

29.
17



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

**Implantación de Programas
Software para Realizar Técnicas
de Escalamiento Multidimensional.**

T E S I S

QUE PARA OBTENER EL TITULO DE:

A C T U A R I O

P R E S E N T A :

Manuel Gerardo Irazos López

México, D. F.

1984.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

PREFACIO

El objetivo de la presente tesis es el exponer los principales tipos de procedimientos para escalamiento:

El método de escalamiento clásico , y

El método de escalamiento ordinal

Las técnicas de escalamiento encaran esencialmente el problema de encontrar una configuración de puntos (o individuos u objetos) a partir de la información acerca de las distancias entre los puntos, las cuales a su vez pueden estar sujetas a error estocástico. Una característica importante de este tipo de procedimientos, es el poco uso de un modelo estocástico inherente. - Para algunos expertos en métodos de análisis estadístico de datos, las técnicas de escalamiento son métodos de naturaleza exploratoria, cuyo fin principal puede enmarcarse dentro del marco de generación de hipótesis, más que de la prueba de ellas. El tipo de enfoque que proveen estos procedimientos se describe generalmente como "Análisis de Datos", más que "Estadística".

Otro aspecto que caracteriza estas técnicas y las hace diferir en forma importante de otras técnicas multivariadas como Análisis Factorial, Análisis de Correlación Canónica, etc., es el hecho de que operan con una función de los puntos dados, i.e., las $n(n-1)/2$ distancias, disimilaridades o similitudes, más que con las coordenadas de los n puntos mismos. Es decir, las técnicas de escalamiento multidimensional operan con una matriz de $n \times n$ cuyos elementos comparan todos los pares de individuos u objetos de interés, ya sea al medir "similitud" o "disimilaridad". El capítulo 1 trata con cierto detalle las medidas de similitud, disimilaridad y distancia utilizadas más frecuentemente.

El capítulo 2 se ocupa del escalamiento clásico, el cual es esencialmente un método algebraico de reconstrucción de coordenadas que supone a las disimilaridades como distancias euclidianas. En el caso de que estas distancias estén sujetas a error, el comportamiento de este procedimiento es aún bastante satisfactorio. Originalmente propuesto por Torgerson (1952, 1968) - fué popularizado por Gower (1966), quien le dió el nombre de "Análisis de Coordenadas Principales". Se expone también en este capítulo la relación de esta técnica con el método llamado "Análisis de Componentes Principales".

El capítulo 3 expone los fundamentos del escalamiento no métrico o escalamiento ordinal . Cuando las disimilaridades observadas son de tal naturaleza que sus valores numéricos o magnitudes son de poco interés, y el rango de orden de éstas es la única información útil , es conveniente la utilización de una técnica alternativa desarrollada por R.N. Shepard y J.B. - - Kruskal a principios de la década de los sesentas, la cual utiliza solo las propiedades ordinales de las disimilaridades. Los desarrollos y alternativas propuestas hasta hoy, siguen los lineamientos originales propuestos por estos dos autores.

La comparación de dos configuraciones obtenidas por diferentes métodos o aun bajo el mismo pero utilizando una métrica diferente, es expuesto bajo el enfoque de una novedosa técnica debida a Gower llamada Rotación de Procrustes, de la cual se ocupará el capítulo 4.

El tema del capítulo 5 son los métodos numéricos utilizados para la implantación automática de estas técnicas que han sido desarrollados utilizando el lenguaje de programación Algol y la computadora B7800 de la UNAM. Algunos ejemplos de aplicación práctica que ilustran el uso de los procedimientos expuestos en los capítulos anteriores es lo tratado en el capítulo 6.

Agradezco al Dr. Alfonso Hernández Avila, coordinador de la Maestría en Estadística del I.I.M.A.S. , por su valiosa ayuda y apoyo en la realización de esta tesis.

Al Dr. Francisco Aranda Ordaz, Investigador del I.I.M.A.S., por la revisión y corrección del texto así como sus atinadas sugerencias.

A mis tíos Arq. Abel Iraizos , así como a su esposa Sra. Ma. de la Luz Ochoa de Iraizos por su apoyo durante la elaboración de este trabajo.

A la Srta. Ma. del Carmen Hernández por su paciente labor al mecanografiar el texto.

C O N T E N I D O

PREFACIO	i
CAPITULO 1	
MEDIDAS DE SIMILARIDAD Y DISIMILARIDAD	1
CAPITULO 2	
ESCALAMIENTO CLASICO	9
CAPITULO 3	
ESCALAMIENTO ORDINAL	17
CAPITULO 4	
ROTACION DE PROCRUSTES	31
CAPITULO 5	
METODOS NUMERICOS	34
CAPITULO 6	
EJEMPLO NUMERICO	49
APENDICE A	61
APENDICE B	63
APENDICE C	65
APENDICE D	68
BIBLIOGRAFIA	69

CAPITULO 1

MEDIDAS DE SIMILARIDAD Y DISIMILARIDAD

En algunas aplicaciones los datos a analizar provienen de una matriz D ($n \times n$) que consiste de similitudes o disimilitudes entre puntos, más que de una matriz X ($n \times p$) de coordenadas de los puntos, antes de continuar, es necesario definir precisamente lo que se considera una medida de similitud.

DEFINICION

Una medida razonable de similitud $s(A,B)$, debe tener las siguientes propiedades

$$i) s(A,B) = s(B,A)$$

$$ii) s(A,B) \geq 0$$

iii) $s(A,B)$ se incrementa a medida que la similitud entre A y B se incrementa, donde A y B son puntos que pueden representar medidas x , y en dos objetos distintos. Las medidas de similitud son frecuentemente llamadas coeficientes de similitud. Las medidas de similitud y disimilitud están cercanamente relacionadas en forma inversa. Supongamos que tenemos una función s , definida en cada par de individuos, que mide similitud. Una posible medida de disimilitud puede construirse como $d = \text{constante} - s$. Tanto las medidas de similitud como las de disimilitud pueden considerarse como funciones $P \times P$ (donde P es un conjunto de objetos) a los números reales. Jardine y Sibson (1971) definen una clase de funciones de disimilitud, llamados "coeficientes de disimilitud". Un coeficiente de disimilitud es una función d de $P \times P$ a los números reales positivos tal que

$$i) d(A,B) \geq 0 \text{ para todo } A, B \text{ en } P$$

$$ii) d(A,A) = 0 \text{ para todo } A \text{ en } P$$

$$iii) d(A,B) = d(B,A) \text{ para todo } A, B \text{ en } P.$$

Muchos de los procedimientos de clasificación automática y escalamiento pueden ser considerados como métodos en los cuales coeficientes de similitud o disimilitud son convertidos uno en otro (i.e., similitud a disimilitud, o, disimilitud a similitud). Un ejemplo es proporcionado por las técnicas de escalamiento multidimensional no métrico, las cuales son métodos para la transformación de una similitud o disimilitud numérica u ordinal, en un

arreglo métrico de distancias representable en un número pequeño de coordenadas euclidianas. Una función de valores reales $d(A,B)$ es una distancia si tiene las siguientes propiedades

- i) $d(A,B) = d(B,A)$ simetría
- ii) $d(A,B) > 0$ no negatividad
- iii) $d(A,A) = 0$ identificación

Para algunas funciones de distancia las siguientes propiedades también se cumplen :

- iv) $d(A,B) = 0$ si y solo si $A = B$
- v) $d(A,B) < d(A,C) + d(B,C)$ desigualdad del triángulo.

Si las propiedades i) - v) se satisfacen d es llamada una métrica. Para algunos propósitos es suficiente considerar funciones de distancia que satisfagan solamente i) - iii), sin embargo es conveniente en general considerar funciones de distancia para las cuales i) - v) son satisfechas. Es de esperarse que $d(P,Q)$ se incremente mientras la disimilaridad entre P y Q se incremente; por lo que $d(A,B)$ es descrito también como un coeficiente de disimilaridad, aun cuando no cumpla las propiedades de una métrica.

DISTANCIA EUCLIDEANA

Posiblemente la medida de disimilaridad más común sea la distancia euclidiana. Sea X una matriz de datos de $n \times p$ (que representa las coordenadas espaciales de cada uno de los n puntos) con filas x_i , $i=1, \dots, n$. Entonces la distancia euclidiana entre los puntos x_i y x_j es d_{ij} , donde

$$d_{ij} = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2} = |x_i - x_j|$$

La función d satisface las propiedades i) - v) además de verificarse:

a) Sea $A = (-\frac{1}{2} d_{ij}^2)$ Entonces HAH es positiva

semidefinida, donde $H = I - n^{-1} 11'$, es la matriz de "centrado" (centering - matrix), $1' (1 \times n) = (1, 1, \dots, 1)$

MÉTRICA DE MINKOWSKY

Una clase importante de funciones de distancia son las llamadas métricas en o métricas r de Minkowsky. Para cualquier $r > 1$, se define la r -distancia entre los puntos x_i ; y_j como

$$d_r(x_j, x_i) = \left\{ \sum_{s=1}^n |x_{is} - x_{js}|^r \right\}^{1/r}$$

d_r es una métrica (i.e., satisfacen i) - v). Las métricas r de Minkowsky comparten algunas propiedades con la métrica euclideana. En particular:

$$i) d_r(x_i, x_j) = d_r(x_i + x_k, x_j + x_k)$$

$$ii) d_r(kx_i, kx_j) = k d_r(x_i, x_j)$$

es decir, estas métricas son invariantes bajo traslaciones y deformaciones uniformes. Un punto de divergencia crucial entre la métrica euclideana y las métricas l_p , consiste en que la distancia bajo la métrica euclideana permanece invariable cuando se involucran rotaciones rígidas, mientras que las únicas rotaciones que dejan las distancias sin cambio bajo las métricas l_p , son aquellas que transforman ejes coordenados en ejes coordenados (i.e., aquellas transformaciones lineales cuya matriz asociada es unitaria), por lo que los ejes coordenados tienen una significancia para las métricas l_p que no tienen para la distancia euclideana.

Existen numerosas medidas de disimilaridad de probada utilidad que no satisfacen la desigualdad métrica. Tal es el caso del cuadrado de la distancia euclideana y la medida D^2 (la función de distancia de Mahalanobis). Es en consecuencia un requerimiento no necesario para una medida de disimilaridad - el cumplimiento de la desigualdad métrica para ser considerada de utilidad, - muy frecuentemente es de mayor importancia. Las propiedades de orden que posea la medida en cuestión, eg. Sibson (1972) define dos coeficientes d_1 , d_2 como de orden global equivalente si

$$d_1(w,x) < d_1(y,z) \text{ si y solo si } d_2(w,x) < d_2(y,z)$$

Si d_1, d_2 son de orden global equivalente entonces d_1, d_2 serán medidas de disimilaridad con el mismo rango de orden. Un ejemplo de este tipo de medidas de disimilaridad está dado por la distancia euclídeana y el cuadrado de ésta.

Hasta aquí consideramos solo medidas o coeficientes de disimilaridad. A continuación se darán algunos ejemplos de coeficientes de similitud.

MEDIDAS DE SIMILARIDAD

Variables Cualitativas

Sean la presencia o la ausencia de p atributos en dos objetos P y Q de notada (x_1, \dots, x_p) , (y_1, \dots, y_p) donde $x_i = 1$ o 0 dependiendo si el i -ésimo atributo está respectivamente ausente o presente en el objeto P .

Hacemos:

$$a = \sum x_i y_i$$

$$b = \sum (1-x_i) y_i$$

$$c = \sum x_i (1-y_i)$$

$$d = \sum (1-x_i) (1-y_i)$$

estos es: a, b, c, d son las frecuencias de $(1,1)$, $(0,1)$, $(1,0)$ y $(0,0)$. La medida más simple de similitud entre p y Q es $s_1(P, Q) = a / p$. Una alternativa es el coeficiente de apareamiento simple (simple matching coefficient).

$$s_2(P, Q) = \frac{a+d}{p}$$

donde $p = a+b+c+d$.

Una dificultad inherente a este coeficiente es la que acarrea la consideración de la ausencia en ambos objetos de cierta característica, en otras palabras, la presencia de un atributo en ambos individuos puede decir más acerca del parecido de estos, que la ausencia del atributo. El coeficiente de Jacquard fué diseñado en forma tal que salve el inconveniente mencionado.

$$S_3(P, Q) = \frac{a}{a+b+c}$$

Sea $X_{(n \times p)}$ una matriz de datos que contenga la información ausencia/ presencia de p atributos para n objetos. La matriz X se llama matriz de incidencia ya que $x_{ij} = 1, 0$.

Considerando el coeficiente de similaridad S_1 , la matriz de similaridades correspondiente es simplemente

$$S_1 = \frac{XX^T}{p}$$

mientras que la matriz de similaridades basada en S_2 es:

$$S_2 = \frac{XX^T + (J - X)(J - X)^T}{p}$$

donde $J = 11^T$

$$S_2 = s_{ij}^2$$

$$1^T = (1, \dots, 1)$$

VARIABLES MIXTAS

Si hay variables tanto cualitativas como cuantitativas, Gower (1971a) ha propuesto el siguiente coeficiente de similaridad entre el i -ésimo y j -ésimo puntos.

$$S^{(3)} = 1 - \frac{1}{p} \sum_{k=1}^p w_k |x_{ik} - x_{jk}|$$

donde $w = 1$ si k es cualitativa, $w = \frac{1}{R_k}$

si k es cuantitativa y donde R_k es el rango de la k -ésima variable.

Existen muchas otras medidas de las cuales no nos ocuparemos ya que no es tal el tema principal de esta exposición. Finalmente podemos citar a Goodman (1954) quien en su artículo clásico dentro de las medidas de asociación argumenta que: "idealmente, cada trabajo de investigación debería contar con una medida de asociación desarrollada para sus necesidades únicas". A través de los años los científicos han desarrollado muchas diferentes medidas. Por lo tanto, cuando se busca una medida de similaridad o asociación, hay raramente necesidad de desarrollar una nueva. Más que otra cosa el problema consiste en escoger sabiamente de entre las medidas ya existentes.

ESCALAMIENTO CLASICO

El escalamiento clásico es un método de reconstrucción algebraico para encontrar una configuración de puntos a partir de las disimilaridades entre los mismos. Este enfoque es particularmente apropiado cuando las disimilaridades son exactas, o, al menos aproximadamente euclidianas.

Supongamos que tenemos una configuración X en un espacio euclideo p -dimensional. Las distancias entre los puntos de la configuración se pueden calcular directamente a partir de la matriz $X = x_{ij}$ $i=1, \dots, n$; $j=1, \dots, n$. Una alternativa más conveniente es proporcionada por el uso de la matriz de $(n \times n)$ $B = XX^T$, la matriz de suma de productos y de cuadrados (la matriz de productos escalares entre vectores puntos) la matriz B , i.e., b_{rs} está dado por la siguiente expresión

$$b_{rs} = \sum_{k=1}^p x_{rk} x_{sk} \quad (1)$$

La matriz D $(n \times n)$ de cuadrados de distancias euclidianas de acuerdo a lo anterior, está dado por los elementos:

$$d_{rs} = \sum_{j=1}^p (x_{rj} - x_{sj})^2$$

$$= \sum_{j=1}^p x_{rj}^2 - 2 \sum_{j=1}^p x_{rj} x_{sj} + \sum_{j=1}^p x_{sj}^2$$

Es decir las distancias d_{ij} se pueden representar en función de los elementos de B o de las coordenadas x_{ij} .

Ahora supongamos que conocemos las distancias (y por consiguiente los cuadrados de estas) pero no las coordenadas de los puntos que las generan. Podemos imaginar el proceso de obtención de coordenadas como el problema inverso al que acabamos de exponer. La determinación consiste en dos etapas:

- i) La construcción de la matriz B
- ii) La factorización de la matriz B en la forma $B = XX^T$

Es evidente que a partir de (1) es posible encontrar b_{ij} en términos de las distancias conocidas, i.e., d_{ij} . Como en muchos otros problemas, la obtención de una solución única depende enteramente en la imposición de ciertas restricciones. En nuestro caso particular, tal restricción es del tipo de localización, y consiste en poner el centro de gravedad \bar{X} en el origen. - Tal restricción puede reformularse en la siguiente condición

$$\sum_{r=1}^n x_{rj} = 0$$

Ahora bajo esta restricción tenemos :

$$\sum_{s=1}^n b_{rs} = \sum_{s=1}^n \sum_{j=1}^p x_{rj} x_{sj} = \sum_{j=1}^p \sum_{r=1}^n x_{rj} x_{sj} = \sum_{j=1}^p x_{sj} \sum_{r=1}^n x_{rj} = 0$$

Esto es, la suma de términos en cualquier columna o fila de B es nula. Sumando (1) sobre r, sobre s y sobre r y s, tenemos:

$$\sum_r d_{rs}^2 = (b_{rr} + b_{ss} - 2b_{rs}) = T + nb_{ss}$$

$$\sum_s d_{rs}^2 = (b_{rr} + b_{ss} - 2b_{rs}) = T + nb_{rr}$$

$$\sum_{r,s} d_{rs}^2 = (b_{rr} + b_{ss} - 2b_{rs}) = nT + nT = 2nT$$

donde T es la traza de la matriz B.

Definimos

$$d_{r.}^2 = \frac{1}{n} \sum_{s=1}^n d_{rs}^2$$

$$d_{s.}^2 = \frac{1}{n} \sum_{r=1}^n d_{rs}^2$$

$$d_{..}^2 = \frac{1}{n} \sum_{r=1}^n \sum_{s=1}^n d_{rs}^2$$

$$b_{rs} = -\frac{1}{2} (d_{rs}^2 - b_{rr} - b_{ss})$$

$$b_{rs} = -\frac{1}{2} (d_{rs}^2 - d_{r.}^2 + \frac{T}{n} - d_{.s}^2 + \frac{T}{n})$$

$$b_{rs} = -\frac{1}{2} (d_{rs}^2 - d_{r.}^2 - d_{.s}^2 + \frac{2T}{n})$$

$$b_{rs} = -\frac{1}{2} (d_{rs}^2 - d_{r.}^2 - d_{.s}^2 + d_{..}^2)$$

Si D consiste en cuadrados de distancias euclidianas exactas, es posible demostrar que B es una matriz simétrica, positiva semidefinida. Si B es de rango $k < n$, entonces tendrá k valores propios distintos de cero. Sean l_1, l_2, \dots, l_p tales valores, los vectores propios de longitud unitaria $\{e_i\}$ son normalizados de tal manera que la suma de los cuadrados de sus entradas sea igual l_i . Esto se logra haciendo $f_i = l_i e_i$. Mostraremos que una posible matriz de coordenadas está dada por

$$X = | f_1, f_2, \dots, f_k |$$

La matriz X es de dimensión $n \times k$, así que en consecuencia tenemos una configuración en k dimensiones. El hecho de que una matriz positiva semidefinida pueda ser factorizada en la forma XX' depende del teorema de factorización de Young-Householder. El hecho de que X es de la forma $X = | f_1, \dots, f_k |$ depende solamente en mostrar que si X tiene esta forma, entonces para $i=1, \dots, k$

$$(XX^*)e_i = X \begin{array}{c} f^* \\ 1 \\ : \\ : \\ : \\ : \\ f^* \\ k \end{array} = X \begin{array}{c} e_i \\ = X \\ 0 \\ : \\ 1 \\ i \\ : \\ 0 \end{array} = X f_i = l_i e_i^*$$

-(i-ésima fila)

Ya que e_i son ortonormales y $f_i = l_i e_i$. Pero $Be_i = l_i e_i$ por ser e_i un vector propio; para $i=1, \dots, k$. Ya que los e_i forman una base ortonormal se concluye que $B = XX^*$.

En resumen, la matriz X puede ser vista en terminos de los vectores propios de B y los puntos correspondientes

Vectores Propios

		l_1	l_2	l_p
u	P_1	x_{11}	$x_{12} \dots$	x_{1p}
n	P_2	x_{21}	$x_{22} \dots$	x_{2p}
o	P_3	x_{31}	$x_{33} \dots$	x_{3p}
s	:	:	:	:
	P_n	x_{n1}	$x_{n2} \dots$	x_{np}

Es decir, la r -ésima fila de X contiene las coordenadas del r -ésimo punto mientras que la r -ésima columna de X contiene al vector propio asociado a l_i . Todo lo dicho hasta aquí es válido mientras B no presente valores propios negativos, debido a la imposibilidad de normalizar un vector que tenga norma cuadrada negativa.

MEDIDAS DE BONDAD DE AJUSTE

Una vez encontrada la configuración asociada a una matriz de distancia D , esperaremos que al escoger una dimensión baja para el espacio en el cual se encuentra, las distancias entre los puntos de ésta, se asemejarán cercanamente a las entradas de la matriz considerada D .

