Estrategias de Construcción sobre Estructuras Métricas para Búsquedas por Similitud*

Roberto Uribe-Paredes[1,2], Claudio Márquez[1,2], and Roberto Solar[1,2]

[1] Depto. de Ingeniería en Computación,
Universidad de Magallanes, Chile
[2] Grupo de Bases de Datos - UART,
Universidad Nacional de la Patagonia Austral, Río Turbio, Argentina
E-mail: {ruribe,clmarque,rsolar}@ona.fi.umag.cl

Resumen La *Lista de Clusters* (LC) es una efectiva técnica para indexar espacios métricos de alta dimensionalidad. LC es una estructura basada en clustering y del tipo arreglo para búsquedas por similitud. El *Sparse Spatial Selection* (*SSS*) es una nueva estructura basada en pivotes para búsqueda por similitud en espacios métricos. Esta estructura es del tipo arreglo y ha demostrado buen rendimiento durante la búsqueda comparado con otros métodos conocidos.
El presente trabajo muestra distintas estrategias de construcción sobre el LC, entre éstas, la utilización del *SSS* como un método general de selección de pivotes o centros El artículo también muestra las ventajas de la utilización de otras técnicas como es la de conservar la distancia entre los objetos y el centro del Cluster, además, ver los efectos de la aplicación recursiva de dichos métodos. Finalmente, se muestra la infuencia de la utilización de particiones de Voronoi para la distribución de los objetos dentro de la estructura.
Resultados experimentales preliminares demuestran que las nuevas versiones del LC tienen mejor desempeño, en términos de evaluaciones de distancia que la estructura original y que otras estructuras conocidas.
Palabras claves: bases de datos, estructuras de datos, algoritmos, espacios métricos, consultas por similitud.

1. Introducción

La búsqueda de objetos similares en un gran conjunto de objetos almacenados en una base de datos métrica se ha convertido en uno de los problemas de gran interés. Por ejemplo, una consulta típica para estas aplicaciones es la *búsqueda por rango* la cual consiste en obtener todos los objetos que están a una determinada distancia del objeto consultado. A partir de esta operación se puede construir otra, como los vecinos más cercanos. La aplicación de estas técnicas pueden ser encontradas, en reconocimiento de voz e imagen, en problemas de minería de datos, detección de plagios y muchas otras.

* Financiado por U. de Magallanes, Chile (PR-F1-02IC-08) y Unidad Académica de Río Turbio, U. Nacional de la Patagonia Austral, Argentina (29/C035-1).

La similitud se modeliza en muchos casos interesantes a través de un espacio métrico, y la búsqueda de objetos más similares a través de una búsqueda por rango o de vecinos más cercanos. Un espacio métrico es un conjunto $\mathbb{X}$ con una función de distancia $d : \mathbb{X}^2 \rightarrow \mathbb{R}$, tal que $\forall x, y, z \in \mathbb{X}$, se debe cumplir las propiedades de: positividad ($d(x, y) \geq 0$ *and* $d(x, y) = 0$ *ssi* $x = y$), simetría ($d(x, y) = d(y, x)$) y desigualdad triangular ($d(x, y) + d(y, z) \geq (d(x, z))$.

Sobre un espacio métrico ($\mathbb{X}$,d), un conjunto de datos finito $\mathbb{Y} \subseteq \mathbb{X}$, se pueden realizar una serie de consultas. La consulta básica es la *consulta por rango*. Sea una consulta $x \in \mathbb{X}$, y un rango $r \in \mathbb{R}$. La consulta de rango alrededor de x con rango r es el conjunto de puntos $y \in \mathbb{Y}$, tal que $d(x, y) \leq r$. Un segundo tipo de consulta, que puede construirse usando la consulta por rango es, *los k vecinos más cercanos*. Sea una consulta $x \in \mathbb{X}$ y un entero k. Los k vecinos más cercanos a x son un subconjunto $\mathbb{A}$ de objetos de $\mathbb{Y}$, donde la $|\mathbb{A}| = k$ y no existe un objeto $y \in \mathbb{A}$ tal que *d(y,x)* sea menor a la distancia de algún objeto de $\mathbb{A}$ a x.

