complejidad de la clase NP, la clase de problemas que pueden ser resueltos por algoritmos de tiempo polinomial no determinista,[8]). La obtención de $\hat{\mathbf{A}}$ (explicada a detalla abajo), requiere identificar su rango (que en este caso lo conocemos por hipótesis), la distribución de sus columnas linealmente independientes en $\hat{\mathbf{A}}$, la muestra multivariada, el valor de las columnas linealmente independientes (obtenidas a través de optimizar L_n) y el valor de los coeficientes que forman las combinaciones lineales que a su vez construyen las restantes columnas linealmente dependientes (obtenidos a través de optimizar L_n), en caso de existir. Las limitaciones de nuestro trabajo reciden en el hecho de tan solo representar una instancia de L_n , ya que como comentamos arriba, L_n modela es un problema NP-hard, así que resulta muy difícil plasmar en un artículo, todos los cálculos para explicarlo, sin embargo, hemos sentamos las bases para dar un ejemplo concreto de un problema NP-hard, en algún estudio posterior. Otra aportación de nuestro trabajo es que mejoramos definición de L_n al definirla más clara de entender, con respecto a las definiciones en [1],[2], al incluir en su argumento, cada uno de los parámetros que la definen, en orden.

La ventaja del algoritmo NQS sobre los algoritmos de búsqueda de computadoras electrónicas, se menciona en la siguiente información presente en [1], [2]. Para una búsqueda combinatoria sobre $D = 2^q$ modelos, un algoritmo de computación eléctrica requiere de $O(D)$ consultas para encontrar el modelo objetivo, por ejemplo, la obtención del estimador $\hat{\mathbf{A}}$ es un ejemplo de una consulta, que al explorar todos los posibles estimadores, elegiríamos el mejor de ellos en cuando al rango, de acuerdo a un criterio aclarado en [2], a saber el Criterio de Información Bayesiana (BIC, Bayesian Information Criterion), por otra parte, el algoritmo NQS requiere de $O(\log_2 D)$ iteraciones para converger con una alta probabilidad a los parámetros que nos producen el modelo objetivo.

II. MODELO Y MÉTODO DE ESTIMACIÓN

Establecemos las notaciones que utilizaremos en cada uno de los elementos que conforman el modelo lineal multi-respuesta propuesto en el artículo, "*Learning High Dimensional Multi-response Linear Models with Non-oracular Quantum Search*" [1].

Usamos \mathbf{Y} para denotar al vector de todas las variables respuesta, \mathbf{X} el vector de todas las covariables, así que de ahora en adelante, sin ninguna confusión, llamamos a \mathbf{Y} la variable respuesta, a \mathbf{X} la covariable. Asumimos que \mathbf{Y} es de dimensión $q \times 1$, \mathbf{X} es de dimensión $p \times 1$. Tanto p como q pueden tender a ∞ , cuando el tamaño de la muestra tiende a ∞ . El modelo lineal multivariado de dimensión alta y rango bajo que se aborda en [1], [2], [7] es

$$\mathbf{Y} = \mathbf{A}^\top \mathbf{X} + \boldsymbol{\epsilon} \quad (1)$$

donde \mathbf{A} es una matriz desconocida de tamaño $p \times q$, de rango desconocido r , con $r \leq q < p$, $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_q)^\top$ es un error aleatorio de dimensión $q \times 1$, junto con las siguientes condiciones

$$\mathbb{E}(\boldsymbol{\epsilon} | X) = \mathbf{0}, \quad \text{cov}(\boldsymbol{\epsilon} | X) = \boldsymbol{\Sigma}$$

asumimos que $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}_q$, con σ^2 desconocido.

Cada una de las columnas de la matriz \mathbf{A} puede ser escrita como una combinación lineal de las r columnas que son linealmente independientes de \mathbf{A} . Esta observación motiva la estimación de \mathbf{A} y r , recuperando las columnas linealmente independientes. A continuación explicamos el método de estimación para \mathbf{A} propuesto en el documento [1], en la sección II, "Model and Estimation Method", para el caso cuando r , el rango de la matriz, es conocido. Se muestran cada uno de los conceptos asociados al método, para posteriormente pasemos a mostrar un ejemplo concreto. Se presenta el método de estimación de \mathbf{A} , de acuerdo con [1], [2].

Sea $\{\mathbf{X}_i, \mathbf{Y}_i\}$, $i = 1, \dots, n$, una muestra tomada de $\{\mathbf{X}, \mathbf{Y}\}$, con $\mathbf{X}_i = (x_{i1}, \dots, x_{ip})^\top$ y $\mathbf{Y}_i = (y_{i1}, \dots, y_{iq})^\top$. Los autores proponen la estimación de \mathbf{A} a través de minimizar la siguiente relación:

$$\begin{aligned} & L_n(r; j_1, \dots, j_r, \mathbf{X}_1, \dots, \mathbf{X}_n, \mathbf{Y}_1, \dots, \mathbf{Y}_n, \mathbf{a}_{j_1}, \dots, \mathbf{a}_{j_r}, b_{k1}, \dots, b_{kr}) \\ &= \sum_{i=1}^n \left\{ \sum_{l=1}^r (y_{ij_l} - \mathbf{X}_i^\top \mathbf{a}_{j_l})^2 + \sum_{k \notin \{j_1, \dots, j_r\}} (y_{ik} - \mathbf{X}_i^\top \sum_{l=1}^r b_{kl} \mathbf{a}_{j_l})^2 \right\} + \sum_{l=1}^r P_\lambda(\|\mathbf{a}_{j_l}\|) \end{aligned} \quad (2)$$

con respecto a \mathbf{a}_{j_l} 's, b_{kl} 's y $\{j_1, \dots, j_r\}$, donde $1 \leq j_1 < \dots < j_r \leq q$. Donde r es el rango de la matriz a estimar $\hat{\mathbf{A}}$, j_l representa l -ésima columna linealmente independiente en $\hat{\mathbf{A}}$ con $1 \leq l \leq r$, \mathbf{a}_{j_l} es el vector columna linealmente independiente que forma $\hat{\mathbf{A}}$ y b_{kl} son los coeficientes que forman las combinaciones lineales tomando a los vectores \mathbf{a}_{j_l} 's, los coeficientes que contruyen las columnas linealmente dependientes, en caso de existir, es decir, $\sum_{l=1}^r b_{kl} \mathbf{a}_{j_l}$. La colección de estos estimadores por columnas, denotado como $\hat{\mathbf{A}}$, nos da el estimador de \mathbf{A} .

En el siguiente párrafo presentamos las mismas definiciones tomadas de [1].

