Un Método para la Fragmentación Vertical de Bases de Datos y su Variante como Evaluador de Particiones

Yurisbel Vega Ortiz[1], Abel Rodríguez Morffi[2]

[1] Universidad de las Ciencias Informáticas (UCI), Carretera a San Antonio de los Baños Km 2 ½ Torrens, Ciudad de La Habana, Cuba
[2] Departamento de Ciencia de la Computación, Universidad Central ¨Marta Abreu¨ de Las Villas, Carretera a Camajuaní km. 5.5, Santa Clara, Cuba
yurisbelv@uci.cu

Resumen. El diseño de bases de datos distribuidas es un problema de optimización que implica la solución de problemáticas como la fragmentación de los datos y su ubicación. Típicamente, los criterios que determinan si la fragmentación y la asignación son óptimas se establecen de manera independiente. Primero se busca la "mejor" fragmentación y luego la "mejor" ubicación de los fragmentos obtenidos. La fragmentación vertical es más complicada que la partición horizontal, debido al incremento del número de posibles alternativas. En este trabajo se presenta un nuevo método para la fragmentación vertical, que se basa fundamentalmente en la Matriz de Atracción entre Atributos, suplantando la conocida Matriz de Afinidad entre Atributos. Se utiliza como heurística el enfoque de agrupamientos jerárquicos y una regla de decisión basada en la homogeneidad interna y la heterogeneidad externa de los grupos obtenidos. También se presenta una variante para que pueda ser usado como evaluador de particiones.

Palabras claves: Bases de datos distribuidas, evaluador de particiones, fragmentación vertical, medida de afinidad, método jerárquico, regla de decisión.

1 Introducción

El interés por el desarrollo de bases de datos distribuidas relacionales ha aumentado en la medida que las organizaciones y empresas han crecido y se han expandido geográficamente; de la mano del vertiginoso progreso en el uso de redes de computadores. El diseño de bases de datos distribuidas es un problema de optimización que implica la solución de problemáticas como la fragmentación de los datos, su ubicación y replicación.

La fragmentación es el proceso mediante el cual una relación global es descompuesta en fragmentos horizontales y/o verticales. Un fragmento vertical atiende al agrupamiento de datos en función de atributos o conjuntos de ellos, mientras que la fragmentación horizontal atiende a dicho agrupamiento en función de

tuplas o conjuntos de tuplas. Típicamente, los criterios que determinan si la fragmentación y la asignación son óptimas se establecen de manera independiente, en dos pasos. En el primero se busca la "mejor" fragmentación y, en el segundo, se busca la "mejor" ubicación de los fragmentos obtenidos en el paso anterior [1].

Comparando las formas de fragmentación, la partición vertical es más complicada que la partición horizontal, debido al incremento del número de posibles alternativas [2]. Como se señala en [3], un objeto con *m* atributos puede ser particionado de *B(m)* diferentes formas, donde *B(m)* es el *m*-ésimo número de Bell, para *m* suficientemente grandes, *B(m)* se aproxima a m^m; para m=15 este es $\approx 10^9$, para m=30 este es $\approx 10^{23}$.

Por lo tanto, es importante contar con una estrategia que reduzca de manera eficiente el número de cálculos. Aunque diferentes, dos enfoques secuenciales que conducen a agrupamientos jerárquicos parecen haber adquirido un interés particular entre los taxónomos. Una de las estrategias es el algoritmo propuesto por Ward (1963). Su idea es aglomerar los puntos o los grupos resultantes, reduciendo su número en uno en cada etapa de un procedimiento de fusión secuencial, hasta que todos los puntos estén en un único clúster. Un algoritmo contrario ha sido propuesto por Edwards y Cavalli-Sforza (1965). La esencia de su método es la partición consecutiva de un conjunto de puntos en dos subconjuntos: primero un conjunto inicial es dividido en dos grupos, cada uno de ellos se subdivide en dos grupos más pequeños por separado, y así sucesivamente, hasta que se alcancen los puntos individuales [4].

En la presente investigación se utiliza el enfoque propuesto por Ward, aunque el método propuesto se puede implementar independientemente con cualquiera de los dos enfoques.

