Una aplicación del modelo Leontief sobre la economía Española con técnicas *big data*

Martínez García, Vicente (martinez@uji.es)
Gregori Huerta, Pablo (gregori@uji.es)
Arnal Pons, Ana (parnal@uji.es)
Departamento de Matemáticas, Universitat Jaume I
Campus del Riu Sec, 12071 Castellón de la Plana

RESUMEN

El modelo de Leontief permite estimar la producción y ha sido utilizado en una amplia variedad de problemas, sobre todo en mercados que deben (o se considera que deben) mantenerse en equilibrio. La intervención matemática sobre el modelo consiste en resolver una ecuación matricial, la cual requiere el cálculo de la inversa de una matriz. Es conocido que para el caso de matrices de gran tamaño este cálculo resulta enormemente complicado, debido a las características de las operaciones que deben realizarse. Utilizaremos operaciones matriciales sparse (matrices de gran tamaño almacenadas de forma eficiente) para calcular la inversa de la matriz que se obtiene al considerar todas las empresas industriales de España. La matriz que resulta, en este caso, es poco densa, debido a que normalmente las empresas suelen tener conexiones en sectores próximos. Diseñaremos una metodología que nos permitirá estimar la producción de un grupo de empresas con supuestos reales con gran cantidad de datos. También, mostraremos como sería posible realizar estimaciones sobre la economía sumergida.

Palabras clave: modelos Leontief, *big data*, economía aplicada.
Área temática: aspectos cuantitativos de problemas económicos y empresariales.

ABSTRACT

The Leontief model allows estimating production and has been used in a wide variety of problems, especially in markets that must be kept in balance, for example those that depend on energy materials. The mathematical intervention on the model consists of solving a matrix equation, which requires the calculation of the inverse of a matrix. It is known that in the case of large matrices this calculation is enormously complicated, due to the characteristics of the operations that must be carried out. We will use sparse matrix operations (large matrices stored efficiently) to calculate the inverse of the matrix obtained when considering all industrial companies in Spain. The resulting matrix, in this case, is not very dense, because companies usually have connections in nearby sectors. We will design a methodology that will allow us to estimate the production of a group of companies with real assumptions with a large amount of data. Also, we will show how it would be possible to make estimates about the submerged economy.

1 INTRODUCCIÓN

El modelo de Leontief aproxima la producción de un grupo de empresas condicionadas por una demanda externa y de sus propias interconexiones de demanda interna. Resulta útil cuando se aplica sobre mercados que se considera deben mantenerse en equilibrio para evitar tensiones competenciales o estrategicas entre sectores sensibles (DYMOVA, SEVASTJANOV, PILAREK; 2013 y PRESLEY, WESSEH, BOQIANG; 2018). La estimación se basa en la resolución de un sistema lineal de ecuaciones, muy fácil de resolver cuando se trata de sistemas de tamaño pequeño, pero enormemente complicado cuando aparecen sistema de gran tamaño. La principal dificultad matemática radica en el cálculo de una matriz inversa, lo cual requiere una enorme cantidad de operaciones, como veremos posteriormente.

Supongamos que la economía de una región depende de n empresas, las cuales tienen que satisfacer una demanda interna y a la vez una demanda externa. Si para cada empresa i consideramos la producción final $\boldsymbol{x} = \{x_i\}_{i=1}^n$ como el vector incógnita de tamaño n, la matriz $A = \{a_{ij}\}_{i,j=1}^n$ representando las fracciones de la producción que la empresa i requiere de la empresa j y por $\boldsymbol{d} = \{d_i\}_{i=1}^n$ el vector de tamaño n representando la demanda externa que tiene la empresa i, el modelo es el siguiente:

$$\boldsymbol{x} = A\boldsymbol{x} + \boldsymbol{d},$$

y por tanto

$$I\boldsymbol{x} - A\boldsymbol{x} = \boldsymbol{d} \Rightarrow (I - A)\boldsymbol{x} = \boldsymbol{d} \Rightarrow \boldsymbol{x} = (I - A)^{-1}\boldsymbol{d}.$$

Si nuestro objetivo es aplicar el modelo de Leontief a la economía española, resulta obvio que el tamaño de la matriz $(I-A)^{-1}$ será grande. Consireando los datos que aparecieron en el periódico Expansión (31 de julio de 2018), España cuenta con 3,34 millones de empresas. Datos avalados por el Instituto Nacional de Estadística de España (INE), el cual restringe a 191.765 el número de empresas del sector industrial (INE 1; 2018).