Para $l \in \{1, \dots, \hat{r}\}$, \hat{r} denota el minimizador de (2), $\hat{\mathcal{D}} = \{j_1, \dots, j_{\hat{r}}\}$, $\hat{\mathbf{a}}_{j_l}$, $j \in \hat{\mathcal{D}}$ y \hat{b}_{kl} , $k \in \hat{\mathcal{D}}^c$, con $\hat{\mathcal{D}}^c$ el complemento de $\hat{\mathcal{D}}$. Entonces la j -ésima columna de \mathbf{A} es estimada por $\hat{\mathbf{a}}_j$, si $j \in \hat{\mathcal{D}}$ o

$\sum_{l=1}^r \hat{b}_{kl} \hat{a}_{jl}$, if $j \in \hat{\mathcal{D}}^c$. La colección de estos estimadores por columnas, denotado como $\hat{\mathbf{A}}$, nos da el estimador de \mathbf{A} .

Cuando r y $\{j_1, \dots, j_r\}$ están fijos, la minimización de L_n es un problema de optimización cuadrática el cual ha sido estudiado en [1], [2]. Sin embargo, la minimización de L_n con respecto a r y $\{j_1, \dots, j_r\}$ es no convexa y envuelve una búsqueda combinatoria sobre $\sum_{r=1}^q \binom{q}{r} = 2^q - 1$ opciones. Sin una relajación convexa o una aproximación por etapas, la optimización es computacionalmente equivalente a un problema NP-hard (una clase de problemas computacionales que son al menos tan difíciles como los problemas más difíciles en complejidad de la clase NP, la clase de problemas que pueden ser resueltos por algoritmos de tiempo polinomial no deterministas, [8]). La regresión lineal multivariada tiene por objetivo minimizar la función de pérdida L_n citada en la ecuación (2), en nuestro caso lo hacemos aplicando iteradamente el método de la gradiente descendente (por ejemplo, ya que hay varios métodos), que provoca en cada iteración se vaya aproximando a los valores de los parámetros que minimizan a L_n , incluso en algunos casos, los parámetros (coeficientes) llevan a L_n a cero, el mínimo absoluto de la función, ya que la misma es no negativa [7], [9]. A continuación, mostramos un diagrama de flujo del método (véase Fig. 1).

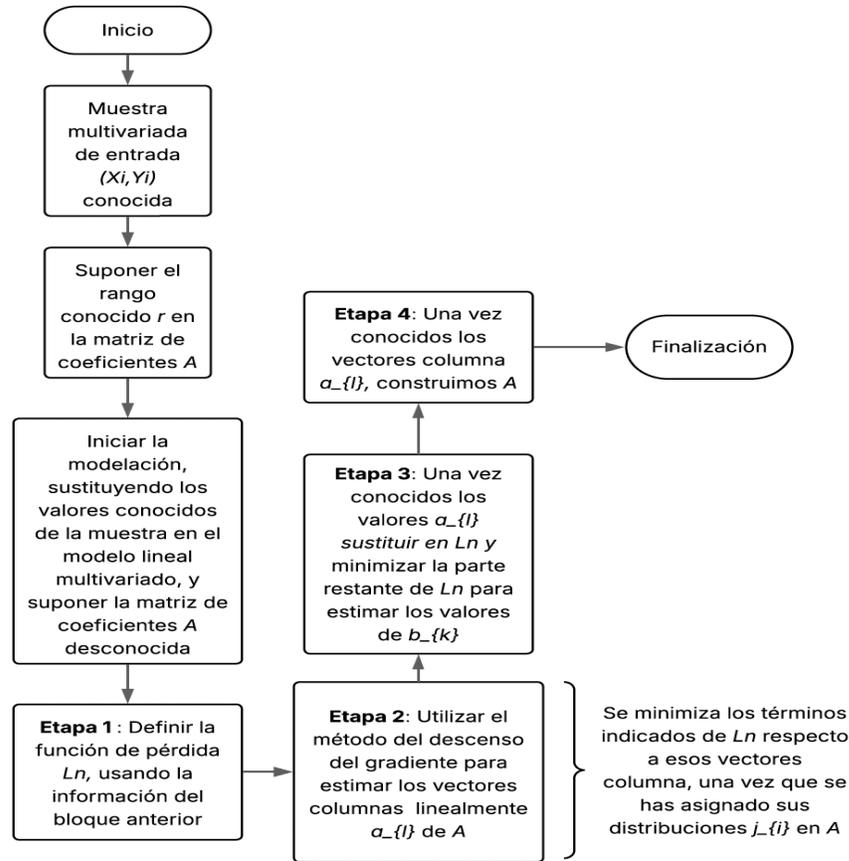


Fig. 1. Diagrama de flujo para la optimización de la función de pérdida L_n .

III. RANGO r DE LA MATRIZ $\widehat{\mathbf{A}}$ ES CONOCIDO

En el siguiente ejemplo ejecutaremos al algoritmo sugerido por los autores de [1], [2], para el caso cuando el rango r de la matriz $\widehat{\mathbf{A}}$ es conocido. Explicaremos a detalle cada uno de los pasos realizados para conseguir el objetivo de encontrar los valores de las entradas que conforman a la matriz en cuestión. Hacemos notar que el ejemplo que presentaremos a continuación es un "problema de optimización cuadrática multivariada", resuelto con ayuda de un algoritmo de la computación cuántica [3], [4], [6], [10] en el siguiente ejemplo concreto.

Ejemplo 1.1. Considere una muestra $\{\mathbf{X}_i, \mathbf{Y}_i\}$, $i=1,2$, tomada de $\{\mathbf{X}, \mathbf{Y}\}$, con los valores de \mathbf{X}_i y \mathbf{Y}_i definidos a continuación:

$$\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2\} \text{ y } \mathbf{Y} = \{\mathbf{Y}_1, \mathbf{Y}_2\}$$

Donde

$$\mathbf{X}_1 = \begin{pmatrix} 3 \\ -1 \\ 4 \\ 2 \\ 1 \end{pmatrix}_{5 \times 1}, \quad \mathbf{X}_2 = \begin{pmatrix} 2 \\ 1 \\ -3 \\ 1 \\ 2 \end{pmatrix}_{5 \times 1}$$

y

$$\mathbf{Y}_1 = \begin{pmatrix} 44 \\ 23 \\ -26 \\ 15 \end{pmatrix}_{4 \times 1}, \quad \mathbf{Y}_2 = \begin{pmatrix} 63 \\ 11 \\ 18 \\ 110 \end{pmatrix}_{4 \times 1}$$

la matriz $\widehat{\mathbf{A}}^\top$ de tamaño 4×5 , que resuelve la modelación de la muestra usando el modelo lineal multivariado y el método propuesto por el artículo [1], es:

$$\widehat{\mathbf{A}}^\top = \begin{pmatrix} 7.977 & 5.751 & -1.960 & 10.636 & 12.387 \\ -0.668 & 3.820 & 2.875 & 5.834 & 5.655 \\ -5.134 & 5.510 & -3.762 & 2.817 & 4.327 \\ -19.082 & 34.608 & -0.339 & 34.631 & 38.952 \end{pmatrix}_{4 \times 5} \quad (3)$$