El objetivo de los algoritmos de búsqueda es minimizar la cantidad de evaluaciones de distancia realizadas para resolver la consulta. Los métodos para buscar en espacios métricos se basan principalmente en dividir el espacio empleando la distancia a uno o más objetos seleccionados. El no trabajar con las características particulares de cada aplicación tiene la ventaja de ser más general, pues los algoritmos funcionan con cualquier tipo de objeto [1].

Existen distintas estructuras para buscar en espacios métricos, las cuales pueden ocupar funciones discretas o continuas de distancia. Algunos son GNAT [2], MTree [3], SAT [4], Slim-Tree [5], EGNAT [6].

Algunas de las estructuras basan la búsqueda en pivotes y otras en clustering. En el primer caso se seleccionan pivotes del conjunto de datos y se precalculan las distancias entre los elementos y los pivotes. Cuando se realiza una consulta, se calcula la distancia de la consulta a los pivotes y se usa la desigualdad triangular para descartar candidatos.

Los algoritmos basados en clustering dividen el espacio en áreas, donde cada área tiene un *centro*. Se almacena alguna información sobre el área que permita descartar toda el área mediante sólo comparar la consulta con su centro. Los algoritmos de clustering son los mejores para espacios de alta dimensión, que es el problema más difícil en la práctica.

Existen dos criterios para delimitar las áreas en las estructuras basadas en clustering, *hiperplanos* y *radio cobertor* (*covering radius*). El primero divide el espacio en particiones de *Voronoi* y determina el hiperplano al cual pertenece la consulta según a qué centro corresponde. El criterio de radio cobertor divide el espacio en esferas que pueden intersectarse y una consulta puede pertenecer a más de una esfera.

Un *diagrama de Voronoi* está definido como la subdivisión del plano en n áreas, una por cada centro c_i del conjunto $\{c_1, c_2, \ldots, c_n\}$ (centros), tal que $q \in$ al área c_i sí y sólo sí la distancia euclidiana $d(q, c_i) < d(q, c_j)$ para cada c_j, con $j \neq i$.

Uno de los problemas que provoca que muchas veces buenas estructuras arrojen malos resultados es la elección poco afortunada de centros o pivotes. En este

sentido, este trabajo propone el uso de un nuevo método denominado *Sparse Spatial Selection* (*SSS*) [7], como un método general para selección de centros o pivotes. Para la aplicación de este método, se eligió la estructura denominada *Lista de Clusters* [8], que es una estructuras basadas en clustering, del tipo arreglo y utiliza el radio cobertor para descartar conjuntos de objetos durante la búsqueda.

Para los experimentos de este artículo se seleccionó un espacio consistente en un diccionario de palabras en castellano de 86.061 objetos con *distancia de edición*. El segundo es un espacio de 100,000 vectores de coordenadas reales de dimensión 10 con distribución de *Gauss* con media 1 y varianza 0.1, para este espacio se utilizó la *distancia Euclidiana*. El tercer espacio es una colección de 40.700 imágenes extraidas de los archivos de imágenes y video de la NASA representadas como vectores de dimensión 20. Para la búsqueda se creó la estructura con el 90 % de los datos y se reservó el 10 % como consultas.

