EL PROCESO ESTADISTICO DE DATOS BAJO EL SISTEMA OPERATIVO MS-DOS: LIMITES Y COMO SOBREPASARLOS

Félix Aparicio Pérez
CIS, Madrid

RESUMEN. Este trabajo trata sobre el tema de las limitaciones con que se encuentra toda persona que intenta realizar procesos estadísticos de datos en ordenadores personales bajo el Sistema Operativo MS-DOS. Se exponen las necesidades de memoria y tiempo de los procesos estadísticos más comunes, se explica dónde está el límite en estos procesos bajo MS-DOS y se exponen estrategias para sobrepasar estos límites.

1. EL PROCESO ESTADISTICO DE DATOS

Comenzaremos por describir cómo es un proceso estadístico de datos en general:

Se dispone de un conjunto de datos que normalmente tienen forma de matriz rectangular; en ella, las filas (p. ej.) representan a los individuos, y las columnas, a las variables. Supongamos que hay n individuos y p variables.

Se desean explotar estadísticamente estos datos; con ello queremos decir que se van a hallar distribuciones marginales de variables, tablas de contingencia entre variables, regresiones, análisis de la varianza, análisis factoriales, etcétera, así como crear gráficos que representan las estadísticas que efectuemos. También pueden crearse ficheros de datos intermedios o de salida.

En primer lugar, el programa estadístico lee la matriz de datos y la almacena traducida a un formato, en el cual le resulta cómoda manejar. En esta fase también almacena información de cuántas variables hay, sus nombres y etiquetas y las etiquetas de sus categorías (variables categóricas).

Después, para cada análisis estadístico pedido vuelve a dar una o varias pasadas de lectura a los datos, aparte de realizar a otros procesos (por ejemplo, en un análisis factorial da una pasada de lectura a los datos para estimar la matriz de correlaciones, después efectúa cálculos matemáticos con esta matriz y, finalmente, si se piden las puntuaciones de los individuos en los factores, tiene que volver a leer los datos y escribir estas puntuaciones).

Vemos, pues, que aparte de los cálculos matemáticos, que pueden ser de gran complicación, se debe acceder repetidamente a la matriz de datos.

2. CONCEPTOS INFORMATICOS NECESARIOS PARA COMPRENDER EL TRABAJO

Se llama bit a una unidad binaria de información (0-1, SI-NO, etc.).

Se llama byte a un conjunto de 8 bits. En ficheros de texto y datos «normales», un byte es lo que ocupa un carácter (letra, número, símbolo especial).

Se llama memoria RAM del ordenador a la memoria central del mismo; ésta puede ser accedida (ya sea para leer en ella o para escribir en ella) a una velocidad muy grande. En el caso de ordenadores personales del tipo de los que funcionan bajo DOS, el tiempo de acceso puede ser tan bajo como 80 nanosegundos (un nanosegundo es una milmillonésima de segundo). En este trabajo, cuando se hable de memoria, a secas, se entenderá memoria RAM.

Se llama memoria periférica del ordenador a la memoria situada en dispositivos distintos de la memoria central (típicamente disco duro en ordenadores personales). El tiempo de acceso a esta memoria es de unos 14 milisegundos o superior, o sea, del orden de un millón de veces superior al de la RAM.

Se llama procesador del ordenador a la unidad central que realiza los cálculos de la máquina. Es el cerebro de la máquina; todas las demás partes de ella son controladas por él. En este trabajo, a veces, llamaré chip al procesador. Para simplificar el trabajo, sólo trataré de los procesadores Intel 8086, Intel 80286 e Intel 80386, que son, respectivamente, procesadores de 16, 16 y 32 bits, aun sabiendo que hay otros importantes, como el Intel 8088.

Coprocesador de la máquina es un procesador auxiliar de la misma, en el sentido de que este procesador alivia de trabajo al procesador principal en

tareas concretas. Un ejemplo típico son coprocesadores de lectura-escritura de datos; mientras el procesador principal efectúa cálculos, los coprocesadores de lectura-escritura escriben los resultados de otros procesos ya finalizados o leen datos a utilizar en procesos futuros, a fin de que el procesador principal los tenga preparados cuando los necesite. En los ordenadores personales que funcionan bajo el Sistema Operativo MS-DOS, el coprocesador más conocido es el matemático, del que hay tres versiones: Intel 8087, Intel 80287 e Intel 80387. Estos coprocesadores realizan los cálculos en coma flotante, liberando de ellos al procesador principal.

Palabra es el número de bits que utilizan los registros internos del procesador. Para variables numéricas es clásico utilizar palabras de 4 y 8 bytes (que, a veces, se llaman palabras de precisión normal y de doble precisión, respectivamente). Con palabras de precisión normal no se suelen obtener más de 7 dígitos significativos en un número, mientras que con palabras de precisión doble se obtienen unos 15. La utilización de palabras de doble precisión se hace necesaria en procesos estadísticos en los que intervienen ficheros de datos con un gran número de variables y/o individuos, para evitar que la acumulación de errores de redondeo y otros haga perder exactitud a los resultados. Debe notarse, a este efecto, que la aritmética en coma flotante no es una ciencia exacta. En Knuth (1981) se pueden estudiar las particularidades de la aritmética en coma flotante.