la cual relaciona a \mathbf{X}_i con \mathbf{Y}_i , $i = 1,2$, a través de las siguientes ecuaciones:

$$\mathbf{Y}_i = \widehat{\mathbf{A}}^\top \mathbf{X}_i + \boldsymbol{\epsilon}_i$$

de acuerdo a los cálculos realizados en RStudio utilizando el método sugerido, se obtiene:

$$\mathbf{Y}_1 = \begin{pmatrix} 44 \\ 23 \\ -26 \\ 15 \end{pmatrix} = \begin{pmatrix} 7.977 & 5.751 & -1.960 & 10.636 & 12.387 \\ -0.668 & 3.820 & 2.875 & 5.834 & 5.655 \\ -5.134 & 5.510 & -3.762 & 2.817 & 4.327 \\ -19.082 & 34.608 & -0.339 & 34.631 & 38.952 \end{pmatrix} \begin{pmatrix} 3 \\ -1 \\ 4 \\ 2 \\ 1 \end{pmatrix} + \begin{pmatrix} 0.001 \\ 0.001 \\ -0.001 \\ 0.000 \end{pmatrix} \quad (4)$$

y

$$Y_2 = \begin{pmatrix} 63 \\ 11 \\ 18 \\ 110 \end{pmatrix} = \begin{pmatrix} 7.977 & 5.751 & -1.960 & 10.636 & 12.387 \\ -0.668 & 3.820 & 2.875 & 5.834 & 5.655 \\ -5.134 & 5.510 & -3.762 & 2.817 & 4.327 \\ -19.082 & 34.608 & -0.339 & 34.631 & 38.952 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ -3 \\ 1 \\ 2 \end{pmatrix} + \begin{pmatrix} 0.005 \\ -0.003 \\ 0.001 \\ 0.000 \end{pmatrix} \quad (5)$$

es decir

$$Y_1 = \begin{pmatrix} 44 \\ 23 \\ -26 \\ 15 \end{pmatrix} = \begin{pmatrix} 43.999 \\ 22.999 \\ -25.999 \\ 15.000 \end{pmatrix} + \begin{pmatrix} 0.001 \\ 0.001 \\ -0.001 \\ 0.000 \end{pmatrix}$$

y

$$Y_2 = \begin{pmatrix} 63 \\ 11 \\ 18 \\ 110 \end{pmatrix} = \begin{pmatrix} 62.995 \\ 11.003 \\ 17.999 \\ 110 \end{pmatrix} + \begin{pmatrix} 0.005 \\ -0.003 \\ 0.001 \\ 0.000 \end{pmatrix}$$

Observe que la matriz \hat{A}^T definida en la ecuación (3), es una pequeña rotación de la matriz A^T definida a continuación:

$$A^T = \begin{pmatrix} 8 & 6 & -2 & 11 & 12 \\ -1 & 4 & 3 & 6 & 6 \\ -5 & 5 & -4 & 3 & 4 \\ -20 & 35 & 0 & 35 & 40 \end{pmatrix} \quad (6)$$

que produce cálculos muy aproximados a los que se encuentran en las ecuaciones (4) y (5). En concreto la matriz A^T ha producido los datos de manera artificial que nos ayudarán a comprobar el método propuesto en [1].

$$Y_1 = \begin{pmatrix} 44 \\ 23 \\ -26 \\ 15 \end{pmatrix} = \begin{pmatrix} 8 & 6 & -2 & 11 & 12 \\ -1 & 4 & 3 & 6 & 6 \\ -5 & 5 & -4 & 3 & 4 \\ -20 & 35 & 0 & 35 & 40 \end{pmatrix} \begin{pmatrix} 3 \\ -1 \\ 4 \\ 2 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

y

$$Y_2 = \begin{pmatrix} 63 \\ 11 \\ 18 \\ 110 \end{pmatrix} = \begin{pmatrix} 8 & 6 & -2 & 11 & 12 \\ -1 & 4 & 3 & 6 & 6 \\ -5 & 5 & -4 & 3 & 4 \\ -20 & 35 & 0 & 35 & 40 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ -3 \\ 1 \\ 2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

una vez establecido un preámbulo de los resultados (la obtención de la matriz \hat{A}^T que en este ejemplo tiene 20 entradas), iniciamos enlistando los pasos generales a seguir, para la aplicación del método sugerido por los doctores Jinyang Chen, Yuan Ke y Cheolwoo Park en [1], para el caso cuando r , el rango de la matriz A es conocido:

Etapla 1: Modelar la muestra de acuerdo al modelo lineal multivariado propuesto en la ecuación (1). Ello incluye la necesidad de modelar todas aquellas incógnitas involucradas en el modelo, hacer uso adecuado de las suposiciones que propone el método, además de tener muy presente cada uno de los parámetros que se obtienen de la muestra, los cuales definen una instancia de la ecuación (2).

Etapa 2: Una vez definida una instancia de la ecuación (2), se han fijado al rango r y al conjunto $\{j_1, \dots, j_r\}$, con estas condiciones iniciales, se minimiza la parte de la función definida en la ecuación (2), que a continuación citamos

$$\sum_{i=1}^n \left\{ \sum_{l=1}^r (y_{ij_l} - \mathbf{X}_i^T \mathbf{a}_{j_l})^2 \right\} + \sum_{l=1}^r P_\lambda(\|\mathbf{a}_{j_l}\|) \quad (7)$$

7

En nuestro caso hemos elegido el *método del descenso del gradiente*, para minimizar los términos indicados en (7), sobre el método sugerido en [2] (estimación estándar por mínimos cuadrados penalizados), ya que los mismos autores enfatizan la no unicidad, del método de minimización seleccionado para (7), con respecto de las \mathbf{a}_{j_l} 's. Una vez encontradas las \mathbf{a}_{j_l} 's que minimicen el término izquierdo de (7), las definiremos como $\mathbf{a}_{j_l}^{(0)}$'s, es claro que la definición de la función de penalización P_λ , influye en la minimización.

Etapa 3: Debemos minimizar la siguiente expresión

$$\sum_{i=1}^n \left\{ \sum_{k \notin \{j_1, \dots, j_r\}} (y_{ik} - \mathbf{X}_i^T \sum_{l=1}^r b_{kl} \mathbf{a}_{j_l}^{(0)})^2 \right\} \quad (8)$$

con respecto a b_{kl} y denotaremos como $b_{kl}^{(0)}$'s aquellos valores que minimicen a (8).

Etapa 4: Considere a $\mathbf{a}_{j_l}^{(0)}$'s y $b_{kl}^{(0)}$'s como valores iniciales y minimice (2). Para definir los términos de la iteración, definimos a $\mathbf{a}_{j_l}^{(i)}$'s y $b_{kl}^{(i)}$'s como las \mathbf{a}_{j_l} 's y b_{kl} 's obtenidas en la i -ésima iteración. Continúe la iteración hasta la convergencia, el límite de las \mathbf{a}_{j_l} 's y b_{kl} 's es el minimizador de (2). Una vez conocidos todos los valores de $\mathbf{a}_{j_l}^{(0)}$'s construimos la matriz $\hat{\mathbf{A}}$ de acuerdo con la distribución asignada de cada uno distinguiendo entre los que son linealmente independientes y los que son linealmente dependientes.