En la mayoría de las situaciones de agrupación de la vida real, un investigador aplicado se enfrenta con el dilema de seleccionar el número de grupos o particiones en la solución final. Los procedimientos no jerárquicos suelen exigir que el usuario especifique este parámetro antes de que la agrupación se lleve a cabo y los métodos jerárquicos habitualmente producen una serie de soluciones que van desde *n* grupos hasta una solución con un único clúster presente (asumir *n* objetos en el conjunto de datos). Cuando aplicamos para los resultados métodos de agrupación jerárquicos, las técnicas para la determinación del número de grupos en un conjunto de datos son muchas veces referidas como reglas de decisión [5].

El dilema de una regla de decisión, que en inglés se denomina comúnmente "stopping rule", es clave ya que en su solución descansa la decisión correcta sobre nuestra estructura grupal [6].

Algunos procedimientos propuestos presentan problemas, ya que sugieren reglas de decisión no automáticas por lo que no eliminan el tema de la subjetividad humana, otros son métodos gráficos que requieren el juicio del investigador y en otros casos son índices con parámetros de control que no han sido totalmente definidos o desarrollados. Otra deficiencia presente en algunas reglas de decisión es su incapacidad de operar cuando el agrupamiento óptimo resulta ser k=1 (los *n* elementos deben pertenecer a un único clúster) o k=*n* (cada uno de los *n* elementos debe permanecer en un clúster individualmente). Y por último, otra dificultad que presentan métodos anteriormente propuestos, es el nivel de cómputo asociado al procedimiento.

Por tanto, el objetivo del presente trabajo es proponer un método sencillo, que incluya una regla de decisión, para la partición vertical de bases de datos; que garantice un índice preciso, que elimine la subjetividad humana del proceso, que no sea susceptible a los casos extremos de k=1 y k=n, anteriormente explicados, con un nivel de cómputo aceptable, que iguale y/o mejore los resultados obtenidos mediante el métodos anteriores. También se presenta una variante de dicho método para que pueda ser usado como evaluador de particiones.

2 Metodología

Como la inmensa mayoría de los algoritmos previos para la partición vertical de bases de datos se usará la Matriz de Uso de Atributos (MUA) como entrada. Esta matriz relaciona las transacciones con los atributos de la relación y contendrá un 1 en una de sus celdas si el atributo A_i es utilizado en la Transacción T_j o un cero en caso contrario. Esta matriz cuenta con una última columna reservada para las frecuencias de acceso de cada transacción, es decir, el número de veces que la transacción se solicita en un intervalo de tiempo definido. Se utilizará un ejemplo que considera 8 transacciones y 10 atributos que producen la MUA que se aprecia en la Tabla I, este ejemplo es utilizado en [7][8][9].

Tabla 1. Ejemplo de Matriz de Uso de Atributos de dimensión 10x8.

ref	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**	**A10**	*acc*
T1	1	0	0	0	1	0	1	0	0	0	25
T2	0	1	1	0	0	0	0	1	1	0	50
T3	0	0	0	1	0	1	0	0	0	1	25
T4	0	1	0	0	0	0	1	1	0	0	35
T5	1	1	1	0	1	0	1	1	1	0	25
T6	1	0	0	0	1	0	0	0	0	0	25
T7	0	0	1	0	0	0	0	0	1	0	25
T8	0	0	1	1	0	1	0	0	1	1	15

Hoffer y Severance en [10] proponen el concepto de afinidad entre pares de atributos. Aplicando este concepto a la MUA se obtiene la Matriz de Afinidad entre atributos (MAA), que es lo que se propone en los 2 primeros pasos del Método Navathe como se indica en [11]. Sin embargo, muchos autores han criticado el uso de esta matriz. Sharma Chakravarthy, Jaykumar Muthuraj, Ravi Varadarajan y Shamkant B. Navathe en [8] aseguran que, debido a que solo un par de atributos son involucrados, esta medida no refleja la cercanía o afinidad cuando más de dos atributos son implicados. Aunque en este trabajo se comparte el enfoque de Jun Du, Ken Barker y Reda Alhajj, quienes fundamentan en [7] las limitaciones de la medida de afinidad como una medida de afinidad local y la necesidad de una medida de afinidad global para lograr que todos los valores de la matriz sean comparables entre sí. No obstante, para esta investigación, no se considera consistente la medida de afinidad global descrita en el trabajo de estos autores, por lo que se realizó un análisis de varias de las medidas de afinidad existentes y que son revisadas por el Doctor en

