Artículo Original / Original Research

Ciencias Matemáticas, Vol. 38, No. 2, 2024, Pag. 71-79
Recibido (*Received*): 04-07-2024, Revisado (*Reviewed*): 12-05-2025
Aceptado (*Accepted*): 20-05-2025, En línea (*Online*): 09-08-2025

Estimación de la matriz de covarianzas mediante la matriz de observaciones vectorizada
Estimation of the covariance matrix using the vectorized observation matrix

Jorge Lemagne Pérez*[1]

Resumen En esta comunicación se describen las características del subprograma Estimacov_2, para estimar la matriz de covarianzas correspondiente con una matriz de observaciones vectorizada; el mismo efectúa una partición de las observaciones en grupos, y brinda diferentes opciones al usuario. Dicho subprograma constituye una mejora de su predecesor, Estimacov (ESTImar MAtriz de COVarianzas), en complejidad algorítmica, utilización de los recursos del lenguaje de programación y robustez. Se presentan todos los pasos principales del algoritmo, y se efectúa un análisis de la complejidad. Finalmente, se muestra un ejemplo. El algoritmo está implementado en MATLAB 9.8.

Palabras Clave: algoritmos para estadística, matriz de covarianzas, métodos multivariados.

Abstract *This paper describes the features of the subprogram Estimacov_2, designed to estimate the covariance matrix corresponding to a vectorized observation matrix. The subprogram partitions the observations into groups and offers various options to the user. This subprogram represents an improvement over its predecessor, Estimacov (ESTImate COVariance MAtrix), in terms of algorithmic complexity, programming language resource utilization, and robustness. All the main steps of the algorithm are presented, along with an analysis of its complexity. Finally, an example is provided. The algorithm is implemented in* MATLAB *9.8.*

Keywords: *algorithms for statistics, covariance matrix, multivariate methods.*

Mathematics Subject Classification: 62H12, 65C60, 68W01, 62-04, 15B48.

[1]Departamento de Matemática, Facultad de Matemática y Computación, Universidad de La Habana, La Habana, Cuba. Email: jorgelemagneperez@gmail.com.
***Autor para Correspondencia (*Corresponding Author*)**

Editado por (*Edited by*): Damian Valdés Santiago, Facultad de Matemática y Computación, Universidad de La Habana, La Habana, Cuba.
Maquetado por (*Layout by*): Lázaro Daniel González Martínez, Facultad de Matemática y Computación, Universidad de La Habana, La Habana, Cuba.

Citar como: Lemagne Pérez, J. (2024). Estimación de la matriz de covarianzas mediante la matriz de observaciones vectorizada. *Ciencias Matemáticas*, 38(2), 71-79. DOI: https://doi.org/10.5281/zenodo.16786510. Recuperado a partir de https://revistas.uh.cu/rcm/article/view/10546.

Introducción

Las técnicas multivariadas permiten el análisis de las observaciones de varias variables aleatorias correlacionadas, para un número de individuos. Estos análisis son necesarios en diversas ramas de las ciencias. Por otra parte, en general, no es adecuado llevar a cabo una serie de análisis estadísticos univariados para cada una de las variables, ya que en ellos se ignoran las correlaciones entre las variables.

Para formular el problema a resolver, supóngase que se tienen $N+1$ elementos de una población (individuos), sobre cada uno de los cuales se observan q variables, atributos o características, entonces los resultados pueden representarse mediante una matriz de dimensiones $(N+1)\times q$:

$$(f_{ih})_{\substack{i=0,1,\cdots,N,\\ h=1,2,\cdots,q}}$$

donde cada fila representa todas las observaciones sobre un individuo y cada columna representa todas las observaciones de una característica. Así, f_{ih} representa el valor de la característica h-ésima en el individuo i-ésimo.

Se define el vector:

$$f^T = \left(f_{\bullet 1}^T, f_{\bullet 2}^T, \cdots, f_{\bullet q}^T\right).$$

Nótese que f contiene $(N+1)\times q$ observaciones.

Es posible suponer que cada observación es un valor realizado de una cierta variable aleatoria [4].

Debido a que se tienen varias variables aleatorias que caracterizan simultáneamente el fenómeno bajo estudio, es conveniente utilizar la covarianza como una medida de la asociación lineal entre dos variables. Al considerar todas las parejas de variables aleatorias posibles, se obtiene la matriz de covarianzas. Las definiciones formales aparecen en [10, pp. 68-69].

Ahora, sea V la matriz de covarianzas correspondiente a f, es decir, con la matriz de observaciones vectorizada. Como – en teoría – se tienen $(N+1)\times q$ variables aleatorias, V será de ese orden. Puede efectuarse una partición de V en bloques:

$$V = \begin{pmatrix} V_{11} & V_{12} & \cdots & V_{1q} \\ V_{21} & V_{22} & \cdots & V_{2q} \\ \vdots & \vdots & \dots & \vdots \\ V_{q1} & V_{q2} & \cdots & V_{qq} \end{pmatrix} = (V_{LH})_{\substack{L=1,2,\cdots,q \\ H=1,2,\cdots,q}}. \quad (1)$$

Cada V_{LH} es un bloque cuadrado de orden $N+1$.

Entonces, dentro de V_{LH}, el elemento de posición (k,j) es la covarianza entre el valor de la característica L del individuo k y el valor de la característica H del individuo j $(k,j = 0,1,\cdots,N)$.

El objetivo de esta contribución es presentar un algoritmo para estimar esta V. En particular, dicha estimación juega un papel muy importante dentro del campo de investigación sobre mínimos cuadrados generalizados (MCG) para funciones vectoriales (generalización con respecto a los mínimos cuadrados ponderados), donde los datos incluyen estimados de la precisión de las mediciones, pero en forma de pseudoinversa de la matriz de covarianzas de las mediciones V [11, pp. 2-3].