Comenzamos con la explicación detallada de cada una de las etapas.

Etapa 1

La matriz \mathbf{A} que relaciona a los vectores \mathbf{X}_i de tamaño 5×1 , con los vectores \mathbf{Y}_i de tamaño 4×1 , para $i = 1, 2$ respectivamente, a través del modelo lineal multivariado en la ecuación (1), está dada por:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \\ a_{51} & a_{52} & a_{53} & a_{54} \end{pmatrix}_{5 \times 4} \quad (9)$$

supongamos que el rango de \mathbf{A} es conocido. Sea $r = 3$ el rango de la matriz \mathbf{A} en (9), así que el modelo lineal multivariado para la muestra dada, que las relaciona respectivamente, está dado por:

$$\mathbf{Y}_1 = \begin{pmatrix} 44 \\ 23 \\ -26 \\ 15 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{21} & a_{31} & a_{41} & a_{51} \\ a_{12} & a_{22} & a_{32} & a_{42} & a_{52} \\ a_{13} & a_{23} & a_{33} & a_{43} & a_{53} \\ a_{14} & a_{24} & a_{34} & a_{44} & a_{54} \end{pmatrix} \begin{pmatrix} 3 \\ -1 \\ 4 \\ 2 \\ 1 \end{pmatrix} + \begin{pmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{14} \end{pmatrix} = \mathbf{A}^T \mathbf{X}_1 + \boldsymbol{\epsilon}_1$$

y

$$\mathbf{Y}_2 = \begin{pmatrix} 63 \\ 11 \\ 18 \\ 110 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{21} & a_{31} & a_{41} & a_{51} \\ a_{12} & a_{22} & a_{32} & a_{42} & a_{52} \\ a_{13} & a_{23} & a_{33} & a_{43} & a_{53} \\ a_{14} & a_{24} & a_{34} & a_{44} & a_{54} \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ -3 \\ 1 \\ 2 \end{pmatrix} + \begin{pmatrix} \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{23} \\ \epsilon_{24} \end{pmatrix} = \mathbf{A}^T \mathbf{X}_2 + \boldsymbol{\epsilon}_2$$

o bien, en su forma más simple

$$\begin{pmatrix} 44 \\ 23 \\ -26 \\ 15 \end{pmatrix} = \begin{pmatrix} 3a_{11} - a_{21} + 4a_{31} + 2a_{41} + a_{51} + \epsilon_{11} \\ 3a_{12} - a_{22} + 4a_{32} + 2a_{42} + a_{52} + \epsilon_{12} \\ 3a_{13} - a_{23} + 4a_{33} + 2a_{43} + a_{53} + \epsilon_{13} \\ 3a_{14} - a_{24} + 4a_{34} + 2a_{44} + a_{54} + \epsilon_{14} \end{pmatrix}$$

y

$$\begin{pmatrix} 63 \\ 11 \\ 18 \\ 110 \end{pmatrix} = \begin{pmatrix} 2a_{11} + a_{21} - 3a_{31} + a_{41} + 2a_{51} + \epsilon_{21} \\ 2a_{12} + a_{22} - 3a_{32} + a_{42} + 2a_{52} + \epsilon_{22} \\ 2a_{13} + a_{23} - 3a_{33} + a_{43} + 2a_{53} + \epsilon_{23} \\ 2a_{14} + a_{24} - 3a_{34} + a_{44} + 2a_{54} + \epsilon_{24} \end{pmatrix}$$

ahora pasamos a la definición del conjunto

$$\{j_1, j_2, \dots, j_r\}$$

de acuerdo con la información tanto de la muestra, como de la suposición relacionada con el rango de la matriz \mathbf{A} , los valores de los parámetros p y r , son $p = 5$ de acuerdo a la información de la muestra y $r = 3$ de acuerdo a la suposición inicial.

Sean $j_1 = 1$, $j_2 = 2$ y $j_3 = 3$, lo que significa en el método, que por una parte j_i , con $1 \leq i \leq 3$, representa la i -ésima columna linealmente independientes de la matriz \mathbf{A} así que al establecer $j_1 = 1$ se indica que el primer vector linealmente independiente de la matriz \mathbf{A} se encuentra en la primera columna de la misma, $j_2 = 2$ se indica que el segundo vector linealmente independiente de la matriz \mathbf{A} se encuentra en la segunda columna de la misma, y como es de esperar, $j_3 = 3$ se indica que el tercer vector linealmente independiente de la matriz \mathbf{A} se encuentra en la tercera columna de la misma. Así que:

$$\mathcal{D}_r = \{j_1, j_2, \dots, j_r\} = \{j_1, j_2, j_3\} = \{1, 2, 3\}$$

Ahora elegimos la función de penalización que no penaliza a ninguna columna de la matriz de rango completo $\mathbf{A}_{\mathcal{D}_r}$, la submatriz de \mathbf{A} de rango 3 formada por los 3 vectores linealmente independientes \mathbf{a}_{j_l} la definimos a continuación:

$$\mathbf{A}_{\mathcal{D}_r} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \\ a_{51} & a_{52} & a_{53} \end{pmatrix}_{5 \times 3} \quad (10)$$

de (10) es claro que \mathbf{a}_{j_l} con $1 \leq l \leq 3$, es alguna de las 3 columnas de la submatriz $\mathbf{A}_{\mathcal{D}_r}$.

Definimos a la función de penalización que no penaliza a ninguna columna de $\mathbf{A}_{\mathcal{D}_r}$ a continuación:

$$P_\lambda(\|\mathbf{a}_{j_l}\|) = 0, \quad \forall 1 \leq l \leq 3, \quad \text{por tanto,} \quad \sum_{l=1}^3 P_\lambda(\|\mathbf{a}_{j_l}\|) = 0$$

en este punto tenemos la información completa de todos los parámetros que posee la muestra y que a su vez define una instancia de la ecuación (2), como se observa en (11).