Consideremos una configuración X en R^p , un enfoque particularmente apropiado al visualizar una configuración en k dimensiones, es considerarla como la proyección de la configuración sobre un espacio de R^p generada por las columnas de una matriz ortogonal L_1 , siendo más explícitos

$$\text{Sea } L_{(p \times p)} = (L_1, L_2), \quad L_1_{(p \times k)}$$

$$\hat{X} = XL_1$$

al ser L ortogonal, las distancias entre las filas de X y aquellas de XL , serán iguales i.e.

$$d_{rs}^2 = \sum (x_{ri} - x_{si})^2 = \sum (x'_r l_{(i)} - x'_s l_{(i)})^2$$

Sea \hat{D} la matriz de distancias entre las filas de XL_1 , entonces

$$\hat{d}_{rs} = \sum_{l=1}^k (x'_r l_{(l)} - x'_s l_{(l)})^2$$

de aquí se sigue en forma directa la desigualdad

$$\hat{d}_{rs} \leq d_{rs}$$

Es un hecho digno de consideración darse cuenta de la importancia de este resultado, pues podríamos considerar la discrepancia entre la configuración original y la configuración ajustada como cuantificada por

$$\emptyset = \sum (d_{rs} - \hat{d}_{rs})^2$$

Es posible demostrar que de entre todas las proyecciones XL_1 de X sobre espacios k -dimensionales de R^p , \emptyset se minimiza cuando X se proyecta sobre -

sus coordenadas principales en k dimensiones, obteniendo que el valor correspondiente para esta proyección en coordenadas principales

$$\phi = 2n (\lambda_{k+1} + \dots + \lambda_p)$$

Cuando D es no necesariamente euclídeana es más conveniente trabajar con la matriz $B = HAH$, $H = I - n^{-1} 1'1$, $1 = (1, \dots, 1)$. Si \hat{X} es una configuración ajustada con matriz asociada \hat{B} entonces una medida de discrepancia entre B y \hat{B} está dada por

$$\Psi = \sum_{r,s=1}^n (b_{rs} - \hat{b}_{rs})^2 = \text{tr}(B - \hat{B})^2$$

encontrándose que al igual que en el caso anterior que para una k fija la medida Ψ es minimizada sobre todas las configuraciones X en k dimensiones - cuando X es la solución clásica al problema del escalamiento multidimensional, hallándose que el valor mínimo de Ψ está dado por

$$\Psi = \lambda_{k+1}^2 + \dots + \lambda_p^2$$

De aquí que la discrepancia total entre la configuración en cuestión X y cualquier otra X ajustada está dada al considerar la suma total de los valores característicos en cada caso y no a partir de k+1.

Lo anterior sugiere posibles medidas de concordancia para "la proporción de matriz de distancia explicada". Suponiendo $\lambda_k > 0$, Mardia (1970) propone

$$\alpha_{1,k} = \left(\sum_{i=1}^k \lambda_i / \sum_{i=1}^n \lambda_i \right) \times 100 \%$$

$$\alpha_{2,k} = \left(\sum_{i=1}^k \lambda_i^2 / \sum_{i=1}^n \lambda_i^2 \right) \times 100 \%$$

RELACION ENTRE EL ESCALAMIENTO CLASICO Y EL
ANALISIS DE COMPONENTES PRINCIPALES

En esta sección se describe la conexión entre el escalamiento clásico y el análisis de componentes principales, suponiendo que una matriz $X_{n \times n}$ corregida por la media es dada, en lugar de una matriz de disimilaridad (para una definición de componentes principales ver el apéndice B). Como se vió en la sección anterior, el problema del escalamiento clasico, se reduce básicamente a encontrar los valores propios y los correspondientes vectores propios - asociados a la matriz XX' . Para efectuar un análisis de componentes principales es necesario encontrar los valores y vectores propios de la matriz de covarianza muestral $X'X$. Una pregunta natural que surge, es si existe alguna - conexión entre los valores y vectores característicos de XX' y $X'X$, si alguna relación existe, entonces habrá una liga interesante entre estas técnicas.

Sean λ_1, a_1 los valores y vectores propios de $X'X$ entonces $X'Xa_1 = \lambda_1 a_1$

Premultiplicando por X , encontramos

$$(XX')Xa_1 = \lambda_1 Xa_1$$

Lo anterior nos permite visualizar que los valores característicos de XX' son los mismos que aquellos de $X'X$, mientras que los vectores característicos de $X'X$, a_1 estarán relacionados por una transformación lineal simple a los de XX' , pues e_1 debe ser proporcional a Xa_1 .

Es importante observar que e_1 es un vector de orden $(n \times 1)$ mientras a_1 es de orden $(p \times 1)$. Si suponemos que tanto a_1 como e_1 han sido normalizados de tal forma que la suma de cuadrados de sus entradas sea unitaria, Xa_1 es un vector que da los puntajes (scores) para cada individuo en relación a la i -ésima componente, y la suma de cuadrados de Xa_1 se sabe que es λ_1 . Por lo tanto debemos tener

$$\lambda_1 a_1 = Xa_1$$

(posiblemente despues de cambiar el signo de e_1 o a_1).

Esto tiene como consecuencia inmediata que, es posible obtener los punta

jes (scores) de las componentes directamente de los vectores propios XX' en lugar de transformar los vectores propios de $X'X$. La fórmula que relaciona lo anterior es:

$$f_i = \frac{1}{\lambda_i} e_i$$

Por lo tanto, los resultados de un análisis de componentes principales son equivalentes (salvo posiblemente una reflexión) a los obtenidos del escalamiento clásico. Gower (1966) muestra que los resultados anteriores se aplican también a una matriz de datos que consista de ceros y unos, si las distancias se calculan con el coeficiente de apareamiento simple (simple matching coefficient). La equivalencia del análisis de componentes principales y el escalamiento clásico significa que, si queremos reducir la dimensionalidad efectiva de una matriz de datos al transformar a nuevos ejes coordenados, entonces, no tiene sentido efectuar ambos análisis. Esto implica que si $n > p$, entonces un análisis de componentes principales se preferirá generalmente, ya que es ciertamente más fácil encontrar los valores propios de una matriz $(X'X)_{p \times p}$ que de una matriz $(XX')_{n \times n}$.

El escalamiento clásico es sorprendentemente robusto a alejamientos de la distancia euclídeana. Errores aleatorios comparativamente grandes pueden ser añadidos a las distancias euclídeanas, sin distorsionar seriamente a la configuración resultante; aun si tomásemos una función monótona de las distancias, obtendríamos una configuración razonable. Sin embargo, si las disimilaridades no son ni siquiera aproximadamente euclídeanas, entonces el escalamiento clásico no es recomendable. Esta situación puede ser indicada al encontrar que B tiene un valor propio negativo "grande" o al encontrar que B tiene varios valores característicos positivos de tamaño medio, así que la solución trata de dispersarse en un número grande de dimensiones. En estas situaciones es preferible una técnica que utiliza el rango de orden de las disimilaridades, la cual será descrita en el capítulo siguiente.

CAPITULO 3

ESCALAMIENTO NO METRICO

Definición del problema

El objetivo del escalamiento no métrico en forma general, es el de encontrar n puntos tales que, las distancias entre ellos se asemejen en algún sentido a las disimilaridades experimentales entre n objetos, utilizando exclusivamente el conocimiento del rango de orden de las disimilaridades observadas entre los objetos de interés. No existe razón por la cual no sea posible el empleo de medidas de proximidad o disociación de cualquier otro tipo. Tales medidas incluirían por ejemplo: coeficientes de correlación, probabilidades de confusión, relaciones de interacción entre grupos, etc. El hecho de que un valor grande implique cercanía o lejanía, es algo que no tiene relevancia alguna. Lo que es esencial para el empleo de esta técnica, es la existencia de una relación monótona, ya sea creciente o decreciente, entre las medidas experimentales y las distancias en la configuración.

UN POCO DE HISTORIA

Durante la época de aparición del libro de Torgerson (1958), los procedimientos de escalamiento en uso corriente podrían describirse como:

i) Procedimientos generales que pueden desglosarse en dos etapas. La primera emplea una técnica de escalamiento unidimensional que tiene como objeto convertir las similaridades o disimilaridades en distancias. La segunda etapa tiene como meta encontrar puntos cuyas distancias entre sí, tengan aproximadamente los valores encontrados en la primera etapa. Un gran avance en este campo fué la aparición de una técnica debida a Shepard, que marcaría el camino a seguir en investigaciones posteriores, que pueden describirse como:

ii) Procedimientos generales que adoptan como característica principal, la obtención de una relación monótona entre las disimilaridades o similaridades experimentales y las distancias en la configuración. Los procedimientos mencionados en i), debido a la naturaleza de las técnicas de escalamiento unidimensional disponibles en aquella época, aceptaban ya sea disimilaridades promediadas o alguna transformación fija de ellas como distancias, o usaban la variabilidad de los datos como elemento crítico en la formación de las distancias.

Los procedimientos en ii) establecen que la satisfactoriedad de una solución debe ser juzgada por el grado en que la condición de monotonía es satisfecha. Shepard muestra que simplemente pidiendo un alto grado de satisfactoriedad en este sentido, y, sin hacer uso de la variabilidad en ninguna forma, es posible obtener soluciones fuertemente restringidas, recuperando al mismo tiempo la forma de las supuestas, pero desconocidas relaciones monótonas; es decir, para la determinación de la solución es necesario solamente conocer el rango de orden de las disimilaridades. No está por demás recalcar que lo anterior convierte a esta clase de procedimientos, ajenos a cualquier suposición distribucional, así como de suponer que las disimilaridades y las distancias están relacionadas por alguna fórmula fija. Esta solución se conoce con el nombre de escalamiento multidimensional.

A pesar de la consideración que se hace acerca de la calidad de la solución a este problema (el de el escalamiento multidimensional), no se incorpora la medida de alejamiento de la condición de monotonía, como criterio principal para la terminación de los procedimientos.

Shepard (1962) describe e implementa un procedimiento iterativo con la ayuda de una computadora cuyos lineamientos generales se ajustan aquellos mencionados en ii).

Kruskal (1964) en un intento por fundamentar y perfeccionar las ideas de Shepard propone una técnica similar a la de regresión por mínimos cuadrados, la que a su vez se inscribe dentro del marco central de una relación monótona entre las disimilaridades y las distancias.

Brevemente la técnica de Kruskal consiste en lo siguiente:

Dada una configuración, efectúa una regresión monótona de distancia con disimilaridad y usa la varianza residual, como una medida cuantitativa de alejamiento de la relación monótona, a la que llama stress. Define una solución el problema de escalamiento multidimensional como la configuración que minimiza el stress, dando de esta manera una definición matemática explícita de una solución, lo cual había hasta ese momento permanecido extrañamente descuidado por muchos autores, incluyendo al propio Shepard.

Sea $\&_{ij}$ la disimilaridad obtenida experimentalmente entre los objetos i y j . Suponemos que el procedimiento experimental es inherentemente simétrico de manera que $\&_{ij} = \&_{ji}$ ignorando las autodisimilaridades (i.e, $\&_{ii}$).

Consideremos n objetos, entonces existen $n(n-1)/2$ números $\&_{ij}$. Suponemos además la siguiente condición: $\&_{ij} \neq \&_{kl}$ para toda $i \neq k$ y para toda $j \neq l$

La condición anterior nos habilita para ordenar las disimilaridades - en estricto orden creciente:

$$\psi \quad \&_{i(1)j(1)} < \dots < \&_{i(M)j(M)}$$

$$\text{donde } M = n(n-1)/2$$

Deseamos representar los n objetos como n puntos en un espacio t -dimensional donde por el momento suponemos que sabemos el valor apropiado de t . Más tarde se abordará la cuestión de la determinación de tal valor.

Sea $X = \{ x_i \mid i=1, \dots, n; x_i \text{ en } R^t \}$ llamamos a X una configuración. Esperamos encontrar la configuración que mejor se ajuste a un conjunto dado de puntos, por lo que es necesario la construcción de un criterio que nos permita juzgar la bondad de cualquier configuración. Tal criterio se desarrolla a continuación:

EL STRESS

Sea d_{ij} la distancia que existe entre x_i y x_j (donde por el momento se considerara la distancia euclídeana y más tarde al igual que con las restricciones hechas a las disimilaridades, serán sustituidas por un esquema de suposiciones más generales).

Intuitivamente la aproximación perfecta a las disimilaridades a partir - de las distancias, se daría en el caso de que se observase lo siguiente:

$$\&_{i(K)j(K)} < \&_{i(1)j(1)} \text{ implica } d_{i(K)j(K)} < d_{i(1)j(1)}$$

en otros términos la aproximación perfecta se da cuando las distancias entre

los puntos de la configuración presentan el mismo orden que las correspondientes disimilaridades

$$d_{i(1)j(1)} < \dots < d_{i(M)j(M)}$$

Examinemos el diagrama que se presenta en la figura 3.1. Como pueden observarse, consta de puntos con abscisa d_{ij} y ordenada \hat{d}_{ij} . Este tipo de diagramas son conocidos como diagramas de dispersión (scatter diagrams). Podemos hacer uso de los diagramas de dispersión para visualizar que también las distancias se asemejan a las disimilaridades. En términos de diagramas de dispersión, el concepto intuitivo de aproximación perfecta, se traduce que al trazar cada punto $(d_{i(k)j(k)}, \hat{d}_{i(k)j(k)})$ empezando con aquel con menor ordenada y continuando de esta forma hasta llegar a aquel con ordenada mayor, nuestra dirección de trazo sea siempre a la derecha, nunca a la izquierda, fig. 3.2. Para medir el alejamiento de una configuración de la aproximación perfecta, es natural ajustar una curva que represente la aproximación perfecta al conjunto de puntos $(d_{i(k)j(k)}, \hat{d}_{i(k)j(k)})$ $k = 1, \dots, n(n-1)/2$ y después medir la desviación de cada punto a la curva. Las desviaciones entre la curva y los puntos se medirán a lo largo del eje de las distancias. Es conveniente hacer una reflexión en cuanto a lo anterior. ¿Que implicaría medir las desviaciones entre la curva y los puntos a lo largo del eje de las disimilaridades?. Implicaría ciertamente, dar un valor numérico a los rangos, es decir, hacer uso de otra información que solo el rango de orden de las disimilaridades. Lo anterior, en consecuencia, no sería la base para la construcción de una medida de bondad de ajuste, en la que no hay efecto de distorsión monótona en el eje de las disimilaridades. La forma en que se medirán las desviaciones implica que de la curva ajustada se necesitan solo M puntos, i.e.,

$$(d_{i(k)j(k)}, \hat{d}_{i(k)j(k)})$$

De aquí que ajustar la curva se reduce a ajustar los valores de $\hat{d}_{i(k)j(k)}$

Por supuesto, es necesario recalcar que los números d_{ij} no son distancias. No existe necesariamente una configuración de puntos cuyas distancias entre sí sean \hat{d}_{ij} . Estos números son meramente una sucesión monótona de números que se escogen tan cerca de las d_{ij} como sea posible, los cuales se usan como referencia de la extensión en la cual los números d_{ij} se alejan de la relación de monotonía. La definición exacta de los números \hat{d}_{ij} se hará más tarde con el -

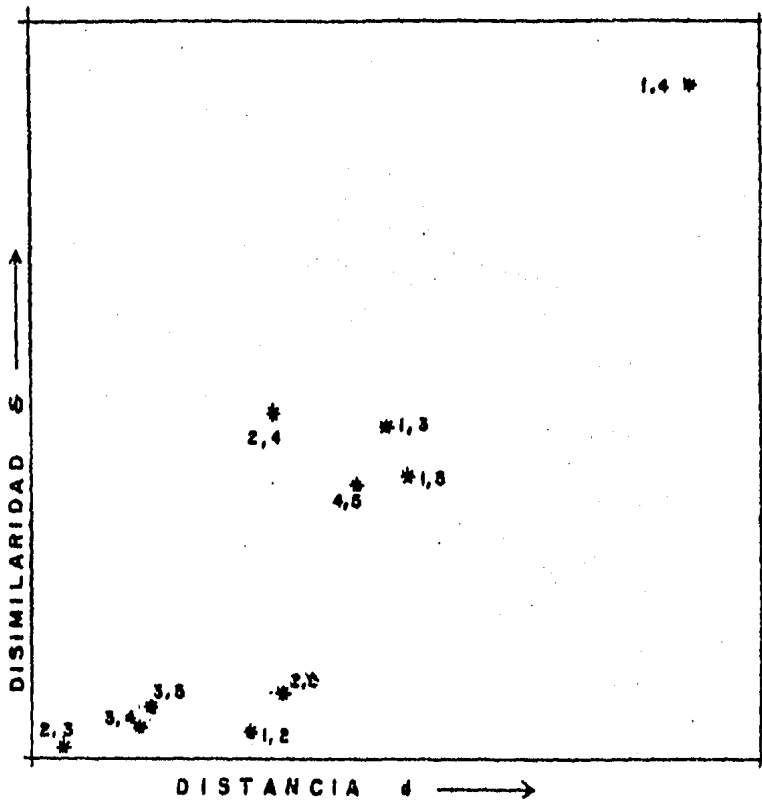


FIG. 3.1

DIAGRAMA DE DISPERSION

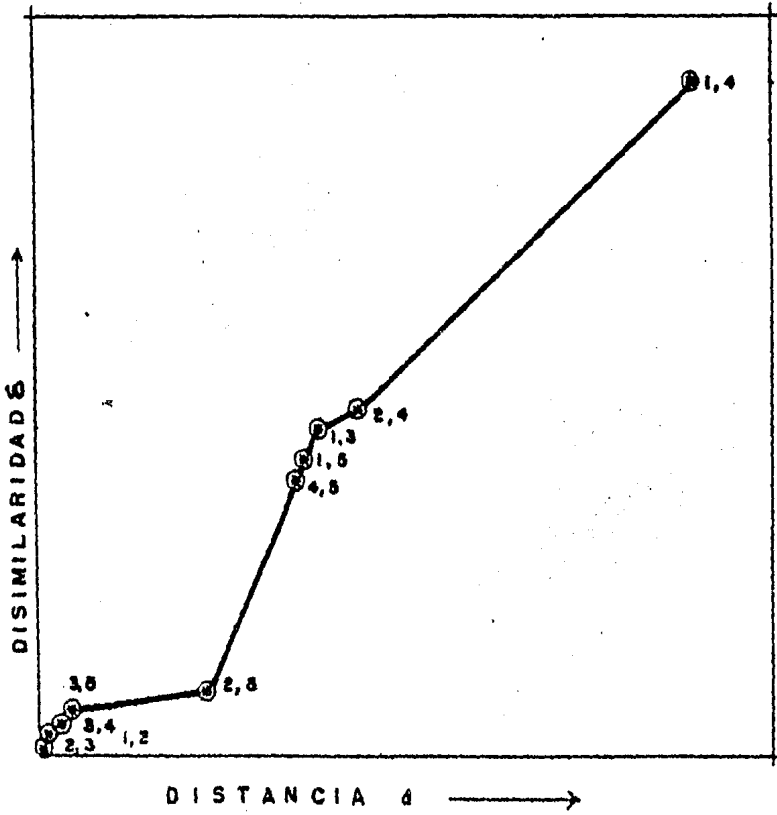


FIG. 32

RELACION ASCENDENTE PERFECTA

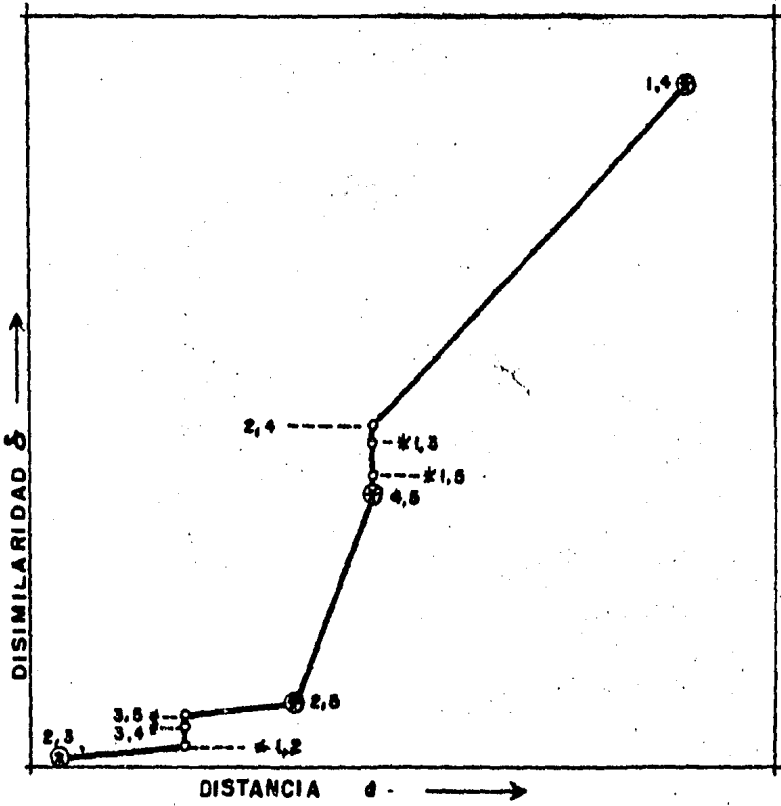


FIG. 3.2 (b)