Al revisar la bibliografía se halla – por ejemplo – que Ebadi et al. [5] presentan una visión general de la literatura sobre gráficos de control para el monitoreo de la matriz de covarianzas en un proceso estadístico multivariado.

Por otra parte, Sabahno y Amiri [16] se orientan a la regresión lineal multivariada. Harrar y Kong [8] introducen pruebas estadísticas para el diseño de mediciones repetidas cuando la dimensión es grande.

En [9], [10] y [3] se trabaja con la matriz de covarianzas Σ correspondiente con las q características. En [7] solo consideran una sola característica.

En investigaciones aplicadas, a menudo es razonable tener en cuenta una o varias covariables cuando se efectúan pruebas en busca de diferencias entre medias de varios grupos. En este sentido, en [17, p. 3], Σ_i es la matriz de covarianzas correspondiente con los n_i individuos del grupo i.

Aparecida y Pereira [2] desarrollan tres análisis estadísticos para someter a prueba la matriz de covarianzas en una población con distribución normal multivariada.

En [15] y [9] se supone que las observaciones correspondientes con diferentes unidades experimentales o individuos son independientes.

Por el contrario, para aplicar el algoritmo que se propondrá en esta investigación no es necesario exigirles propiedades especiales a las observaciones.

En los documentos revisados no se reportó ningún algoritmo para estimar V, la matriz de covarianzas correspondiente con la matriz de observaciones vectorizada.

Relevancia del estudio

Este trabajo presenta una mejora significativa en la estimación de matrices de covarianzas para datos multivariados, problema fundamental en estadística y análisis de datos. El algoritmo propuesto, implementado en `Estimacov_2`, supera las limitaciones de su predecesor al reducir la complejidad computacional de $\mathscr{O}\left(\frac{N^4q^2}{m_{\text{mín}}^2}\right)$ a $\mathscr{O}(N^2q^2)$, lo que lo hace más eficiente para conjuntos de datos grandes. Esta optimización es crucial en aplicaciones como modelado estadístico, aprendizaje automático e investigación biomédica, donde el cálculo robusto de covarianzas es esencial. La flexibilidad del algoritmo para manejar diferentes estructuras de grupos y su validación integrada lo convierten en herramienta valiosa para la comunidad científica, con impacto en econometría, genómica e ingeniería, donde la precisión en estimaciones es crítica para decisiones basadas en datos.

1. Dificultades para calcular V

Dado que el orden de V es $(N+1)q$, es probable que a un usuario le resulte difícil o trabajoso obtener dicha matriz, aunque sea aproximadamente. Aún para el caso $q=1$, Greene [6] reconoce esta dificultad cuando afirma: “Obviamente, hay que imponerle al modelo alguna estructura, si queremos avanzar.”. Es por ello que originalmente se creó el subprograma `Estimacov` (ESTImar MAtriz de COVarianzas) y – más recientemente – `Estimacov_2` para obtenerla.

Para el cálculo de V utilizando la conocida fórmula muestral:

$$\mathrm{Cov}(Y,X) = \frac{\sum_{i=1}^{n}(y_i-\bar{y})(x_i-\bar{x})}{n-1}, \quad (2)$$

es necesario tener $(N+1)q$ vectores, lo que, suponiendo que cada vector tenga n componentes, conlleva conocer $n(N+1)q$ escalares, donde n puede ser algo grande, de manera que la estimación calculada sea suficientemente buena. Sin embargo, en las observaciones solo se dispone de $(N+1)q$ escalares, que es mucho menos.

Al atenerse estrictamente a la información con que se cuenta, para cada individuo y cada característica solo se tiene un vector trivial, que es simplemente un escalar. De esta manera se individualizaría completamente la información en cada una de las características, pero por supuesto, el cálculo de las covarianzas carecería totalmente de confiabilidad.

Yendo al otro extremo, si se aplica:

$$V = \Sigma \otimes I, \quad (3)$$

se utilizarían muestras suficientemente grandes, pero todas las características quedarían globalizadas al no reconocerse las posibles diferencias entre individuos.

Ante la imposibilidad de generar una V individualizada completamente (por no contar con suficiente información) se adoptará una estrategia de compromiso entre ambas posiciones extremas.

Para ello, se ordena previamente la información de acuerdo con los individuos, y posteriormente se hace una partición de las observaciones en grupos (o *clusters*), de manera que todos los individuos correspondientes con un mismo grupo (es decir, aquellos cuyas observaciones están contenidas en un mismo grupo) sean relativamente cercanos o afines. Entonces, para el cálculo de la covarianza correspondiente con dos características específicas de un par dado de individuos, se tomarán como muestras los grupos en los que se encuentran y esta covarianza será la misma para todos los individuos de dichos grupos.

Para más detalles sobre estos fundamentos, ver [11].

2. Características generales del algoritmo

El algoritmo que lleva a cabo la idea bosquejada en el párrafo anterior fue implementado en MATLAB 9.8 [13], en esta ocasión mediante la función `Estimacov_2` (Estimar matriz de covarianzas 2). Este subprograma le brinda al usuario diferentes opciones, la primera de ellas es decidir el nivel de detalle para la estimación de V:

1. **'Individuos no correlacionados'**: Corresponde con la ecuación (3), en este caso no se forman grupos.
2. **'Grupos no correlacionados'**: Para cada grupo se calculan las covarianzas, pero no hay correlación entre características de individuos de grupos distintos.
3. **'Grupos correlacionados'**: Para cada grupo se calculan las covarianzas, y en general hay correlación entre características de individuos de grupos distintos.

En los casos 2 y 3, se puede imponer la cantidad mínima de observaciones que se desea que haya en cada grupo, y así alcanzar mayor precisión al calcular (2). También puede exigirse que todos los grupos tengan el mismo tamaño; esto tiene ventajas y desventajas.

El espacio de las unidades experimentales o individuos puede tener una sola dimensión (como en el caso de unidades de tiempo) o varias dimensiones, cuando por ejemplo se trate de puntos del plano o del espacio tridimensional.