Por software se entiende cualquier programa o conjunto de programas de ordenador, es decir, sentencias que el ordenador comprende y que hacen que éste efectúe una serie de tareas.

Por hardware se entiende el ordenador como conjunto de componentes electrónicos o algunos de estos componentes.

3. NECESIDADES DE MEMORIA Y TIEMPO

A continuación expondré las necesidades de memoria y tiempo para efectuar algunos de los procesos estadísticos más habituales.

Notar antes que, normalmente, a más memoria utilizada menos tiempo empleado, y viceversa. Los requerimientos de memoria y tiempo los haré suponiendo que los datos nunca están en memoria RAM, sino que se leen de memoria periférica tantas veces como sea necesario (como expliqué en la sección 1). En cambio, supondré que las matrices y vectores que se emplean en los algoritmos estadísticos sí están en memoria RAM. Esta es la forma de funcionamiento de casi todos los programas estadísticos bajo MS-DOS.

Para almacenar matrices de correlaciones o de covarianzas se emplean sólo $p \cdot (p + 1)/2$ palabras, pues estas matrices son simétricas y basta con almacenar su parte triangular inferior o superior en un vector.

En general, los aspectos computacionales de los modelos estadísticos más habituales se encuentran en Maindonald (1984) y Kennedy y Gentle (1980).

1) *Tablas de contingencia*

Bajo este epígrafe estoy incluyendo implícitamente las distribuciones de frecuencia marginales.

Está claro que las necesidades de memoria son de tantas palabras como celdas tenga la tabla que se calcula, más alguna memoria adicional para operaciones auxiliares del tipo de definir variables acumuladoras, etc.

En cuanto al tiempo, éste es, si cada variable no tiene demasiadas categorías, proporcional a n y al número de variables que intervienen en la tabla.

Siempre y cuando se disponga de suficiente memoria, se puede reducir el tiempo, incluyendo en un mismo procedimiento tabulador cuantas tablas sean posibles, pues de esta forma con la misma pasada de lectura de los datos se crean más tablas.

2) *Análisis factorial*

Referencias: Cuadras (1981), Harman (1980), Morrison (1976).

Memoria: Se necesitan $p \cdot (p + 1)/2$ palabras para la matriz de correlaciones, p^2 palabras para la matriz factorial, $p \cdot (p + 1)/2$ palabras para la matriz de correlaciones residual, en algunos modelos, $p \cdot (p + 1)/2$ palabras para la inversa de la matriz de correlaciones. Con algunos modelos y, después, con las rotaciones se necesita más memoria.

Tiempo: Sea f el número de factores retenidos. Sea i el número de iteraciones efectuadas en la estimación del modelo factorial y j el número de iteraciones en la rotación de los factores.

El tiempo de cálculo de la matriz de correlaciones es, aproximadamente, proporcional a $n \cdot p^2$.

El tiempo de estimación de las comunalidades *a priori* oscila, según el método empleado, entre ser proporcional a p^2 y a p^3.

El tiempo de extracción de factores es, en el peor de los casos, proporcional, por un lado, a $i \cdot p^3$ y, por otro, a $i \cdot f \cdot p^2$.

El tiempo de rotación es, en el peor de los casos, proporcional a $j \cdot f^2 \cdot p$.

3) *Análisis de componentes principales*

Referencias: Aparicio (1988), Cuadras (1981), Morrison (1976).

Si se efectúa una aproximación a este modelo mediante Análisis Factorial con Extracción de Componentes Principales, me remito al apartado anterior. Sin embargo, es mejor, si el software lo permite, realizar el Análisis de Componentes Principales tal cual (Aparicio, 1988). En este caso, los requerimientos son:

Memoria: $p \cdot (p + 1)/2$ palabras para la matriz de correlaciones o covarianzas, p^2 palabras para la matriz de las componentes principales, p^2 palabras para la matriz de correlaciones entre las variables y las componentes principales, más algún gasto menor de vectores y variables.

Tiempo: El tiempo necesario para estimar la matriz de correlaciones o covarianzas es, aproximadamente, proporcional a $n \cdot p^2$. El tiempo de cálculo de autovalores y autovectores es, aproximadamente, proporcional a p^3.

4) *Modelos lineales*

Referencias: Anderson (1984), Finn (1977), Searle (1971), Seber (1984).

Englobo en este epígrafe el Análisis de la Varianza, el Análisis de la Covarianza, la Regresión Lineal Múltiple y Multivariante y cuantos modelos se reducen a éstos.

Memoria: El principal gasto de memoria en estos modelos consiste en la creación de la matriz de sumas de cuadrados y de sumas de productos de las variables que intervienen, con el agravante de que las variables cualitativas incrementan la dimensión de esta matriz en el mismo número que categorías tienen. Más adelante, en la sección 4 se ve un ejemplo del gasto de memoria en un modelo lineal.