Para solventar el problema que se plantea cuando se tienen que invertir matrices de gran tamaño, nosotros utilizaremos operaciones matriciales *sparse* de la libreria de **R, Matrix** (BATES *et. al.*; 2018), combinado con aproximaciones que resulten eficientes para dicho cálculo. En este sentido, y previamente a realizar operaciones con supuestos reales, precisamos conocer propiedaes matriciales que nos permitan justificar nuestro análisis, lo cual haremos en la Sección 2. En la Sección 3, realizaremos una simulación para estimar la producción de las empresas industriales de España. Posteriormente, en la Sección 4, mostraremos la posibilidad de estimar el fraude producido por la economía sumergida, también en España. Y finalmente en la Sección 5 detallaremos las conclusiones obtenidas.

2 ALGUNAS PROPIEDADES MATRICIALES

Las propiedades que enumeramos en esta sección pueden encontarse en la ámplia literatura existente sobre cálculo matricial, nosotros por motivos de simplicidad, solamente reseñamos tres de ellas (CONTRERAS, CORNETERO; 2010, FORSYTHE, MOLER; 1967 y HARTFIEL; 2000). También, en aras de una mayor claridad obviamos las demostraciones de las mismas.

Proposición 2.1 *Si M y N son matrices no singulares, entonces siempre que su producto tenga sentido se verifica:*

$$||M \cdot N|| \leq ||M|| \cdot ||N||$$

para cualquier norma matricial.

Proposición 2.2 *Si M es una matriz $n \times n$ no singular, entonces para todo vector $\boldsymbol{x}$ de n coordenadas se verifica:*

$$||M\boldsymbol{x}|| \geq \frac{1}{||M^{-1}||} \cdot ||\boldsymbol{x}||.$$

Proposición 2.3 *Si la serie numérica $\sum_{k=0}^{\infty} ||M_k||$ converge para alguna norma matricial, también converge la serie matricial $\sum_{k=0}^{\infty} M_k$.*

Proposición 2.4 *Si M es una matriz $n \times n$ con $||M||_p < 1$ entonces*

a) *$I - M$ es no singular.*

b) $\lim_{k\to\infty} M^k = \boldsymbol{0}$.

c) $(I - M)^{-1} = \sum_{n=0}^{\infty} M^k$.

d) $||(I - M)^{-1}||_p \leq \dfrac{1}{1 - ||M||_p}$.

Nota.- Observar que en la anterior proposición se ha utilizado la norma p, debido a que algunos resultados no son válidos para todas las normas. Para una mayor facilidad de seguimiento de la argumentación, a continuación recordamos algunas características de la norma p, $(1 \leq p < \infty)$.

Dado un vector $\boldsymbol{x} = \{x_i\}_{i=1}^n$, la norma p se define como

$$\|\boldsymbol{x}\|_p = \sqrt[p]{|x_1|^p + |x_2|^p + ... + |x_n|^p}.$$

Para el caso $p = 1$, se obtiene

$$\|\boldsymbol{x}\|_1 = |x_1| + |x_2| + ... + |x_n|,$$

y para $p = 2$, obtenemos la norma euclídea:

$$\|\boldsymbol{x}\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

Otra norma vectorial, también muy utilizada, es la norma infinito:

$$\|\boldsymbol{x}\|_\infty = \max(|x_1|, |x_2|, ..., |x_n|) = \max_{i \in \{1,...,n\}} |x_i|.$$

La norma p matricial inducida sería:

$$\|M\|_p = \max_{\boldsymbol{x} \neq 0} \frac{\|M\boldsymbol{x}\|_p}{\|\boldsymbol{x}\|_p}.$$

Para el caso $p = 1$, se verifica

$$\|M\|_1 = \max_{1 \leq i \leq n} \left\{ \sum_{j=1}^{n} |a_{ij}| \right\},$$

que resulta ser el máximo de la suma de los valores absolutos de los elementos de cada fila de la matriz M.