2. *Lista de Clusters*

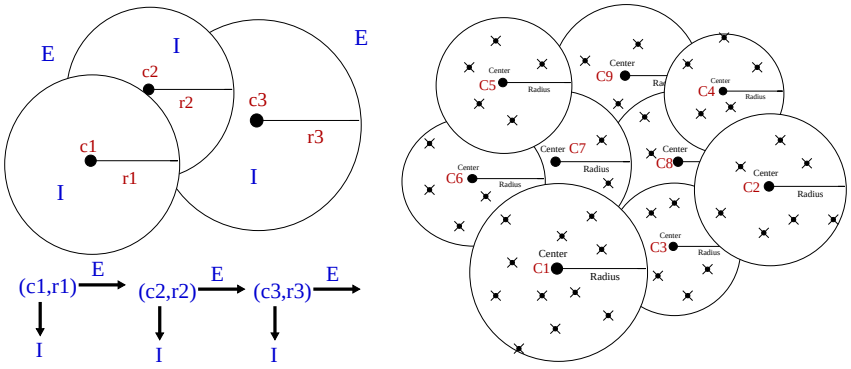
La ***Lista de Clusters*** [8] es una estructura basada en Clustering o particiones compactas, la cual es muy similar a una Lista Enlazada. Diseñada para tener un buen desempeño en espacios de altas dimensiones.

En la *Lista de Clusters* se selecciona un centro c perteneciente a la base de datos $\mathbb{Y}$ y un radio r el cual determina la fracción del espacio que abarca la esfera (c, r) definida como el subconjunto de elementos de $\mathbb{Y}$ los cuales están a una distancia no mayor a r del centro c. Luego se define como I a los elementos que están dentro de la esfera de centro c también llamado *Bucket*, y E definido como el resto de los elementos externos a la esfera de centro c. Este proceso se repite recursivamente. En consecuencia se obtiene una lista compuesta por un centro, un radio y un Bucket (c,r,I) denominado Cluster (ver figura 1(a)).

Comparado con otros algoritmos de Clustering, la *Lista de Clusters* solamente usa el criterio de radio cobertor y no áreas como en el *Voronoi-Tree*. también es posible ver la *Lista de Clusters* como un caso particular de *Voronoi-tree* o un *M-tree*, considerando I y E como los sub-árboles izquierda y derecha de raíz c, con las diferencias de que las estructuras recién mencionadas tratan de construir un árbol balanceado y que además poseen estructuras internas, en cambio la *Lista de Clusters* es extremadamente desbalanceada y no posee estructura interna alguna.

La estructura de datos construida no es simétrica. El primer centro escogido tiene preferencia sobre los centros subsecuentes por lo que se provoca solapamiento entre clusters. La figura 1(b) lo ilustra. Todos los elementos que están dentro del cluster del primer centro (c1 en la figura 1(b)) se guardan en su Bucket I, a pesar de eso ellos pueden quedar también dentro de los Buckets I de centros subsecuentes (c2, c3, etc. figura 1).

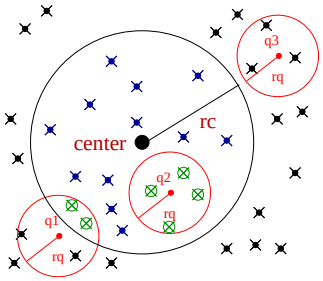
El algoritmo para la búsqueda se muestra en la figura 2(a), aplicado a una consulta q y un radio de búsqueda r sobre la Lista de Cluster L.



(a) Construcción (b) Distribución general en el espacio

Figura 1. Construcción y distribución del espacio en la Lista de Clusters.

```
Search(L,q,r)
    if L is empty then
        return
    Let L=(c,rc,I) : E
    Compute d(c,q)
    if d(c,q) <= r then
        add c to the results
    if d(c,q) <= rc + r then
        search I exhaustively
    if d(c,q) > rc − r then
        Search(E,q,r)
```



(a) Algoritmo de búsqueda (b) Tres casos

Figura 2. Búsqueda sobre la Lista de Clusters.