Ciencias Biológicas Alejandro Herrera Moreno en [6]. Se decide que el Índice de Jaccard, una expresión de similitud, es una medida de afinidad global apropiada para el tema de la partición vertical de bases de datos, eliminando de su fórmula el factor que representa las ausencias conjuntas o ceros compartidos, debido a que el hecho de que en una transacción no se usen ninguno de los dos atributos del par analizado, no brinda ninguna información para el caso que ocupa. A continuación la ecuación del Índice de Jaccard que se sugiere:

$$S = a/(a + b + c) . \quad (\mathbf{1})$$

donde:

S: valor de atracción entre el atributo Ai y Aj,

a: suma de las frecuencias de aquellas transacciones que usan tanto el atributo Ai como el Aj,

b: suma de las frecuencias de aquellas transacciones que usan el atributo Ai y no el Aj y

c: suma de las frecuencias de aquellas transacciones que usan el atributo Aj y no el Ai.

Al aplicar la medida de afinidad global seleccionada, se obtiene una nueva matriz, se nombrará Matriz de Atracción entre Atributos (MAA*).

Tabla 2. Ejemplo de Matriz de Atracción entre Atributos de dimensión 10x10.

Sij	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**	**A10**
A1	1	0,156	0,15	0	1	0	0,45	0,156	0,15	0
A2	0,156	1	0,5	0	0,156	0	0,4	1	0,5	0
A3	0,15	0,5	1	0,107	0,15	0,107	0,142	0,5	1	0,107
A4	0	0	0,107	1	0	1	0	0	0,107	1
A5	1	0,156	0,15	0	1	0	0,45	0,156	0,15	0
A6	0	0	0,107	1	0	1	0	0	0,107	1
A7	0,45	0,4	0,142	0	0,45	0	1	0,4	0,142	0
A8	0,156	1	0,5	0	0,156	0	0,4	1	0,5	0
A9	0,15	0,5	1	0,107	0,15	0,107	0,142	0,5	1	0,107
A10	0	0	0,107	1	0	1	0	0	0,107	1

Como se puede apreciar, la matriz es simétrica, los valores quedan normalizados entre cero y uno, donde uno representa el valor máximo de similitud y cero el mínimo, por lo que todos los valores son comparables entre sí. Es obvio que la diagonal principal esté formada por unos, debido a que la atracción de un atributo consigo mismo es máxima.

A partir de este punto se comienza a aplicar el método de agrupamiento jerárquico que propone Ward (1963). Recordemos que este procedimiento parte de que los *k* atributos estén individualmente en un grupo y en cada etapa reduce el número de grupos en 1, mediante fusiones, hasta que todos los atributos se encuentren en un único clúster. Para esta altura, se habrán obtenido *k*-1 posibles agrupaciones. Tras cada fusión deben ser recalculados los valores de "atracción" entre grupos-atributos o pares de atributos que provoca el nuevo grupo recién formado.

Lance y Williams. Boesch (1977) citan ocho estrategias aglomerativas que pueden ser utilizadas con este propósito, entre las que se encuentran: ligamiento simple o vecino más cercano, ligamiento completo o vecino más lejano, promedio simple, promedio de grupos y estrategia flexible. El "promedio simple" es un método que se considera como conservativo del espacio ya que introduce poca distorsión en las afinidades originales, propiedad que la hace una estrategia muy recomendada [6]. Por tal motivo esta es la estrategia seleccionada para el presente método. A continuación, en la Tabla 3, se presenta en resumen los resultados obtenidos al aplicar los procesos antes mencionados al ejemplo desarrollado.

Tabla 3. Resultados de aplicar el método de agrupamiento de Ward con la estrategia aglomerativa "Promedio Simple".