$$\begin{aligned} & L_n(3; j_1, j_2, j_3, \mathbf{X}_1, \mathbf{X}_2, \mathbf{Y}_1, \mathbf{Y}_2, \mathbf{a}_{j_1}, \mathbf{a}_{j_2}, \mathbf{a}_{j_3}, b_{41}, b_{42}, b_{43}) \\ &= \sum_{i=1}^2 \left\{ \sum_{l=1}^3 (y_{ij_l} - \mathbf{X}_i^\top \mathbf{a}_{j_l})^2 + \sum_{k=4 \notin \{j_1=1, j_2=2, j_3=3\}} (y_{ik} - \mathbf{X}_i^\top \sum_{l=1}^3 b_{kl} \mathbf{a}_{j_l})^2 \right\} + \sum_{l=1}^3 P_\lambda(\|\mathbf{a}_{j_l}\|) \\ &= \sum_{i=1}^2 \left\{ \sum_{l=1}^3 (y_{ij_l} - \mathbf{X}_i^\top \mathbf{a}_{j_l})^2 + \sum_{k=4 \notin \{j_1=1, j_2=2, j_3=3\}} (y_{ik} - \mathbf{X}_i^\top \sum_{l=1}^3 b_{kl} \mathbf{a}_{j_l})^2 \right\} \quad (11) \\ &= (44 - (3a_{11} - a_{21} + 4a_{31} + 2a_{41} + a_{51}))^2 \quad \text{--- (11.1)} \\ &\quad + (23 - (3a_{12} - a_{22} + 4a_{32} + 2a_{42} + a_{52}))^2 \quad \text{--- (11.2)} \\ &\quad + (-26 - (3a_{13} - a_{23} + 4a_{33} + 2a_{43} + a_{53}))^2 \quad \text{--- (11.3)} \\ &\quad + (63 - (2a_{11} + a_{21} - 3a_{31} + a_{41} + 2a_{51}))^2 \quad \text{--- (11.4)} \\ &\quad + (11 - (2a_{12} + a_{22} - 3a_{32} + a_{42} + 2a_{52}))^2 \quad \text{--- (11.5)} \\ &\quad + (18 - (2a_{13} + a_{23} - 3a_{33} + a_{43} + 2a_{53}))^2 \quad \text{--- (11.6)} \\ &\quad + \sum_{i=1}^2 \sum_{k=4 \notin \{j_1=1, j_2=2, j_3=3\}} (y_{ik} - \mathbf{X}_i^\top \sum_{l=1}^3 b_{kl} \mathbf{a}_{j_l})^2 \quad \text{--- (11.7)} \end{aligned}$$

Se ha terminado con la etapa 1, ahora pasaremos a los cálculos en RStudio que llevan a cabo el método de aproximación elegido para obtener un mínimo local de la expresión matemática en (11).

Etapa 2

En esta etapa buscaremos minimizar a la expresión (11), que en parte está de manera explícita en los términos del (11.1) al (11.6). Es decir, deberemos de encontrar los valores de a_{ij} con $1 \leq i \leq 5$ y $1 \leq j \leq 4$ que minimicen los términos del (11.1) al (11.6). Como lo habíamos mencionado anteriormente, utilizaremos el *método de la gradiente descendente*.

Iniciamos con los cálculos del vector \mathbf{a}_{j_1} , la primer columna de la matriz $\hat{\mathbf{A}}$ que se desea estimar, para ello definimos la función a minimizar, la gradiente de la función con respecto a cada parámetro a estimar, que en este caso, para simplificar la notación en RStudio, establecemos que $a_{11} = a$, $a_{21} = b$, $a_{31} = c$, $a_{41} = d$ y $a_{51} = e$ (haremos una equivalencia semejante para las siguientes estimaciones), establecemos los valores iniciales de la iteración, establecemos el valor de la razón de aprendizaje para el método de minimización elegido, establecemos el número de iteraciones hasta obtener un valor convergente.

Los cálculos fueron realizados en RStudio, y los presentamos para facilitar la comprobación de los resultados obtenidos, así como comprobar la predicción afirmada por la teoría.

A continuación, se presentan en las figuras los códigos en RStudio de las simulaciones correspondientes a los 3 vectores linealmente independientes que la hipótesis considera.

Cálculos para $\mathbf{a}_{j_1}^{(0)}$

```

Y11=44
Y12=63

# Define the function to minimize
f <- function(a, b, c, d, e) {return((Y11 - 3*a + b - 4*c - 2*d - e)^2 +
(Y12 - 2*a - b + 3*c - 1*d - 2*e)^2 )}

# Gradient of the function with respect to each parameter
gradient <- function(a, b, c, d, e) {
  grad_a <- -6*(Y11 - 3*a + b - 4*c - 2*d - e) -4*(Y12 - 2*a - b + 3*c - 1*d - 2*e)
  grad_b <- 2*(Y11 - 3*a + b - 4*c - 2*d - e) -2*(Y12 - 2*a - b + 3*c - 1*d - 2*e)
  grad_c <- -8*(Y11 - 3*a + b - 4*c - 2*d - e) + 6*(Y12 - 2*a - b + 3*c - 1*d - 2*e)
  grad_d <- -4*(Y11 - 3*a + b - 4*c - 2*d - e) -2*(Y12 - 2*a - b + 3*c - 1*d - 2*e)
  grad_e <- -2*(Y11 - 3*a + b - 4*c - 2*d - e) -4*(Y12 - 2*a - b + 3*c - 1*d - 2*e)
  return(c(grad_a, grad_b, grad_c, grad_d, grad_e))
}

# Initial parameter values

a <- -5
b <- 5
c <- -2
d <- 3
e <- 4

# Learning rate
alpha <- 0.01

# Number of iterations
iterations <- 100
    
```

Fig. 2. Cálculos para $\mathbf{a}_{j_1}^{(0)}$.

Se realizaron iteraciones necesarias hasta alcanzar valores de convergencia, que se encuentran al final del código en color azul en la Figura 2.

11

```
# Perform gradient descent
for (i in 1:iterations) {
  # Compute gradient
  grad <- gradient(a, b, c, d, e)

  # Update parameters
  a <- a - alpha * grad[1]
  b <- b - alpha * grad[2]
  c <- c - alpha * grad[3]
  d <- d - alpha * grad[4]
  e <- e - alpha * grad[5]

  # Print current iteration and parameter values
  cat("Iteration:", i, "\n")
  cat("a:", a, " b:", b, " c:", c, " d:", d, " e:", e, "\n")
}

# Print final result
cat("Final result:\n")
cat("a:", a, " b:", b, " c:", c, " d:", d, " e:", e, "\n")

a: 7.977586 b: 5.751724 c: -1.960345 d: 10.63621 e: 12.38793
```

Fig. 3. Iteraciones para valores de convergencia.

Cálculos para $\alpha_{j_2}^{(0)}$

```
Y12=23
Y22=11

# Define the function to minimize
f <- function(a, b, c, d, e) {return((Y21 - 3*a + b - 4*c - 2*d - e)^2 +
(Y22 - 2*a - b + 3*c - 1*d - 2*e)^2 )}

# Gradient of the function with respect to each parameter
gradient <- function(a, b, c, d, e) {
  grad_a <- -6*(Y12 - 3*a + b - 4*c - 2*d - e) -4*(Y22 - 2*a - b + 3*c - 1*d - 2*e)
  grad_b <- 2*(Y12 - 3*a + b - 4*c - 2*d - e) -2*(Y22 - 2*a - b + 3*c - 1*d - 2*e)
  grad_c <- -8*(Y12 - 3*a + b - 4*c - 2*d - e) + 6*(Y22 - 2*a - b + 3*c - 1*d - 2*e)
  grad_d <- -4*(Y12 - 3*a + b - 4*c - 2*d - e) -2*(Y22 - 2*a - b + 3*c - 1*d - 2*e)
  grad_e <- -2*(Y12 - 3*a + b - 4*c - 2*d - e) -4*(Y22 - 2*a - b + 3*c - 1*d - 2*e)
  return(c(grad_a, grad_b, grad_c, grad_d, grad_e))
}

# Initial parameter values

a <- -5
b <- 5
c <- -2
d <- 3
e <- 4

# Learning rate
alpha <- 0.01

# Number of iterations
iterations <- 100
```

Fig. 4. Cálculos para $\alpha_{j_2}^{(0)}$.