Una característica esencial ausente en otros algoritmos de Clustering, en donde la búsqueda necesita entrar en todos los Clusters que son intersectados por la esfera de consulta. En esta estructura la búsqueda sobre los Clusters restantes puede ser cancelada en cuanto la esfera de consulta este totalmente contenida en un Cluster. En la figura 2(b) se puede apreciar tres casos de consultas sobre un Cluster, en el caso de $q1$ se debe considerar el Bucket actual y el resto de los Clusters. La bola de consulta para $q2$ está totalmente contenida en el cluster, entonces se realiza la búsqueda sólo en este Bucket. Para $q3$ evitamos la búsqueda en el Bucket actual.

En las características generales de esta estructura no se especifica como se seleccionan los centros y radios en cada punto del algoritmo de construcción, ya que esto se relaciona con la eficacia y no a la exactitud de la estructura de datos. Una buena selección de centros podría ser más bien costosa, además se debe hacer una selección adecuada del radio para cada centro de cluster. En este

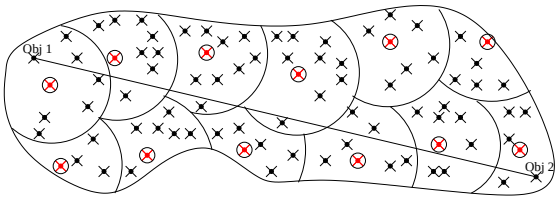


Figura 3. Representación de pivotes seleccionados. M se define como la distancia entre Obj 1 y Obj 2, los más alejados en el espacio.

sentido resulta interesante aplicar el método *SSS* para la selección de centros versus la opción de selección aleatoria. La estructura posee dos alternativas de construcción, la primera es seleccionar un radio fijo para todos los clusters de la lista y la segunda es seleccionar un tamaño fijo para todos los clusters de la lista, *Cluster de Radio Fijo* y *Cluster de tamaño Fijo* respectivamente.

3. Selección de Pivotes y Centros

En particular, la elección de pivotes o centros según sea la estructura resulta relevante para obtener un mayor rendimiento durante la búsqueda, lo que queda demostrado empíricamente en [9]. Distintas estrategias han sido propuestas como adecuadas para la elección de pivotes. En [10] se propone seleccionar como pivotes aquellos objetos que maximicen la suma de las distancias a los pivotes ya seleccionados. En [2] se siguen heurísticas que tratan de seleccionar pivotes que están lejanos entre sí. En [9] se presenta un criterio de comparación de la eficiencia entre dos conjuntos de pivotes, así también se presentan varias estrategias de selección de conjuntos donde se usa el criterio de eficiencia anterior.

3.1. *Sparse Spatial Selection (SSS)*

El método denominado *Sparse Spatial Selection* (*SSS*) [7], es una nueva técnica propuesta para la selección de pivotes, fue originalmente implementada sobre una estructura del tipo arreglo y usa la desigualdad triangular para discriminar objetos durante la búsqueda. Dicho método se comportó igual o mejor en términos de eficiencia que los propuestos en [9], con la ventaja adicional que escoge un conjunto dinámico de pivotes bien distribuidos en el espacio.

Sea $(\mathbb{X}, d)$ un espacio métrico, $\mathbb{Y} \subset \mathbb{X}$ una colección de objetos y M la distancia entre los dos objetos más alejados. En un principio el conjunto de pivotes está formado por el primer objeto de la colección. Luego, para cada elemento de la colección se verifica si está a una distancia mayor o igual a $M * \alpha$ de los pivotes seleccionados, si es así, se agrega al conjunto de pivotes, siendo α una constante cuyos valores están cercanos a 0,4[7]. La figura 3 muestra la obtención de los pivotes en un espacio cualquiera.

La construcción es similar a la estructura *FQA*[11] y a otras del tipo arreglo, pero se diferencia en la forma de elegir los pivotes y en la manera de buscar.

Básicamente se tiene un arreglo donde la cantidad de filas es la cantidad total de objetos en la Base de Datos y la cantidad de columnas es el número de pivotes.