En general, por convención, las columnas correspondientes con los individuos van a ser las primeras columnas de `Valores_var_indep` (valores de la variable independiente) en el algoritmo; por lo tanto, es posible que un mismo individuo aparezca repetido de manera que le correspondan varias observaciones (dicha posibilidad constituye una extensión de la usual relación individuo-observación).

Si no se le exige al algoritmo que todos los grupos tengan el mismo tamaño, este tratará, como es natural, de incluir en un mismo grupo a todas las observaciones correspondientes con un mismo individuo (lo que además aumenta el tamaño de esa muestra para el posterior cálculo de las covarianzas). Por esa razón, en este caso el subprograma le pregunta al usuario cuántas dimensiones tiene el espacio de los individuos (`dim_indiv`). Entonces, los individuos están determinados por las primeras `dim_indiv` columnas de `Valores_var_indep`.

Suponiendo que ya están formados todos los grupos (posteriormente se verá cómo se hace esto); si ahora se supone además que G_I denota el grupo I-ésimo, y C es la cantidad de grupos, entonces se puede decir que G_I es una matriz de n_I filas y q columnas ($I = 1, \cdots, C$). Cuando se habla de "tamaño" del grupo I-ésimo, se hace referencia al valor de n_I.

Si al ejecutar `Estimacov_2` se selecciona para la estimación de V la opción 'Grupos correlacionados', el bloque genérico V_{LH} de la matriz resultante (recordar (1)) quedará a su vez particionado en los subbloques S_{IJ}; $I, J = 1, \ldots, C$, y tendrá la siguiente estructura:

$$\begin{pmatrix} S_{11} & S_{12} & \cdots & S_{1J} & \cdots & S_{1C} \\ S_{21} & S_{22} & \cdots & S_{2J} & \cdots & S_{2C} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ S_{I1} & S_{I2} & \cdots & S_{IJ} & \cdots & S_{IC} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ S_{C1} & S_{C2} & \cdots & S_{CJ} & \cdots & S_{CC} \end{pmatrix}.$$

Estos C^2 subbloques, en general, no son del mismo tamaño.

Sea $m = \text{mín}\{n_I, n_J\}$, entonces:

$$S_{IJ} = \text{Cov}\left(G_I(1 \cdots m, L), G_J(1 \cdots m, H)\right) \cdot \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}_{n_I \times n_J}$$

Como se aprecia, los subbloques de la diagonal son cuadrados.

En el caso particular que todos los grupos tengan el mismo tamaño m, es posible brindar una expresión más directa para V. Sea:

$$G = (G_1(\bullet, 1) \ldots G_C(\bullet, 1)\, G_1(\bullet, 2) \cdots G_C(\bullet, 2) \cdots G_1(\bullet, q) \cdots G_C(\bullet, q)),$$

y

$$A = (a_{ij})_{Cq \times Cq} = \text{Cov}(G).$$

Entonces,

$$V = A \otimes \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}_{m \times m}.$$

Si además se utiliza la opción 'Grupos no correlacionados', a partir de A en (2) se define la matriz:

$$B = (b_{ij})_{Cq \times Cq} : \; b_{ij} = \begin{cases} a_{ij} & \text{si } i \equiv j \pmod{C} \\ 0 & \text{en otro caso} \end{cases} \qquad i, j = 1, \cdots, C_q,$$

y entonces,

$$V = B \otimes \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}_{m \times m}.$$

En resumen, los grupos se forman de acuerdo con el orden en que aparecen los individuos. Si se requiere igualdad de tamaños, los grupos quedan definidos directamente de acuerdo con el tamaño y la V queda definida implícitamente, conforme a (2) o a (2), conociendo A o B, respectivamente, y además m. En el algoritmo que se está explicando, para cada par I, J, se calcula el mínimo correspondiente, pero para evitar complicación innecesaria en la notación, escribiremos m sin doble subíndice. Este ahorro de memoria es aprovechado por este software.

Si no se requiere igualdad de tamaños, entonces, en general, cada grupo que se está formando se engrosa hasta que su tamaño alcance al menos el valor de $m_{\text{mín}}$, cuidando que los individuos repetidos (si los hay) pertenezcan al mismo grupo. Con respecto a V, esta se calcula explícitamente: el subbloque S_{IJ} del bloque genérico V_{LH} de V viene dado por (2).

Dentro del campo de investigación MCG para funciones vectoriales (que se trató en la Introducción), la estimación de V constituye solo una parte que se articula con el resto. En esta comunicación trataremos exclusivamente la estimación de V.

3. Pasos principales de `Estimacov`

1. Asignación de los parámetros de entrada: `Valores_var_indep` y `Observ` (Observaciones). Similarmente, en el caso simple de (3), basta conocer $\sum$ y N, por lo que el resultado también viene dado implícitamente.

2. Asignar a p el número de columnas de `Valores_var_indep`, y a q el número de columnas de `Observ`.

3. Si `Valores_var_indep` y `Observ` tienen diferente número de filas, indicar error y retornar (al programa que invoca).

4. Hacer `Info` = [`Valores_var_indep` `Observ`] (concatenación de matrices). Si no se va a efectuar ningún ajuste de datos, simplemente puede tomarse `Val_var_indep` como $(1,\cdots,N+1)^T$, donde T indica transpuesta.

5. Se ordena `Info` de manera ascendente.

6. Asignar a `Val_var_indep_ord` las primeras p columnas de `Info`, y a `Observ_ord` las últimas q columnas de `Info`. Estos son parámetros de salida de `Estimacov`.

7. Preguntarle al usuario con qué nivel de detalle desea estimar V y registrar la respuesta en `nivel_detalle`. Esta es una información inicial.

8. Si el usuario no respondió, retornar.

9. En el caso que `nivel_detalle` corresponda con '`Individuos no correlacionados`':

   9.1 Calcular $\sum = \text{cov}(\texttt{Observ_ord})$ y asignarla a `Matr` (este último es un parámetro de salida) conforme a la ecuación (3).