Se realizaron iteraciones necesarias hasta alcanzar valores de convergencia, que se encuentran al final del código en color azul en la Figura 4.

```
# Perform gradient descent
for (i in 1:iterations) {
  # Compute gradient
  grad <- gradient(a, b, c, d, e)
  |
  # Update parameters
  a <- a - alpha * grad[1]
  b <- b - alpha * grad[2]
  c <- c - alpha * grad[3]
  d <- d - alpha * grad[4]
  e <- e - alpha * grad[5]

  # Print current iteration and parameter values
  cat("Iteration:", i, "\n")
  cat("a:", a, " b:", b, " c:", c, " d:", d, " e:", e, "\n")
}

# Print final result
cat("Final result:\n")
cat("a:", a, " b:", b, " c:", c, " d:", d, " e:", e, "\n")

a: -0.6689655 b: 3.82069 c: 2.875862 d: 5.834483 e: 5.655172
```

Fig. 5. Iteraciones para valores de convergencia.

Cálculos para $\alpha_{j_3}^{(0)}$

```
Y13=-26
Y23= 18

# Define the function to minimize
f <- function(a, b, c, d, e) {return((Y31 - 3*a + b - 4*c - 2*d - e)^2 +
(Y32 - 2*a - b + 3*c - 1*d - 2*e)^2 )}

# Gradient of the function with respect to each parameter
gradient <- function(a, b, c, d, e) {
  grad_a <- -6*(Y13 - 3*a + b - 4*c - 2*d - e) -4*(Y23 - 2*a - b + 3*c - 1*d - 2*e)
  grad_b <- 2*(Y13 - 3*a + b - 4*c - 2*d - e) -2*(Y23 - 2*a - b + 3*c - 1*d - 2*e)
  grad_c <- -8*(Y13 - 3*a + b - 4*c - 2*d - e) + 6*(Y23 - 2*a - b + 3*c - 1*d - 2*e)
  grad_d <- -4*(Y13 - 3*a + b - 4*c - 2*d - e) -2*(Y23 - 2*a - b + 3*c - 1*d - 2*e)
  grad_e <- -2*(Y13 - 3*a + b - 4*c - 2*d - e) -4*(Y23 - 2*a - b + 3*c - 1*d - 2*e)
  return(c(grad_a, grad_b, grad_c, grad_d, grad_e))
}

# Initial parameter values

a <- -5
b <- 5
c <- -2
d <- 3
e <- 4

# Learning rate
alpha <- 0.01

# Number of iterations
iterations <- 100
```

Fig. 6. Cálculos para $\alpha_{j_3}^{(0)}$.

Se realizaron iteraciones necesarias hasta alcanzar valores de convergencia, que se encuentran al final del código en color azul en la Figura 6.

```
# Perform gradient descent
for (i in 1:iterations) {
  # Compute gradient
  grad <- gradient(a, b, c, d, e)

  # Update parameters
  a <- a - alpha * grad[1]
  b <- b - alpha * grad[2]
  c <- c - alpha * grad[3]
  d <- d - alpha * grad[4]
  e <- e - alpha * grad[5]

  # Print current iteration and parameter values
  cat("Iteration:", i, "\n")
  cat("a:", a, " b:", b, " c:", c, " d:", d, " e:", e, "\n")
}

# Print final result
cat("Final result:\n")
cat("a:", a, " b:", b, " c:", c, " d:", d, " e:", e, "\n")

a: -5.134483 b: 5.510345 c: -3.762069 d: 2.817241 e: 4.327586
```

Fig. 7. Iteraciones para valores de convergencia.

Obsérvese que en las tres iteraciones realizadas, han sido elegidos los mismos valores para los parámetros iniciales. La teoría afirma sobre la libertad de elegir cualesquiera parámetros iniciales, sin embargo, en esta investigación, para entender este método sugerido, nos dimos cuenta que hay parámetros iniciales que aproximan mejor que otros.

Enlistamos las tres columnas estimadas para $\hat{\mathbf{A}}$ correspondientes a la hipótesis inicial del problema analizado, que supone tener un rango igual a 3:

$$\mathbf{a}_{j_1}^{(0)} = \begin{pmatrix} 7.977586 \\ 5.751724 \\ -1.960345 \\ 10.63621 \\ 12.38793 \end{pmatrix}, \mathbf{a}_{j_2}^{(0)} = \begin{pmatrix} -0.6689655 \\ 3.82069 \\ 2.875862 \\ 5.834483 \\ 5.655172 \end{pmatrix}, \mathbf{a}_{j_3}^{(0)} = \begin{pmatrix} -5.134483 \\ 5.510345 \\ -3.762069 \\ 2.817241 \\ 4.327586 \end{pmatrix} \quad (12)$$

Una vez determinados los vectores linealmente independientes de $\hat{\mathbf{A}}$, nos referimos a los vectores $\mathbf{a}_{j_1}^{(0)}$, $\mathbf{a}_{j_2}^{(0)}$ y $\mathbf{a}_{j_3}^{(0)}$, podemos pasar a la siguiente etapa para poder minimizar la expresión (11.7).

Etapa 3

Sustituimos los vectores $\mathbf{a}_{j_1}^{(0)}$, $\mathbf{a}_{j_2}^{(0)}$ y $\mathbf{a}_{j_3}^{(0)}$ en la expresión (11.7) y minimizamos la expresión resultante, a través del hecho de que la función resultante es convexa y así, cualquier punto crítico de la función, resulta ser mínimo local. Una vez encontrados los valores de b_{41} , b_{42} y b_{43} , estamos en condiciones de construir a $\mathbf{a}_{j_4}^{(0)}$, como una combinación lineal de $\mathbf{a}_{j_1}^{(0)}$, $\mathbf{a}_{j_2}^{(0)}$ y $\mathbf{a}_{j_3}^{(0)}$. A continuación, presentamos la función que modela la parte de la muestra relacionada con la expresión (11.7), cuyos parámetros

desconocidos deseamos estimar, se refiere a la que nos permite la estimación de los coeficientes b_{41} , b_{42} y b_{43} , de la combinación lineal afirmada por la hipótesis, es decir

$$\mathbf{a}_{j_4}^{(0)} = b_{41}\mathbf{a}_{j_1}^{(0)} + b_{42}\mathbf{a}_{j_2}^{(0)} + b_{43}\mathbf{a}_{j_3}^{(0)} \quad (13)$$