# de fusiones	# de grupos	Grupos	Valores de fusión
0	10	(A1)(A2)(A3)(A4)(A5)(A6)(A7)(A8)(A9)(A10)	-
1	9	(A1, A5)(A2)(A3)(A4)(A6)(A7)(A8)(A9)(A10)	1
2	8	(A1, A5)(A2, A8)(A3) (A4)(A6)(A7)(A9)(A10)	1
3	7	(A1, A5)(A2, A8)(A3, A9)(A4)(A6)(A7)(A10)	1
4	6	(A1, A5)(A2, A8)(A3, A9)(A4, A6)(A7)(A10)	1
5	5	(A1, A5) (A2, A8) (A3, A9) (A4, A6, A10) (A7)	1
6	4	(A1, A5)(A2, A8, A3, A9) (A4, A6, A10) (A7)	0,5
7	3	(A1, A5, A7) (A2, A8, A3, A9) (A4, A6, A10)	0,45
8	2	(A1, A5, A7, A2, A8, A3, A9) (A4, A6, A10)	0,212
9	1	(A1, A5, A7, A2, A8, A3, A9, A4, A6, A10)	0,02675

Nótese que los valores de fusión decrecen a medida que aumenta el número de fusiones debido a que siempre se selecciona el mayor valor de "atracción" para realizar la fusión. El principal problema a enfrentar ahora es escoger cuál de estos agrupamientos parece ser mejor, aquí es donde juega un papel fundamental la "regla de decisión".

Muchos intentos por definir qué es un grupo emplean propiedades como la *cohesión interna* y el *aislamiento externo* lo cual está más cerca de la definición de clasificación que pretende de manera objetiva crear grupos muy homogéneos entre sí y bien diferentes de otros. Esto es lo que nos dicen Hair, Anderson, Tatham y Black en [12] cuando explican que los grupos deben poseer una homogeneidad interna muy alta ("withincluster") y una heterogeneidad externa ("betweencluster") también muy alta [6].

Basado en estos principios en el presente artículo se proponen varias ecuaciones para calcular la Homogeneidad Interna (HI) y la Heterogeneidad Externa (HE). A continuación se muestra la fórmula para calcular la HI de un grupo específico:

$$HI_l = \begin{cases} 1 \; si \; n = 1 \\ \dfrac{\sum A_{ij}}{n(n-1)/_2} \quad en \; otro \; caso \; . \end{cases} \tag{2}$$

donde:

HI_l: valor de la homogeneidad interna del clúster *l*,

n: cantidad de atributos del clúster *l* y

$\sum A_{ij}$: sumatoria de los valores de atracción entre cada par de atributos que pertenecen al clúster *l*, sin tener en cuenta la atracción de cada atributo consigo mismo.

Para calcular la Homogeneidad Interna Total (HIT) de un agrupamiento completo, se usará la siguiente ecuación:

$$HI_T = \sum_{l=1}^{m} HI_l / m \,. \tag{3}$$

donde:

HI_l: valor de la homogeneidad interna de cada clúster y

m: cantidad de grupos del agrupamiento.

De igual forma se define una manera de calcular la heterogeneidad externa de cada grupo:

$$HE_k = \frac{\sum_{i=1}^{n}(1 - Max\ A_{ij})}{n}. \tag{4}$$

donde:

$Max\ A_{ij}$: máximo valor de afinidad o atracción para cada atributo del clúster k con los atributos de otros grupos distintos a k y

n: número de atributos del clúster k.

Para calcular la Heterogeneidad Externa Total (HET) de un agrupamiento completo, se usará la siguiente ecuación:

$$HE_T = \begin{cases} 1\ si\ k = 1 \\ \dfrac{\sum_{z=1}^{k} HE_z}{k} \quad en\ otro\ caso\ . \end{cases} \tag{5}$$

donde:

k: número de grupos del agrupamiento y

HE_z: valor de heterogeneidad externa del clúster *z*.

A continuación se ilustran los resultados del cálculo de la HI_T y HE_T para cada uno de los esquemas de fragmentación obtenidos para el ejemplo que nos ocupa.

Tabla 4. Resultados del cálculo de HIT y de HET.