RELACION ASCENDENTE IMPERFECTA

fin de simplificar la exposición. La curva ajustada es desde luego de carácter ascendente, esto es, con las mismas características que las expuestas en términos del diagrama de dispersión para una configuración con aproximación perfecta. Haciendo referencia a los números \hat{d}_{ij} , esto significa que cuando son arreglados en el orden normal, i.e.,

$$d_{i(1)j(1)}, \dots, d_{i(M)j(M)}$$

entonces se verificará la siguiente relación

$$\hat{d}_{i(1)j(1)} < \dots < \hat{d}_{i(M)j(M)}$$

De acuerdo a lo anterior el diagrama de la figura 3.2(b) muestra una relación ascendente imperfecta, debido a que el conjunto de distancias no guarda la misma relación ascendente que las correspondientes disimilaridades, en otras palabras, $\&$ y d no son de orden global equivalente (Sibson (1972)). Un ejemplo lo proporciona el par de puntos 1,2 ; 3,4 correspondientes a $(d_{12}, \&_{12})$, $(d_{34}, \&_{34})$. Para este par en particular observamos

$$\&_{12} < \&_{34} \text{ no implica } d_{12} < d_{34}$$

La misma situación se hace evidente al observar los pares de puntos $(d_{24}, \&_{24}), (d_{15}, \&_{15}) ; (d_{12}, \&_{12}), (d_{35}, \&_{35})$. En la figura 3.2 (b) se muestra además la curva ajustada por regresión monótona a los puntos $(d_{i(k)j(k)}, \&_{i(k)j(k)})$. Los puntos $(d_{i(k)j(k)}, \&_{i(k)j(k)})$ se han representado en el diagrama por medio de un *, y los puntos $(\hat{d}_{ij}, \hat{\&}_{ij})$ por medio de un pequeño círculo. En caso de que el punto ajustado coincida con el punto observado, se representa por un asterisco encerrado en un círculo. La curva ajustada es finalmente representada por un conjunto de líneas rectas que unen los puntos ajustados. la figura 3.2 muestran una relación ascendente perfecta, donde, de acuerdo a la simbología usada, se aprecia que los puntos ajustados coinciden perfectamente con los puntos observados, esto quiere decir simplemente que, dado cualquier par de disimilaridades $\&_{i(K)j(K)}$, $\&_{i(1)j(1)}$ tales que $\&_{i(K)j(K)} < \&_{i(1)j(1)}$ las correspondientes distancias cumplen $d_{i(K)j(K)} < d_{i(1)j(1)}$ esto es $\&$ y d son de orden global equivalente o están relacionadas monótonamente. Cuando cualquier conjunto de números satisface las desigualdades anteriores, se dice que están relacionados monótonamente a las distancias d_{ij} . Supongamos que tenemos los valores ajustados

\hat{d}_{ij} , definimos

$$S^* = \sum_{i < j} (d_{ij} - \hat{d}_{ij})^2$$

como nuestra medida tentativa de bondad de ajuste (stress no normalizado), que no es otra cosa que la suma de cuadrados residuales asociados a muchas técnicas de ajuste. Esta medida presenta dos particularidades especiales

1) el medir desviaciones en el eje de las distancias

2) la curva ajustada se escoge no de una familia de curvas paramétricas (i.e., polinomios o series trigonométricas), sino de una familia de curvas no paramétricas: la familia de curvas monótonas ascendentes. Esta medida es invariante bajo movimientos rígidos de la configuración (i.e., rotaciones, traslaciones y reflexiones), sin embargo, no es invariante bajo deformaciones uniformes, esto es si afectamos a cada punto de la configuración por el escalar k ($f(x_i) = kx_i$), el stress cambia de S^* a $k^2 S^*$ pues los números d_{ij} , \hat{d}_{ij} se afectan por el mismo factor. Cualquier deformación uniforme (expansión, reflexión o contracción) al no afectar la relación de orden entre las distancias, debería por lo tanto mantener inalterada la medida de bondad de ajuste. Una forma obvia de remediar este defecto en el stress, es dividirlo por un factor de escala, que tenga la misma dependencia cuadrática en la escala que éste. Un factor adecuado es:

$$T^* = \sum_{i < j} d_{ij}^2$$

Por lo tanto

$$S^* / T^* = \frac{\sum_{i < j} (d_{ij} - \hat{d}_{ij})^2}{\sum_{i < j} d_{ij}^2}$$

es una medida que posee las propiedades originales de S^* con las ventajas de ser invariante bajo cambios de escala. Finalmente, es deseable utilizar la raíz cuadrada de esta expresión, lo cual es equivalente a utilizar la desviación standard en lugar de la varianza, con lo que la definición de stress nor-

malizado es :

$$S^2 = \frac{S^*}{T^*} = \frac{\sum_{i < j} (d_{ij} - \hat{d}_{ij})^2}{\sum_{i < j} d_{ij}^2}$$

Esta medida nos representa la bondad con la que una configuración representa los datos. Mientras menor stress, mayor ajuste. Un stress nulo representa un ajuste perfecto dentro de nuestro contexto particular.

La definición de los números \hat{d}_{ij} que fué aplazada hasta este momento, será dada en terminos de la función stress a la que acabamos de llegar. Las cantidades \hat{d}_{ij} son aquellas que minimizan S (y en consecuencia S^*) sujetas a la restricción de monotonía. Utilizando la anterior es posible condensar la definición de stress en la siguiente fórmula:

$$S^2(x_1, \dots, x_n) = \min_D \frac{\sum_{i < j} (d_{ij} - \hat{d}_{ij})^2}{\sum_{i < j} d_{ij}^2}$$

$$D = \{ d_{ij} \mid d_{i(1)j(1)} < \dots < d_{i(M)j(M)} \}$$

Esta minimización es lograda por un algoritmo rápido y eficiente que se describe en el capítulo 6. No está por demás recalcar que, la definición de stress nos provee de medios para la evaluación de cualquier configuración. La configuración que se considera solución al problema del escalamiento multidimensional es obviamente aquella a la cual está asociada el stress mínimo para un valor particular de t (la dimensión del espacio modelo). Teóricamente la obtención de la configuración con mínimo stress, es equivalente a minimizar la función $S(x_1, x_2, \dots, x_n)$ sujeta a la restricción

$$d_{ij} < d_{i'j'} \text{ cuando } \&_{ij} < \&_{i'j'}$$

Este problema será abordado en el capítulo siguiente.

EL PROBLEMA DE LA DIMENSION

En la práctica es por lo general desconocido el número de dimensiones a emplear. Es finalmente el criterio científico del investigador en donde reside la determinación del número de coordenadas a recuperar a través de los datos. Kruskal (1964) sugiere dos lineamientos generales para la determinación de t o que suponen cierta ayuda para tal fin. Estos dos criterios son intuitivamente obvios y de sentido común.

El primer criterio sugiere efectuar varios análisis utilizando diferentes valores de t , la construcción de una gráfica que muestre la dependencia del stress mínimo y la dimensión. Un punto importante debe ser tomado en cuenta en este paso. A medida que el valor de t aumenta, el stress asociado disminuye. Para $t > n-1$, el stress asociado es nulo. (Siempre se puede lograr un ajuste perfecto con n puntos en $n - 1$ dimensiones). Es razonable establecer un compromiso entre el valor de t y el stress asociado, consistente en la elección de un valor de t aceptablemente pequeño, para el cual, valores mayores de t no reduzcan en forma significativa el stress. Datos cualitativamente buenos exhiben un "codo" notable en la curva, apuntando el valor apropiado de t . El segundo criterio descansa enteramente en la interpretación de las coordenadas. Si la solución t -dimensional provee una interpretación satisfactoria, pero si la solución $(t+1)$ -dimensional no muestra más una estructura, puede ser entonces conveniente usar solamente la solución t -dimensional.

VIOLACION A LAS HIPOTESIS Y DATOS FALTANTES

El desarrollo anterior se hizo bajo ciertas hipótesis básicas en cuanto a las disimilaridades. Se expone a continuación el procedimiento a seguir en el caso de violación a la simetría e igualdad de disimilaridades.

NO SIMETRIA

Supongamos que el procedimiento de medida no es inherentemente simétrico i.e., $d_{ij} \neq d_{ji}$. Si suponemos que la diferencia observada se debe simplemente a fluctuación estadística, podemos proceder utilizando dos enfoques naturales:

i) Formar medidas simétricas promediando los valores de las disimilaridades iguales \hat{d}_{ij} , \hat{d}_{ji}

ii) Generalizar la definición de stress dejando correr las sumatorias tanto en el numerador como en el denominador sobre $i \neq j$ (en lugar de solo $i < j$) En algunos casos pueden tener significado las autodisimilaridades y ser deseable la inclusión del caso $i=j$.

DISIMILARIDADES IGUALES

Los números \hat{d}_{ij} fueron definidos como aquellos que minimizan S sujetos a la restricción que estén monótonamente relacionados a las disimilaridades \hat{d}_{ij} . El problema de interpretación de esta restricción tiene dos enfoques, a saber primario y secundario.

a) Enfoque Primario

En el caso $\hat{d}_{ij} = \hat{d}_{kl}$, el valor que tomen las correspondientes distancias \hat{d}_{ij} , \hat{d}_{kl} no es considerado relevante, en consecuencia, no afectará la configuración al ocurrir $\hat{d}_{ij} \neq \hat{d}_{kl}$, y es natural pensar que el stress no debe reflejar tal situación. La forma de lograr esto, es no restringiendo \hat{d}_{ij} , \hat{d}_{kl} , por lo que los términos pueden ser cero, excepto por otras restricciones. Resumiendo, para el caso de enfoque primario las únicas restricciones sobre \hat{d}_{ij}

$$\text{Cuando } \hat{d}_{ij} < \hat{d}_{kl} \text{ entonces } \hat{d}_{ij} < \hat{d}_{kl}$$

Nótese que en este enfoque se consideran los rangos de todas las disimilaridades y esto es algunas veces conocido como escalamiento global.

b) Enfoque Secundario

El caso $\hat{d}_{ij} = \hat{d}_{kl}$ es evidencia de que la \hat{d}_{ij} debe ser igual a \hat{d}_{kl} y no afectar la configuración si este no es el caso. El stress por su parte debe reflejar esta situación. La forma de lograr esto es imponiendo la restricción $\hat{d}_{ij} = \hat{d}_{kl}$. Si $\hat{d}_{ij} \neq \hat{d}_{kl}$ los términos no pueden ser cero y reflejan directamente nuestro malestar por la desigualdad de \hat{d}_{ij} y \hat{d}_{kl} .

Resumiendo, para el caso de enfoque secundario las restricciones sobre las \hat{d}_{ij} son:

Cuando $\&_{ij} < \&_{kl}$ entonces $\hat{d}_{ij} < \hat{d}_{kl}$

Cuando $\&_{ij} = \&_{kl}$ entonces $\hat{d}_{ij} = \hat{d}_{kl}$

Este enfoque es demasiado restrictivo y conduce frecuentemente a problemas de convergencia.

La implantación para el uso opcional de ambos enfoques es muy simple.

La parte en la que la diferencia en estos dos enfoques tiene lugar es dentro del algoritmo para el ajuste de las \hat{d}_{ij} .

DATOS FALTANTES

En algunos casos no todas las disimilaridades son susceptibles de ser obtenidas por causas de diversa índole (diseño, error, etc.) Para valores grandes de n esto puede ser bastante común. Ante esta situación se hace necesario replantear la definición original de stress, generalizándola de tal manera que cubra esta contingencia. La forma natural que surge para la solución de este problema, es la simple omisión tanto en el numerador S^* , como en el denominador T^* , de los términos correspondientes a las disimilaridades faltantes.

Si no existen suficientes disimilaridades el método expuesto fallará. - Esto es, habrá una configuración con stress cero que no tendrá ninguna relación real con los datos. Es importante tener presente que la cuestión del número considerado de disimilaridades, no es determinante en la obtención de resultados con significado real, sino depende de que disimilaridades se consideren.

En muchos casos de importancia práctica, la consideración de un medio o cuarta parte, o aun menos, del total de disimilaridades son más que suficientes, si están distribuidas propiamente en la matriz de disimilaridades.

DISTANCIA EUCLIDEANA

Otra posible generalización en el planteamiento original, es como se -

mencionó, la utilización de funciones de distancia más generales que la euclídeana . En principio, no parece existir razón por la que la definición de stress, no pueda ser usada con cualquier distancia. La única restricción al parecer, es la dificultad de implantación en un programa de computadora.

EL PROCEDIMIENTO DE SHEPARD

La técnica expuesta, tiene su origen en los intentos de Kruskal por racionalizar y perfeccionar el procedimiento iterativo desarrollado por Shepard. La comparación de estos dos procedimientos es un punto interesante que puede ser descrito dentro del contexto de esta exposición.

Sea d_{ij} la m -ésima distancia en orden creciente, definimos \hat{d}_{ij} como la m -ésima disimilaridad . En lugar de hacer la influencia de x_i sobre x_j proporcional a $d_{ij} - \hat{d}_{ij}$, Shepard la hace proporcional a $\hat{d}_{ij} - \hat{d}_{ij}$. No parece posible describir bajo estas circunstancias su procedimiento como aquel que minimice alguna medida particular de no monotonía.

ROTACION DE PROCRUSTES

El problema de la evaluación de una configuración obtenida por alguno de los dos métodos expuestos, ha quedado resuelto por las medidas propuestas por Mardia de proporción de matriz de distancia explicada para el escalamiento clásico y por el stress de Kruskal para el escalamiento ordinal.

El carácter individual de estas medidas dado principalmente por el enfoque diferente necesario para la solución del mismo problema siguiendo un distinto esquema inicial de premisas o aun bajo el mismo método, pero utilizando alguna variación permitida dentro del mismo, hace evidente la conveniencia de establecer algún método de evaluación para la bondad de ajuste al comparar dos configuraciones.

De contarse con alguna forma de evaluar dos configuraciones, el método en cuestión no estará limitado para su uso en el escalamiento, sino se aplicaría en cualesquiera métodos de naturaleza semejante.

(Green 1952; Gower 1971) han desarrollado una medida de bondad de ajuste para comparar dos configuraciones. Consideremos a la matriz $X_{(n \times p)}$ de coordenadas de n puntos obtenida de la matriz D por una técnica.

Supongamos que $Y_{(n \times q)}$ es la matriz de coordenadas de otro conjunto de puntos obtenidos por otra técnica, o utilizando otra medida de distancia. Podemos considerar que las matrices son del mismo orden, al añadir columnas de ceros a la matriz de orden menor.

La medida de bondad de ajuste adoptada es obtenida al mover los puntos y_r con respecto a los puntos x_r hasta que la suma de cuadrados residuales

$$\sum_{r=1}^n (x_r - y_r)' (x_r - y_r)$$

sea mínimo. Podemos mover y_r con respecto a x_r a través de rotación, reflexión y traslación, i.e., por,

$$A'y_r + b \quad r=1, \dots, n$$

donde $A_{(p \times p)}$ es una matriz ortogonal, por lo que necesitamos resolver

$$R^2 = \min_{A,b} \sum_{r=1}^n (x_r - A'y_r - b)'(x_r - A'y_r - b)$$

para A y b.

Es posible demostrar que los valores de A y b que minimizan la medida están dados por

$$b = \bar{X} - A'\bar{Y}, \quad A = VU'$$

donde las matrices V, U' provienen de la descomposición en valores singulares de la matriz $Z = Y'X = UV'$

La obtención de estos valores particulares, vienen de resolver para A y b la expresión equivalente, (suponiendo que \bar{X} , \bar{Y} son ceros)

$$R^2 = \min \text{tr} (X - A'Y)'(X - A'Y) = \text{tr}XX' + \text{tr}YY' - \max \text{tr}X'YA$$

Hemos supuesto que los vectores medias de X, Y son ceros. Entonces la mejor rotación de Y con respecto a X es $\hat{Y}A$, donde \hat{A} está definida como anteriormente, y \hat{A} es llamada la Rotación de Procrustes de Y con respecto a X.

FACTOR DE ESCALA

Si las escalas de las dos configuraciones son diferentes, entonces la transformación debe ser de la forma

$$cA'y_r + b \quad \text{con } c > 0.$$

Puede demostrarse que

$$\hat{c} = \frac{\text{tr} \Gamma}{\text{tr}YY'} \quad (1)$$

y los otros estimadores permanecen igual.

Esta transformación se llama rotación de Procrustes con escalamiento de Y con respecto a X. La nueva mínima suma de cuadrados residuales está dada por

$$R^2 = \text{tr}(XX') + \hat{c}^2 \text{tr}(YY') - 2 \hat{c} \text{tr}(\Gamma)$$

donde c está dada por (1). Este procedimiento no es simétrico con respecto a X Y . La simetría puede ser obtenida al seleccionar el escalamiento de tal forma que

$$\text{tr}(XX') = \text{tr}(YY').$$

Escalamiento Clásico

Sea D una matriz de distancia que represente aproximadamente las distancias entre puntos de una configuración en un espacio euclideo de dimensión baja, teniendo en cuenta que D no necesariamente debe ser euclidea. Como sabemos la configuración asociada a esta matriz por el método clásico se obtiene al tomar los primeros k vectores propios de la matriz B . Si los primeros k valores propios de B son "grandes" y los restantes son cercanos a cero, entonces las distancias entre los puntos de la configuración se aproximarán cercanamente a las entradas de la matriz D .

El método numérico utilizado para la obtención de la configuración asociada a la matriz D es bastante simple, la única dificultad es de hecho, el conseguir una buena rutina para encontrar los valores propios y los correspondientes vectores propios de una matriz simétrica. Los cálculos involucrados en el método numérico pueden considerarse comprendidos en 4 etapas.

a) La construcción a partir de la matriz de distancia D de la matriz A - cuyo elemento genérico es $(-\frac{1}{2} d_{rs}^2)$.

En caso de presentarse una matriz de similitudes, será necesario transformar las similitudes a distancias. Sea C la matriz de similitudes, puede demostrarse que la transformación de similitudes a distancia está dada - por $d_{rs} = (c_{rr} - 2c_{rs} + c_{ss})^{1/2}$

b) La obtención de la matriz B con elementos $b_{rs} = a_{rs} - a_{r.} - a_{.s} + a_{..}$

c) La obtención de los k primeros valores propios de la matriz B $\lambda_1 > \lambda_2 > \dots > \lambda_k$, con correspondientes vectores propios $X = (x_{(1)}, \dots, x_{(k)})$

d) La normalización de los vectores $x_{(i)}$, en forma tal que $x'_{(i)} x_{(i)} = 1$ (suponiendo que los primeros k valores propios son positivos)

e) Los puntos requeridos son de la forma $x_r = (x_{r1}, \dots, x_{rk})'$, $r = 1, \dots, k$, las filas de X .

ESCALAMIENTO ORDINAL

El método numérico expuesto en esta sección tiene como finalidad, la obtención de la configuración de mejor ajuste para un valor fijo de t (dimensión del espacio de configuración), esto es, la configuración que minimice el stress

$$S^2 = \frac{S^*}{T^*} = \frac{\sum_{i < j} (d_{ij} - \hat{d}_{ij})^2}{\sum_{i < j} d_{ij}^2}$$

donde el mínimo es tomado sobre d_{ij}^{mon} tales que $d_{ij}^{\text{mon}} \sim \hat{d}_{ij}$ donde \sim significa relacionado monótonamente con. Un problema secundario de interés independiente, es la obtención de \hat{d}_{ij} a partir de los valores fijos de d_{ij} , este es el problema computacional de regresión monótona.

ENTRADAS FALTANTES

Este método es fácilmente adaptable en la situación en la que existen valores faltantes. Simplemente se omiten las disimilaridades en el orden

$$\hat{d}_{i(1)j(1)} < \dots < \hat{d}_{i(M)j(M)}$$

y se suprimen los términos correspondientes del numerador y denominador. Mientras no haya demasiados valores faltantes, el método parece todavía trabajar bien.