3 ESTIMACIÓN DE LA PRODUCCIÓN DE LAS EMPRESAS INDUSTRIALES ESPAÑOLAS

Hemos comentado anteriormente que las empresas industriales de España suman un total de 191.765 y dado que no se espera que todas tengan contactos con todas, resulta factible que la matriz A, la cual contendría los datos sobre las fracciones de sus interconexiones, sea una matriz con muchos ceros, es decir poco densa. Esto nos va a permitir utilizar librerias de cálculo de matrices sparse, para conseguir realizar los cálculos sin agotar la memoría RAM de nuestro ordenador.

Consideremos $\boldsymbol{x} = \{x_i\}_{i=1}^n$ como el vector incógnita que indica la producción de las empresas industriales españolas. Por tanto, $n = 191.765$. Consideremos conocida la matriz $A = \{a_{ij}\}_{i,j=1}^n$ representando las fracciones de la producción de las interconexiones entre empresas. Y finalmente, consideremos el vector $\boldsymbol{d} = \{d_i\}_{i=1}^n$ el vector representando la demanda externa.

Conviene aclarar, que el cálculo que vamos a realizar es un supuesto real, recalcando la palabra supuesto, ya que la matriz A es conocida únicamente por la Agencia Tributaria. Nuestro análisis, por tanto debería ser útil para las agencias tributarias de las instituciones públicas.

Algunas restricciones que deben verificarse:

R1. El vector demanda externa $\boldsymbol{d}$ y el vector solución $\boldsymbol{x}$, deben ser no negativos. No pueden haber demandas ni producciones negativas:

$$d_i \geq 0 \text{ y } x_i \geq 0, \ \forall i = 1, 2, \cdots, n.$$

R2. $a_{ij} \geq 0$, $\forall i, j = 1, 2, \ldots n$. Ya que la matriz A contiene fracciones, es decir porcentajes positivos en tanto por 1.

R3. La matriz A es poco densa. Las empresas mantienen interconexiones con empresas de sectores próximos y con el resto de sectores de forma muy esporádica.

R4. $\sum_{j=1}^{n} a_{ij} < 1, \; \forall i = 1, 2, \ldots, n$. Ya que los elementos de la fila i de la matriz A se corresponden con la producción de una empresa i en porcentajes de tanto por 1. De esta manera, la suma de cada fila no puede superar el 100%, en este caso 1.

R5. Además, obviamente no todas las componentes de los vectores y los coeficientes matriciales pueden ser todos nulos. Por que entonces no tendríamos una economía en funcionamiento.

La restricción R3 nos permite el uso de operaciones sparse dentro del paquete **Matrix** de **R**. Si hacemos un simple cálculo con este paquete generando una matriz diagonal de tamaño 180.000×180.000 utilizando, por una lado matrices en el sentido usual y por otro matrices con la utilidad *sparse:*

```
#Comprobar el uso de memoria RAM
> library(Matrix) #uso del paquete Matrix
> M<-Diagonal(180000, x = 1) #matriz sparse.
> dim(M)
[1] 180000 180000
> object.size(M)
1168 bytes
> N<-diag(180000, x=1) #matriz en el sentido usual.
Error: cannot allocate vector of size 241.4 Gb
```

Una vez ejecutadas en **R** los comandos indicados, comprobamos que la memoria RAM requerida para la matriz usual es de 241,4 gigabytes, mientras que para la matriz *sparse* es solamente de 1168 bytes. Esto nos da una idea de lo que podemos hacer y lo que no podenos hacer desde el punto de vista computacional.