En el presente trabajo se considera que el *SSS* es básicamente un método de selección de objetos bien distribuidos en el espacio, por lo que puede ser aplicado a cualquier estructura, independiente del tipo y de los criterios para delimitar áreas. Se considera también, que es posible construir una estructura plenamente basada en el *SSS*, es decir, una estructura que puede ajustarse completamente al espacio métrico sobre el cual es implementada.

4. *List of Clusters* y *Sparse Spatial Selection*

Durante la construcción, la *Lista de Clusters* selecciona inicialmente un centro y un radio, por lo que resulta natural elegir como radio $M * \alpha$ y cada centro usando el algoritmo *SSS*, es decir, los centros son elegidos si están ubicados a una distancia $M * \alpha$ de todos los centros anteriores. SSS puede ser utilizado para radio fijo, como para tamaño fijo, sin embargo, pruebas preliminares determinaron un comportamiento superior en listas de cluster con radio fijo. El cálculo del valor de M se realiza sobre todos los objetos de la base de datos, este proceso es costoso, sin embargo, es off-line y en este trabajo no se lo considerá como costo de construcción.

Se pudo observar experimentalmente, que para la estructura Lista de Clusters en su versión de radio fijo con selección aleatoria de centros versus la alternativa de selección de centros usando SSS y radio $M * \alpha$, la diferencia fue ínfima, sobre todo en el espacio de palabras. Esto último debido a que la función de distancia es discreta, por lo que el mejor radio, es muy parecido al mejor α. En el caso del espacio de vectores, existe una pequeña mejora al aumentar los rangos de búsqueda, en este experimento, el mejor radio es aquel que recupera el 0,1 % de los datos. Para ambos casos se utilizaron los mejores valores para radio fijo y para α.

4.1. *Lista de Clusters* Recursiva

En los indicados anteriormente no se logra una mejora importante en el desempeño de la estructura. Esto resulta así, dado que se seleccionaron los mejores métodos para ambos experimentos. El mejor radio fijo y α fueron obtenidos experimentalmente.

Considerando que el espacio es dividido una sola vez en N partes, es posible aplicar esa división en los subespacios generados, es decir, cada cluster de la estructura puede ser a su vez una *Lista de Clusters*. Entonces, una segunda alternativa de construcción es aplicar recursivamente el proceso de construcción sobre cada cluster formado en la estructura original. Para la construcción de esta estructura se utiliza el método SSS.

Finalmente, lo que se obtiene es una estructura de tipo árbol donde se seleccionan los centros espacialmente dispersos, usando un radio de $M * \alpha$.

Cada cluster de la estructura original representa un subespacio con características distintas al original, de hecho los tamaños de dicho espacio son mucho menores. Si este árbol se construye usando el M original, podría ser desventajoso debido a que el espacio quedaría s obredimensionado, provocando una baja en la eficiencia del método. Ahora, es posible calcular nuevamente el M para el nuevo subespacio, pero implicaría un costo demasiado elevado durante la construcción. Sin embargo, es posible utilizar el mismo radio cobertor del subespacio para calcular un M aproximado, sin pagar costos adicionales.

El radio cobertor es la distancia desde el centro a su elemento más alejado, por lo que se garantiza que M siempre sería menor o igual a $2 * rc$ (dos veces el radio cobertor o diametro del cluster). Esto puede ser utilizado cada vez que se realiza el proceso de construcción recursivamente.

La utilización recursiva del método SSS sobre cada cluster, modificando el valor de M, provoca que la estructura se vaya adaptando siempre a la nueva forma del espacio. Lo anterior implica que la cantidad de centros en cada nodo el árbol será dinámica, es decir, los nodos usualmente no tendrán la misma cantidad de objetos.

Finalmente, el proceso termina cuando el cluster tiene una cantidad de datos inferior a una cota determinada, por ejemplo a una página de disco.

Trabajos similares a listas de Clusters recursivas se han realizado en [12], donde uno de los métodos de selección de centros propuesto, es el de ir eligiendo el siguiente centro mas alejado a cada uno de los anteriores.