No obstante, algunos modelos concretos, como el Análisis de la Varianza con igual número de observaciones en cada celda y otros con diseños sencillos, se pueden resolver con menos necesidades de memoria. A este efecto, varios fabricantes de software proporcionan procedimientos concretos para los diseños sencillos que requieren muchos menos recursos que un modelo lineal en general.

Tiempo: Sea m la dimensión del problema (suma del número de variables cuantitativas, categorías de cualitativas e interacciones en el peor de los casos).

El tiempo necesario para calcular la matriz de sumas de cuadrados y productos de los datos es, en el peor de los diseños, proporcional a $n \cdot m^2$.

El tiempo necesario para resolver las ecuaciones normales (cálculo de inversa generalizada) es proporcional a m^3.

A esto hay que añadir el tiempo de contraste de hipótesis, que depende del tipo de función estimable que se desee utilizar.

5) *Análisis discriminante*

Referencias: Anderson (1984), Escudero (1977), Kshirsagar (1972), Mardia *et al.* (1979).

El Análisis Discriminante o, más en general, el Reconocimiento de Patrones no son uno o unos pocos modelos aislados, sino un gran número de técnicas de muy diverso tipo que persiguen la clasificación de objetos según una norma o patrón dados. No se puede, por tanto, dar unos consumos de memoria y tiempo generales. Lo haré sólo para el modelo más usual de Análisis Discriminante mediante funciones discriminantes lineales, que supone igualdad entre las matrices de covarianzas de cada grupo. Sobre reconocimiento de patrones en general puede consultarse Escudero (1977).

Memoria: Sea g el número de grupos en que están divididos los n individuos ya clasificados.

Se necesitan $p \cdot g$ palabras para los coeficientes de las g funciones discriminantes lineales, $p \cdot g$ palabras para los vectores de medias de cada grupo, $p \cdot (p + 1)/2$ palabras para la matriz de covarianzas y alguna memoria más para vectores y variables adicionales.

6) *Análisis de conglomerados*

Referencias: Anderberg (1973), Cuadras (1981), Sánchez (1978).

Al igual que en Análisis Discriminante, el tema es tan extenso que sólo podré hacer referencia a dos de los grupos de algoritmos más utilizados (que suelen ser los implementados en casi todos los programas estadísticos); estos dos grupos son los algoritmos aglomerativos y los algoritmos de tipo centroide.

Algoritmos aglomerativos.

Memoria: Si se almacena la matriz de similaridades entre individuos, las necesidades de memoria de esta matriz son de $n \cdot (n + 1)/2$ palabras, lo que hace el problema intratable para valores elevados de n. Si, en cambio, se mantienen los datos mismos en memoria central, el problema también se hace intratable, con no muchos casos, bajo MS-DOS; una tercera alternativa

es almacenar en disco duro la matriz de similaridades ordenada (Anderberg, 1973). Los algoritmos de clasificación no son excesivamente costosos computacionalmente en comparación con los algoritmos de análisis de conglomerados mediante algoritmos aglomerativos. Una buena descripción de algoritmos de clasificación se puede encontrar en Knuth (1973). En general, como bajo MS-DOS no hay mucha memoria disponible, se deberán emplear estrategias de crear ficheros de trabajo donde almacenar cosas que los algoritmos aglomerativos suponen que estarán en memoria central; con ello, el problema cabe en memoria, pero a costa de unos tiempos de proceso muy superiores a los normales.

Tiempo: El tiempo depende del tipo de enlace entre clusters que se utilice; es proporcional, en los mejores casos, a n^2 y a $n \cdot ln(n)$ y, en los peores casos, a n^3. Vemos, pues, que no se puede aplicar un modelo aglomerativo a un problema con un número elevado de observaciones.

Algoritmos de tipo centroide.

Supondremos algoritmos sencillos, en los que el número de conglomerados se establece de antemano y no cambia a lo largo de la ejecución del procedimiento. Sea c el número de conglomerados especificado. Sea i el número de iteraciones del algoritmo.

Memoria: $c \cdot p$ palabras para almacenar el centroide de cada conglomerado, más gastos menores de vectores de estadísticas para cada conglomerado, etcétera.

Tiempo: Con estos algoritmos, el tiempo es proporcional a $n \cdot p \cdot i \cdot c$, una vez calculadas las semillas iniciales. Para calcular éstas se debe dar una pasada a los datos; el tiempo empleado es proporcional a $n \cdot p \cdot c + p \cdot c^2$.

Existen otros muchos modelos estadísticos, pero con los citados, que son los más usuales, es suficiente para comprender los problemas que se presentan al realizar procesos estadísticos. Este será el objeto de la siguiente sección.

4. LOS LIMITES BAJO MS-DOS

El Sistema Operativo MS-DOS de Microsoft, y los demás Sistemas Operativos DOS, creados por los distintos fabricantes a partir del kit que les es suministrado por Microsoft, es un Sistema Operativo monousuario y monotarea que fue creado para ser utilizado por el procesador Intel 8086, sin utilizar todas las posibilidades de este chip, sobre todo en cuanto se refiere a direccionamiento de memoria, pues sólo es capaz de direccionar 640 K-bytes.