DISTANCIA

La posibilidad de utilización de una función de distancia más general que la euclídeana se incluye en este método. Específicamente, las técnicas numéricas y las formulas dadas cubren una clase de funciones introducidas en el capítulo 1, conocidas como métricas L_p o L_p , ocasionalmente llamadas métricas r de Minkowsky.

Entramos ahora a la descripción del metodo computacional de obtención de la configuración que minimice el stress para valores fijos de r y t , es decir, para una métrica particular, en un espacio t -dimensional.

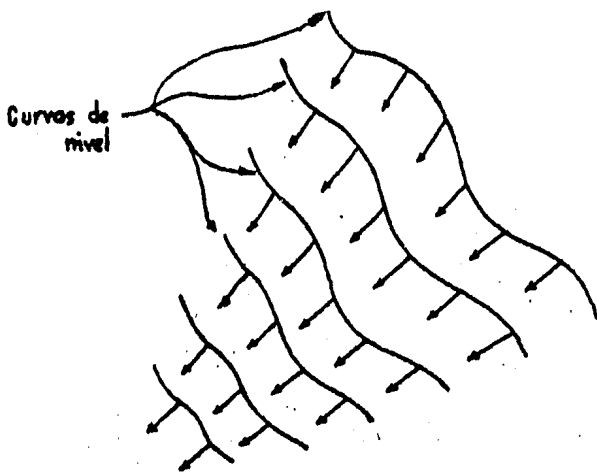
Hasta aquí se ha descrito una configuración como n puntos en un espacio t -dimensional; es conveniente, e igualmente válido, considerar tal configuración como un solo punto en un espacio nt -dimensional, i.e.,

$$(x_{11}, \dots, x_{1t}, \dots, x_{n1}, \dots, x_{nt})$$

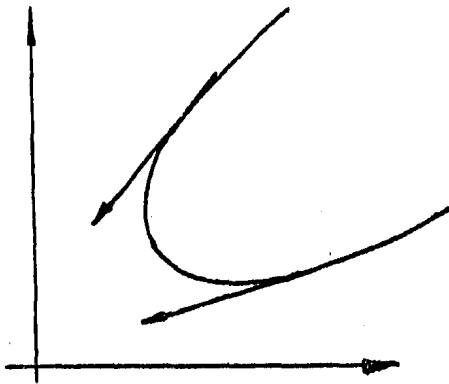
Nos referimos al espacio nt -dimensional como el espacio de configuración y al espacio t -dimensional como espacio modelo.

Supongamos que están dados los valores ϵ_{ij} . Para cada configuración existe un valor de S asociado. En forma equivalente $S : R^{nt} \rightarrow R$ es una función definida en cada punto del espacio de configuración. El problema a resolver es la obtención del punto que minimice a S . Nos enfrentamos con un problema común en análisis numérico, minimizar una función de varias variables. El método adoptado aquí es el metodo de gradientes.

Este método inicialmente considera un punto arbitrario en el espacio de configuración. El paso siguiente consiste en intentar mejorar un poco la configuración moviéndola ligeramente. El procedimiento adoptado por el método para lograr este fin, es seguir la dirección en el espacio de configuración en la cual S decrece más rápido. Tal dirección es proporcionada por el gradiente negativo de la función S (ver apéndice C). El gradiente, por supuesto, cambia de dirección de punto en punto, pero si pudiésemos seguir la dirección del gradiente negativo local, seríamos guiados al mínimo local. Ya que todas las direcciones cercanas conducen al mismo local, no es necesario seguir el gradiente negativo exactamente, podemos ser muy descuidados en este aspecto y todavía esperar el mínimo local.



El método de gradiente simple tiene la desventaja de que cerca del mínimo, el gradiente negativo apunta solo débilmente (debido al error de redondeo) al mínimo.



Sin embargo esta debilidad no debe ser muy exagerada; si todas las tangentes son pequeñas, hay probablemente poca ganancia al refinar aun más la estimación de la posición del mínimo. El método en su forma más sencilla consiste en tomar

$$x_j^{(i+1)} := x_j^{(i)} - h \frac{\partial S}{\partial x_j}$$

el superíndice indica el número de iteración , y h el tamaño del paso.

El primer problema es adivinar un paso razonable h (tal punto es abordado

más adelante) , teniendo en cuenta que no necesitamos ser exactos, pues todas las trayectorias cercanas conducen al mismo mínimo local.

Un punto importante en la discusión de cualquier metodo iterativo, ya sea de minimización o de cualquier otra índole, es el criterio de terminación del proceso. Las respuestas existentes en lo referente a este método en particular, no son satisfactorias. Existen varios criterios, entre los más populares están:

Sea $f(x)$ la función a minimizar, para cualquier real positivo ϵ tenemos

$$\begin{array}{l}
 \text{i) } \sum_{j=1}^n \left| \frac{\partial f}{\partial x_j} \right| < \epsilon \\
 \text{ii) } \sum_{j=1}^n \left| f^{(i+1)}(x)_j - f^{(i)}(x)_j \right| < \epsilon
 \end{array}$$

Como en el caso de medidas de similitud, el criterio de terminación de un proceso debe idealmente ser diseñado en forma específica para cada problema. Kruskal sugiere una forma modificada del criterio del cuadrado de la norma del gradiente i). El criterio en cuestión relaciona la norma del gradiente y la norma de la configuración normalizada, es decir, lo que Kruskal llama la magnitud relativa del gradiente.

Lo anterior es compatible con los criterios de obtención de mínimos de funciones vectoriales de valores reales dados en los textos standard de Cálculo Vectorial. Específicamente, se afirma que dado un subconjunto de un espacio euclideo U , y una función $f : U \rightarrow R$ y $x \in R^n$, es un extremo local si, y solo si, $\text{grad}(x_0) = 0$. La demostración de este hecho y la justificación acerca de la dirección en la cual decrece con mayor rapidez una función de valores reales, es dada en los apéndices.

LA DIFICULTAD DE MINIMOS LOCALES

Una dificultad general que padecen los métodos numéricos de minimización - incluyendo por supuesto del que nos ocupamos, es el de arribar a un mínimo local

en lugar de un mínimo global. En el problema de minimización que tratamos, esta dificultad no llega a alcanzar dimensiones de consideración, por las razones que se exponen a continuación.

i) Es posible iniciar el método de gradientes con configuraciones iniciales diferentes, las cuales pueden conducir a diferentes mínimos locales. Mientras sólo el menor de estos puede ser el mínimo verdadero, podríamos preguntarnos por mínimos más pequeños. Generalmente la mayoría de las configuraciones conducen al mismo mínimo local, el cual es mucho menor que los restantes que encontremos. Parece innecesario preocuparse cuando esto sucede.

ii) La configuración de mínimo local no es en sí el producto final del análisis, la cual deba ser aceptada ciegamente. En la mayoría de los casos la configuración de mínimo local es de interés solo si tiene algún sentido o pueda ser interpretada. Si la configuración posee una de estas propiedades, no es muy probable que sea un error considerarla.

iii) A menos que el stress de la supuesta configuración con mínimo local sea lo suficientemente bajo, esta configuración tendrá algún valor. Kruskal sugiere la siguiente clasificación de configuraciones con respecto a su stress asociado.

stress	bondad de ajuste
20 %	pobre
10 %	aceptable
5 %	buena
2 %	excelente
0 %	perfecta

A la fecha esta consideración acerca de la calidad de la configuración se considera demasiado simple ya que el stress depende de n y p .

iv) Existe la posibilidad de hacer varias pruebas de la configuración y los datos, así como de análisis separados de subconjuntos de datos.

DESCRIPCION DE LA IMPLANTACION

La técnica empleada para la minimización del stress es de carácter itera

tivo. En terminos generales consiste en empezar con una configuración arbitra-
ria , calcular el gradiente negativo, mover la configuración en esta direc-
ción una distancia conveniente y repetir estos dos pasos un número suficiente
de veces. Lo anterior se puede lograr dividiendo la tarea en siete subtareas
principales:

- 1) Creación de una configuración inicial
- 2) Normalización de la configuración
- 3) Cálculo de las distancias entre los puntos de la configuración
- 4) Ajuste de una regresión monótona creciente a las distancias
- 5) Cálculo del gradiente de la función stress
- 6) Cálculo de la magnitud en que se moverá la configuración
- 7) Creación de una nueva configuración

La descripción detallada de cada subtask se hará a continuación

1) La configuración inicial que se construya debe reunir solo dos cuali-
dades, siendo arbitraria de otra manera.

a) La configuración no puede estar en un espacio de dimensión menor al -
que se haya escogido para el análisis.

Una buena elección para la configuración inicial es aquella que resulta
del escalamiento clásico en la dimensión requerida, si no hay configuración -
disponible conveniente , una configuración arbitraria debe ser generada. Una
forma satisfactoria de conseguirlo es tomar los n primeros puntos de la forma

$$\begin{aligned} &(1,0,\dots,0), \\ &(0,1,\dots,0), \\ &\cdot \\ &\cdot \\ &\cdot \\ &\cdot \\ &(0,0,\dots,1), \\ &(2,0,\dots,0), \\ &(0,2,\dots,0), \text{etc.} \end{aligned}$$

Otra solución igualmente satisfactoria es la generación de los puntos en
la configuración con la ayuda de un generador de números aleatorios.

2) La normalización de la configuración se logra en dos pasos.

a) Trasladando el centro de gravedad de la configuración al origen.

b) Haciendo que la raíz cuadrada del promedio de las distancias de los puntos al origen sea unitaria.

La normalización de acuerdo a lo anterior implica la transformación de los puntos originales en puntos de la forma

$$x = \frac{x - \bar{x}}{fn}$$

donde

$$fn = \frac{1}{n} \sum_{i,s} x_{is}^2$$

3) El cálculo de la distancia entre los puntos en la configuración se hará utilizando la métrica de Minkowsky para un valor dado de r , el parámetro de esta métrica, la fórmula a utilizar, como es bien sabido es la siguiente:

$$d_{ij}^r = \left(\sum_{s=1}^p |x_{is} - x_{js}|^r \right)^{1/r}$$

4) Ajuste de una regresión monótona creciente a las distancias. Supongamos que las disimilaridades $d_{i(k)j(k)}$ presentan las siguientes relaciones

$$d_{i(1)j(1)} < \dots < d_{i(M)j(M)}$$

Los valores correctos de \hat{d}_{ij} pueden ser descritos de la siguiente manera:

Existe una partición de las disimilaridades en bloques consecutivos $b_i, i=1, \dots, u$ tal que, dentro de cada bloque el valor de \hat{d}_{ij} es constante, este valor común \hat{d}_{bk} es el promedio de las d_{ij} en ese bloque. Es por lo tanto,

suficiente encontrar la partición correcta con el fin de encontrar los -- números \hat{d}_{ij} . En un principio consideramos la partición en bloques más fina -- posible, uniendo según sea preciso, paso a paso los bloques apropiados hasta encontrar la partición correcta. La partición más fina posible consiste -- obviamente, de M bloques cada uno conteniendo una sola disimilaridad.

Supongamos que tenemos cualquier partición en bloques consecutivos. -- Denotemos \hat{d}_b como el promedio de las d_{ij} en el bloque b. Sean $b-1, b, b+1$ tres bloques consecutivos en orden creciente.

Definición

Se dice que el bloque b es satisfecho superiormente (up-satisfied) si $\hat{d}_b < \hat{d}_{b+1}$ o si es el último elemento en orden creciente de la partición.

Definición

Se dice que el bloque b es satisfecho inferiormente (down-satisfied) si $\hat{d}_b < \hat{d}_{b-1}$ o si es el primer bloque en orden creciente en partición.

En cada paso del algoritmo tenemos una partición en bloques, de los cuales solo uno se considera activo. Este bloque puede ser ya sea activo superior (up-active) o activo inferior (down-active). Al principio el primer bloque en la partición en orden ascendente, consistente solo de $d_{i(1)j(1)}$, es activo superior. El algoritmo procede posteriormente de la manera siguiente:

Si el bloque activo, es activo superior, verifica si es satisfecho superiormente. En caso afirmativo la partición en curso permanece inalterada; en caso contrario el bloque activo es unido al siguiente bloque superior, alterando la partición. En ambos casos el bloque resultante se considera ahora activo inferior.

Si el bloque activo, es activo inferior, verifica si es satisfecho inferiormente. En caso afirmativo la partición en curso permanece inalterada; en caso contrario, el bloque activo se une al siguiente bloque inferior, alterando la partición. En ambos casos, el bloque resultante se considera ahora activo inferior.

vo superior.

Esta alternación de estados, resulta eventualmente en un bloque activo el cual es simultáneamente satisfecho superior e inferiormente. Cuando esto ocurre, ninguna unión adicional puede hacerse por este procedimiento. El procedimiento se repite hasta que el último bloque en la partición en orden creciente es simultáneamente satisfecho superior e inferiormente. El caso de disimilaridades iguales puede ser manejado enteramente por el algoritmo, el cual sólo debe ser modificado por un preproceso. En caso de haberse adoptado el enfoque primario, el preproceso consiste en arreglar las disimilaridades de tal manera que las distancias formen una sucesión creciente en cada bloque donde aparezcan disimilaridades iguales. En caso de que se haya adoptado el enfoque secundario, es decir, si restringimos a los números \hat{d}_{ij} a que sean iguales cuando los correspondientes d_{ij} son iguales, el preproceso consiste en formar bloques de tal manera que el bloque que contenga a d_{ij} contenga además a todas las disimilaridades que igualen a esta. De ésta forma el algoritmo no empezará con la partición más fina sino con la partición formada por bloques con disimilaridades iguales.

4) Procederemos a encontrar la forma explícita del gradiente de la función stress, para el caso particular de la función de distancia de Minkowsky.

$$S^2 = \frac{S^*}{T^*} = \frac{\sum_{i < j} (d_{ij} - \hat{d}_{ij})^2}{\sum_{i < j} d_{ij}^2}$$

el stress asociado a la configuración.

$$S : R^{nt} \rightarrow R$$

La componente k_l del gradiente de S es:

$$\frac{\partial S}{\partial x_{kl}} = S \sum_{i,j} (d_{ij}^{kl} - \hat{d}_{ij}^{kl}) \left(\frac{d_{ij} - \hat{d}_{ij}}{S^*} - \frac{d_{ij}}{T^*} \right) \frac{|x_{i1} - x_{j1}|^{r-1}}{d^{r-1}} \text{signum}(x_{i1} - x_{j1})$$

donde δ^{ki} es la delta de Kronecker

$$\delta^{ki} = 1,0 \text{ si } k=i, k \neq i \text{ respectivamente}$$

La derivación completa puede ser encontrada en el apéndice A

5) La magnitud h ("tamaño de paso" en que se moverá la configuración a lo largo del gradiente) se varía en cada iteración. La magnitud h escogida en cada iteración, no afecta la solución obtenida al final. Lo que si es -- afectado, es el número de iteraciones requeridas para llegar a la solución, lo que en efecto es, un factor computacional importante.

El procedimiento para el calculo de h , provee pasos grandes durante las primeras iteraciones, y, pasos pequeños al final. El valor inicial de h para una configuración inicial arbitraria debe ser aproximadamente 0.02. Despues, el valor h es determinado por la formula:

$$h := h * f_a * f_r * f_b \quad \text{donde}$$

$$f_a = \frac{\cos^3 \theta}{4}$$

θ = angulo entre el gradiente actual y el gradiente anterior

$$f_r = \frac{1.3}{1 + c^{\frac{5}{5}}}$$

$$c_5 = \min \left\langle 1, \frac{S}{S_5} \right\rangle$$

$$f_b = \min \left\langle 1, \frac{S}{S} \right\rangle$$

S_5 = Stress de la configuración obtenida hace cinco iteraciones

S = Stress de la configuración obtenida en la iteración anterior

Si no han sido calculadas todavía 5 iteraciones, S_5 puede ser tomado como el primer stress calculado. Artificios similares pueden ser utilizados en relación a S' y θ en la primera iteración. Recordando el significado geo

métrico del gradiente de una función de valores reales f como el vector

$$\text{grad } f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)$$

El ángulo entre $\text{grad } S$ y $\text{grad } S$ está dado por :

$$\cos \theta = \frac{\langle \text{grad } S, \text{grad } S \rangle}{|S| |S|}$$

6) Supongamos que hemos llegado a una configuración X en R^S , entonces — formamos la nueva configuración siguiendo la dirección del vector $X + t \text{ grad}(S)$ donde

$$t = - \frac{h}{\text{mag}(\text{grad}(S))}$$

i.e., la componente x_{is} de la nueva configuración es la dada por

$$x_{is} := x_{is} + \frac{h}{\text{mag}(\text{grad}(S))} \frac{\partial S}{\partial x_{is}}$$

Si suponemos que X está normalizada

$$\text{mag}^2(\text{grad } S) = \frac{1}{n} \sum_{is} \left\langle \frac{\partial S}{\partial x_{is}}, \frac{\partial S}{\partial x_{is}} \right\rangle \quad 2$$

La ejecución de los procedimientos descritos en los párrafos anteriores, constituye el aspecto computacional central de la implantación del método descrito en el capítulo anterior. Mientras este proceso se desarrolla se logran sucesivamente, valores más pequeños de la función stress. Eventualmente el valor del stress desciende en forma significativa, de tal forma que, iteraciones adicionales resultan en una mejora nula o insignificante. Es conveniente contar con criterios adicionales para la terminación del proceso, como puede ser el número de iteraciones, etc. Para una configuración arbitraria, la condición de mínimo local sugerida es aproximadamente del 2% (i.e., $\text{mag}(\text{grad}(S)) = 0.02$). Este criterio dependerá en algunos casos de las condiciones específicas de los datos. Para datos con gran variación estadística (i.e., $n=40, t=2, o, n=15, t=5$) — un valor mayor es apropiado. Con más precisión lo anterior significa: un valor grande del criterio podría representar tanto como $e=0.05$ o posiblemente tan alto como $e=0.10$. Un valor bajo podría ser $e=0.005$ o cualquier valor menor, incluyen

do por supuesto cero.

La complicada naturaleza de la rutina iterativa arroja como resultado, - que la cantidad de cómputo involucrado en la obtención de una configuración - por el método ordinal, es mucho mayor que la necesaria utilizando el método - clásico. Para un conjunto típico de datos, el escalamiento ordinal podría tomar de 10 a 15 veces más tiempo que el escalamiento clásico para la obtención de una configuración asociada a una matriz dada.

ROTACION DE PROCRUSTES

Abordamos ahora la descripción del método numérico utilizado en la técnica Procrustes.

Podemos considerar que la única dificultad en la implementación de esta técnica, reside en la descomposición de la matriz $Z=Y'X$ en la forma $Z=U\Gamma V'$ (descomposición de valores singulares). Un excelente algoritmo puede encontrarse en Wilkinson (1970) bajo el nombre de SVD:

Para propósitos computacionales sumaremos los cálculos involucrados.

Sean X, Y dos configuraciones

a) A partir de X, Y construir la matriz $Z=Y'X$.

b) Encontrar las matrices U, Γ, V tales que $Z=U\Gamma V'$ donde U es una matriz con columnas ortonormalizadas, V es una matriz ortogonal y Γ es una matriz diagonal de valores positivos.

c) Calcular el factor de escala c

$$c = \frac{\text{tr}(\Gamma)}{\text{tr}(YY')}$$

d) Calcular la medida de bondad de ajuste

$$R^2 = c^2 \text{tr}(XX') + \text{ctr}(YY') - 2\text{ctr}(\Gamma)$$

e) Encontrar la matriz de Rotación A

$$A = VU'$$

f) Encontrar el vector de rotación b

$$b = \bar{X} - A'\bar{Y}$$

g) Rotar y trasladar la configuración Y i.e

$$y_r^* = cA'y_r + b$$

CAPITULO 6

Es el propósito de esta sección el mostrar los reportes impresos de los subprogramas que conforman MDS, las tablas que componen la subsección siguiente fueron en su totalidad generadas por MDS, y servirán de base en la discusión del ejemplo numérico que ilustrará el empleo práctico de los métodos descritos en los capítulos anteriores.

Ejemplo Numérico

	1	2	3	4	5	6	7	8	9	10	11	12
1	0											
2	244	0										
3	218	350	0									
4	284	77	369	0								
5	197	167	347	242	0							
6	312	444	94	463	441	0						
7	215	221	150	236	279	245	0					
8	469	583	251	598	598	169	380	0				
9	166	242	116	257	269	210	55	349	0			
10	212	53	298	72	170	392	168	531	190	0		
11	253	325	57	340	359	143	117	264	91	273	0	
12	270	168	284	164	277	378	143	514	173	111	256	0

La tabla se utilizó como matriz de prueba para los subprogramas Escalamiento Clásico y Escalamiento Ordinal. Los datos utilizados corresponden a las distancias por carretera entre doce poblaciones de la Gran Bretaña.