# de fusiones	**# de grupos**	**Grupos**	**HI_T**	**HE_T**
0	10	(A1)(A2)(A3)(A4)(A5)(A6)(A7)(A8)(A9)(A10)	1	0,055

1	9	(A1, A5)(A2)(A3)(A4)(A6)(A7)(A8)(A9)(A10)	1	0,12
2	8	(A1, A5)(A2, A8)(A3) (A4)(A6)(A7)(A9)(A10)	1	0,2
3	7	(A1, A5)(A2, A8)(A3, A9)(A4)(A6)(A7)(A10)	1	0,3
4	6	(A1, A5)(A2, A8)(A3, A9)(A4, A6)(A7)(A10)	1	0,4244
5	5	(A1, A5) (A2, A8) (A3, A9) (A4, A6, A10) (A7)	1	0,5986
6	4	(A1, A5)(A2, A8, A3, A9) (A4, A6, A10) (A7)	0,9	0,6795
7	3	(A1, A5, A7) (A2, A8, A3, A9) (A4, A6, A10)	0,743	0,7935
8	2	(A1, A5, A7, A2, A8, A3, A9) (A4, A6, A10)	0,695	0,93121
9	1	(A1, A5, A7, A2, A8, A3, A9, A4, A6, A10)	0,263	1

Es obvio que la homogeneidad interna de las agrupaciones sea decreciente. Este es un criterio que parte de un máximo de homogeneidad interna, cuando los atributos individualmente forman un clúster, pues nadie más parecido a un elemento que él mismo. Luego esta medida se debe ir deteriorando con el propio crecimiento de los grupos, pues los valores de fusión van empeorando y la probabilidad de convivir con un atributo no tan parecido, en el mismo grupo, aumenta, hasta hacerse mínima cuando todos los atributos pertenecen a un mismo grupo. Exactamente lo contrario ocurre con el concepto de heterogeneidad externa, que es creciente, pues cuando los grupos son pequeños, lo más probable es que existan en otros grupos atributos muy parecidos y se obtiene un máximo cuando todos los atributos forman parte del mismo grupo, pues sencillamente, no existe otro clúster con quien diferenciarse.

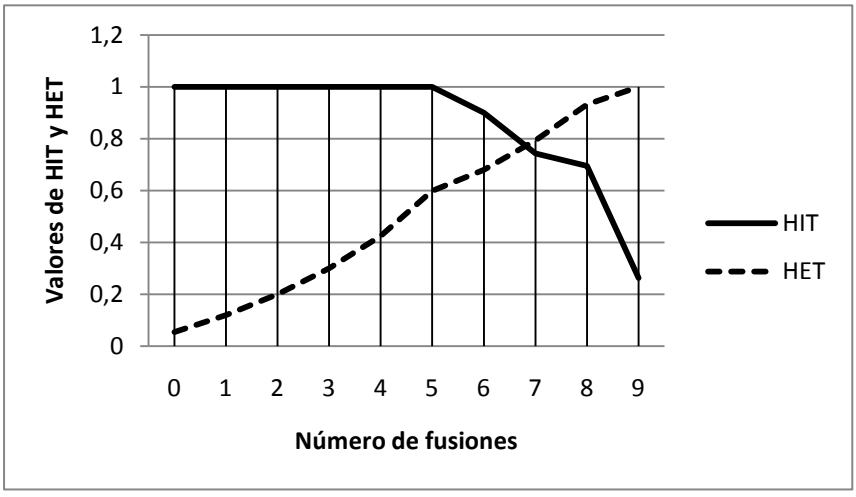

Fig. 1. Funciones HIT y HET y su punto de intersección (*punto de balance*).

Estos elementos llevan a pensar en que estas dos funciones, HI_T y HE_T, en algún momento se cortan. Véase la Fig. 1. Al punto de intersección de ambas funciones se le denominará "punto de balance", ya que en él se alcanza el mejor balance de homogeneidad interna y heterogeneidad externa de los grupos y constituye la base de la regla de decisión que se propone en el presente trabajo. En la práctica, el proceso

de cómputo pararía en el preciso momento en que nos encontremos en un agrupamiento donde el valor de HE_T supere al valor de HI_T, evitando cálculos innecesarios. Entonces la regla de decisión indica que se tomará como mejor agrupamiento aquel definido por la fusión más cercana al punto de balance, lo que matemáticamente es muy sencillo de determinar. En el ejemplo desarrollado la mejor opción, según este nuevo enfoque, resulta ser la agrupación en 3 grupos de la siguiente formas: (A1, A5, A7) (A2, A8, A3, A9) (A4, A6, A10).