   9.2 Hacer `tam`=0. Este es el otro parámetro de salida; tamaño de los grupos = 0 indica que no se formaron grupos.

   9.3 Retornar.

   Casos de `Grupos no correlacionados` y `Grupos correlacionados`:

10. Asignar a `cant_obs` (cantidad de observaciones) el número de filas de `Valores_var_indep`.

11. Preguntar cuál será la cantidad mínima de observaciones en cada grupo.

12. Si la respuesta fue vacía, regresar a la instrucción 7.

13. Si la respuesta no corresponde con un número entero entre 1 y `cant_obs`, indicar que la respuesta es inadecuada y regresar a 11.

14. Guardar en `cant_min` dicha respuesta.

15. Preguntar si se desea que todos los grupos tengan el mismo tamaño.

16. Si el usuario canceló o la respuesta fue vacía, regresar a 11.

17. Caso de 'Sí' (grupos de igual tamaño):

    17.1 Hacer $\texttt{cant_fil_g} = \text{mín}\{j \mid \texttt{cant_min} \leq j \leq \texttt{cant_obs} \mid \frac{\texttt{cant_obs}}{j}\}$, Cantidad de filas que tendrá cada grupo.

    17.2 Asignar a `Cant_gru` la cantidad de grupos.

    17.3 Si `Cant_gru` = 1:

    17.3.1 `Hay_un_solo_grupo`; se indica, debe ejecutarse nuevamente el programa.

    17.3.2 Retornar.

    17.4 Calcular A según (2) y asignarla a "`matriz`".

    17.5 En el caso que `nivel_detalle` corresponda con '`Grupos no correlacionados`': A partir de "`matriz`" y de (2), construir B y asignarla a `matriz`.

    17.6 Hacer `Matr` = `matriz`.

    17.7 Hacer `tam` = `cant_fil_g` (tamaño de cada grupo).

    17.8 Retornar.

    Caso de 'No'; los grupos no son de igual tamaño:

18. Preguntar cuántas dimensiones tiene el espacio al que pertenecen los individuos.

19. Si la respuesta fue vacía, ir a 15.

20. Si la respuesta no corresponde con un número entero entre 1 y p, indicar que la respuesta es inadecuada y regresar a 18.

21. Asignar dicha respuesta a `dim_indiv`.

    Formación de los grupos:

22. Iniciar "`Grupos`" como matriz de `cant_obs` filas y `cant_obs` $\times q$ columnas de elementos `NaN` (*Not a Number*, no es un número). En "`Grupos`" se irán colocando todos los grupos de izquierda a derecha.

23. Iniciar `gru` como matriz de `cant_obs` filas y q columnas de elementos `NaN`. En `gru` se guardará el grupo en formación.

24. Hacer `gru`(1,:) = `Info`(1,p+1:p+q). Aquí se asigna la fila 1, desde la columna $p+1$ hasta la $p+q$, o sea, la primera observación.

25. Hacer $k = 1$; contador interno de grupo.

26. Hacer `cont_grup` = 0; Contador de grupos.

27. Para i desde 2 hasta `cant_obs` repetir:

    27.1 Si $(k \geq$ `cant_min`$)$ e `Info`$(i, 1:$ `dim_indiv`$) \neq$ `Info`$(i-1, 1:$ `dim_indiv`$)$), cerrar grupo antes de tomar la nueva observación. Si hay individuos repetidos no se cierra el grupo, se ejecuta "De lo contrario...".

    27.1.1 `Cerrar_grupo` (ver final de este listado).

    27.1.2 Iniciar `gru` como matriz de `cant_obs` filas y q columnas de elementos `NaN`. Tomar la nueva observación:

    27.1.3 Hacer `gru`(1,:) = `Info`(i,p+1:p+q). La i-ésima observación pasa a ser la primera del grupo en formación.

    27.1.4 Hacer $k = 1$; contador interno de grupo.

    De lo contrario: (Mantener el grupo)

    27.1.5 Hacer $k = k+1$.

    27.1.6 Hacer `gru`(k,:) = `Info`(i,p+1:p+q). La i-ésima observación pasa a ser la k-ésima del grupo en formación.

28. Si $(k \geq$ `cant_min`$)$. `Cerrar_grupo` (ver final de este listado); también se cierra el grupo.

29. Hacer `Cant_gru` = `cont_grup` (cantidad de grupos).

    Para añadir las k observaciones remanentes al último grupo:

    30 Si $k <$ `cant_min`.

    30.1 Hacer `Grupos`(`cant_filas_grup`(`Cant_gru`)+1: `cant_filas_grup`(`Cant_gru`)+k, `Cant_gru`$\times$ q-q+1:`Cant_gru`$\times$ q) = `gru`(1:k,:).

    30.2 Hacer `cant_filas_grup`(`Cant_gru`) = `cant_filas_grup`(`Cant_gru`) + k.

Cálculo de las covarianzas:

31 Eliminar todas las columnas de `Grupos` cuyos elementos sean todos `NaN` (últimas columnas).

32 `covaria` = cov(`Grupos`). Si dos columnas tienen tamaños n y m, donde $n \leq m$, el cálculo de la covarianza se efectúa con las primeras n componentes de las dos columnas.

33 Hacer `tam` = -1; indica que se formaron grupos de diferente tamaño.

34 Iniciar `Matr` como matriz vacía.

35 En el caso que `nivel_detalle` corresponda con '`Grupos no correlacionados`':

35.1 Iniciar "`matriz`" como una matriz cuadrada de ceros, de orden `cant_obs` $\times q$.

35.2 Para L desde 1 hasta q repetir:

35.21 Para H desde 1 hasta q repetir:

35.211 `fila` = $(L-1) *$ `cant_obs` $+ 1$.

35.212 `columna` = $(H-1) *$ `cant_obs` $+$ 1.