Por otro lado, la proposición 2.4, conjuntamente con la restricción R4, nos permite obtener que $||A||_1 < 1$. Y además asegurar que la matriz inversa $(I - A)^{-1}$

existe y puede aproximarse por una suma finita de matrices:

$$(I-A)^{-1} \approx \sum_{k=0}^{r} A^k. \tag{1}$$

Veamos el algoritmo que hemos utilizado para justificar la viabilidad de nuestras estimaciones:

Paso 1: Construimos una matriz cuadrada de tamaño $n = 180.000$ de manera aleatoria con las restricciones del modelo indicadas. En este caso hemos tomado 540.000 valores no nulos, $d_i = 10.000$ para todas las empresas $i = 1, \ldots, n$ y una tolerancia $\varepsilon > 0$.

Paso 2: Calculamos $\boldsymbol{x}(1) = (I + A)\boldsymbol{d}$ y $\boldsymbol{x}(2) = (I + A + A^2)\boldsymbol{d}$.

Paso 3: Para $r \geq 1$, comparamos: $||\boldsymbol{x}(r) - \boldsymbol{x}(r+1)|| < \varepsilon$.

- Se cumple la comparación, detenemos las iteraciones y tomamos $\boldsymbol{x}(r+1)$ como solución del problema.
- En caso contrario calculamos $\boldsymbol{x}(r+2) = \left(\sum_{k=0}^{r+2} A^k\right) \boldsymbol{d} = \boldsymbol{x}(r+1) + A^{r+2}\boldsymbol{d}$. Hacemos $\boldsymbol{x}(r) = \boldsymbol{x}(r+1)$, $\boldsymbol{x}(r+1) = \boldsymbol{x}(r+2)$ y regresamos al inicio del punto 3.

Salida: $\boldsymbol{x}(r+1)$ solución aproximada.

De esta manera obtenemos una solución acorde con la restricción R1 del modelo. Obviamente, el vector obtenido $\boldsymbol{x}(r+1)$ es positivo, ya que es suma de matrices positivas multiplicadas por un vector positivo. En el peor de los casos $\boldsymbol{x}(r+1)$ tendrá algunos valores cero (alguna empresa puede que no tenga actividad), pero el resto deberá ser mayor que cero. Por tanto, resulta muy ventajoso, desde el punto de vista computacional, la posibilidad de utilizar la ecuación (1) para el cálculo de la inversa.

En nuestro experimento hemos utilizado una tolerancia $\varepsilon = 10$ unidades producidas y hemos precisado realizar 17 iteraciones para obtener una solución aproximada con esta tolerancia. El tiempo de computación ha sido de aproximadamente de 2 segundos utilizando un ordenador personal con procesador *intel CORE i5* de 8 Gb de memoria RAM. Lo cual confirma la viabilidad computacional del algoritmo utilizado.

4 ESTIMACIÓN DE LA ECONOMÍA SUMERGIDA

Es habitual encontrar en la prensa escrita noticias como las siguientes:

- **ELDIARIO.ES** (25 de novimbre de 2014): "El Gobierno canario cifra la economía sumergida en el 28% del PIB".
- **LA INFORMACIÓN** (1 de marzo 2016): "La ex presidenta del INE cifra la economía sumergida en el 20% o 25% del PIB".
- **LA VANGUARDIA** (20 de enero de 2018): "Un dinero que se escapa de las arcas del Estado porque no tributa y que en el caso español llega a representar hasta el 18% del PIB".
- **EXPANSIÓN** (21 de junio de 2018): "La economía sumergida resta un 23% a la recaudación en España".
- **EL PAÍS** (1 de julio de 2018): "Fraude fiscal y economía sumergida: una realidad mal conocida .... Los estudios más recientes calculan la economía negra en más del 20%".

Siguiendo con nuestro objetivo de considerar un supuesto realista, podríamos suponer que el conjunto de actividades económicas no declaradas que se encuentran fuera del control de la Agencia Tributaria es de un 20% de PIB. Si además sabemos que según los últimos datos del INE, el PIB de España es de 1.163.662 millones de euros, podemos concluir que la economía ilegal en España es de aproximadamente de 232.732 millones de euros (INE 2, 2018). Entre otras consideraciones delictivas graves, estos recursos dejan de cotizar a la Seguridad Social contribuyendo así a su déficit.