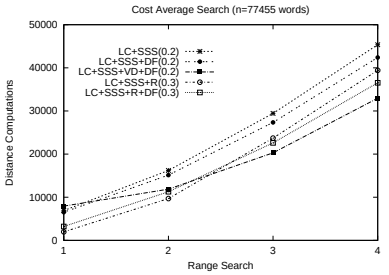
4.2. *Lista de Clusters* y otras técnicas

Inicialmente, la aplicación de la selección de centros usando el método SSS no dió los resultados esperados, sin embargo, combinando la aplicación recursiva de LC con el método *SSS*, estos mejoran notablemente el desempeño de la versión original.

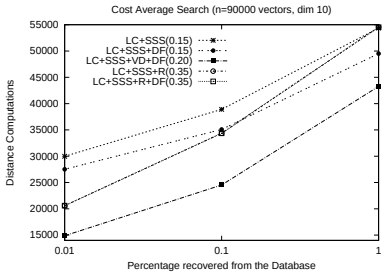
En este sentido, en este trabajo se analiza experimentalmente sobre la estructura diversas técnicas conocidas[1]. Una de las técnicas adicionadas fue la de mantener la distancia de un objeto al padre (o centro del Cluster), de tal manera de poder usar esta información durante la búsqueda. Esta técnica se aplica sobre la versión con SSS (LC+SSS+DF) y con SSS Recursiva (LC+SSS+R+DF). En la figura 4 se puede ver el comportamiento de las distintas versiones versus la original con SSS (LC+SSS). En todos los experimentos se uso un α calculado experimentalmente para cada espacio, usándose el de mejor rendimiento en cada uno de ellos. El valor usado como α aparece entre parentesis. Finalmente, la última técnica aplicada fue modificar la LC+SSS pero distribuyendo los objetos en los clusters usando diagramas de Voronoi y manteniendo la distancia al padre (LC+SSS+VD+DF). Los resultados presentados en este artículo son preliminares, sin embargo, resultan prometedores. Importante es notar que todas las versiones mejoran el rendimiento de la *Lista de Clusters* original. De los gráficos se desprende que la alternativa de mantener la distancia al padre resulta una

[1] DF: Distance to Father, R: Recursive, VD: Voronoi Diagrams

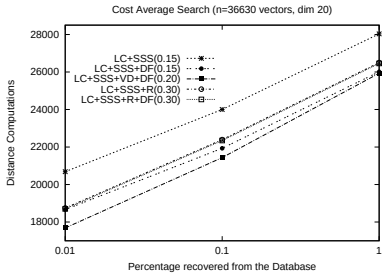
buena idea mejorando los resultados obtenidos en todos los espacios. En general, las versiones recursivas son la segunda mejor alternativa en todos los espacios, excepto en el de vectores de dimensión 20. De los gráficos, sorprende que la versión con distribución de los datos usando diagramas de Voronoi supera todas las versiones, siendo notable la diferencia en el espacio de Gauss. Se considera que la diferencia a favor puede radicar en la distribución del espacio, viendose favorecido por esta distribución de datos.



(a) Diccionario en Español



(b) Gauss dimensión 10



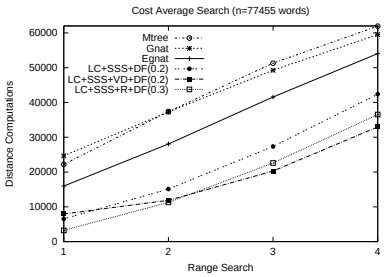
(c) Imágenes de la NASA, vectores de dim. 20

Figura 4. Costos promedios de búsqueda para las diferentes versiones de la *LC*.