35.213 Para I desde 1 hasta `Cant_gru` repetir: ($J = I$ en (2))

35.2131 `fil` = $(I-1) \times q + L$.

35.2132 `col` = $(I-1) \times q + H$.

35.2133 `matriz`(`fila`:`fila`+ `cant_filas_grup`(I)-1, `columna`:`columna`+ `cant_filas_grup`(I)-1) = `covaria`(`fil`, `col`)* `unos`(`cant_filas_grup`(I)), de acuerdo con (2).

35.2134 `fila` = `fila` + `cant_filas_grup`(I).

35.2135 `columna` = `columna` + `cant_filas_grup`(I)

(El bloque V_{LH} es diagonal por bloques)

35.3 `Matr` = `matriz`.

35.4 Retornar.

`Grupos correlacionados`:

36 Iniciar "`matriz`" como matriz vacía que tendrá un espacio máximo de `cant_obs` $\times q$ filas y `cant_obs` $\times q$ columnas.

37 Reservar para la matriz M un espacio máximo de `cant_obs` $\times q$ filas y `cant_obs` columnas, mediante elementos `NaN`.

38 Para H desde 1 hasta q repetir: (se inicia una columna de bloques)

38.1 Iniciar M como matriz vacía.

38.2 Para L desde 1 hasta q repetir:

38.21 Iniciar S como un arreglo `Cant_gru` × `Cant_gru` de matrices vacías.

38.22 Para J desde 1 hasta `Cant_gru` repetir:

38.221 `col` = $(J-1)*q+H$.

38.222 Para I desde 1 hasta `Cant_gru` repetir:

38.2221 `fil` = $(I-1)*q+L$.

38.2222 $S(I,J) =$ `covaria(fil,col)`$*$ `unos(cant_filas_grup(`I`), cant_filas_grup(`J`))`; de acuerdo con (2).

38.23 Expandir M hacia abajo con S.

38.3 Expandir "`matriz`" a la derecha con M.

39 `Matr` = `matriz`.

40 Retornar.

`Cerrar_grupo` (subprograma):

S1 Hacer `cont_grup` = `cont_grup` + 1.

S2 Hacer `Grupos`(1:k, `cont_grup`× q-q+1: `cont_grup`× q) = `gru`(1:k,:); colocación en `Grupos` del nuevo grupo formado.

S3 Hacer `cant_filas_grup(cont_grup)` = k; el grupo número `cont_grup` tiene k filas.

En el listado anterior – en aras de la claridad – la validación de la información de entrada correspondiente con `nivel_detalle`, `cant_min`, grupos de igual tamaño o no y `dim_indiv` se ha efectuado de manera sencilla y directa; en realidad, en el programa esta se realiza mediante cuatro ciclos `while` anidados.

4. Complejidad

Antes de determinar la complejidad (orden de magnitud o razón de crecimiento) de `Estimacov_2`, es conveniente recordar que el orden de un algoritmo puede ocultar términos de menor orden, como puede verse en [12, 14]. Además, aunque el mismo no depende de detalles de implementación, el tiempo verdadero de ejecución del algoritmo para los mismos datos de entrada sí depende de factores tales como rapidez del procesador, conjunto de instrucciones, rapidez del disco, tipo de compilador, etc. [1].

A continuación se obtiene la razón de crecimiento de `Estimacov_2`. En el caso sencillo en que `nivel_detalle` corresponda con '`Individuos no correlacionados`' (instrucción 9), se calcula $\sum$ mediante cov(`Observ_ord`), utilizando la función de MATLAB correspondiente, cuyo orden de magnitud es $O(\texttt{cant_obs} \times q^2)$, o lo que es lo mismo, $O(Nq^2)$, ya que $\texttt{cant_obs} = N+1 = N'$ [13].

Para los otros casos basta analizar las instrucciones 17, 22, 27, 31, 32, 35, 37, 38 y 39, que son las cruciales; entre ellas se encuentran las de mayor complejidad algorítmica y, por tanto, determinan el orden de magnitud general. Es necesario distinguir si el usuario exige o no que todos los grupos tengan el mismo tamaño.

Grupos de igual tamaño

En este caso se ejecutará la instrucción 17. Sus instrucciones más complejas son la 17.4 y la 17.5:

La razón de crecimiento de la instrucción 17.4 es $O(\texttt{cant_obs} \times q \times \texttt{Cant_gru}^2)$, o lo que es lo mismo, $O\left(\frac{q^2 \times \texttt{cant_obs}^2}{m}\right)$. Ahora, si se toma en cuenta que $m \geq m_{\text{mín}}$; en consecuencia, el orden de magnitud de la instrucción 17.4 es $O\left(\frac{N^2q^2}{m_{\text{mín}}}\right)$.

La instrucción 17.5 tiene un orden de magnitud $O(C^2q^2)$, o sea, el de la cantidad de elementos de B en (2). Este orden también puede expresarse como $O\left(\frac{N^2q^2}{m^2_{\text{mín}}}\right)$, inferior al de 17.4.

Por lo tanto, si se pide que los grupos tengan el mismo tamaño, `Estimacov_2` tiene una complejidad de $O\left(\frac{N^2q^2}{m_{\text{mín}}}\right)$.

Grupos de diferentes tamaños

Instrucción 22:

La instrucción 22 tienen un orden de complejidad $O(N^2q)$, acorde con la cantidad de elementos de `Grupos`.

Instrucción 27:

Como `dim_indiv` está acotado por p, en la instrucción 27.1, la evaluación del "y" lógico inicial tiene un costo $O(p)$.

Dentro de la propia instrucción 27.1, el orden del costo de la formación de "`gru`" es el de 27.1.6, multiplicado por el número de veces que se efectúa, o sea, se acota por qN veces.

Una vez formado `gru`, el orden del costo del cierre es el de 27.1.1 o 27.1.2 (ambos iguales a Nq).