Escalamiento Clásico

* * * * ESCALAMIENTO CLASICO * * * *

No	VALOR PROPIO	PROPORCION DE MATRIZ DE DISTANCIA EXPLICADA	PROPORCION DE MATRIZ DE DISTANCIA EXPLICADA
1	394472.79	79.73	97.25
2	63634.37	92.59	99.78
3	13544.07	95.33	99.90
4	10245.31	97.40	99.96
5	2466.68	97.90	99.97
6	1450.27	98.19	99.97
7	500.89	98.29	99.97
8	-0.00	98.29	99.97
9	-16.94	98.30	99.97
10	-214.08	98.34	99.97
11	-1141.22	98.57	99.97
12	-7063.42	100.00	100.00

La tabla muestra los valores propios asociados a la matriz B así como las medidas de bondad de ajuste sugeridas por Mardia para la proporción de matriz de distancia explicada por una configuración obtenida por el método de escalamiento clásico, la expresión numérica correspondiente a cada medida es respectivamente

$$w_{1,s} = \frac{\sum_{r=1}^s |1_r|}{n}$$

$$w_{2,s} = \frac{\sum_{r=1}^s 1_r^2}{n}$$

el empleo de cada medida depende del carácter de la matriz de datos considerada, recordando que la segunda medida es más conveniente en el caso de que la matriz de distancia D no sea necesariamente Euclídeana. Un examen a la tabla revela enseguida el carácter bidimensional de la configuración

obtenida. Ambas medidas reportan una proporción superior al 90% y en el caso concreto de la segunda medida, la configuración en dos dimensiones explica prácticamente la totalidad de la matriz de distancia.

La configuración obtenida utilizando esta técnica se muestra en la tabla siguiente

* * * * ESCALAMIENTO CLASICO * * * *

CONFIGURACION DE PUNTOS, SOLUCION CLASICA

1	2
0.249325	1.921810
1.117395	-0.242210
-0.762621	0.424228
1.168113	-1.047268
1.042611	1.925208
-1.290233	0.426846
-0.042854	-0.691027
-2.104742	-0.353027
-0.174756	-0.075266
0.842838	-0.369057
-0.661968	-0.462649
0.616892	-1.457587

Escalamiento Ordinal

* * * * ESCALAMIENTO ORDINAL * * * *

ESCALAMIENTO ORDINAL,
CORRIDO BAJO LAS SIGUIENTES OPCIONES

MATRIZ DE DATOS SIMETRICA
AJUSTE BAJO ENFOQUE PRIMARIO
CONDICION DE MINIMO LOCAL = 0.000100
MAXIMO DE ITERACIONES = 100

DIMENSION REQUERIDA	=	2
METRICA DE MINKOWSKY	=	=2.000000

PARAMETROS ASOCIADOS A LA CONFIGURACION OBTENIDA		
ITERACION EN CURSO	=	101
MAGNITUD DE GRADIENTE	=	=0.000126
STRESS	=	=0.010643

El metodo de escalamiento ordinal fué ahora utilizado para la obtención de una configuración asociada a la matriz de datos de la tabla. La tabla nos muestra algunos datos correspondientes a la configuración obtenida impresos por el subprograma ESCALAMIENTO ORDINAL. En específico se informa el tipo de matriz de datos analizada, el criterio de ajuste en caso de medidas experimentales con el mismo valor numérico, las condiciones que de cumplirse resultarán en la terminación del proceso iterativo, i.e., la magnitud relativa del gradiente de la función stress especificado como condición de mínimo local y el número máximo de iteraciones permitidas para la búsqueda de la configuración con stress mínimo. A continuación se muestran las condiciones que resultaron al término del proceso, i.e., la iteración en curso y la magnitud relativa del gradiente de la función stress.

De acuerdo con el reporte mostrado, el método no convergió, pues la terminación del proceso fué causada por excederse en el número máximo de iteraciones permitido. Esto se debió al criterio tan restringido pedido como condición de mínimo local (0.000100). Por otra parte, el stress asociado a la configuración obtenida por este metodo nos lleva a clasificarla según el criterio de Kruskal (considerado hoy en día demasiado simple) como excelente (stress = 1.06 %). La siguiente tabla muestra la configuración de puntos a la que se llegó por el metodo que tratamos

* * * * ESCALAMIENTO ORDINAL * * * *

* CONFIGURACION DE PUNTOS, SOLUCION ORDINAL *

0.723483	0.336022
-0.270878	1.035944
0.275689	-0.650876
-0.630301	0.964504
0.477298	1.145919
0.432855	-1.123869
-0.221228	-0.107837
0.149550	-1.907132
0.062437	-0.105478

-0.285246	0.734465
0.001730	-0.650266
-0.715389	0.328602

En cierto sentido esto podría parecer una muestra inútil del uso de las técnicas, pues se conoce la configuración real que dió origen a la matriz - analizada, sin embargo, servirá para mostrar algunos aspectos de interés - inherentes a los métodos de escalamiento.

Las observaciones hechas sobre las configuraciones obtenidas ilustran el hecho evidente acerca de la carencia completa de significado particular de los ejes resultantes, por lo que es perfectamente factible la búsqueda - de otros que conduzcan a una interpretación obvia. El problema de encontrar dimensiones con significado es uno de los propósitos principales del escala - miento, Kruskal y Wish (1978) sugieren que una regresión lineal múltiple - entre alguna dirección en la configuración y varias características de los puntos, puede ser útil al tratar de interpretar las dimensiones.

Otra forma en la que la gráfica de una configuración puede ser usada - es para localizar grupos o conglomerados de puntos que estén cercanos. Si - se encuentra una configuración en una dimensión mayor a dos, los problemas de interpretación son considerablemente más complejos. Puede ser de ayuda - graficar las dimensiones dos a la vez, una regresión lineal múltiple puede ser también de utilidad.

COMPARACION DE CONFIGURACIONES

Las configuraciones obtenidas por cada uno de los metodos tratados fue - ron comparadas por el subprograma COMPARACION DE CONFIGURACIONES utilizando la técnica de Rotación de Procrustes. En este ejemplo en particular al con - tarse con la configuración real, fué posible la comparación de cada configu - ración generada con dicha configuración.

El subprograma COMPARACION DE CONFIGURACIONES reportó en cada caso la suma de cuadrados residuales.

$$\sum_{i=1}^n (x_i - (Ay_i + b))^2 (x_i - (Ay_i + b))^2$$

donde x_1 es un punto de la configuración con respecto a la cual se compara la configuración Y de la cual y_1 forma parte.

Asimismo se imprimió el factor de escala correspondiente, la matriz de Rotación A y el vector de traslación b que minimizan la suma de cuadrados residuales. Por último el subprograma en base al factor de escala, la matriz de rotación y el vector de traslación calculados, escala rota y -- traslada la configuración que se ha comparado. Los reportes impresos por este subprograma se muestra a continuación en forma condensada.

* * * * ROTACION DE PROCRUSTES * * * *

COMPARACION DE CONFIGURACION CLASICA CON CONFIGURACION VERDADERA

* SUMA DE CUADRADOS RESIDUALES = 4.0381E+06 *
 * FACTOR DE ESCALA = 1.7557E+02 *

*MATRIZ DE ROTACION *

1	2
0.32	-0.95
-0.95	-0.32

*VECTOR DE TRASLACION *

1.
 424.25
 383.42

* * * * ROTACION DE PROCRUSTES * * * *

COMPARACION DE CONFIGURACION ORDINAL CON CONFIGURACION VERDADERA

* SUMA DE CUADRADOS RESIDUALES = 3.9357E+06 *
 * FACTOR DE ESCALA = 2.6492E+02 *

* MATRIZ DE ROTACION *

1	2
-0.99	0.13
-0.13	-0.99

*VECTOR DE TRASLACION *

1
 424.25
 383.42

Un examen a la suma de cuadrados residuales correspondiente a cada comparación en este ejemplo en particular muestra una superioridad del método ordinal sobre el clásico, aún cuando tal superioridad es casi insignificante pues al hacer el cociente de los cuadrados residuales este resultado ser casi 1 (0.9746).

La última tabla muestra por una parte a la configuración real y por otra a las configuraciones obtenidas por el método clásico y el método ordinal en ese orden. El subprograma COMPARACION DE CONFIGURACIONES imprime opcionalmente la configuración que ha sido comparada escalada, rotada y trasladada con respecto a la otra configuración.

* * * * ROTACION DE PROCRUSTES * * * *

257.00	279.00	118.90	233.36	222.36	321.07
529.00	104.00	527.66	211.37	458.33	101.80
339.00	554.00	310.63	486.20	375.14	564.12
629.00	142.00	664.35	248.44	555.24	107.71
292.00	90.00	163.17	101.31	258.01	99.67
259.00	665.00	280.38	573.76	350.79	693.90
508.00	433.00	536.69	429.59	486.18	403.82
265.00	842.00	363.98	753.22	453.15	889.39
433.00	438.00	426.28	416.72	411.63	413.34
533.00	183.00	533.23	264.18	472.88	180.42
420.00	563.00	463.74	519.60	447.03	554.17
627.00	308.00	701.40	363.26	600.30	271.59

Comparación de Configuraciones

Compara dos configuraciones por la técnica Rotación de Procrustes desarrollada por Gower.

Escalamiento Clásico

Obtiene la configuración asociada a una matriz de datos por el método de escalamiento clásico debido a Torgerson y Gower.

Escalamiento Ordinal

Obtiene la configuración asociada a una matriz de datos por el método de escalamiento ordinal desarrollado por Kruskal.

Ejecución

Transmitir RUN MDS en respuesta al prompt de CANDE. El programa mostrará una lista de subprogramas en la terminal. El usuario puede seleccionar uno a la vez. El control será transferido al subprograma seleccionado, el cual a su vez desplegará en la terminal una lista de opciones o información requerida para el proceso deseado. Cualquier número de requerimientos pueden ser ejecutados en el mismo o diferentes archivos, así como en los archivos creados por el programa. Cuando la ejecución del subprograma seleccionado haya sido completada, el subprograma devolverá el control al programa principal, mostrará nuevamente el menú en la pantalla y esperará el siguiente requerimiento. Si el requerimiento no puede ser completado se mostrará el mensaje de error apropiado y se devolverá el control al programa principal en espera del siguiente requerimiento. En esta descripción cualquier referencia a un archivo debe ser un identificador de archivo válido asociado a un archivo existente en el sistema.

Requerimiento

Comparación de Configuraciones

Definición

Compara dos configuraciones por la técnica de Rotación de Procrustes desarrollada por Gower.

Discusión

La exposición detallada de esta técnica se encuentra en el capítulo 4 de este trabajo. Este subprograma debe ser ejecutado cuando se tengan dos configuraciones 'X', 'Y' obtenidas por diferentes técnicas, o con una diferente función de distancia y se desee comparar la configuración 'Y' con respecto a la configuración 'X'.

Entrada y Salida

El subprograma requiere dos archivos. Cada uno conteniendo una configuración. El formato requerido es (8F10.6). El subprograma al término de su ejecución provee la facilidad de imprimir o crear un archivo en disco con la configuración 'Y' rotada y trasladada, el formato será también (8F10.6)

Opciones o Información

Requerida

Dimensión

Indicará la dimensión del espacio en que se encuentra cada configuración, la cual podrá diferir para cada configuración.

Tamaño de Muestra

Precisará el número de puntos en cada configuración. Deberá tenerse en cuenta que este número deberá coincidir en ambas configuraciones.

Requerimiento

Escalamiento Clásico

Definición

Obtiene la configuración asociada a una matriz de datos por el método de escalamiento clásico debido a Torgerson y Gower.

Discusión

La fundamentación teórica de esta técnica puede ser encontrada en el capítulo 2 de esta exposición. Este subprograma debe ser usado cuando se tenga una matriz de distancia que sea exacta o al menos aproximadamente euclídeana, o en el caso de una matriz de similaridad.

Entrada y Salida

El subprograma requiere solo un archivo de datos, el cual contendrá la matriz de datos analizar. El formato esperado es (8F10.6). Al completar la tarea el programa está en posibilidad de imprimir o crear un archivo en disco con la configuración obtenida por este método, el formato utilizado será también (8F10.6).

Opciones o Información Requerida

Espacio de Observaciones

Proporcionará el total de puntos considerados los cuales deben coincidir con la dimensión de la matriz de datos.

Espacio Modelo

Indicará el número de coordenadas o dimensiones requeridas en la configuración por obtener asociada a la matriz de datos en cuestión.

Matriz de Datos

El carácter de la matriz de datos determinará el tipo de proceso a seguir según el caso. Si no se proporciona dato al respecto se asumirá una matriz de distancia, la otra opción disponible es considerar una matriz de similaridad.

Requerimiento

Escalamiento Ordinal

Definición

Obtiene una configuración asociada a una matriz de datos por el método ordinal debido a Kruskal.

Discusión

Una exposición detallada de esta técnica puede ser encontrada en el capítulo 3 de este trabajo.

Este subprograma deberá utilizarse cuando las disimilaridades observadas son tales que sus valores - numéricos son de poco significado y el rango de orden de éstas es la única información considerada - relevante .

Entrada y Salida

Este subprograma opera con un archivo de datos conteniendo la matriz a procesar. El formato esperado es el usual (8F10.6). Al término del procesamiento de la matriz el programa podrá a elección del usuario, imprimir o crear un archivo en disco con la - configuración obtenida por este método. En este último caso el formato a utilizar será (8F10.6)

Opciones o Información Requerida

Dimensión

La dimensión de la matriz de datos es necesaria para la lectura de esta, por lo que respecta al espacio de observaciones. Para el espacio modelo, la - dimensión del espacio en el que se encontrará la - configuración a obtener, lo cual indicará el núme- ro de coordenadas requeridas en la configuración - buscada.

Configuración

Informará al programa de la disponibilidad o ausencia de una configuración inicial. Si esta opción - no es considerada se supondrá la ausencia de ésta.

Medida experimental

Definirá el carácter de la medida experimental encontrada en la matriz de datos, esto es, si se trata de un coeficiente de disimilaridad o de un coeeficiente de similaridad. Al ignorar esta opción se

asumirá que la medida utilizada corresponde a un -
coeficiente de disimilaridad.

Distancia

Seleccionará un elemento de la familia de funciones de distancia Minkowsky al especificar el parámetro R de esta clase de métricas. Esta función de distancia se utilizará en el cálculo de distancias entre los puntos de la configuración. La métrica default es la euclideana (correspondiente a $R=2$).

Ajuste

Determinará el enfoque a utilizar en el caso de encontrar medidas que coincidan en valor numérico dentro de las entradas de la matriz de datos. Los enfoques a elegir son el primario o secundario, teniendo por default el primero de estos.

Criterio de Terminación

Determinará el criterio de terminación del proceso iterativo. Se elegirá entre un número máximo de iteraciones o la magnitud relativa del gradiente de la función Stress. El proceso finalizará cuando alguna de las dos condiciones se satisfagan. Los valores - default considerados por el subprograma son de 100 iteraciones y 0.025 como condición de mínimo local.

EL GRADIENTE DE LA FUNCION STRESS

$$\nabla S^* = \sum_{i,j} (d_{ij} - \hat{d}_{ij})^2$$

$$\nabla S^* = 2 \sum_{i,j} (d_{ij} - \hat{d}_{ij}) D (d_{ij} - \hat{d}_{ij})$$

$$\nabla S^* = 2 \sum_{i,j} (d_{ij} - \hat{d}_{ij}) D (d_{ij} - \hat{d}_{ij})$$

$$\nabla S^* = 2 \sum_{i,j} (d_{ij} - \hat{d}_{ij}) D (d_{ij})$$

$$D^* = \sum_{i,j} d_{ij}^2$$

$$\nabla D^* = 2 \sum_{i,j} d_{ij} D (d_{ij})$$

$$\nabla S = \frac{1}{2} S \left\{ \frac{\nabla (S^*)}{S^*} - \frac{\nabla (T^*)}{T^*} \right\} = \frac{1}{2} S \left\{ \frac{2 \sum_{i,j} (d_{ij} - \hat{d}_{ij}) D (d_{ij})}{S^*} - \frac{2 \sum_{i,j} d_{ij} D (d_{ij})}{T^*} \right\}$$

$$\nabla S = \sum_{i,j} S \left\{ \frac{d_{ij} - \hat{d}_{ij}}{S^*} - \frac{d_{ij}}{T^*} \right\} D (d_{ij})$$

$$\begin{aligned} D (d_{ij}) &= D \left(\left(\sum_{s=1}^t |x_{is} - x_{js}|^r \right)^{\frac{1}{r}} \right) \\ &= \frac{1}{r} \left(\sum_{s=1}^t |x_{is} - x_{js}|^r \right)^{\frac{1}{r} - 1} D \left(\sum_{s=1}^t |x_{is} - x_{js}|^r \right) \\ &= \frac{1}{r} \left(\sum_{s=1}^t |x_{is} - x_{js}|^r \right)^{\frac{1}{r} - 1} \sum_{s=1}^t D (|x_{is} - x_{js}|^r) \end{aligned}$$

$$D (|x_{is} - x_{js}|^r) = r |x_{is} - x_{js}|^{r-1} D (|x_{is} - x_{js}|)$$

$$D (|x_{is} - x_{js}|) = \frac{|x_{is} - x_{js}|}{(x_{is} - x_{js})} D (x_{is} - x_{js})$$

$$\left(\sum_{s=1}^t |x_{is} - x_{js}|^r \right)^{\frac{1}{r} - 1} = \frac{\left(\sum_{s=1}^t |x_{is} - x_{is}| \right) d_{ij}}{\left(\sum_{s=1}^t |x_{is} - x_{js}| \right)^r d_{ij}^r} = \frac{1}{d_{ij}^{r-1}}$$

$$\frac{|x_{is} - x_{js}|}{|x_{is} - x_{js}|} = \text{signum}(x_{is} - x_{js})$$

$$s_{ij} = \frac{1}{r} \frac{1}{d_{ij}^{r-1}} r \sum_{s=1}^t |x_{is} - x_{js}|^{r-1} \text{signum}(x_{is} - x_{js}) \cdot D(x_{is} - x_{js})$$

De acuerdo a lo anterior la componente k l del gradiente de S es

$$\frac{\partial S}{\partial x_{kl}} = S \sum_{ij} (\delta^{kl} - \delta^{kj}) \left(\frac{\hat{d}_{ij} - d_{ij}}{S^*} - \frac{d_{ij}}{r^*} \right) \frac{|x_{i1} - x_{j1}|^{r-1}}{d_{ij}^{r-1}} \text{signum}(x_{i1} - x_{j1})$$

EL ANALISIS DE COMPONENTES PRINCIPALES

La relación que existe entre el método de escalamiento clásico y el análisis de componentes principales se discutió en el capítulo 3. Tratando de evitar una complicación innecesaria en la exposición se ha diferido hasta aquí la presentación de los fundamentos de tal técnica.

Definición

Si x es un vector aleatorio con media u y matriz de covarianza S , entonces la transformación de componentes principales es la transformación

$$T : x \rightarrow y = G'(x - u)$$

donde t es ortogonal, $G'SG = L$ es diagonal y $l_1 > l_2 > \dots > l_p > 0$. El tener valores de l_i no negativos está garantizado si S es positiva definida. La representación de S se sigue del teorema de descomposición espectral. La i -ésima componente principal puede ser definida como el i -ésimo elemento del vector y , denotado como:

$$y_i = g_{(i)}'(x - u)$$

Aquí $g_{(i)}$ es la i -ésima columna de G , y puede ser llamado el i -ésimo vector de cargas de componentes principales.

Las propiedades principales que poseen las nuevas variables generadas vía la transformación T son enunciadas por el siguiente teorema:

Teorema

Sea T la transformación definida arriba; entonces se cumple

- i) $E(Y_i) = 0$
- ii) $V(Y_i) = 1$
- iii) $C(Y_i, Y_j) = 0, i \neq j$
- iv) $V(y_1) > V(y_2) > \dots > V(y_p) > 0$

$$v) \sum_{i=1}^p V(y_i) = \text{tr } S$$

$$vi) \prod_{i=1}^p V(y_i) = |S|$$

las propiedades i) a iv) se siguen a partir de la definición de la transformación T y las propiedades del operador Esperanza. La propiedad v) se sigue a partir de ii) y el hecho de que $\text{tr } S$ es la suma de los valores característicos, vi) se sigue de ii) y del hecho de que $|S|$ es el producto de los valores característicos.