Para mantener la compatibilidad con este chip y, por consiguiente, con los IBM PC, IBM XT y compatibles, las sucesivas versiones del DOS han seguido siendo sistemas operativos incapaces de direccionar más de 640 K-bytes, aunque capaces de trabajar con los chips Intel 80286 e Intel 80386. Sin embargo, esta capacidad se basa en la facultad que tienen estos chips de emular (imitar) al Intel 8086.

Esta es la mayor grandeza y, a la vez, la mayor miseria del DOS. Grandeza en cuanto a que se ha mantenido la compatibilidad de máquinas muy diferentes y se ha creado una verdadera jungla de programas de aplicación que funcionan bajo DOS. Miseria en el sentido de que se ha imposibilitado el desarrollo bajo DOS de aplicaciones pensadas para utilizar toda la potencia de chips de 16 y 32 bits, con las grandes ventajas que esto hubiera conllevado.

La más seria limitación del DOS consiste, como ya hemos dicho, en que sólo es capaz de direccionar 640 K-bytes de memoria (un K-byte son 1.024 bytes, o bien 8.192 bits). En este exiguo espacio deben convivir el Sistema Operativo DOS, cuya parte residente ocupa unos 60 K-bytes de memoria, el programa que realiza las estadísticas y las matrices, vectores, etc., que este programa utiliza (véanse los requerimientos de memoria expuestos en la sección anterior).

Un paquete de programas estadístico tipo utiliza, bajo MS-DOS, unos 380 K-bytes por el solo hecho de funcionar, más la memoria que utiliza el procedimiento que queramos utilizar, más las matrices, vectores, etc., a que acabamos de hacer referencia. Como vemos, no hay espacio para hacer grandes cosas.

Un ejemplo, un Análisis de Componentes Principales con 100 variables: El Sistema Operativo emplea unos 60 K-bytes, un programa estadístico normal, unos 380 K-bytes, la matriz de correlaciones tiene dimensión 100, con lo que si trabajamos con palabras de 8 bytes (necesario para mantener la precisión en un problema de dimensión elevada como éste), nos ocupa 5.050 palabras, o sea, 40.400 bytes o unos 39 K-bytes. La matriz de correlaciones entre las componentes principales nos ocupa otros 39 K-bytes. Finalmente, la matriz de las 100 componentes principales nos ocupa 10.000 palabras o 80.000 bytes, es decir, unos 78 K-bytes. Todo esto suma casi 600 K-bytes, con lo cual estamos muy cerca del límite de 640 del DOS. Teniendo en cuenta que existen otros vectores y variables que consumen alguna memoria más, vemos que no seremos capaces de realizar un Análisis de Componentes Principales con muchas más de 100 variables por falta de memoria bajo DOS.

Pero, a su vez, esta escasez tan notable de memoria fuerza a realizar la tarea con mayor lentitud. En efecto, en un Sistema Operativo que aproveche las posibilidades de un chip de 16 ó 32 bits, se pueden direccionar 16 Megabytes o 4 Gigabytes de memoria, respectivamente (un Megabyte son 1.000 K-bytes y un Gigabyte son 1.000 Megabytes). Suponiendo que en la

máquina dispongamos de, al menos, 6 Megabytes de memoria RAM, aun teniendo en cuenta los que ocupe el Sistema Operativo y el Programa Estadístico y las matrices, vectores, etc., que utilicemos, nos quedarán varios Megabytes libres de memoria. Pues bien, los programas estadísticos que funcionan bajo estos sistemas operativos aprovechan esta memoria libre para, cada vez que tienen que dar una pasada de lectura a los datos, no hacerlo tomando sólo uno o unos pocos individuos de cada vez, sino tomar muchos (p. ej., 100, 500 ó 1.000). Esto hace que se divida por 100 o más el número de accesos que se deben hacer a disco duro y que, en su lugar, se acceda a memoria RAM. Resulta que el acceso a memoria RAM es del orden de un millón de veces más rápido que el acceso a disco duro, como dijimos en la sección 2.

Esto da idea de por qué un programa estadístico que funciona bajo un sistema operativo adecuado a un chip de 16 ó 32 bits es mucho más rápido que otro que funciona bajo un Sistema Operativo como el MS-DOS.

Por otra parte, el chip Intel 80386 está todavía mucho más infrautilizado bajo DOS. Aparte del direccionamiento de más memoria, existen, al menos, tres factores que harían más rápido a este chip bajo un sistema operativo de 32 bits; cada uno de estos factores multiplica la velocidad del chip aproximadamente por 2. Los tres factores son:

— Utilización eficiente de los registros de 32 bits del 80386 para almacenar objetos de 32 bits.
— Utilización de aritmética de 32 bits del chip, en lugar de llamadas a funciones de biblioteca.
— El doble de rendimiento en la utilización efectiva del bus de datos del sistema.

Veremos ahora cuáles son los límites prácticos de los análisis.