Para encajar estos datos en un modelo Leontief, vamos a considerar la matriz A de dimensiones $n \times n$ con $n = 3,24$ millones (número aproximado del total de empresas españolas) como las fracciones del dinero que cada empresa gestiona de forma legal, y por tanto conocidas por la Agencia Tributaria. El vector $\boldsymbol{x}$, en este caso será el total real de la producción en España, considerando en euros está producción, es decir, en nuestro análisis no vamos a considerar unidades producidas, sino dinero generado. El vector $\boldsymbol{x}$, en este caso, será la suma del PIB conocido por la Agencia Tributaria, $\boldsymbol{y}$, más la parte que aportaría el 20% de dinero fuera de su control $\boldsymbol{z}$. Así, $\boldsymbol{x} = \boldsymbol{y} + \boldsymbol{z}$. Con estos supuestos, nuestro objetivo se centra en encontrar $\boldsymbol{d}$ que en este caso será la parte que cada empresa aporta a la economía ilegal. Es decir queremos averiguar la cantidad d_i que cada empresa i, $(i = 1, 2, \cdots, n)$ "defrauda" a la Agencia Tributaria.

$$\boldsymbol{x} = A\boldsymbol{x} + \boldsymbol{d} \Rightarrow I\boldsymbol{x} - A\boldsymbol{x} = \boldsymbol{d} \Rightarrow \boldsymbol{d} = (I - A)\boldsymbol{x}.$$

En teoría el cálculo resulta más sencillo, sin embargo en este caso el tamaño de la matriz A es enorme y será necesario de nuevo utilizar el paquete **Matrix** de **R** para matrices *sparse.*

Los supuestos del apartado anterior son válidos, también en este caso, así consideraremos que dicha matriz es poco densa. Dado que los datos de la Agencia Tributaria no son públicos, nosotros los generaremos de manera aleatoria.

Por otra parte, el vector $\boldsymbol{y}$ es conocido, pero debe satisfacer

$$\sum_{i=1}^{n} y_i \;=\; 1.163.662 \times 10^6.$$

El vector $\boldsymbol{z}$ será la primera estimación del vector $\boldsymbol{d}$, y dado que en un principio no conocemos el fraude individual de cada empresa, supondremos que es el 20% del dinero generado por ella misma de manera legal. Resulta obvio decir que ésta es una estimación inicial, la mayoría de las empresas cumplen de manera rigurosa con la Agencia Tributaria, por tanto muchas componentes del vector $\boldsymbol{d}$ serán cero.

El algoritmo utilizado para justificar un supuesto realista será:

Paso 1: Construimos la matriz cuadrada A y el vector $\boldsymbol{y}$ con $n = 3.240.000$ de manera aleatoria y las restricciones del modelo indicadas. En este caso, 9.180.000 valores no nulos para la matriz A y 3.240.000 para el vector $\boldsymbol{y}$. Además tomamos una tolerancia $\varepsilon > 0$.

Paso 2: Calculamos $\boldsymbol{d}(1) = 0,2 \cdot \boldsymbol{y}$ y $\boldsymbol{d}(2) = (I - A)(\boldsymbol{y} + \boldsymbol{d}(1))$.

Paso 3: Para $r \geq 1$, comparamos: $||\boldsymbol{d}(r) - \boldsymbol{d}(r+1)|| < \varepsilon$.

- Se cumple la comparación, detenemos las iteraciones y tomamos $\boldsymbol{d}(r+1)$ como solución del problema.
- En caso contrario calculamos $\boldsymbol{d}(r+2) = (I - A)(\boldsymbol{y} + \boldsymbol{d}(r+1))$.

  Hacemos $\boldsymbol{d}(r) = \boldsymbol{d}(r+1)$, $\boldsymbol{d}(r+1) = \boldsymbol{d}(r+2)$ y regresamos al inicio del punto 3.

Salida: $\boldsymbol{d}(r+1)$ solución aproximada.