De los tres párrafos anteriores, se infiere que el orden de 27.1 es $O(Nq)$, y el orden de 27 es $O(N^2q)$.

Instrucción 31:

En la instrucción 31, el número de columnas de `Grupos` con `NaN` exclusivamente es $(N+1-C)q$ y, como cada columna tiene N elementos (de acuerdo con la instrucción 22), la cantidad de posiciones a eliminar se puede acotar por N^2q. Luego, el orden de la instrucción 31 es $O(N^2q)$.

Instrucción 32:

Aquí, las covarianzas se calculan entre los C grupos existentes, por lo que su complejidad es del orden de $N^2C^2q^2$, que puede acotarse por $O\left(\frac{N^3q^2}{m^2_{\text{mín}}}\right)$.

Instrucción 35:

Las instrucciones 35.1 y 35.3 son del orden de $O(N^2q^2)$.

Análisis de 35.2: Para cada L y cada H, sea T el tiempo total necesario para ejecutar el ciclo "Para I..." (`for I`). Si denotamos mediante T_I el tiempo de la iteración I-ésima de dicho ciclo, tenemos que, debido a 35.2133:

$$T_I \leq c_I m_I^2 : m_I = \texttt{cant_filas_grup}(I),$$

o sea, el tamaño del grupo I, y c_I es una constante independiente de los tamaños, $I = 1, \cdots, C$. (Note que las demás instrucciones del ciclo `for I` son $O(1)$). Entonces,

$$T = \sum_{I=1}^{C} T_I \leq \sum_{I=1}^{C} c_I m_I^2.$$

Si se toma $c = \text{máx}_I\{c_I\}$, se obtiene que:

$$T \leq c \sum_{I=1}^{C} m_I^2 \leq c \left(\sum_{I=1}^{C} m_I \right)^2 = cN^2.$$

Como 35.211 y 35.212 son $O(1)$, el tiempo de una iteración de H se puede acotar por $d^* N^2$. Considerando ahora los ciclos `for L` y `for H`, el costo de 35.2 es $T^* \leq d^* N^2 q^2 \therefore T^* = O(N^2 q^2)$.

Entonces, 35.1, 35.2 y 35.3 son $O(N^2 q^2)$, por lo que la instrucción 35 es $O(N^2 q^2)$.

Instrucción 37:

Al igual que las instrucciones 22 y 27, la 37 es $O(N^2 q)$, acorde con la cantidad de elementos de M.

Instrucción 38:

Para cada H y cada L, sea $T(J)$ el tiempo total necesario para ejecutar el ciclo `for I`. Si se denota mediante $T_I(J)$ el tiempo de la iteración I-ésima de dicho ciclo, se tiene que, debido a 38.2222: $T_I(J) \leq c_{JI} m_I m_J$, donde c_{JI} es una constante independiente de los tamaños, $I = 1, \cdots, C$.

Notar que la otra instrucción del ciclo `for I`, 38.2221 es $O(1)$). Entonces,

$$T(J) = \sum_{I=1}^{C} T_I(J) \leq \sum_{I=1}^{C} c_{JI} m_I m_J.$$

Si se hace $c_J = \text{máx}_I\{c_{JI}\}$, se obtiene:

$$T(J) \leq c_J m_J \sum_{I=1}^{C} m_I = c_J m_J N.$$

Como la instrucción 38.221 es $O(1)$, el tiempo de una iteración de J se puede acotar por $d_J m_J N$ y el tiempo que requiere el ciclo `for J` completo (instrucción 38.22) es

$$T \leq \sum_{J=1}^{C} d_J m_J N.$$

Si se toma $d^* = \text{máx}_J\{d_J\}$, entonces,

$$T \leq d^* N \sum_{J=1}^{C} m_J = d^* N^2.$$

Luego, el orden de la instrucción 38.22 es $O(N^2)$.

Por otra parte, la instrucción 38.21 es del orden de la cantidad de matrices vacías creadas, o sea,

$$C^2 \leq \frac{N^2}{m_{\text{mín}}^2}.$$

En cuanto a la instrucción 38.23, esta ocasiona el crecimiento de la columna de bloques en la medida que aumenta L; cuando alcanza su tamaño máximo, la cantidad de elementos que se asigna es del orden de $O(N^2 q)$.

Como consecuencia de los tres párrafos anteriores, la instrucción 38.2 es de orden $O(N^2 q)$.

Ocasionado por la instrucción 38.3, el número de columnas de bloques va creciendo en la medida que aumenta H. Cuando alcanza su valor máximo, la cantidad de elementos que se asigna es de orden $N^2 q^2$, es decir, el de la matriz V completa.

A partir de los dos párrafos anteriores se tiene que la instrucción 38 es de orden $O(N^2 q^2)$.

Instrucción 39:

Se le asigna explícitamente al parámetro de salida `Matr` la matriz de covarianzas; el orden de la instrucción 39 es, también, $O(N^2 q^2)$.

Se ha visto entonces que el orden de la instrucción 32 es $O\left(\frac{N^3 q^2}{m_{\text{mín}}^2}\right)$ y el de las instrucciones 35, 38 y 39 es $O(N^2 q^2)$. Puede observarse que, para valores típicos de N, q y $m_{\text{mín}}$, la instrucción 32 tiene menor orden que las otras tres; por lo tanto, estas determinan la complejidad de `Estimacov_2`, que es $O(N^2 q^2)$, menor que $O\left(\frac{N^4 q^2}{m_{\text{mín}}^2}\right)$ en `Estimacov` [11].

En el caso que el usuario decida que todos los grupos sean del mismo tamaño, debido a la instrucción 17, `Estimacov_2` tiene complejidad $O\left(\frac{N^2 q^2}{m_{\text{mín}}}\right)$, la misma de `Estimacov`. Sin embargo, en general, `Estimacov_2` es superior en eficiencia, lo que se manifiesta, además, por ejemplo, en el proceso de formación de los grupos y en la utilización de los recursos del lenguaje.