Dejaré a un lado el Análisis de Conglomerados mediante algoritmos aglomerativos, pues es el único de los modelos vistos que incumple lo que viene a continuación. En cuanto a este modelo, se debe evitar utilizarlo para problemas de muchos individuos; en Anderberg (1973) se encuentran estrategias para conseguirlo. Debe notarse que éste es el único modelo de los vistos en que el tiempo de ejecución es superior a una función lineal del número de casos. Aun así, para la mayoría de tipos de enlace entre conglomerados, esta función es polinómica de orden bajo. Por ejemplo, en Anderberg (1973, pp. 150-151) se comprueba cómo el enlace sencillo entre conglomerados es equivalente al problema del Arbol Minimal de Expansión (Minimal Spanning Tree, o MST) de un grafo. En Garey y Johnson (1979, pp. 130-131) se clasifica este problema como de resolución en tiempo polinómico de orden bajo del número de vértices del grafo (o sea, de individuos del análisis de con-

glomerados). Pero, por ser en procesos estadísticos el número de individuos normalmente elevado, no resultan tratables muchos problemas.

Tiempo

En el resto de modelos que hemos visto en la sección anterior, su tiempo de ejecución es sólo lineal en el número de individuos y polinómico de orden tres o inferior en el número de variables, con lo cual, habida cuenta de que el número de variables suele ser muy inferior al de individuos, estos modelos serán tratables. No tenemos algoritmos con tiempo de ejecución exponencial, ni siquiera algoritmos de la clase NP. La definición de esta clase de algoritmos y otras propiedades se encuentra en Garey y Johson (1979). Es cierto que algunos algoritmos que se aplican a modelos estadísticos no tienen demostrada la convergencia, p. ej., el algoritmo iterativo para estimar la matriz factorial por el método del factor principal (Cuadras, 1981), pero en la práctica estos algoritmos suelen converger con mucha rapidez y, en caso de no convergencia, ésta se puede suponer a partir de un número razonable de iteraciones. Como, por otro lado, los problemas con matrices de dimensiones muy superiores a 100 no son tratables, parece claro que el único problema práctico que encontraremos en cuanto a tiempos consiste en la realización de un gran número de análisis, pues uno o unos pocos análisis, por complicados que sean, no nos supondrán un dispendio excesivo de tiempo. Esta realización masiva de análisis se presenta, por ejemplo, cuando se efectúa un plan de tabulación de una encuesta. En este caso, no es infrecuente que se soliciten más de 1.000 tablas, cada una de las cuales requiere una pasada de lectura a todos los datos. Si, además, hay muchos individuos, podemos necesitar que un ordenador equipado con un chip Intel 80386 y con coprocesador Intel 80387 necesite varios días para terminar la tarea. Estos tiempos son claramente inaceptables en la mayoría de las situaciones.

Memoria

Ya vimos en un ejemplo de la sección 4 que no podremos realizar un Análisis de Componentes Principales con muchas más de 100 variables por falta de memoria. Tampoco será posible realizar un Análisis Factorial con muchas más de 100 variables.

La situación es mucho más dramática en el caso de modelos lineales, pues en estos modelos se necesita hallar la matriz inversa (o inversa generalizada) de una matriz simétrica, donde la dimensión de esta matriz simétrica es de una fila y columna por cada variable de tipo continuo y, para cada variable de tipo discreto, una fila y columna más por cada una de sus cate-

gorías y por cada una de las categorías que se cruce con otra(s) categoría(s) de otra(s) variable(s) (si especificamos efectos cruzados). Pongamos un ejemplo: sea un análisis multivariante de la covarianza en el cual hay tres variables dependientes (y_1, y_2 e y_3), dos variables independientes cuantitativas (x_1 y x_2) y tres variables independientes cualitativas (z_1, z_2 y z_3). Supongamos que z_1 tiene 8 categorías, z_2 tiene 7 categorías y z_3 tiene 5 categorías. La matriz de la cual deberemos hallar la inversa generalizada si el modelo estudia todos los efectos cruzados posibles es de dimensión 437 (en efecto, 1 por el término independiente, 3 por las variables dependientes, 2 por las variables independientes cuantitativas, 8 por z_1, 7 por z_2, 5 por z_3, 56 por el efecto cruzado entre z_1 y z_2, 40 y 35 por los otros efectos cruzados entre pares de variables y 280 por los efectos cruzados entre las tres variables). Es decir, esta matriz tiene $437 \cdot (437 + 1)/2 = 95.703$ elementos; suponiendo longitud de palabra 8 (necesaria con esta dimensión), ocupa $95.703 \times 8 = 765.624$ bytes, o bien unos 747 K-bytes, cantidad ella sola superior a los 640 K-bytes disponibles para todo el MS-DOS. Así, pues, un problema de 8 variables, que parecería, en principio, modesto, resulta ser intratable. Es cierto que no hay por qué especificar todos los efectos cruzados, sobre todo el de la interacción entre las tres variables cualitativas a la vez, pero este efecto cruzado puede ser importante y, sobre todo, si no lo incluimos por falta de memoria, nunca tendremos la seguridad de que no fuera importante. Por consiguiente, este análisis no se va a poder efectuar con garantías.