La función que nos permite estimar b_{41} , b_{42} y b_{43} , a partir de conocer a las cuartas entradas de la muestra, y_{14} , y_{24} , y de los vectores estimados $\mathbf{a}_{j_1}^{(0)}$, $\mathbf{a}_{j_2}^{(0)}$ y $\mathbf{a}_{j_3}^{(0)}$, es

$$f(a, b, c) = (15 - b_{41}(43.999) - b_{42}(22.999) + b_{43}(25.999))^2 + (110 - b_{41}(62.995) - b_{42}(11.003) - b_{43}(17.999))^2 \quad (14)$$

Veamos cómo fue obtenida la mencionada función. Para ello, tal como lo recomiendan los autores del manuscrito [1], estimaremos a $\mathbf{a}_{j_4}^{(0)}$ (la cuarta columna de la matriz $\hat{\mathbf{A}}$ que deseamos estimar) a partir de la expresión en (11.7), junto con los valores de los parámetros que contiene la muestra dada, $n = 2$, el tamaño de la muestra, $k = 4$, la cuarta coordenada de las variables resultantes \mathbf{Y}_1 y \mathbf{Y}_2 , y $r = 3$, el tamaño del rango de la matriz a estimar $\hat{\mathbf{A}}$.

$$\begin{aligned} & \sum_{i=1}^2 \sum_{k=4 \notin \{j_1=1, j_2=2, j_3=3\}} (\mathcal{Y}_{ik} - \mathbf{X}_i^T \sum_{l=1}^r b_{kl})^2 \\ &= \sum_{i=1}^2 \left(\mathcal{Y}_{i4} - \left(\sum_{l=1}^3 b_{4l} \mathbf{X}_i^T \mathbf{a}_{j_l}^{(0)} \right) \right)^2 \\ &= \sum_{i=1}^2 \left(\mathcal{Y}_{i4} - b_{41}(\mathbf{X}_i^T \mathbf{a}_{j_1}^{(0)}) - b_{42}(\mathbf{X}_i^T \mathbf{a}_{j_2}^{(0)}) - b_{43}(\mathbf{X}_i^T \mathbf{a}_{j_3}^{(0)}) \right)^2 \\ &= \left(\mathcal{Y}_{14} - b_{41}(\mathbf{X}_1^T \mathbf{a}_{j_1}^{(0)}) - b_{42}(\mathbf{X}_1^T \mathbf{a}_{j_2}^{(0)}) - b_{43}(\mathbf{X}_1^T \mathbf{a}_{j_3}^{(0)}) \right)^2 \\ &+ \left(\mathcal{Y}_{24} - b_{41}(\mathbf{X}_2^T \mathbf{a}_{j_1}^{(0)}) - b_{42}(\mathbf{X}_2^T \mathbf{a}_{j_2}^{(0)}) - b_{43}(\mathbf{X}_2^T \mathbf{a}_{j_3}^{(0)}) \right)^2 \\ &= (15 - b_{41}(43.999) - b_{42}(22.999) + b_{43}(25.999))^2 \\ &+ (110 - b_{41}(62.995) - b_{42}(11.003) - b_{43}(17.999))^2 \end{aligned}$$

En la investigación realizada para intentar minimizar los términos en la expresión (11.7), ha resultado ser poco efectivo el método de la gradiente descendente, ya que los valores a los que convergen, en el método son valores extremadamente grandes, a saber, $b_{41} = -1.5319 \times 10^{211}$, $b_{42} = -4.6550 \times 10^{210}$ y $b_{43} = 1.9027 \times 10^{210}$, cantidades que resultan ser poco prácticas para nuestros fines, además de estar truncadas. Así que para minimizar los términos en la ecuación (14), utilizaremos el hecho que la misma es una función convexa, así que, si un punto crítico al ser evaluado en la función me da 0 (o virtualmente 0, debido a manejo de valores truncados), nos aseguramos de encontrar un mínimo local.

15

A continuación, los cálculos necesarios para minimizar los términos en la ecuación (14) (véase Fig. 8).

```
# Define the function to minimize
f(a, b, c) = (15 - 43.999*a - 22.999*b + 25.999*c)^2 + (110 - 62.995*a - 11.003*b - 17.999*c)^2

# To find a critical point
grad_a <- -87.998*(15 - 43.999*a - 22.999*b + 25.999*c) - 125.99*(110 - 62.995*a - 11.003*b - 17.999*c) = 0
grad_b <- -45.998*(15 - 43.999*a - 22.999*b + 25.999*c) - 22.006*(110 - 62.995*a - 11.003*b - 17.999*c) = 0
grad_c <- 51.998*(15 - 43.999*a - 22.999*b + 25.999*c) - 35.998*(110 - 62.995*a - 11.003*b - 17.999*c) = 0

# An equivalent system
11808.56*a + 3410.134*b - 20.16599*c = 15178.87
3410.134*a + 1300.04*b - 799.816*c = 3110.63
-20.16599*a - 799.816*b + 1999.824*c = 3179.81

# Solution parameter values
a= 0.071848
b= 4.2217
c=3.27921

# A value almost 0
(15 - 43.999*a - 22.999*b + 25.999*c)^2 + (110 - 62.995*a - 11.003*b - 17.999*c)^2 = 8.695449e-09
=0.000008695449
```

Fig. 8. Cálculos para minimizar los términos de la ecuación (14).

De acuerdo con los cálculos, los valores de los coeficientes que satisfacen la ecuación (14) son

$$a = b_{41}^{(0)} = 0.071848, \quad b = b_{42}^{(0)} = 4.2217 \quad y \quad c = b_{43}^{(0)} = 3.27921$$

y con estos valores estimados, podemos calcular la cuarta columna de la matriz $\hat{\mathbf{A}}$, nos referiremos al vector $\mathbf{a}_{j_4}^{(0)}$,

$$\begin{aligned} \mathbf{a}_{j_4}^{(0)} &= b_{41}^{(0)} \mathbf{a}_{j_1}^{(0)} + b_{42}^{(0)} \mathbf{a}_{j_2}^{(0)} + b_{43}^{(0)} \mathbf{a}_{j_3}^{(0)} \\ &= (0.071848) \begin{pmatrix} 7.977 \\ 5.751 \\ -1.960 \\ 10.636 \\ 12.387 \end{pmatrix} + (4.2217) \begin{pmatrix} -0.668 \\ 3.820 \\ 2.875 \\ 5.834 \\ 5.655 \end{pmatrix} + (3.27921) \begin{pmatrix} -5.134 \\ 5.510 \\ -3.762 \\ 2.817 \\ 4.327 \end{pmatrix} \end{aligned}$$

que al simplificar obtenemos

$$\mathbf{a}_{j_4}^{(0)} = \begin{pmatrix} -19.082 \\ 34.608 \\ -0.339 \\ 34.631 \\ 38.952 \end{pmatrix} \quad (15)$$

Finalmente, con los vectores estimados en la ecuaciones (12) y (15), hemos completado la búsqueda de las 20 incógnitas que conforman la estimación de la matriz $\hat{\mathbf{A}}$ que se aproxima al modelo lineal multivariado en (1), cuya transpuesta ya habíamos presentado en (3).