También, `Estimacov_2` valida la información de entrada. Para ello se recomienda leer instrucción 3 y el párrafo al final del listado del algoritmo.

5. Ejemplo

Dadas las usuales dimensiones de V, se considera un ejemplo con N pequeño.

Sea `Observ`:

$$\begin{bmatrix} 0 & 0{,}6585 \\ 0{,}2086 & 0{,}6172 \\ 0{,}3699 & 0{,}4465 \\ 0{,}5929 & 0{,}3533 \\ 0{,}6815 & 0{,}1732 \\ 0{,}8835 & 0 \\ 0{,}1887 & 0{,}7490 \\ 0{,}4089 & 0{,}7278 \end{bmatrix}.$$

El coeficiente de correlación muestral es $-0{,}8847$, lo que indica una alta correlación entre las dos columnas.

Como no se va a efectuar ningún ajuste de datos, simplemente se tomará `Val_var_indep` como $[1:8]^T$.

Al ejecutar `Estimacov_2` con grupos de igual tamaño y `cant_min=3`, obtenemos: `Val_var_indep_ord`

= `Val_var_indep` y `Observ_ord = Observ` (por supuesto) y además:

```
>> Matr
0.0629 -0.0456 -0.0348  0.0751
-0.0456 0.0929  0.0336 -0.1117
-0.0348 0.0336  0.0206 -0.0502
0.0751 -0.1117 -0.0502 0.1467

>> tam
4
```

Esto quiere decir que todos los grupos formados (en este caso dos) son de tamaño 4. La matriz V se obtuvo implícitamente. Explícitamente:

$$V = \text{Matr} \otimes \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}_{4\times 4} .$$

Al ejecutar nuevamente `Estimacov_2` con los mismos datos, pero ahora sin exigir que los grupos sean de igual tamaño; `cant_min=3` y `dim_indiv=1`, los parámetros `Val_var_indep_ord` y `Observ_ord` son los mismos de la corrida anterior. El valor `tam = -1`, indica que los grupos formados fueron de distinto tamaño. V = `Matr` se obtuvo explícitamente; debido a su orden, se presentará por bloques, de acuerdo con (1) para $q = 2$. El lector puede verificar la estructura descrita para un bloque genérico V_{LH}, con dos grupos de tamaños 3 y 5, en correspondencia también con (2).

```
>> Matr(1:8,1:8)
0.0344  0.0344  0.0344  0.0264  0.0264  0.0264  0.0264  0.0264
0.0344  0.0344  0.0344  0.0264  0.0264  0.0264  0.0264  0.0264
0.0344  0.0344  0.0344  0.0264  0.0264  0.0264  0.0264  0.0264
0.0264  0.0264  0.0264  0.0702  0.0702  0.0702  0.0702  0.0702
0.0264  0.0264  0.0264  0.0702  0.0702  0.0702  0.0702  0.0702
0.0264  0.0264  0.0264  0.0702  0.0702  0.0702  0.0702  0.0702
0.0264  0.0264  0.0264  0.0702  0.0702  0.0702  0.0702  0.0702
0.0264  0.0264  0.0264  0.0702  0.0702  0.0702  0.0702  0.0702
```

```
>> Matr(1:8,9:16)
-0.0191 -0.0191 -0.0191 -0.0327 -0.0327 -0.0327 -0.0327 -0.0327
-0.0191 -0.0191 -0.0191 -0.0327 -0.0327 -0.0327 -0.0327 -0.0327
-0.0191 -0.0191 -0.0191 -0.0327 -0.0327 -0.0327 -0.0327 -0.0327
-0.0166 -0.0166 -0.0166 -0.0844 -0.0844 -0.0844 -0.0844 -0.0844
-0.0166 -0.0166 -0.0166 -0.0844 -0.0844 -0.0844 -0.0844 -0.0844
-0.0166 -0.0166 -0.0166 -0.0844 -0.0844 -0.0844 -0.0844 -0.0844
-0.0166 -0.0166 -0.0166 -0.0844 -0.0844 -0.0844 -0.0844 -0.0844
-0.0166 -0.0166 -0.0166 -0.0844 -0.0844 -0.0844 -0.0844 -0.0844
```

```
>> Matr(9:16,1:8)
-0.0191 -0.0191 -0.0191 -0.0166 -0.0166 -0.0166 -0.0166 -0.0166
-0.0191 -0.0191 -0.0191 -0.0166 -0.0166 -0.0166 -0.0166 -0.0166
-0.0191 -0.0191 -0.0191 -0.0166 -0.0166 -0.0166 -0.0166 -0.0166
-0.0327 -0.0327 -0.0327 -0.0844 -0.0844 -0.0844 -0.0844 -0.0844
-0.0327 -0.0327 -0.0327 -0.0844 -0.0844 -0.0844 -0.0844 -0.0844
-0.0327 -0.0327 -0.0327 -0.0844 -0.0844 -0.0844 -0.0844 -0.0844
-0.0327 -0.0327 -0.0327 -0.0844 -0.0844 -0.0844 -0.0844 -0.0844
-0.0327 -0.0327 -0.0327 -0.0844 -0.0844 -0.0844 -0.0844 -0.0844
```

```
>> Matr(9:16,9:16)
0.0126 0.0126 0.0126 0.0186 0.0186 0.0186 0.0186 0.0186
0.0126 0.0126 0.0126 0.0186 0.0186 0.0186 0.0186 0.0186
0.0126 0.0126 0.0126 0.0186 0.0186 0.0186 0.0186 0.0186
0.0186 0.0186 0.0186 0.1107 0.1107 0.1107 0.1107 0.1107
0.0186 0.0186 0.0186 0.1107 0.1107 0.1107 0.1107 0.1107
0.0186 0.0186 0.0186 0.1107 0.1107 0.1107 0.1107 0.1107
0.0186 0.0186 0.0186 0.1107 0.1107 0.1107 0.1107 0.1107
0.0186 0.0186 0.0186 0.1107 0.1107 0.1107 0.1107 0.1107
```

Comentarios finales

En esta comunicación se han descrito las características del subprograma `Estimacov_2`, para estimar la matriz de covarianzas correspondiente con una matriz de observaciones vectorizada, el cual brinda diferentes opciones al usuario.