Como resumen de esta sección, vemos que los límites de tiempo se sobrepasan, sobre todo, si se desea realizar una explotación masiva (o sea, un gran número de análisis) y los de memoria se sobrepasan al realizar un análisis de dimensión elevada.

5. ESTRATEGIAS PARA SOBREPASAR LOS LIMITES DEL MS-DOS

Para un investigador que desea realizar análisis serios con datos voluminosos, el panorama visto en la sección anterior no resulta alentador en absoluto. Puede llegar a pensar que, en un futuro próximo, no podrá realizar análisis estadísticos que sobrepasen los límites que hemos establecido.

Esto no es así. En esta sección veremos varias estrategias para hacerlo. Algunas de ellas permiten mejorar ligeramente los límites vistos y sirven, tan sólo, para problemas cercanos a esos límites, mientras que otras son verdaderas soluciones que sitúan los límites en los de miniordenadores que cuestan diez o hasta cien veces más que un ordenador personal basado en el procesador Intel 80386.

5.1. *Soluciones que mejoran ligeramente los límites*

5.1.1. *Sistemas aceleradores.*

Entendemos por sistemas aceleradores a dispositivos basados en software, hardware o una mezcla de ambos, que mejoran la velocidad de proceso del ordenador. Aparte del evidente uso de coprocesador matemático que, al liberar al procesador principal de las tediosas operaciones en coma flotante, da mayor velocidad a los procesos estadísticos, tenemos otros tipos de sistemas aceleradores.

Uno de ellos es el uso de tarjetas aceleradoras, que incorporan un procesador superior al que la máquinas traía consigo. En el terreno que nos ocupa, el límite de esta solución está en sustituir un chip Intel 8086 ó 80286 por una tarjeta aceleradora con el chip Intel 80386; el sistema resultante será siempre más lento que un ordenador equipado de fábrica con el chip Intel 80386; por tanto, ésta es sólo solución para personas que tengan un ordenador de la gama media-baja y no permite sobrepasar los límites expuestos en la sección 4.

Otra forma de acelerar los procesos es la utilización de sistemas de memoria caché. Un sistema de memoria caché es un sistema en el cual se mejoran los tiempos de acceso a memoria no central (disco duro), mediante la utilización de parte de la memoria RAM o de otra memoria de velocidad semejante a la RAM para guardar en ella los sectores más y más recientemente utilizados del disco duro. En efecto, el sistema de memoria caché realiza básicamente la función de interceptar los accesos a disco duro para ver si el sector de disco duro del que se desea leer se encuentra en su memoria (la del sistema caché); si es así, se evita el acceso a disco duro y, en caso contrario, accede a disco duro para leer el sector, pero se guarda una copia en su memoria, pasando a borrar (si es preciso) el sector que tenga las estadísticas peores de utilización y a actualizar estas estadísticas para los demás sectores que están en su memoria. De esta forma se evitan muchos accesos a disco duro, pues la probabibilidad de que, al ser pedido un sector concreto, éste se encuentre en la memoria caché es elevada. Hay muchos refinamientos de que gozan algunos sistemas de memoria caché; por ejemplo, los hay predictivos que, sobre todo en accesos secuenciales, son capaces de prever qué sectores del disco se les reclamará más adelante y los leen de antemano.

Algunos sistemas de memoria caché tienen el inconveniente de trabajar dentro de los 640 K-bytes de RAM del MS-DOS, con lo cual limitan la ya muy exigua memoria disponible; otros, mejores, trabajan sobre la memoria extendida o expandida de la máquina, y los mejores incluyen una placa propia de memoria RAM, junto con un coprocesador.

En cuanto a su rendimiento, éste es modesto en aplicaciones estadísticas. En otro tipo de aplicaciones, como diseño asistido por ordenador, es muy

elevado, pues en estas aplicaciones se accede casi siempre a los mismos sectores del disco, pero en aplicaciones estadísticas la experiencia me demuestra que no se mejora gran cosa.

Personalmente he probado dos sistemas de memoria caché. El primero de ellos estaba dotado de una placa de 128 K-bytes de memoria RAM junto con un coprocesador. Hacía que un disco duro con tiempo de acceso medio de 48 milisegundos devolviera un índice de 1 milisegundo (según un programa de los más usados para chequeo del tiempo de acceso a disco duro). Sin embargo, al aplicarlo a una tarea estadística seria fracasó enormemente y se produjo una modesta mejora en el tiempo de ejecución del programa que, en ningún caso, superaba el 10 por 100.

El segundo sistema de memoria caché era más sofisticado (y caro) que éste. Disponía de 2 Megabytes de RAM, junto con un potente coprocesador Motorola 68000. Devolvió un índice de velocidad del disco duro de medio milisegundo, pero, de nuevo, al aplicarlo a la misma tarea estadística que antes, sólo redujo el tiempo de ejecución en un 20 por 100 (sobre el tiempo original, sin ningún sistema de memoria caché).