Para llevar a cabo nuestro experimento, en este caso, hemos utilzado el mismo ordenador personal que se utilizó en el experimento anterior. Tomando una tolerancia de 100 euros, para llegar a la salida del algoritmo precisamos realizar 20 iteraciones, utilizando 43.05 segundos de tiempo de computación. Si queremos una mayor precisión, por ejemplo de 1 euro, precisamos 47 iteraciones y 99.11 segundos de tiempo de computación.

El resultado es satisfactorio, demostrando la viabilidad de la metodología presentada: los cálculos son viables desde el punto de vista de la precisión y del tiempo de computación.

5 CONCLUSIONES

Se han llevado a cabo varios experimentos con supuestos reales, considerando datos de la economía española, lo cual nos ha conducido a manejar una gran cantidad de datos y la necesidad de utilizar el paquete **Matrix** de **R** para el tratamiento eficiente de dichos datos. Los datos empresariales utilizados en el modelo son datos fiscales, como no puede ser de otra manera son datos confidenciales y únicamente conocidos por la Agencia Tributaria Española. Por este motivo, para nuestro supuesto hemos generado dichos datos de manera aleatoria. La razón de utilizar esta gran cantidad de datos, ha sido con objeto de diseñar una metodología viable desde el punto de vista computacional.

Hemos utilizado datos poco densos para que fuese posible realizar la experimentación con un ordenador personal. Sin embargo, el resultado de la experimentación, ha puesto de manifiesto la utilidad de nuestra metodología, la cual pensamos podría ser aprovechable para las agencias tributarias oficiales. Dichas agencias podrán utilizar sus propios datos, éstos seguirán siendo poco densos, no en la medida en la que lo hemos supuesto nosotros para visualizar la efectividad del procedimiento. Sin embargo podrán utilizar una potencia de cálculo mucho mayor, dado que sus recursos informáticos son muy avanzados.

Cabe resaltar que el tiempo de cálculo requerido para la computación, ha sido irrisorio: solamente unos pocos segundos. Este hecho confirma que el uso de una metodología eficiente para manejar enormes cantidades de datos resulta viable.

Un aspecto matemático que no ha sido tratado en este trabajo, es el referente al estudio teórico sobre la convergencia de los algoritmos propuestos. Para ello sería necesario conocer los datos concretos de cada caso y analizar las propiedades de las matrices intervinientes.

6 REFERENCIAS BIBLIOGRÁFICAS

- BATES, D.; MAECHLER, M.; DAVIS, T.A.; OEHLSCHLÄGEL, J. RIEDY, J. y R-CORE-TEAM, (2018). "Matrix: Sparse and Dense Matrix Classes and Methods". CRAN Repository, `https://cran.r-project.org/web/packages/Matrix/index.html`.

- CONTRERAS, CH.D. y CORNETERO, E.J. (2010). "Solución del modelo input-output de Leontief aplicando la forma canónica de Jordan". Tesina, Universidad Nacional Pedro Ruíz Gallo, Perú.

- DYMOVA, L.; SEVASTJANOV, P. y PILAREK, M. (2013). "A method for solving systems of linear interval equations applied to the Leontief input–output model of economics". Expert Systems with Applications, 40, pp. 222–230.

- FORSYTHE, G. y MOLER, C.B. (1967). "Computer Solution of Linear Algebraic Systems". Prentice Hall. ISBN: 9780131657793.

- HARTFIEL, D.J. (2000). "Matrix Theory and Applications with MATLAB". CRC Press. ISBN: 1584881089.

- INE 1 (2018). `https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica\_C\&cid=1254736143952\&menu=ultiDatos\&idp=1254735576715.`

- INE 2 (2018). “España en cifras”. `https://www.ine.es/prodyser/espa_cifras/2018/files/assets/common/downloads/publication.pdf?uni=4f7e7b429c56ccbc4bf56b3e93ebc47b.`

- PRESLEY, K.; WESSEH, Jr. y BOQIANG, L. (2018). “Carbon pricing and general equilibrium under Leontief production technology”. Journal of Cleaner Production, 190, pp. 368–377.