2.1 Variante para que el método pueda ser usado como "Evaluador de Particiones"

Sharma Chakravarthy, Jaykumar Muthuraj, Ravi Varadarajan y Shamkant B. Navathe en [8], explican que los resultados de los algoritmos de partición vertical de bases de datos son a veces diferentes, incluso para la misma entrada de datos. Estos autores justifican la necesidad de contar con una función objetivo que permita evaluar la "calidad" o "bondad" de las particiones que estos algoritmos producen; y desarrollan una propuesta a la que llaman "Evaluador de particiones". Siguiendo este mismo enfoque, el método presentado en este artículo también pudiera ser utilizado con estos fines. En principio, como información de entrada no bastaría con la MUA, se necesitaría también el esquema de fragmentación a evaluar. Del método general se obvia todo el proceso para generar los agrupamientos jerárquicos. Se procede a obtener la MAA* y a calcular la HI_T y la HE_T del esquema de fragmentación a evaluar, de la misma forma en que ya se ha explicado. Luego calculamos el "balance", mediante la siguiente ecuación:

$$B = |HI_T - HE_T| \quad \textbf{(6)}$$

El valor absoluto del balance es un índice que nos permitirá comparar la "calidad" de esta partición con otra obtenida con otra obtenida con otro algoritmo para la misma MUA. Mientras menor sea el valor de B mejor calidad tendrá el esquema de partición analizado, debido a que posee un mejor balance entre la homogeneidad interna y la heterogeneidad externa de sus grupos.

3 Resultados y discusión

Para el ejemplo utilizado, en particular, otros algoritmos como el de Ra [13], Zahn [14] y Partición Vertical Binaria [3] identifican el esquema de fragmentación mencionado (en 3 fragmentos) como óptimo. En el enfoque de Zahn, una vez que el árbol de expansión máxima se obtiene, dos condiciones diferentes producen dos diferentes esquemas de partición. Uno de ellos es el mismo que se identifica como óptimo. Aplicando el Evaluador de Particiones de Chakravarthy [8] se obtiene el mismo resultado.

El método aquí expuesto también ha sido probado, con resultados satisfactorios, para los ejemplos que aparecen en [9], [16] y otros casos. También se han empleado ejemplos diseñados intencionalmente que demuestran que no es sensible a los casos extremos (k=1 y k=n).

Tabla 5. Matriz de Uso de Atributos de dimensión 6x6 (*ejemplo 2*).

ref	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	*acc*
T1	1	1	1	1	1	0	20
T2	1	1	1	1	0	1	21
T3	1	1	1	0	1	1	19
T4	1	1	0	1	1	1	22
T5	1	0	1	1	1	1	18
T6	0	1	1	1	1	1	20

La anterior MUA, altamente poblada de unos y con valores de frecuencia de accesos muy similares, nos sugiere de antemano que el agrupamiento debería definir un solo grupo compuesto por todos los atributos. Al aplicar el método propuesto, se distingue claramente que el punto de balance se encuentra entre la fusión 4 y 5. Al calcular la ordenada de dicho punto, x=4,595115, se comprueba que se encuentra ligeramente más cercano a la fusión 5, por lo que, siguiendo la regla de decisión anteriormente descrita, el agrupamiento seleccionado es aquel definido por la última fusión, quedando todos los atributos en un mismo grupo, como se vaticinaba desde el principio. Este resultado es corroborado al aplicar el Evaluador de Particiones de Chakravarthy [8].

Por lo tanto, este ejemplo demuestra que la regla de decisión propuesta no es sensible a los casos en que los atributos deberían aparecer todos en un mismo grupo.

4 Conclusiones

Con el desarrollo de la presente investigación, se ha logrado proponer un método y una regla de decisión, capaces de eliminar, en un alto grado, la subjetividad humana, en el proceso de particionado vertical de bases de datos. Como se ha podido comprobar, el centro de la propuesta se basa en la existencia de un punto de balance entre la HI_T y HE_T, y este punto es único. La regla de decisión es clara, precisa y fácil de aplicar. En futuras publicaciones se presentarán los resultados en comparación con otros métodos y reglas de decisión, como es el caso del índice de Mojena [6][17].