Parece, pues, claro que estos sistemas tienen una eficiencia reducida y que no permiten sobrepasar los límites de velocidad del MS-DOS en forma sustancial.

5.1.2. *Aumento de la memoria disponible.*

Como ha quedado claro desde el principio, el Sistema Operativo MS-DOS no direcciona más de 640 K-bytes de memoria RAM. Esto es estrictamente cierto; no obstante, la acuciante necesidad de memoria ha hecho a los fabricantes de software agudizar el ingenio y conseguir que, de alguna forma, determinados programas puedan direccionar más memoria. Me estoy refiriendo, en concreto, a la memoria expandida, según las especificaciones LIM (Lotus, Intel, Microsoft).

Estos fabricantes han establecido un estándar de hardware y software que hace que se pueda acceder hasta 4 Megabytes (quizá en el momento de publicarse este trabajo ya sea más), de memoria expandida. Pero esta memoria expandida es memoria paginada y, por tanto, mucho más lenta que la RAM.

Algunos fabricantes de programas estadísticos han hecho ya que sus programas soporten esta memoria expandida. Esto es una solución poco eficiente al problema de la escasez de memoria, pues la velocidad de ejecución de los programas utilizando esta memoria es exasperante. Como mucho, se puede pensar en esta solución para ser utilizada esporádicamente, teniendo presente que se puede precisar el dejar el ordenador funcionando toda una noche para conseguir finalizar una tarea no demasiado complicada. En ningún caso esta

solución es válida para analistas cuyos problemas sobrepasen con frecuencia los límites establecidos en la sección 4.

5.2. *Soluciones que alcanzan límites de miniordenadores*

Veamos ahora cómo sobrepasar de verdad los límites de la sección 4.

La primera y evidente solución es eliminar la raíz del problema. La causa de todos los males es, como hemos visto, la utilización de un Sistema Operativo que sólo direcciona 640 K-bytes de memoria. Entonces, ¿por qué no utilizar un sistema operativo pensado para un chip de 16 ó 32 bits? La respuesta a la enorme utilización del MS-DOS ya la vimos antes: su enorme éxito lo ha hecho prisionero de sí mismo. Sin embargo, una persona que desee sobrepasar verdaderamente los límites de la sección 4 no podrá evitar el tener que utilizar otro sistema operativo. La cuestión es conseguir esto sin desviarse mucho del MS-DOS, sin tener que comprar una máquina distinta de la que se disponga y sin tener que aprender un sistema operativo muy distinto al DOS. En esta sección veremos cómo conseguir todo esto.

Mencionar antes que utilizando un ordenador personal tipo Macintosh no hay problema; su sistema operativo es de 32 bits y el software estadístico que funciona bajo él tiene prestaciones de tipo miniordenador, que son las que deseamos alcanzar. Pero esto incumple claramente lo que acabamos de decir, pues debemos cambiar de ordenador y de sistema operativo.

Sin cambiar de ordenador (si se dispone de uno con el chip Intel 80286 o el Intel 80386), pero sí cambiando de sistema operativo, se resuelve el problema utilizando Xenix 286 o Xenix 386, respectivamente, con el punto en contra de que no existe gran cosa en software estadístico bajo estos sistemas operativos.

Más prometedor es el nuevo sistema operativo que se supone que será el estándar en los años noventa para ordenadores personales IBM y compatibles, el OS/2. Es éste un sistema operativo de 16 bits, pensado para el 80286 y capaz de direccionar hasta 16 MB de memoria RAM. Con él, en teoría, quedan resueltos los problemas que nos ocupan. Pero hoy día no existe prácticamente ningún software estadístico para el OS/2. No obstante, el OS/2 es tan sencillo de usar como el DOS y sí existen compiladores de Fortran y C para este sistema operativo. El investigador que realice sus procesos estadísticos a base de programación, utilizando bibliotecas de subrutinas matemático-estadísticas y que disponga de los programas-fuente de estas subrutinas, tiene resuelto su problema: le basta con compilar las subrutinas bajo uno de estos compiladores del OS/2 y ejecutarlas. Si las subrutinas son de buena calidad, no deben dar problema alguno al hacer esto (a lo sumo, cambio de algunas constantes). Pero el resto de analistas que no desean hacer esto y que no están dispuestos a programar a base de subrutinas, no tienen

hoy día solución con el OS/2. Esta forma de trabajo, a base de programar en lenguaje de alto nivel y llamar a subrutinas, fue muy utilizada en los años sesenta; hoy día está cayendo en desuso debido a la mayor comodidad de los paquetes de programas, pero en modo alguno debe ser despreciada; es más, para aplicaciones serias es, en general, mucho más potente que el uso de paquetes de programas. Por otra parte, parece ser que uno de los paquetes de programas estadísticos más difundidos va a dejar de ser soportado bajo MS-DOS, debido precisamente a problemas de escasez de memoria RAM, y va a pasar a ser soportado en OS/2. Esto, no obstante, no ocurrirá, previsiblemente, hasta dentro de, al menos, un año.