IV. CONCLUSIONES

Este artículo introduce parcialmente a un algoritmo en computación cuántica no oracular NQS aplicado a un modelo lineal multi-respuesta (el único conocido hasta ahora, de acuerdo a sus autores), aunque no es una innovación la construcción de una matriz de coeficientes en un modelo lineal multivariado a través de obtener las columnas linealmente independientes de la misma, y el resto de columnas construirlas como una combinación lineal de las mencionadas, sí lo es el concepto de *no oracular* dentro de un algoritmo cuántico (por ejemplo el algoritmo de Grover, que tiene la desventaja notable de inicialmente conocer, o bien, suponer conocido el estado solución, dentro de un conjunto de estados cuánticos o bien vectores, situación que les pasa a los algoritmos cuánticos oraculares conocidos, en la vida real usualmente no se dispone de tal información). La implementación completa del algoritmo cuántico NQS resuelve un problema NP-hard.

Nuestra contribución ofrece una alternativa en el tratado de modelos lineales multi-respuesta respecto del tradicional método de descomposición en valores singulares o del método mínimo cuadrados penalizados, nos referimos a la optimización de la función de pérdida L_n . A diferencia de muchos trabajos teóricos, el nuestro implementa detalladamente el proceso de optimización de un modelo complejo utilizando datos realistas calculando de manera explícita a cada uno de sus estimadores, incluyendo un ejemplo de los códigos utilizados para obtenerlos, para impulsar la reproducibilidad y aplicabilidad del método, para científicos o para quienes quisiesen explorar el mismo sin que sea la falta de la comprensión teórica o técnica computacional un factor para su estudio o aplicación. Nuestro trabajo vive en la intersección de disciplinas tales como computación cuántica, optimización computacional, física aplicada, matemáticas aplicadas, ingeniería y estadística; así que la presente investigación puede impactar en una amplia comunidad. Prepara un marco de trabajo para la optimización de un problema NP-hard (cuando el rango de la matriz de coeficientes no es conocido), punto de inicio para futuras exploraciones en el tema. Nuestro documento tiene la didáctica suficiente para guiar al lector desde los supuestos teóricos hasta el uso de práctico de codificación computacional, lo que lo convierte en una herramienta para los investigadores al abordar o comprender modelos similares y para los docentes una herramienta excelente.

Otra aplicación de gran importancia radica en la predicción de valores de una muestra no conocida, a partir de valores de una muestra conocida. Esta aplicación podría ser de gran interés para áreas de estudio poco involucradas con las matemáticas avanzadas, tales como la medicina, psicología, demografía, etc.

Este modelo nos permite manejar variables de manera simultánea, así el análisis para obtener la solución es versátil. Por supuesto que resolver el modelo nos permite predecir las respuestas para muestras de nuestro interés. Las hipótesis que requiere el método propuesto en [1] son muy claras, tales como la independencia lineal de las columnas, el rango y el tamaño de la matriz, etcétera.

En términos generales, el modelo lineal multivariado ofrece muchas ventajas, sin embargo, se debe tener cuidado en su uso, teniendo en cuenta sus alcances, supuestos y limitaciones, para poder aplicarlos adecuadamente.

CRedit (Contributor Roles Taxonomy)

Contribuciones de los autores: Conceptualización / Metodología / Software / Investigación / Redacción y preparación del borrador original / Redacción, revisión y edición / Supervisión / Análisis formal / Administración del proyecto / Adquisición de fondos: **VHHC**.

Financiamiento: El autor declara que este trabajo se realizó con recursos propios.

Declaración de disponibilidad de datos: Los datos se encuentran en el artículo.

Agradecimientos: I would like to express my sincere gratitude to Dr. Jinyang Chen from The University of Georgia Athens and Dr. Yuan Ke from The University of Georgia Athens, for kindly taking the time to answer my query regarding a function presented in their article. Their clarification was essential to ensuring the accuracy and clarity of the dissemination work I developed based on their research.

Conflicto de interés: El autor declara que no existe conflicto de interés.

REFERENCIAS

- [1] J. Chen, C. Park, Y. Ke, “Learning High Dimensional Multi-response Linear Models with Non-oracular Quantum Search,” IEEE International Conference on Quantum Computing and Engineering (QCE), 18-23 September 2022, Broomfield, CO, USA. <https://doi.org/10.1109/QCE53715.2022.00018>
- [2] C. Zou, Y. Ke, W. Zhang, “Estimation of Low Rank High-Dimensional Multivariate Linear Models for Multi-Response Data.” *Journal of the American Statistical Association*, vol. 117, no. 538, pp. 693–703. <https://doi.org/10.1080/01621459.2020.1799813>
- [3] D. McMahon, *Quantum computing explained*, USA: John Wiley & Sons, 2007.
- [4] A. Lopatnikova, T. Minh-Ngoc, S. Sisson, “An introduction to quantum computing for statisticians and data scientists”, 2021, arXiv preprint: https://ui.adsabs.harvard.edu/link_gateway/2021arXiv211206587L/doi:10.48550/arXiv.2112.06587
- [5] A. Lounsbury Andrew, *Singular Value Decomposition*, technical report, Tennessee Technological University, 2018. <https://www.tntech.edu/cas/pdf/math/techreports/TR-2018-2.pdf>
- [6] D. K. Ferry, “Quantum computing and probability,” *Journal of Physics: Condensed Matter*, vol. 21, no. 47, 2009, <https://doi.org/10.1088/0953-8984/21/47/474201>
- [7] X. Bing, M. Wegkamp, “Adaptive estimation of the rank of the coefficient matrix in high-dimensional multivariate response regression models,” *The Annals of Statistics*, JSTOR, 2018. <https://doi.org/10.48550/arXiv.1704.02381>
- [8] W. J. Welch, “Algorithmic complexity: three NP-hard problems in computational statistics,” *Journal of Statistical Computation and Simulation*, vol. 15, no. 1, 1980. <https://doi.org/10.1080/00949658208810560>
- [9] T. W. Anderson, *An introduction to multivariate statistical analysis*, USA: Wiley New York, 1958.
- [10] A. Prakash, *Quantum algorithms for linear algebra and machine learning*, University of California, Berkeley, 2014. <https://escholarship.org/uc/item/5v9535q4>
- [11] T. W. Anderson, “Estimating linear restrictions on regression coefficients for multivariate normal distributions,” *The Annals of Mathematical Statistics*, JSTOR, vol. 22, no. 3, pp. 327-351, 1951. <https://www.jstor.org/stable/2236620>