TEOREMA DE DESCOMPOSICION ESPECTRAL

Cualquier matriz simétrica $A_{p \times p}$ puede ser escrita como

$$A = GLG^T = \sum l_i g_{(i)} g_{(i)}^T$$

donde L es una matriz diagonal de valores propios de A , y G es una matriz ortogonal cuyas columnas son vectores propios estandarizados.

Demostración

Supongamos que podemos encontrar vectores ortonormales $g_{(1)}, \dots, g_{(p)}$ tales que $A g_{(i)} = l_i g_{(i)}$ para algunos números l_i . Entonces

$$g_{(i)}^T A g_{(i)} = l_i g_{(i)}^T g_{(i)} = \begin{cases} l_i & i = j \\ 0 & i \neq j \end{cases}$$

o en forma matricial $G^T A G = L$ por lo que A y L tienen los mismos valores propios, así que los elementos de L son exactamente los valores propios de A con la misma multiplicidad. Por lo tanto debemos encontrar una base ortonormal de vectores propios. Hay que notar que si $l_i \neq l_j$ son valores propios distintos con vectores propios asociados x y y respectivamente, entonces

$$l_i x^T y = x^T A y = y^T A x = l_j y^T x, \text{ así que } y^T x = 0$$

De aquí que para una matriz simétrica, vectores propios correspondientes a valores propios distintos son ortogonales entre si. Supongamos que hay k valores propios de A distintos con correspondientes espacios propios H_1, \dots, H_k de dimensiones r_1, \dots, r_k

$$\text{Sea } r = \sum_{j=1}^k r_j$$

Ya que espacios propios distintos son ortogonales, existe un conjunto ortonormal de vectores e_1, \dots, e_r tal que los vectores

$$\begin{array}{c}
 j-1 \qquad \qquad j \\
 \Sigma r \quad \dots, \Sigma r \\
 i=1 \quad i+1 \qquad \qquad i=1 \quad i
 \end{array}$$

forman una base para H_j , de tal manera que r_j es menor o igual a la multiplicidad del valor propio correspondiente. De aquí que reordenando los valores propios l_i si es necesario, podemos suponer

$$A e_i = l_i e_i, \quad i = 1, \dots, r \text{ y } r \leq p$$

Si $r = p$, hagamos $g_i = e_i$ y el teorema está probado.

Si $r < p$ entonces nos lleva a una contradicción y por lo tanto no puede aparecer tal situación. Supongamos sin pérdida de generalidad que todos los valores propios de A son estrictamente positivos (si no, podemos reemplazar A por $A + aI$ para un escalar a apropiado, debido a que A y $A + aI$ tiene los mismos valores propios.

$$\text{Sea } B = A - \sum_{i=1}^r l_i e_i e_i'$$

entonces

$$\text{tr } B = \text{tr } A - \sum_{i=1}^r l_i (e_i' e_i) = \sum_{i=r+1}^p l_i > 0$$

ya que $r < p$. Por lo tanto B tiene al menos un valor propio no nulo, digamos f .

Sea $x \neq 0$ el vector propio correspondiente, entonces para $1 \leq j \leq r$

$$f e_j' x = e_j' B x = (l_j e_j' - \sum_{i=1}^r l_i (e_j' e_i) e_i') x = 0$$

así que es ortogonal a e_j , $j = 1, \dots, r$

Entonces

$$fx = Bx = (A - \sum l_i e_i e_i')x = Ax - \sum l_i (e_i' x) e_i = Ax$$

así que x es vector propio de A también. Por lo tanto $f = l_i$ para alguna i y x es una combinación lineal de las e_i , lo cual contradice la ortogonalidad entre las x y las e_i .

Teorema

Si UCR^n es abierto y $f: UCR^n \rightarrow R$ es diferenciable y $x_0 \in U$ es un extremo local, entonces $\nabla f(x_0) = \underline{0}$.

Demostración

Supongamos que f alcanza mínimo local en x_0 . Entonces para cualquier $h \in R^n$, la función $g(t) = f(x_0 + th)$, $t \in R$ tiene un mínimo local en $t = 0$, esto implica que $g'(0) = 0$. Por otra parte por la regla de la cadena

$$g'(0) = \nabla f(x_0) \cdot h = 0$$

Por lo tanto $\nabla f(x_0) \cdot h = 0$, para cualquier h , por lo que $\nabla f = \underline{0}$

Teorema

Supongamos que $\nabla f \neq \underline{0}$. Entonces $-\nabla f$ apunta en la dirección en que f decrece más rápidamente.

Demostración

Si n es un vector unitario, la rapidéz de crecimiento de f en la dirección n es $\langle -\nabla f, n \rangle = \|\nabla f\| \cos \theta$ donde θ es el ángulo entre n y $-\nabla f$. Este es máximo cuando $\theta = 0$; esto es cuando n y $-\nabla f$ son paralelos. En otras palabras, si se desea moverse en la dirección en que f decrece más rápidamente se debe proceder en la dirección de $-\nabla f$.

B I B L I O G R A F I A

Chattfield, C. and Collins, A. J. (1980)

Introduction to Multivariate Analysis

London: Chapman and Hall.

Gower, J.C. (1966)

Some distance properties of latent roots and vector methods used in multivariate analysis

Biometrika 53, 325-8.

Hamming, R. W. (1971)

Introduction to Applied Numerical Analysis

New York: McGraw-Hill.

Jardine, N. and Sibson, R. (1971)

Mathematical Taxonomy

London: Wiley Interscience.

Kruskal, J. B. (1964)

Multidimensional Scaling by optimizing goodness of fit to a nonmetric hypothesis

Psychometrika, 29, 1-27.

Kruskal, J. B. (1964)

Nonmetric Multidimensional Scaling: A Numerical Method

Psychometrika, 29, 115-29.

Kruskal, J. B. and Wish, M. (1978)

Multidimensional Scaling

London, Sage University Paper Series on Quantitative Applications in the Social Sciences.

Mardia, K. V., Kenneth, J.T. and Bibby, J. M. (1979)

Multivariate Analysis

London: Academic Press.

Sibson, R. (1972)

Order Invariant Methods for Data Analysis

J. R. Statist. Soc., B, 34, 311-49.

Torgerson, W. S. (1958)

Theory and Methods of Scaling

New York: Wiley.

Shepard, R. N. (1962)

The Analysis of Proximities: Multidimensional Scaling with an unknown distance function.

Psychometrika, 27, 125-139, 219-46.

Wilkinson, J. H. and Reinsch, C. (1971)

Linear Algebra

Handbook for Automatic Computation Vol II

New York, Springer Verlag.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

```

BEGIN
  FILE
  TTY(KIND=REMOTE, MAXRECSIZE=22, MYUSE=10, PAGESIZE=25),
  PRN(KIND=PRINTER, PAGESIZE=55),
  DATAFILE(KIND=DISK, FILETYPE=7),
  OUTFILE(KIND=DISK, MAXRECSIZE=14, BLOCKSIZE=420);
  SWITCH FILE F:=TTY, PRN, OUTFILE;
  PROCEDURE STOPPAGE(DDV);
  INTEGER DDV;
  IF DDV = 1 THEN
    BEGIN
      FORMAT
      DISP(" * TRANSKITA CUALQUIER CARACTER PARA CONTINUAR *");
      STRING C;
      WRITE(TTY, DDV, DISP);
      READ(TTY, <A>, C);
      WHILE C = EMPTY DO
        BEGIN
          WRITE(TTY, DISP);
          READ(TTY, <A>, C);
        END;
      END;
    END;
  PROCEDURE QUICKSORT(D, IND, N);
  ARRAY D, IND(1);
  INTEGER N;
  BEGIN
    PROCEDURE SORT(L, R);
    INTEGER L, R;
    BEGIN
      INTEGER I, J;
      REAL X, M, Z;
      I:=L;
      J:=R;
      X:=D(L+R) DIV 2;
      DO
        BEGIN
          WHILE D(I) > X DO
            I:=I + 1;
          WHILE X > D(J) DO
            J:=J - 1;
          IF I < J THEN
            BEGIN
              M:=D(I);
              Z:=IND(I);
              D(I):=D(J);
              IND(I):=IND(J);
              D(J):=M;
              IND(J):=Z;
              I:=I + 1;
              J:=J - 1;
            END;
        END;
      UNTIL I > J;
      IF I < J THEN SORT(L, J);
      IF I < R THEN SORT(I, R);
    END;
    SORT(L, N);
  END;