Con relación a su predecesor, `Estimacov`, cuya complejidad algorítmica es:

$$O\left(\frac{N^4 q^2}{m_{\text{mín}}^2}\right),$$

constituye una mejora, pues la de este nuevo subprograma es $O(N^2 q^2)$. Además, su mayor eficiencia se debe en parte a una mejor utilización de los recursos de MATLAB.

Por otra parte, se realiza una validación de la información de entrada, lo que contribuye a la robustez de este software.

El ejemplo final ilustra la idea central del algoritmo de formar grupos y asignar colectivamente covarianzas dentro de los mismos.

En un futuro trabajo se aplicará `Estimacov_2`, pero dentro del campo de los mínimos cuadrados generalizados.

Suplementos

Este artículo no contiene información suplementaria.

Conflictos de interés

El autor declara que no tiene conflicto de interés en relación con este artículo. No hubo ninguna subvención para este artículo.

Referencias

[1] Adamchik, V.S.: *Algorithmic Complexity*. Carnegie Mellon University, 2009. https://viterbi-web.usc.edu/~adamchik/15-121/lectures/Algorithmic%20Complexity/complexity.html.

[2] Aparecida, S. and L. Pereira: *Alternative Hypothesis Tests for the Covariance Matrix Based on Eigenvalues and Multivariate Normality*. Brazilian Journal of Operations & Production Management, 9(1):9–27, 2012. https://doi.org/10.4322/bjopm.2013.002.

[3] Branco, A.F. and M.A. Guerreiro: *Monitoring the mean vector and the covariance matrix multivariate processes with sample means and sample ranges*. Produção, 21(2):197–208, 2011. https://doi.org/10.1590/S0103-65132011005000029.

[4] Brockwell, P.J. and R.A. Davis: *Introduction to Time Series and Forecasting*. Springer Science+Business Media, Inc., NY, USA, second edition, 2002, ISBN 0-387-95351-5. https://doi.org/10.1007/978-3-319-29854-2.

[5] Ebadi, M., S. Chenouri, D.K.J. Lin, and S.H. Steiner: *Statistical Monitoring of the Covariance Matrix in Multivariate Processes: A Literature Review*. Journal of Quality Technology, 54(3):269–289, 2021. https://doi.org/10.1080/00224065.2021.1889419.

[6] Greene, W.H.: *Econometrics Analysis*. Prentice Hall, New Jersey, eigth edition, 2018, ISBN 978-0-13-446136-6. https://pages.stern.nyu.edu/~wgreene/Text/econometricanalysis.htm.

[7] Hair, J.F., W.C. Black, B.J. Babin, and R.E. Anderson: *Multivariate Data Analysis*. Pearson Prentice Hall, 7th edition, 2010. https://opac.utssurabaya.ac.id/uploaded_files/temporary/DigitalCollection/Mjk2M2I0ZTZlOTFhYTE3ODE1YTJkODVkMWJiMDllZGQzYmNiNzI5MQ==.pdf.

[8] Harrar, S.W. and X. Kong: *High-dimensional multivariate repeated measures analysis with unequal covariance matrices*. Journal of Multivariate Analysis, 145:1–21, 2016. https://doi.org/10.1016/j.jmva.2015.11.012.

[9] Janiszewska, M.: *Structures of the covariance matrix: An overview*. Biometrical Letters, 59(2):141–157, 2022. https://doi.org/10.2478/bile-2022-0010.

[10] Johnson, R.A. and D.W. Wichern: *Applied Multivariate Statistical Analysis*. Pearson Education Inc, NJ, USA, 6th edition, 2007. https://www.webpages.uidaho.edu/~stevel/519/Applied%20Multivariate%20Statistical%20Analysis%20by%20Johnson%20and%20Wichern.pdf.

[11] Lemagne, J.: *Mínimos Cuadrados Generalizados no lineales para funciones vectoriales: Un enfoque desde el Análisis Numérico*. Tesis de Doctorado, Universidad de La Habana, Facultad de Matemática y Computación, 2011.

[12] Martignon, L.: *Algorithms, 3 the complexity of an algorithm*. In *International Encyclopedia of the Social & Behavioral Sciences*. Elsevier Ltd., 2001, ISBN 978-0-08-043076-8. https://www.sciencedirect.com/topics/computer-science/algorithmic-complexity.

[13] MathWorks: *MATLAB R2020a (9.8.0.1323502)*, 2020. https://www.mathworks.com.

[14] Perret, B.: *Algorithm complexity*. ESIEE Paris, 2022. https://perso.esiee.fr/~perretb/AlgorithmDesign/complexity/index.html.

[15] Rencher, A.C. and W.F. Christensen: *Methods of Multivariate Analysis*. John Wiley & Sons, Inc., third edition, 2012, ISBN 978-0-470-17896-6. https://www.wiley-vch.de/en/areas-interest/mathematics-statistics/statistics-16st/data-analysis-16st5/multivariate-analysis-16st51/methods-of-multivariate-analysis-978-0-470-17896-6.

[16] Sabahno, H. and A. Amiri: *Simultaneous monitoring of the mean vector and covariance matrix of multivariate multiple linear profiles with a new adaptive Shewhart-type control chart*. Quality Engineering, 35(4):600–618, 2023. https://doi.org/10.1080/08982112.2022.2164725.

[17] Zimmermann, G., M. Pauly, and A.C. Bathke: *Multivariate analysis of covariance with potentially singular covariance matrices and non-normal responses*. Journal of Multivariate Analysis, Volume 177(C), 2020. http://doi.org/10.1016/j.jmva.2020.104594.

BY NC ND