El hecho de que al encontrar el punto de balance, no es necesario continuar el proceso de obtención de agrupaciones, es un elemento que repercute directamente en el nivel de cómputo que requiere el método.

La propuesta supera también la limitante de varias reglas de decisión previas, de ser sensibles a los casos extremos (k=1 y k=n), donde k representa el número de grupos y n el número de atributos.

Este método puede ser usado también para evaluar la “calidad” de las particiones obtenidas con otros métodos de fragmentación vertical.

Referencias

1. Taddei, E., Kury, A.: Fragmentación vertical y asignación simultánea en BDD usando algoritmos genéticos, (2000), http://cursos.itam.mx/akuri/PUBLICA.CNS/2000/Fragmentaci%F3n%20Vertical%20usando%20AGs.pdf
2. Özsu, M.T., Valduriez, P.: Principles of Distributed Database Systems (2nd ed.). Prentice-Hall, USA (1999)
3. Navathe, S., Ceri, S., Wiederhold, G., Dou, J.: Vertical Partitioning algorithms for database design. In: ACM Transactions on Database Systems, vol. 9, No. 4, pp. 680--710.(1984)
4. Caliński, T., Harabasz, J.: A Dendrite Method for Cluster Analysis. In: Communications in Statistics - Theory and Methods, vol. 3, No. 1, pp. 1 -- 27. (1974)
5. Milligan, G.W., Cooper, M.C.: An Examination of Procedures for Determining the Number of Clusters in a Data Set. In: Psychometrika, vol. 50, No. 2, pp. 159 --179. (1985)
6. Herrera, A.: La clasificación numérica y su aplicación en la ecología. Sammenycar C. x A., Santo Domingo, República Dominicana (2000)
7. Du, J., Barker, K., Alhajj, R.: Attraction-A global affinity measure for data base vertical partitioning, (2003), http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/b/barker:ken.html
8. Chakravarthy, S., Muthuraj, J., Varadarajan, R., Navathe, S.B.: An Objective Function for Vertically Partitioning Relations in Distributed Databases and its Analysis. In: Distributed and Parallel Databases, vol. 2, No. 1. (1993)
9. Runceanu, A.: TOWARDS VERTICAL FRAGMENTATION IN DISTRIBUTED DATABASES. In: Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering, pp. 57--62. (2008), http://springerlink3.metapress.com/content/u8nn1622r5108q6v/
10. Hoffer, J. A., Severance, D.G.: The Use of Cluster Analysis in Physical Database Design. In: Proceedings of the 1st International Conference on Very Large Databases. (1975), http://portal.acm.org/citation.cfm?id=1282480
11. Zorrilla, M.E., Mora, E., Corcuera, P., Fernández, J.: Vertical Partitioning Algorithms in Distributed Databases. In: Computer Aided Systems Theory - EUROCAST'99. LNCS, vol. 1798/2000, pp. 465—474. Springer (2000) http://www.springerlink.com/content/b01725637014666t/
12. Hair, J.F., Anderson, R.E., Tatham, R.L., Black,W.C.: Cluster Analysis. In: Multivariate Data Analysis with Readings, pp. 420--483. Prentice-Hall, New Jersey (1995)
13. Navathe, S., Ra, M.: Vertical Partitioning for Database Design: A Graphical Algorithm. ACM SIGMOD, Portland (1989)
14. Zahn, C.: Graph-theoretical methods for detecting and describing Gestalt Clusters. IEEE Transactions on Computers, vol. 20, pp. 68--86 (1971)
15. Mojena, R.: Hierarchical grouping methods and stopping rules: An evaluation. Computer Journal, vol. 20, pp. 359--363 (1977)
16. Runceanu, A.: Fragmentation in Distributed Database. In: Innovations and advanced techniques in systems computing sciences and software engineering, pp. 57--62 (2008)
17. Mojena, R.: Hierarchical grouping methods and stopping rules: an evaluation. Computer Journal, No. 20, pp. 353--363. (1977).