```

0000001
0010000
0010010
0010020
0010030
0010040
0010050
0010060
0010070
0010080
0010090
0010100
0010110
0010120
0010130
0010140
0010150
0010160
0010170
0010180
0010190
0010200
0010210
0010220
0010230
0010240
0010250
0010260
0010270
0010280
0010290
0010300
0010310
0010320
0010330
0010340
0010350
0010360
0010370
0010380
0010390
0010400
0010410
0010420
0010430
0010440
0010450
0010460
0010470
0010480
0010490
0010500
0010510
0010520
0010530
0010540
0010550
0010560


```

4421      FOR J=1 STEP 1 UNTIL M DO
4422          S:=A + UEK,J1 * UEK,J2;
4423          FOR K=1 STEP 1 UNTIL M DO
4424              UEK,J3:=S + F * UEK,J3;
4425          END;
4426      END;
4427      Q11:=G;
4428      FOR J=L STEP 1 UNTIL M DO
4429          S:=A + UEI,J1 * UEI,J2;
4430          IF S < TOL THEN
4431              G:=1;
4432          ELSE
4433              BEGIN
4434                  S:=UEI,J1 * J1 THEN SQR(S) ELSE - SQR(S);
4435                  H:=F * G - S;
4436                  UEI,J1 + 1:=S - G;
4437                  FOR J=L STEP 1 UNTIL M DO
4438                      UEI,J3:=UEI,J3 / H;
4439                  FOR J=L STEP 1 UNTIL M DO
4440                      BEGIN
4441                          FOR K=L STEP 1 UNTIL M DO
4442                              S:=A + UEI,K1 * UEI,K2;
4443                              FOR K=L STEP 1 UNTIL M DO
4444                                  UEI,K3:=S + S * UEI,K3;
4445                          END;
4446                      END;
4447                  Y:=ABS(Q11) + ABS(Q12);
4448                  IF Y < 4 THEN
4449                      G:=1;
4450                  END;
4451      END;
4452      X-----ACUMULACION DE TRANSFORMACIONES DERECHAS -----
4453      FOR I=M STEP -1 UNTIL 1 DO
4454          BEGIN
4455              G:=UEI,I;
4456              IF G NEQ 1 THEN
4457                  BEGIN
4458                      H:=UEI,I + 1 + G;
4459                      FOR J=L STEP 1 UNTIL M DO
4460                          UEI,J1:=UEI,J1 / H;
4461                      FOR J=L STEP 1 UNTIL M DO
4462                          BEGIN
4463                              S:=A;
4464                              FOR K=L STEP 1 UNTIL M DO
4465                                  S:=S + UEI,K1 * UEK,J2;
4466                              FOR K=L STEP 1 UNTIL M DO
4467                                  UEK,J3:=S + S * UEK,J3;
4468                              END;
4469                          END;
4470                      END;
4471                  END;
4472              FOR J=L STEP 1 UNTIL M DO
4473                  VEI,J1:=VEI,J1;
4474              G:=UEK,I;
4475              L:=I;
4476          END;
4477      X-----ACUMULACION DE TRANSFORMACIONES IZQUIERDAS -----
4478      FOR I=M STEP -1 UNTIL 1 DO
4479          BEGIN
4480              G:=UEK,I;
4481              FOR J=L STEP 1 UNTIL M DO
4482                  UEK,J1:=UEK,J1;
4483              IF G NEQ 1 THEN
4484                  BEGIN
4485                      H:=UEI,I * G;
4486                      FOR J=L STEP 1 UNTIL M DO
4487                          S:=A;
4488                          FOR K=L STEP 1 UNTIL M DO
4489                              S:=S + UEK,J1 * UEK,J2;
4490                          FOR K=L STEP 1 UNTIL M DO
4491                              UEK,J3:=S + F * UEK,J3;
4492                          END;
4493                      END;
4494                  END;
4495              H:=UEI,I / G;
4496              FOR J=L STEP 1 UNTIL M DO
4497                  UEI,J1:=UEI,J1 * H;
4498              END;
4499          END;
4500      END;
4501      Y---- DIAGONALIZACION DE LA FORMA HIDIAGONAL -----
4502      EPS:=A * X;
4503      FOR K=M STEP -1 UNTIL 1 DO
4504          SPLI:=UEK,K;
4505          BEGIN
4506              FOR I=K STEP -1 UNTIL 1 DO
4507                  IF ABS(E11) < EPS THEN GO FCONVERGENCE;
4508                  IF ABS(E12) < EPS THEN GO CANCELLATION;
4509              END;
4510          CANCELLATION :
4511              L:=K - 1;
4512              FOR I=L STEP 1 UNTIL K DO
4513                  BEGIN
4514                      E11:=E11;
4515                      E12:=E12;
4516                      IF ABS(E12) < EPS THEN GO FCONVERGENCE;
4517                      G:=E12;
4518                      H:=SQR(F * F + G * G);
4519                      S:=F / H;
4520                      FOR J=L STEP 1 UNTIL M DO
4521                          BEGIN
4522                              Y:=UEI,L1;
4523                              Z:=UEI,L2;
4524                              UEI,L1:=Y * C + Z * S;
4525                              UEI,L2:=Y * S + Z * C;
4526                          END;
4527                      FCONVERGENCE :
4528                          Z:=UEK,L2;
4529                          E11:=E11 * K THEN GO CONVERGENCE;
4530          X-----
4531          G:=UEK,L2;
4532          H:=UEK,L2;
4533          F:=SQR(F * F + G * G);
4534          S:=F / H;
4535          FOR J=L STEP 1 UNTIL M DO
4536              Y:=UEI,L1;
4537              Z:=UEI,L2;
4538              UEI,L1:=Y * C + Z * S;
4539              UEI,L2:=Y * S + Z * C;
4540          END;
4541          FCONVERGENCE :
4542              Z:=UEK,L2;
4543              E11:=E11 * K THEN GO CONVERGENCE;
4544          X-----
4545          G:=UEK,L2;
4546          H:=UEK,L2;
4547          F:=SQR(F * F + G * G);
4548          S:=F / H;
4549          FOR J=L STEP 1 UNTIL M DO
4550              Y:=UEI,L1;
4551              Z:=UEI,L2;
4552              UEI,L1:=Y * C + Z * S;
4553              UEI,L2:=Y * S + Z * C;
4554          END;
4555          FCONVERGENCE :
4556              Z:=UEK,L2;
4557              E11:=E11 * K THEN GO CONVERGENCE;
4558          X-----
4559          G:=UEK,L2;
4560          H:=UEK,L2;
4561          F:=SQR(F * F + G * G);
4562          S:=F / H;
4563          FOR J=L STEP 1 UNTIL M DO
4564              Y:=UEI,L1;
4565              Z:=UEI,L2;
4566              UEI,L1:=Y * C + Z * S;
4567              UEI,L2:=Y * S + Z * C;
4568          END;
4569          FCONVERGENCE :
4570              Z:=UEK,L2;
4571              E11:=E11 * K THEN GO CONVERGENCE;
4572          X-----
4573          G:=UEK,L2;
4574          H:=UEK,L2;
4575          F:=SQR(F * F + G * G);
4576          S:=F / H;
4577          FOR J=L STEP 1 UNTIL M DO
4578              Y:=UEI,L1;
4579              Z:=UEI,L2;
4580              UEI,L1:=Y * C + Z * S;
4581              UEI,L2:=Y * S + Z * C;
4582          END;
4583          FCONVERGENCE :
4584              Z:=UEK,L2;
4585              E11:=E11 * K THEN GO CONVERGENCE;
4586          X-----
4587          G:=UEK,L2;
4588          H:=UEK,L2;
4589          F:=SQR(F * F + G * G);
4590          S:=F / H;
4591          FOR J=L STEP 1 UNTIL M DO
4592              Y:=UEI,L1;
4593              Z:=UEI,L2;
4594              UEI,L1:=Y * C + Z * S;
4595              UEI,L2:=Y * S + Z * C;
4596          END;
4597          FCONVERGENCE :
4598              Z:=UEK,L2;
4599              E11:=E11 * K THEN GO CONVERGENCE;
4600          X-----
4601          G:=UEK,L2;
4602          H:=UEK,L2;
4603          F:=SQR(F * F + G * G);
4604          S:=F / H;
4605          FOR J=L STEP 1 UNTIL M DO
4606              Y:=UEI,L1;
4607              Z:=UEI,L2;
4608              UEI,L1:=Y * C + Z * S;
4609              UEI,L2:=Y * S + Z * C;
4610          END;
4611          FCONVERGENCE :
4612              Z:=UEK,L2;
4613              E11:=E11 * K THEN GO CONVERGENCE;
4614          X-----
4615          G:=UEK,L2;
4616          H:=UEK,L2;
4617          F:=SQR(F * F + G * G);
4618          S:=F / H;
4619          FOR J=L STEP 1 UNTIL M DO
4620              Y:=UEI,L1;
4621              Z:=UEI,L2;
4622              UEI,L1:=Y * C + Z * S;
4623              UEI,L2:=Y * S + Z * C;
4624          END;
4625          FCONVERGENCE :
4626              Z:=UEK,L2;
4627              E11:=E11 * K THEN GO CONVERGENCE;
4628          X-----
4629          G:=UEK,L2;
4630          H:=UEK,L2;
4631          F:=SQR(F * F + G * G);
4632          S:=F / H;
4633          FOR J=L STEP 1 UNTIL M DO
4634              Y:=UEI,L1;
4635              Z:=UEI,L2;
4636              UEI,L1:=Y * C + Z * S;
4637              UEI,L2:=Y * S + Z * C;
4638          END;
4639          FCONVERGENCE :
4640              Z:=UEK,L2;
4641              E11:=E11 * K THEN GO CONVERGENCE;
4642          X-----
4643          G:=UEK,L2;
4644          H:=UEK,L2;
4645          F:=SQR(F * F + G * G);
4646          S:=F / H;
4647          FOR J=L STEP 1 UNTIL M DO
4648              Y:=UEI,L1;
4649              Z:=UEI,L2;
4650              UEI,L1:=Y * C + Z * S;
4651              UEI,L2:=Y * S + Z * C;
4652          END;
4653          FCONVERGENCE :
4654              Z:=UEK,L2;
4655              E11:=E11 * K THEN GO CONVERGENCE;
4656          X-----
4657          G:=UEK,L2;
4658          H:=UEK,L2;
4659          F:=SQR(F * F + G * G);
4660          S:=F / H;
4661          FOR J=L STEP 1 UNTIL M DO
4662              Y:=UEI,L1;
4663              Z:=UEI,L2;
4664              UEI,L1:=Y * C + Z * S;
4665              UEI,L2:=Y * S + Z * C;
4666          END;
4667          FCONVERGENCE :
4668              Z:=UEK,L2;
4669              E11:=E11 * K THEN GO CONVERGENCE;
4670          X-----
4671          G:=UEK,L2;
4672          H:=UEK,L2;
4673          F:=SQR(F * F + G * G);
4674          S:=F / H;
4675          FOR J=L STEP 1 UNTIL M DO
4676              Y:=UEI,L1;
4677              Z:=UEI,L2;
4678              UEI,L1:=Y * C + Z * S;
4679              UEI,L2:=Y * S + Z * C;
4680          END;
4681          FCONVERGENCE :
4682              Z:=UEK,L2;
4683              E11:=E11 * K THEN GO CONVERGENCE;
4684          X-----
4685          G:=UEK,L2;
4686          H:=UEK,L2;
4687          F:=SQR(F * F + G * G);
4688          S:=F / H;
4689          FOR J=L STEP 1 UNTIL M DO
4690              Y:=UEI,L1;
4691              Z:=UEI,L2;
4692              UEI,L1:=Y * C + Z * S;
4693              UEI,L2:=Y * S + Z * C;
4694          END;
4695          FCONVERGENCE :
4696              Z:=UEK,L2;
4697              E11:=E11 * K THEN GO CONVERGENCE;
4698          X-----
4699          G:=UEK,L2;
4700          H:=UEK,L2;
4701          F:=SQR(F * F + G * G);
4702          S:=F / H;
4703          FOR J=L STEP 1 UNTIL M DO
4704              Y:=UEI,L1;
4705              Z:=UEI,L2;
4706              UEI,L1:=Y * C + Z * S;
4707              UEI,L2:=Y * S + Z * C;
4708          END;
4709          FCONVERGENCE :
4710              Z:=UEK,L2;
4711              E11:=E11 * K THEN GO CONVERGENCE;
4712          X-----
4713          G:=UEK,L2;
4714          H:=UEK,L2;
4715          F:=SQR(F * F + G * G);
4716          S:=F / H;
4717          FOR J=L STEP 1 UNTIL M DO
4718              Y:=UEI,L1;
4719              Z:=UEI,L2;
4720              UEI,L1:=Y * C + Z * S;
4721              UEI,L2:=Y * S + Z * C;
4722          END;
4723          FCONVERGENCE :
4724              Z:=UEK,L2;
4725              E11:=E11 * K THEN GO CONVERGENCE;
4726          X-----
4727          G:=UEK,L2;
4728          H:=UEK,L2;
4729          F:=SQR(F * F + G * G);
4730          S:=F / H;
4731          FOR J=L STEP 1 UNTIL M DO
4732              Y:=UEI,L1;
4733              Z:=UEI,L2;
4734              UEI,L1:=Y * C + Z * S;
4735              UEI,L2:=Y * S + Z * C;
4736          END;
4737          FCONVERGENCE :
4738              Z:=UEK,L2;
4739              E11:=E11 * K THEN GO CONVERGENCE;
4740          X-----
4741          G:=UEK,L2;
4742          H:=UEK,L2;
4743          F:=SQR(F * F + G * G);
4744          S:=F / H;
4745          FOR J=L STEP 1 UNTIL M DO
4746              Y:=UEI,L1;
4747              Z:=UEI,L2;
4748              UEI,L1:=Y * C + Z * S;
4749              UEI,L2:=Y * S + Z * C;
4750          END;
4751          FCONVERGENCE :
4752              Z:=UEK,L2;
4753              E11:=E11 * K THEN GO CONVERGENCE;
4754          X-----
4755          G:=UEK,L2;
4756          H:=UEK,L2;
4757          F:=SQR(F * F + G * G);
4758          S:=F / H;
4759          FOR J=L STEP 1 UNTIL M DO
4760              Y:=UEI,L1;
4761              Z:=UEI,L2;
4762              UEI,L1:=Y * C + Z * S;
4763              UEI,L2:=Y * S + Z * C;
4764          END;
4765          FCONVERGENCE :
4766              Z:=UEK,L2;
4767              E11:=E11 * K THEN GO CONVERGENCE;
4768          X-----
4769          G:=UEK,L2;
4770          H:=UEK,L2;
4771          F:=SQR(F * F + G * G);
4772          S:=F / H;
4773          FOR J=L STEP 1 UNTIL M DO
4774              Y:=UEI,L1;
4775              Z:=UEI,L2;
4776              UEI,L1:=Y * C + Z * S;
4777              UEI,L2:=Y * S + Z * C;
4778          END;
4779          FCONVERGENCE :
4780              Z:=UEK,L2;
4781              E11:=E11 * K THEN GO CONVERGENCE;
4782          X-----
4783          G:=UEK,L2;
4784          H:=UEK,L2;
4785          F:=SQR(F * F + G * G);
4786          S:=F / H;
4787          FOR J=L STEP 1 UNTIL M DO
4788              Y:=UEI,L1;
4789              Z:=UEI,L2;
4790              UEI,L1:=Y * C + Z * S;
4791              UEI,L2:=Y * S + Z * C;
4792          END;
4793          FCONVERGENCE :
4794              Z:=UEK,L2;
4795              E11:=E11 * K THEN GO CONVERGENCE;
4796          X-----
4797          G:=UEK,L2;
4798          H:=UEK,L2;
4799          F:=SQR(F * F + G * G);
4800          S:=F / H;
4801          FOR J=L STEP 1 UNTIL M DO
4802              Y:=UEI,L1;
4803              Z:=UEI,L2;
4804              UEI,L1:=Y * C + Z * S;
4805              UEI,L2:=Y * S + Z * C;
4806          END;
4807          FCONVERGENCE :
4808              Z:=UEK,L2;
4809              E11:=E11 * K THEN GO CONVERGENCE;
4810          X-----
4811          G:=UEK,L2;
4812          H:=UEK,L2;
4813          F:=SQR(F * F + G * G);
4814          S:=F / H;
4815          FOR J=L STEP 1 UNTIL M DO
4816              Y:=UEI,L1;
4817              Z:=UEI,L2;
4818              UEI,L1:=Y * C + Z * S;
4819              UEI,L2:=Y * S + Z * C;
4820          END;
4821          FCONVERGENCE :
4822              Z:=UEK,L2;
4823              E11:=E11 * K THEN GO CONVERGENCE;
4824          X-----
4825          G:=UEK,L2;
4826          H:=UEK,L2;
4827          F:=SQR(F * F + G * G);
4828          S:=F / H;
4829          FOR J=L STEP 1 UNTIL M DO
4830              Y:=UEI,L1;
4831              Z:=UEI,L2;
4832              UEI,L1:=Y * C + Z * S;
4833              UEI,L2:=Y * S + Z * C;
4834          END;
4835          FCONVERGENCE :
4836              Z:=UEK,L2;
4837              E11:=E11 * K THEN GO CONVERGENCE;
4838          X-----
4839          G:=UEK,L2;
4840          H:=UEK,L2;
4841          F:=SQR(F * F + G * G);
4842          S:=F / H;
4843          FOR J=L STEP 1 UNTIL M DO
4844              Y:=UEI,L1;
4845              Z:=UEI,L2;
4846              UEI,L1:=Y * C + Z * S;
4847              UEI,L2:=Y * S + Z * C;
4848          END;
4849          FCONVERGENCE :
4850              Z:=UEK,L2;
4851              E11:=E11 * K THEN GO CONVERGENCE;
4852          X-----
4853          G:=UEK,L2;
4854          H:=UEK,L2;
4855          F:=SQR(F * F + G * G);
4856          S:=F / H;
4857          FOR J=L STEP 1 UNTIL M DO
4858              Y:=UEI,L1;
4859              Z:=UEI,L2;
4860              UEI,L1:=Y * C + Z * S;
4861              UEI,L2:=Y * S + Z * C;
4862          END;
4863          FCONVERGENCE :
4864              Z:=UEK,L2;
4865              E11:=E11 * K THEN GO CONVERGENCE;
4866          X-----
4867          G:=UEK,L2;
4868          H:=UEK,L2;
4869          F:=SQR(F * F + G * G);
4870          S:=F / H;
4871          FOR J=L STEP 1 UNTIL M DO
4872              Y:=UEI,L1;
4873              Z:=UEI,L2;
4874              UEI,L1:=Y * C + Z * S;
4875              UEI,L2:=Y * S + Z * C;
4876          END;
4877          FCONVERGENCE :
4878              Z:=UEK,L2;
4879              E11:=E11 * K THEN GO CONVERGENCE;
4880          X-----
4881          G:=UEK,L2;
4882          H:=UEK,L2;
4883          F:=SQR(F * F + G * G);
4884          S:=F / H;
4885          FOR J=L STEP 1 UNTIL M DO
4886              Y:=UEI,L1;
4887              Z:=UEI,L2;
4888              UEI,L1:=Y * C + Z * S;
4889              UEI,L2:=Y * S + Z * C;
4890          END;
4891          FCONVERGENCE :
4892              Z:=UEK,L2;
4893              E11:=E11 * K THEN GO CONVERGENCE;
4894          X-----
4895          G:=UEK,L2;
4896          H:=UEK,L2;
4897          F:=SQR(F * F + G * G);
4898          S:=F / H;
4899          FOR J=L STEP 1 UNTIL M DO
4900              Y:=UEI,L1;
4901              Z:=UEI,L2;
4902              UEI,L1:=Y * C + Z * S;
4903              UEI,L2:=Y * S + Z * C;
4904          END;
4905          FCONVERGENCE :
4906              Z:=UEK,L2;
4907              E11:=E11 * K THEN GO CONVERGENCE;
4908          X-----
4909          G:=UEK,L2;
4910          H:=UEK,L2;
4911          F:=SQR(F * F + G * G);
4912          S:=F / H;
4913          FOR J=L STEP 1 UNTIL M DO
4914              Y:=UEI,L1;
4915              Z:=UEI,L2;
4916              UEI,L1:=Y * C + Z * S;
4917              UEI,L2:=Y * S + Z * C;
4918          END;
4919          FCONVERGENCE :
4920              Z:=UEK,L2;
4921              E11:=E11 * K THEN GO CONVERGENCE;
4922          X-----
4923          G:=UEK,L2;
4924          H:=UEK,L2;
4925          F:=SQR(F * F + G * G);
4926          S:=F / H;
4927          FOR J=L STEP 1 UNTIL M DO
4928              Y:=UEI,L1;
4929              Z:=UEI,L2;
4930              UEI,L1:=Y * C + Z * S;
4931              UEI,L2:=Y * S + Z * C;
4932          END;
4933          FCONVERGENCE :
4934              Z:=UEK,L2;
4935              E11:=E11 * K THEN GO CONVERGENCE;
4936          X-----
4937          G:=UEK,L2;
4938          H:=UEK,L2;
4939          F:=SQR(F * F + G * G);
4940          S:=F / H;
4941          FOR J=L STEP 1 UNTIL M DO
4942              Y:=UEI,L1;
4943              Z:=UEI,L2;
4944              UEI,L1:=Y * C + Z * S;
4945              UEI,L2:=Y * S + Z * C;
4946          END;
4947          FCONVERGENCE :
4948              Z:=UEK,L2;
4949              E11:=E11 * K THEN GO CONVERGENCE;
4950          X-----
4951          G:=UEK,L2;
4952          H:=UEK,L2;
4953          F:=SQR(F * F + G * G);
4954          S:=F / H;
4955          FOR J=L STEP 1 UNTIL M DO
4956              Y:=UEI,L1;
4957              Z:=UEI,L2;
4958              UEI,L1:=Y * C + Z * S;
4959              UEI,L2:=Y * S + Z * C;
4960          END;
4961          FCONVERGENCE :
4962              Z:=UEK,L2;
4963              E11:=E11 * K THEN GO CONVERGENCE;
4964          X-----
4965          G:=UEK,L2;
4966          H:=UEK,L2;
4967          F:=SQR(F * F + G * G);
4968          S:=F / H;
4969          FOR J=L STEP 1 UNTIL M DO
4970              Y:=UEI,L1;
4971              Z:=UEI,L2;
4972              UEI,L1:=Y * C + Z * S;
4973              UEI,L2:=Y * S + Z * C;
4974          END;
4975          FCONVERGENCE :
4976              Z:=UEK,L2;
4977              E11:=E11 * K THEN GO CONVERGENCE;
4978          X-----
4979          G:=UEK,L2;
4980          H:=UEK,L2;
4981          F:=SQR(F * F + G * G);
4982          S:=F / H;
4983          FOR J=L STEP 1 UNTIL M DO
4984              Y:=UEI,L1;
4985              Z:=UEI,L2;
4986              UEI,L1:=Y * C + Z * S;
4987              UEI,L2:=Y * S + Z * C;
4988          END;
4989          FCONVERGENCE :
4990              Z:=UEK,L2;
4991              E11:=E11 * K THEN GO CONVERGENCE;
4992          X-----
4993          G:=UEK,L2;
4994          H:=UEK,L2;
4995          F:=SQR(F * F + G * G);
4996          S:=F / H;
4997          FOR J=L STEP 1 UNTIL M DO
4998              Y:=UEI,L1;
4999              Z:=UEI,L2;
5000              UEI,L1:=Y * C + Z * S;
5001              UEI,L2:=Y * S + Z * C;
5002          END;
5003          FCONVERGENCE :
5004              Z:=UEK,L2;
5005              E11:=E11 * K THEN GO CONVERGENCE;
5006          X-----
5007          G:=UEK,L2;
5008          H:=UEK,L2;
5009          F:=SQR(F * F + G * G);
5010          S:=F / H;
5011          FOR J=L STEP 1 UNTIL M DO
5012              Y:=UEI,L1;
5013              Z:=UEI,L2;
5014              UEI,L1:=Y * C + Z * S;
5015              UEI,L2:=Y * S + Z * C;
5016          END;
5017          FCONVERGENCE :
5018              Z:=UEK,L2;
5019              E11:=E11 * K THEN GO CONVERGENCE;
5020          X-----
5021          G:=UEK,L2;
5022          H:=UEK,L2;
5023          F:=SQR(F * F + G * G);
5024          S:=F / H;
5025          FOR J=L STEP 1 UNTIL M DO
5026              Y:=UEI,L1;
5027              Z:=UEI,L2;
5028              UEI,L1:=Y * C + Z * S;
5029              UEI,L2:=Y * S + Z * C;
5030          END;
5031          FCONVERGENCE :
5032              Z:=UEK,L2;
5033              E11:=E11 * K THEN GO CONVERGENCE;
5034          X-----
5035          G:=UEK,L2;
5036          H:=UEK,L2;
5037          F:=SQR(F * F + G * G);
5038          S:=F / H;
5039          FOR J=L STEP 1 UNTIL M DO
5040              Y:=UEI,L1;
5041              Z:=UEI,L2;
5042              UEI,L1:=Y * C + Z * S;
5043              UEI,L2:=Y * S + Z * C;
5044          END;
5045          FCONVERGENCE :
5046              Z:=UEK,L2;
5047              E11:=E11 * K THEN GO CONVERGENCE;
5048          X-----
5049          G:=UEK,L2;
5050          H:=UEK,L2;
5051          F:=SQR(F * F + G * G);
5052          S:=F / H;
5053          FOR J=L STEP 1 UNTIL M DO
5054              Y:=UEI,L1;
5055              Z:=UEI,L2;
5056              UEI,L1:=Y * C + Z * S;
5057              UEI,L2:=Y * S + Z * C;
5058          END;
5059          FCONVERGENCE :
5060              Z:=UEK,L2;
5061              E11:=E11 * K THEN GO CONVERGENCE;
5062          X-----
5063          G:=UEK,L2;
5064          H:=UEK,L2;
5065          F:=SQR(F * F + G * G);
5066          S:=F / H;
5067          FOR J=L STEP 1 UNTIL M DO
5068              Y:=UEI,L1;
5069              Z:=UEI,L2;
5070              UEI,L1:=Y * C + Z * S;
5071              UEI,L2:=Y * S + Z * C;
5072          END;
5073          FCONVERGENCE :
5074              Z:=UEK,L2;
5075              E11:=E11 * K THEN GO CONVERGENCE;
5076          X-----
5077          G:=UEK,L2;
5078          H:=UEK,L2;
5079          F:=SQR(F * F + G * G);
5080          S:=F / H;
5081          FOR J=L STEP 1 UNTIL M DO
5082              Y:=UEI,L1;
5083              Z:=UEI,L2;
5084              UEI,L1:=Y * C + Z * S;
5085              UEI,L2:=Y * S + Z * C;
5086          END;
5087          FCONVERGENCE :
5088              Z:=UEK,L2;
5089              E11:=E11 * K THEN GO CONVERGENCE;
5090          X-----
5091          G:=UEK,L2;
5092          H:=UEK,L2;
5093          F:=SQR(F * F + G * G);
5094          S:=F / H;
5095          FOR J=L STEP 1 UNTIL M DO
5096              Y:=UEI,L1;
5097              Z:=UEI,L2;
5098              UEI,L1:=Y * C + Z * S;
5099              UEI,L2:=Y * S + Z * C;
5100          END;
5101          FCONVERGENCE :
5102              Z:=UEK,L2;
5103              E11:=E11 * K THEN GO CONVERGENCE;
5104          X-----
5105          G:=UEK,L2;
5106          H:=UEK,L2;
5107          F:=SQR(F * F + G * G);
5108          S:=F / H;
5109          FOR J=L STEP 1 UNTIL M DO
5110              Y:=UEI,L1;
5111              Z:=UEI,L2;
5112              UEI,L1:=Y * C + Z * S;
5113              UEI,L2:=Y * S + Z * C;
5114          END;
5115          FCONVERGENCE :
5116              Z:=UEK,L2;
5117              E11:=E11 * K THEN GO CONVERGENCE;
5118          X-----
5119          G:=UEK,L2;
5120          H:=UEK,L2;
5121          F:=SQR(F * F + G * G);
5122          S:=F / H;
5123          FOR J=L STEP 1 UNTIL M DO
5124              Y:=UEI,L1;
5125              Z:=UEI,L2;
5126              UEI,L1:=Y * C + Z * S;
5127              UEI,L2:=Y * S + Z * C;
5128          END;
5129          FCONVERGENCE :
5130              Z:=UEK,L2;
5131              E11:=E11 * K THEN GO CONVERGENCE;
5132          X-----
5133          G:=UEK,L2;
5134          H:=UEK,L2;
5135          F:=SQR(F * F + G * G);
5136          S:=F / H;
5137          FOR J=L STEP 1 UNTIL M DO
5138              Y:=UEI,L1;
5139              Z:=UEI,L2;
5140              UEI,L1:=Y * C + Z * S;
5141              UEI,L2:=Y * S + Z * C;
5142          END;
5143          FCONVERGENCE :
5144              Z:=UEK,L2;
5145              E11:=E11 * K THEN GO CONVERGENCE;
5146          X-----
5147          G:=UEK,L2;
5148          H:=UEK,L2;
5149          F:=SQR(F * F + G * G);
5150          S:=F / H;
5151          FOR J=L STEP 1 UNTIL M DO
5152              Y:=UEI,L1;
5153              Z:=UEI,L2;
5154              UEI,L1:=Y * C + Z * S;
5155              UEI,L2:=Y * S + Z * C;
5156          END;
5157          FCONVERGENCE :
5158              Z:=UEK,L2;
5159              E11:=E11 * K THEN GO CONVERGENCE;
5160          X-----
5161          G:=UEK,L2;
5162          H:=UEK,L2;
5163          F:=SQR(F * F + G * G);
5164          S:=F / H;
5165          FOR J=L STEP 1 UNTIL M DO
5166              Y:=UEI,L1;
5167              Z:=UEI,L2;
5168              UEI,L1:=Y * C + Z * S;
5169              UEI,L2:=Y * S + Z * C;
5170          END;
5171          FCONVERGENCE :
5172              Z:=UEK,L2;
5173              E11:=E11 * K THEN GO CONVERGENCE;
5174          X-----
5175          G:=UEK,L2;
5176          H:=UEK,L2;
5177          F:=SQR(F * F + G * G);
5178          S:=F / H;
5179          FOR J=L STEP 1 UNTIL M DO
5180              Y:=UEI,L1;
5181              Z:=UEI,L2;
5182              UEI,L1:=Y * C + Z * S;
5183              UEI,L2:=Y * S + Z * C;
5184          END;
5185          FCONVERGENCE :
5186              Z:=UEK,L2;
5187              E11:=E11 * K THEN GO CONVERGENCE;
5188          X-----
5189          G:=UEK,L2;
5190          H:=UEK,L2;
5191          F:=SQR(F * F + G * G);
5192          S:=F / H;
5193          FOR J=L STEP 1 UNTIL M DO
5194              Y:=UEI,L1;
5195              Z:=UEI,L2;
5196              UEI,L1:=Y * C + Z * S;
5197              UEI,L2:=Y * S + Z * C;
5198          END;
5199          FCONVERGENCE :
5200              Z:=UEK,L2;
5201              E11:=E11 * K THEN GO CONVERGENCE;
5202          X-----
5203          G:=UEK,L2;
5204          H:=UEK,L2;
5205          F:=SQR(F * F + G * G);
5206          S:=F / H;
5207          FOR J=L STEP 1 UNTIL M DO
5208              Y:=UEI,L1;
5209              Z:=UEI,L2;
5210              UEI,L1:=Y * C + Z * S;
5211              UEI,L2:=Y * S + Z * C;
5212          END;
5213          FCONVERGENCE :
5214              Z:=UEK,L2;
5215              E11:=E11 * K THEN GO CONVERGENCE;
5216          X-----
5217          G:=UEK,L2;
5218          H:=UEK,L2;
5219          F:=SQR(F * F + G * G);
5220          S:=F / H;
5221          FOR J=L STEP 1 UNTIL M DO
5222              Y:=UEI,L1;
5223              Z:=UEI,L2;
5224              UEI,L1:=Y * C + Z * S;
5225              UEI,L2:=Y * S + Z * C;
5226          END;
5227          FCONVERGENCE :
5228              Z:=UEK,L2;
5229              E11:=E11 * K THEN GO CONVERGENCE;
5230          X-----
5231          G:=UEK,L2;
5232          H:=UEK,L2;
5233          F:=SQR(F * F + G * G);
5234          S:=F / H;
5235          FOR J=L STEP 1 UNTIL M DO
5236              Y:=UEI,L1;
5237              Z:=UEI,L2;
5238              UEI,L1:=Y * C + Z * S;
5239              UEI,L2:=Y * S + Z * C;
5240          END;
5241          FCONVERGENCE :
5242              Z:=UEK,L2;
5243              E11:=E11 * K THEN GO CONVERGENCE;
5244          X-----
5245          G:=UEK,L2;
5246          H:=UEK,L2;
5247          F:=SQR(F * F + G * G);
5248          S:=F / H;
5249          FOR J=L STEP 1 UNTIL M DO
5250              Y:=UEI,L1;
5251              Z:=UEI,L2;
5252              UEI,L1:=Y * C + Z * S;
5253              UEI,L2:=Y * S + Z * C;
5254          END;
5255          FCONVERGENCE :
5256              Z:=UEK,L2;
5257              E11:=E11 * K THEN GO CONVERGENCE;
5258          X-----
5259          G:=UEK,L2;
5260          H:=UEK,L2;
5261          F:=SQR(F * F + G * G);
5262          S:=F / H;
5263          FOR J=L STEP 1 UNTIL M DO
5264              Y:=UEI,L1;
5265              Z:=UEI,L2;
5266              UEI,L1:=Y * C + Z * S;
5267              UEI,L2:=Y * S + Z * C;
5268          END;
5269          FCONVERGENCE :
5270              Z:=UEK,L2;
5271              E11:=E11 * K THEN GO CONVERGENCE;
5272          X-----
5273          G:=UEK,L2;
5274          H:=UEK,L2;
5275          F:=SQR(F * F + G * G);
5276          S:=F / H;
5277          FOR J=L STEP 1 UNTIL M DO
5278              Y:=UEI,L1;
5279              Z:=UEI,L2;
5280              UEI,L1:=Y * C + Z * S;
5281              UEI,L2:=Y * S + Z * C;
5282          END;
5283          FCONVERGENCE :
5284              Z:=UEK,L2;
5285              E11:=E11 * K THEN GO CONVERGENCE;
5286          X-----
5287          G:=UEK,L2;
5288          H:=UEK,L2;
5289          F:=SQR(F * F + G * G);
5290          S:=F / H;
5291          FOR J=L STEP 1 UNTIL M DO
5292              Y:=UEI,L1;