Incluso, sin necesidad de utilizar el OS/2, existen compiladores de Fortran y C que funcionan bajo DOS «mejorado», con lo cual, utilizando uno de estos compiladores (con un chip 80286 ó 80386), también se resuelve el problema para los analistas dispuestos a manejar subrutinas. El DOS «mejorado» más conocido es el llamado *Phar Lap Extended DOS,* que es DOS desde el punto de vista del usuario, pero aprovecha el chip Intel 80386 como un sistema operativo de 32 bits (o sea, direcciona hasta 4 Gigabytes de memoria RAM). Incluso una empresa especializada de EE. UU. vende un paquete estadístico de los más difundidos preparado para funcionar bajo esta extensión del DOS. Esta sí es una verdadera solución para el analista que disponga de un chip 80386. El coste del *Phar Lap Extended DOS* y del paquete preparado para él es, en EE. UU., aproximadamente, de 200.000 pesetas, lo cual quiere decir que, importándolo, costaría unas 300.000 pesetas, precio que considero más que razonable por alcanzar prestaciones de miniordenador.

Existen más herramientas finales que funcionan bajo *Phar Lap Extended DOS;* por ejemplo, varios de los programas matemáticos que han venido proliferando últimamente lo hacen. Estos programas suelen tener un lenguaje matricial de muy alto nivel, junto con funciones estadísticas y matemáticas, lo cual hace que sea fácil programar en ellos los modelos estadísticos que se desee. Incluso algunos de estos programas admiten llamadas a subrutinas en Fortran y C, con lo cual, si se dispone de los programas fuente de subrutinas estadísticas y se compilan adecuadamente, no es necesario programar los modelos estadísticos. Un resumen de características de varios de estos programas se puede encontrar en Simon (1989).

En resumen, cualquiera de las soluciones que hemos visto en la sección 5.2 son verdaderas soluciones que permiten que una máquina dotada con el procesador Intel 80386 alcance y, a veces, sobrepase las prestaciones de un miniordenador. Desde mi punto de vista, el analista que de verdad necesite realizar procesos estadísticos complejos y no tenga acceso a un gran ordenador ni a un miniordenador deberá pasar por una de estas soluciones.

6. COMENTARIOS FINALES DEL AUTOR

Este trabajo se terminó de escribir en junio de 1989. Debido a la vertiginosa evolución en el terreno informático, es posible que se hayan producido, entre esa fecha y la de llegada del trabajo al lector, algunos cambios en lo expuesto.

Por otra parte, agradeceré cualquier comunicación sobre el estado del arte en el tema objeto de este trabajo.

BIBLIOGRAFIA

ANDERBERG, M. R. (1973): *Cluster Analysis for Aplications,* Academic Press.

ANDERSON, T. W. (1984): *Multivariate Statistical Analysis,* 2.ª ed., John Wiley.

APARICIO, F. (1988): «La difícil realización de un Análisis de Componentes Principales mediante los programas estadísticos más difundidos en el mercado», *Estadística Española,* vol. 30, núm. 117, pp. 99-114.

CUADRAS, C. M. (1981): *Métodos de Análisis Multivariante,* Editorial Universitaria de Barcelona, Barcelona.

ESCUDERO, L. F. (1977): *Reconocimiento de Patrones,* Paraninfo, Madrid.

FINN, J. D. (1977): «Multivariate Analysis of Variance and Covariance», publicado en el libro *Statistical Methods for Digital Computers,* vol. III, editado por Enslein *et al.,* John Wiley.

GAREY, R., y JOHNSON, D. S. (1979): *Computers and Intractability. A Guide to the Theory of NP-Completeness,* W. H. Freeman, Nueva York.

HARMAN, H. H. (1980): *Análisis Factorial Moderno,* Ed. Saltés, Madrid.

KENNEDY, W. J., y GENTLE, J. E. (1980): *Statistical Computing,* Marcel Dekker, Nueva York.

KNUTH, D. (1973): *The Art of Computer Programming,* vol. 3: *Sorting and Searching,* Addison Wesley.

— (1981): *The Art. of Computer Programming,* vol. 2: *Seminumerical Algorithms.*

KSHIRSAGAR, A. M. (1972): *Multivariate Analysis,* Marcel Dekker, Nueva York.

MAINDONALD, J. H. (1984): *Statistical Computation,* John Wiley.

MARDIA, K. V.; KENT, J. T., y BIBBY, J. M. (1979): *Multivariate Analysis,* Academic Press.

MORRISON, D. F. (1976): *Multivariate Statistical Methods,* 2.ª ed., McGraw-Hill.

SÁNCHEZ, M. (1978): *Modelos Estadísticos Aplicados a Tratamiento de Datos,* Centro de Cálculo de la Universidad Complutense, Madrid.

SEARLE, S. R. (1971): *Linear Models,* John Wiley.

SEBER, G. A. F. (1984): *Multivariate Observations,* John Wiley.

SIMON, B. (1989): «Better Tools for Higher Math: When Numbers Count», *PC-Magazine,* vol. 8, núm. 5, pp. 289-310.

TEXTOS CLASICOS