```



```

JCC:=L;
END;
END;
IF M=0 = J THEN
BEGIN
WRITE(TTY, '<' * MATRIZ DE DATOS VACIA * '>');
TAREA ABORTADA-
GO OSEOT;
END;
FOR K:=1 STEP 1 UNTIL N DO
DEALLOCATE(WK, *);
END;
PROCEDURE INITCODE;
IF CT = 1 THEN * CONFIGURACION INICIAL DISPONIBLE
BEGIN
INPUTDESK(OSFPT);
TINPUTDATA(X, N, T, CSEOT);
END
ELSE
BEGIN
INTEGER T, J;
REAL SEED;
SEED:=TIME(11);
FOR I:=1 STEP 1 UNTIL N DO
FOR J:=1 STEP 1 UNTIL T DO
XC(I, J) := RANDOP(SEED);
END;
PROCEDURE TILS;
----- PARA CADA ENTRADA DE LA MATRIZ DE DATOS, ENCUENTRA EL NUMERO -----
X----- DE FUERA DE LA MATRIZ A ESTE TIE(K) = 0 SI TIE(K) < 0 -----
X----- DISSIM(L) < DISSIM(K) PARA TODA K < L -----
BEGIN
DO
BEGIN
K := K + 1;
DISSIM(K) := DISSIM(K) + 1;
WHILE DISSIM(K) = DISSIM(L) DO
TIE(K) := L - K;
K := L;
END;
UNTIL K > N;
END;
PROCEDURE DISTANCES;
BEGIN
REAL EXP;
EXP:=1;
FOR L:=1 STEP 1 UNTIL M DO
BEGIN
DIST(L) := 0;
FOR S:=1 STEP 1 UNTIL T DO
DIST(L) := * * RES(XC(L, S) - XC(L, S)) * * R;
DIST(L) := * * EXP;
END;
END;
PROCEDURE NORMALIZE;
BEGIN
INTEGER T, S;
REAL MEAN(S);
FOR S:=1 STEP 1 UNTIL T DO
BEGIN
FOR I:=1 STEP 1 UNTIL N DO
XC(I, S) := * * XC(I, S);
MEAN(S) := * * N;
END;
FOR I:=1 STEP 1 UNTIL N DO
FOR S:=1 STEP 1 UNTIL T DO
XC(I, S) := * * MEAN(S);
XC(I, S) := * * XC(I, S) * * 2;
SE := SE * (CE / N);
FOR I:=1 STEP 1 UNTIL N DO
FOR S:=1 STEP 1 UNTIL T DO
XC(I, S) := * * SE;
END;
END;
PROCEDURE PRIMARY;
BEGIN
INTEGER K, NT, F;
K:=1;
F:=1;
DO UNTIL = 0 THEN -1 ELSE 1;
WHILE K <= N DO
IF K >= N THEN
BEGIN
INVERT(D, INDEF, TIE(K));
INVERT(L, U);
S:=K + TIE(K) - 1;
NT:=TIE(K);
TIE(K) := 0;
FOR I:=K STEP 1 UNTIL S DO
BEGIN
U:= * * + 1;
D(I) := DIST(L) * F;
INVC(SORT(CO, IND, NT));
END;
U:=1;
FOR I:=K STEP 1 UNTIL S DO
BEGIN
U:= * * + 1;
OSTYLE(D)=DUAL(L) := DCU) * F;
DOTHER(L) := 0;
LJLL:=INDCU);
END;
TIE(K) := NT;
K := S + 1;
END;
ELSE
BEGIN
WRITE(K) := DIST(K);
DOTHER(K) := 0;
K := * * + 1;
END;
END;
PROCEDURE SECONDARY;
BEGIN
INTEGER K;
K:=1;
DO
BEGIN
M:=M+1;
IF T(K) < 2 THEN
M := M + 1;
WRITE(K) := DIST(K);
END;
END;

```

```

2110 DOOTHERK:=0;
2120 K:=K+1;
2130 END
2140 BEGIN
2150 INTEGER L;
2160 REAL AC;
2170 INTEGER S;
2180 S:=K+TIE(K)-1;
2190 DOOTHERK:=K;
2200 FOR L:=K STEP 1 UNTIL S DO
2210 AC:=*+DSTEL;
2220 DHATE(K)=AC/TIE(K);
2230 DHATE(K+1):=AC;
2240 W:=S+1;
2250 END;
2260 UNTIL K > 4;
2270 END;
2280 PROCEDURE MONOTONE;
2290 BEGIN
2300 INTEGER FIRST,K,LAST;
2310 ACCEM STATE UP,DOWN;
2320 INTEGER PROCEDURE EPB(L);
2330 INTEGER L;
2340 IF DOOTHER(L) = 0 THEN
2350 DO:=1;
2360 ELSE
2370 DO:=DOOTHER(L);
2380 REAL PROCEDURE FIT(L);
2390 INTEGER L;
2400 IF DOOTHER(L) = 0 THEN
2410 FIT:=DHATE(L);
2420 ELSE
2430 FIT:=DHATE(L+1);
2440 INTEGER PROCEDURE INDICE(L);
2450 INTEGER L;
2460 IF DOOTHER(L) = 0 THEN
2470 INDICE:=L;
2480 ELSE
2490 INDICE:=DOOTHER(L);
2500 PROCEDURE JOIN;
2510 BEGIN
2520 LACT:=FIRST+(IF DOOTHER(FIRST) = 0 THEN 0
2530 ELSE DOOTHER(FIRST)-1);
2540 DHATE(LACT)=EPB(K)+EPB(FIRST);
2550 DOOTHER(KACT)=EPB(K)+EPB(FIRST);
2560 DHATE(K)=DHATE(K+1)/DOOTHER(K);
2570 DOOTHER(LACT)=K;
2580 END;
2590 OR FINE
2600 UPACTIVE = TRUE #;
2610 DOWNACTIVE = FALSE#;
2620 W:=1;
2630 DO
2640 BEGIN
2650 STATE:=UPACTIVE;
2660 DOWN:=FALSE;
2670 DO
2680 IF STATE THEN
2690 BEGIN
2700 IF FIRST=(K+(IF DOOTHER(K)=0 THEN 0
2710 ELSE DOOTHER(K)-1)) = W THEN
2720 UP:=TRUE
2730 ELSE
2740 IF ACT UP:=DHATE(K) < DHATEFIRST:=*+1 THEN
2750 JOIN
2760 STATE:=DOWNACTIVE;
2770 END
2780 ELSE
2790 BEGIN
2800 IF K = 1 THEN
2810 DOWN:=TRUE
2820 ELSE
2830 IF ACT DOWN:=DHATE(INDICE(K)-1) < DHATE(K) THEN
2840 FIRST:=K;
2850 K:=INDICE(K-1);
2860 JOIN;
2870 END;
2880 STATE:=UPACTIVE;
2890 END
2900 UNTIL UP AND DOWN;
2910 K:=*+(IF DOOTHER(K) = 1 THEN 1 ELSE DOOTHER(K));
2920 END
2930 UNTIL K > 4;
2940 K:=1;
2950 BEGIN
2960 REAL FITTED;
2970 INTEGER L;
2980 S:=K+TIE(K)-1;
2990 FIT:=DHATE(K);
3000 FOR L:=K STEP 1 UNTIL S DO
3010 DHATE(L):=FITTED;
3020 K:=S+1;
3030 END
3040 UNTIL K > 4;
3050 END;
3060 PROCEDURE FIT;
3070 BEGIN
3080 IF 1 = 0 THEN
3090 PRIMARY;
3100 ELSE
3110 SECONDARY;
3120 MONOTONE;
3130 END;
3140 PROCEDURE TRACE;
3150 BEGIN
3160 IF IT = 1 THEN
3170 BEGIN
3180 PSI:=FIAS:=S;
3190 FILL SAVE[4] WITH S(0);
3200 END;
3210 IF IT > 5 THEN
3220 BEGIN
3230 INTEGER I;
3240 I:=IT MOD 5;
3250 FIAS:=SAVE[I];
3260 SAVE[I]:=S;
3270 END
3280 ELSE
3290 SAVE[IT MOD 5]:=S;
3300 END;
3310 PROCEDURE STRESS;
3320 BEGIN
3330 INTEGER P;
3340 SASTK:=;
3350 TASTK:=;

```

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```
FOR K:=1 STEP 1 UNTIL M DO  
  BEGIN  
    SASTK:= * + (DISTEK) - DHATK) * * 2;  
    TASTK:= * + DISTEK) * * 2;  
  END;  
  PS:= *;  
  S:=SORT(SASTK / TASTK);  
  TRACE;  
END;  
PROCEDURE ASSIGN;  
  BEGIN  
    INTEGER I,S;  
    FOR I:=1 STEP 1 UNTIL N DO  
      FOR S:=1 STEP 1 UNTIL T DO  
        BEGIN  
          OLDGRAD1,S:=GRAD1,S;  
          GRAD1,S:=I;  
        END;  
      END;  
    END;  
  END;  
PROCEDURE GRADIENT;  
  BEGIN  
    REAL X,L;  
    FOR L:=1 STEP 1 UNTIL N DO  
      FOR X:=1 STEP 1 UNTIL T DO  
        BEGIN  
          REAL TERM;  
          TERM:=(DISTEK) - DHATK) / SASTK;  
          TERM:= * - DISTEK) / TASTK;  
          TERM:= * S / DISTEK) * * EXP;  
          FOR S:=1 STEP 1 UNTIL T DO  
            BEGIN  
              REAL D,ADD;  
              D:=X(K),L) - X(K),L);  
              ADD:=X(K),L) * (ABS(D) * * EXP) * SIGN(D);  
              GRAD1(K),L:= * + ADD;  
              GRAD1(K),L:= * + ADD;  
            END;  
          END;  
        END;  
      END;  
    END;  
  END;  
REAL PROCEDURE RELMAG;  
  BEGIN  
    INTEGER I,S;  
    REAL A;  
    FOR I:=1 STEP 1 UNTIL N DO  
      FOR S:=1 STEP 1 UNTIL T DO  
        BEGIN  
          A:=SORT(A / N);  
          RELMAG:=A;  
        END;  
      END;  
    END;  
  END;  
REAL PROCEDURE COS;  
  BEGIN  
    IF IT = 1 THEN  
      COS:=1  
    ELSE  
      BEGIN  
        INTEGER I,S;  
        REAL GNORM,GNORM;  
        FOR I:=1 STEP 1 UNTIL N DO  
          FOR S:=1 STEP 1 UNTIL T DO  
            BEGIN  
              GNORM:= * + GRAD1,S) * OLDGRAD1,S);  
              GNORM:= * + GRAD1,S) * * 2;  
              GNORM:= * + OLDGRAD1,S) * * 2;  
            END;  
          END;  
        COS:=GNORM / SORT(GNORM * GNORM);  
      END;  
    END;  
  END;  
REAL PROCEDURE FINDALPHA;  
  BEGIN  
    REAL AF,FR,GF,FF;  
    AF:= * + (EGG * * 3);  
    FR:= * + (FGS / FLS);  
    GF:= * + (G / FS);  
    FF:= * + (F / F);  
    AF:= * + (F * FR * * 3);  
    AF:= * + (F * GF);  
    FINDALPHA:=AF;  
  END;  
PROCEDURE INITNEW;  
  BEGIN  
    INTEGER I,S;  
    REAL STEP,IT;  
    IF OK THEN MAG:=RELMAG * LRC.INEN  
    BEGIN  
      STEP:=FINDALPHA / MAG;  
      FOR I:=1 STEP 1 UNTIL N DO  
        FOR S:=1 STEP 1 UNTIL T DO  
          X(I),S:= * + STEP * GRAD1,S);  
        END;  
      END;  
    END;  
  END;  
PROCEDURE UNSP(COP);  
  REAL COP;  
  CASE OF  
    BEGIN  
      I := BOOLEAN(COP);  
      I := BEGIN UP:=BOOLEAN(COP);INITNEW;END;  
    ELSE  
      I:=0;  
    END;  
  END;  
PROMPT;  
INPUT(COP);  
INPUT(N,M,T,SEOT);  
VALIDATE;  
INITCON;  
DSSORT;  
TRACE;  
UP:= *;  
INITNEW;  
WHILE UP DO  
  BEGIN  
    D:= *;  
    BEGIN  
      NORMALIZE;  
      DIFFERENCES;  
      FILE;  
      STEP;  
      GRADIENT;  
      UNSP(COP);  
    END;  
    END;  
    UNTIL NOT OK OR I) > ITHAX;  
  PRINT(I);
```

U234400
U234401
U234402
U234403
U234404
U234405
U234406
U234407
U234408
U234409
U234410
U234411
U234412
U234413
U234414
U234415
U234416
U234417
U234418
U234419
U234420
U234421
U234422
U234423
U234424
U234425
U234426
U234427
U234428
U234429
U234430
U234431
U234432
U234433
U234434
U234435
U234436
U234437
U234438
U234439
U234440
U234441
U234442
U234443
U234444
U234445
U234446
U234447
U234448
U234449
U234450
U234451
U234452
U234453
U234454
U234455
U234456
U234457
U234458
U234459
U234460
U234461
U234462
U234463
U234464
U234465
U234466
U234467
U234468
U234469
U234470
U234471
U234472
U234473
U234474
U234475
U234476
U234477
U234478
U234479
U234480
U234481
U234482
U234483
U234484
U234485
U234486
U234487
U234488
U234489
U234490
U234491
U234492
U234493
U234494
U234495
U234496
U234497
U234498
U234499
U234500
U234501
U234502
U234503
U234504
U234505
U234506
U234507
U234508
U234509
U234510
U234511
U234512
U234513
U234514
U234515
U234516
U234517
U234518
U234519
U234520
U234521
U234522
U234523
U234524
U234525
U234526
U234527
U234528
U234529
U234530
U234531
U234532
U234533
U234534
U234535
U234536
U234537
U234538
U234539
U234540
U234541
U234542
U234543
U234544
U234545
U234546
U234547
U234548
U234549
U234550
U234551
U234552
U234553
U234554
U234555
U234556
U234557
U234558
U234559
U234560
U234561
U234562
U234563
U234564
U234565
U234566
U234567
U234568
U234569
U234570
U234571
U234572
U234573
U234574
U234575
U234576
U234577
U234578
U234579
U234580
U234581
U234582
U234583
U234584
U234585
U234586
U234587
U234588
U234589
U234590
U234591
U234592
U234593
U234594
U234595
U234596
U234597
U234598
U234599
U234600
U234601
U234602
U234603
U234604
U234605
U234606
U234607
U234608
U234609
U234610
U234611
U234612
U234613
U234614
U234615
U234616
U234617
U234618
U234619
U234620
U234621
U234622
U234623
U234624
U234625
U234626
U234627
U234628
U234629
U234630
U234631
U234632
U234633
U234634
U234635
U234636
U234637
U234638
U234639
U234640
U234641
U234642
U234643
U234644
U234645
U234646
U234647
U234648
U234649
U234650
U234651
U234652
U234653
U234654
U234655
U234656
U234657
U234658
U234659
U234660
U234661
U234662
U234663
U234664
U234665
U234666
U234667
U234668
U234669
U234670
U234671
U234672
U234673
U234674
U234675
U234676
U234677
U234678
U234679
U234680
U234681
U234682
U234683
U234684
U234685
U234686
U234687
U234688
U234689
U234690
U234691
U234692
U234693
U234694
U234695
U234696
U234697
U234698
U234699
U234700
U234701
U234702
U234703
U234704
U234705
U234706
U234707
U234708
U234709
U234710
U234711
U234712
U234713
U234714
U234715
U234716
U234717
U234718
U234719
U234720
U234721
U234722
U234723
U234724
U234725
U234726
U234727
U234728
U234729
U234730
U234731
U234732
U234733
U234734
U234735
U234736
U234737
U234738
U234739
U234740
U234741
U234742
U234743
U234744
U234745
U234746
U234747
U234748
U234749
U234750
U234751
U234752
U234753
U234754
U234755
U234756
U234757
U234758
U234759
U234760
U234761
U234762
U234763
U234764
U234765
U234766
U234767
U234768
U234769
U234770
U234771
U234772
U234773
U234774
U234775
U234776
U234777
U234778
U234779
U234780
U234781
U234782
U234783
U234784
U234785
U234786
U234787
U234788
U234789
U234790
U234791
U234792
U234793
U234794
U234795
U234796
U234797
U234798
U234799
U234800
U234801
U234802
U234803
U234804
U234805
U234806
U234807
U234808
U234809
U234810
U234811
U234812
U234813
U234814
U234815
U234816
U234817
U234818
U234819
U234820
U234821
U234822
U234823
U234824
U234825
U234826
U234827
U234828
U234829
U234830
U234831
U234832
U234833
U234834
U234835
U234836
U234837
U234838
U234839
U234840
U234841
U234842
U234843
U234844
U234845
U234846
U234847
U234848
U234849
U234850
U234851
U234852
U234853
U234854
U234855
U234856
U234857
U234858
U234859
U234860
U234861
U234862
U234863
U234864
U234865
U234866
U234867
U234868
U234869
U234870
U234871
U234872
U234873
U234874
U234875
U234876
U234877
U234878
U234879
U234880
U234881
U234882
U234883
U234884
U234885
U234886
U234887
U234888
U234889
U234890
U234891
U234892
U234893
U234894
U234895
U234896
U234897
U234898
U234899
U234900
U234901
U234902
U234903
U234904
U234905
U234906
U234907
U234908
U234909
U234910
U234911
U234912
U234913
U234914
U234915
U234916
U234917
U234918
U234919
U234920
U234921
U234922
U234923
U234924
U234925
U234926
U234927
U234928
U234929
U234930
U234931
U234932
U234933
U234934
U234935
U234936
U234937
U234938
U234939
U234940
U234941
U234942
U234943
U234944
U234945
U234946
U234947
U234948
U234949
U234950
U234951
U234952
U234953
U234954
U234955
U234956
U234957
U234958
U234959
U234960
U234961
U234962
U234963
U234964
U234965
U234966
U234967
U234968
U234969
U234970
U234971
U234972
U234973
U234974
U234975
U234976
U234977
U234978
U234979
U234980
U234981
U234982
U234983
U234984
U234985
U234986
U234987
U234988
U234989
U234990
U234991
U234992
U234993
U234994
U234995
U234996
U234997
U234998
U234999
U235000