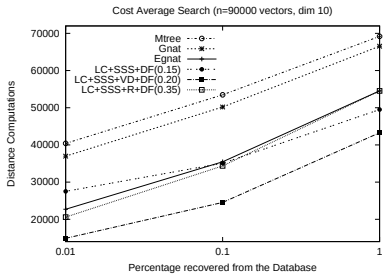
4.3. *Lista de Clusters* y otras Estructuras

Las figuras 5(a) y 5(b) muestran la disminución de evaluaciones de distancia de las nuevas versiones de *LC* versus estructuras de datos ya conocidas como son *M-Tree*, *GNAT* y *EGNAT*. En estas figuras se muestran las mejores tres versiones del *LC*. Se puede notar que en el espacio de palabras es donde mejor se comporta *LC*, en el espacio con distribución de Gauss la única estructura que resulta competitiva es el *EGNAT*, la que resulta similar a todas la otras versiones del *LC*, superada sólo por la versión con diagramas de Voronoi.

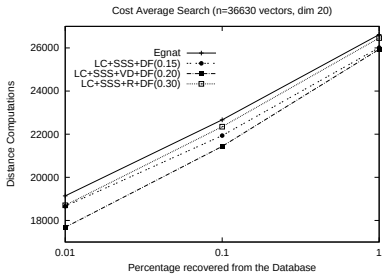
En la figura 5(c) se muestra un gráfico comparativo para el espacio de dimensión 20 entre el *EGNAT*, la mejor de las tres estructuras anteriores, y las 3



(a) Diccionario Español



(b) Vectores de Gauss, Dim. 10



(c) Imágenes de la NASA, vectores de dim. 20

Figura 5. Búsqueda: diferentes estruturas v/s versiones de LC).

versiones del LC. Para este espacio, las diferencias son menores, pero aún el LC se comporta mejor que el $EGNAT$.

5. Conclusiones

Una buena elección de pivotes o centros durante la construcción de estructuras métricas siempre será relevante para los procesos de búsqueda. Considerando que los mejores centros serán dependientes del espacio, es ideal contar con mecanismos que permitan recolectar, independiente de la forma del espacio, la mejores alternativas de centros. En este sentido, los autores consideran que el método *SSS* permite, efectivamente, obtener un conjunto adecuado de centros, lo que queda demostrado claramente en el presente artículo.

Se considera que el principal aporte del presente trabajo es desarrollar distintas versiones de la estructura *Lista de Clusters*, donde todas las nuevas propuestas tienen mejor desempeño que la versión original. Las nuevas versiones son todas basadas en clustering y usan radio cobertor para discriminar durante la búsqueda.

Se demuestra que el uso de la distancia al padre (centro del cluster) permite rebajar evaluaciones de distancia, la cual es una técnica usual en estructuras

basadas en pivotes. Es importante mencionar que la versión recursiva de la estructura se va adaptando a la forma del espacio, dado el uso de *SSS*. Esto también es posible debido a la estimación del valor de M durante la construcción, lo que no tiene costos adicionales. también, se puede decir, preliminarmente, que en determinados espacios, el uso de diagramas de Voronoi permiten una mejor distribución del espacio y con ello aumentar la eficiencia en las búsquedas.

Los autores estiman relevante, que durante el diseño de nuevas estructuras se considere los efectos que pudiesen tener algunas técnicas para una selección adecuada de centros o pivotes, la utilización de la distancia al centro del cluster y el uso de particiones de Voronoi. Los resultados experimentales proporcionan una visión de las enormes ventajas de las nuevas versiones de Lista de Clusters frente a otras estructuras prometedoras.

Referencias

1. Chávez, E., Navarro, G., Baeza-Yates, R., Marroquín, J.L.: Searching in metric spaces. In: ACM Computing Surveys. (September 2001) 33(3):273–321
2. Brin, S.: Near neighbor search in large metric spaces. In: the 21st VLDB Conference, Morgan Kaufmann Publishers (1995) 574–584
3. Ciaccia, P., Patella, M., Zezula, P.: M-tree : An efficient access method for similarity search in metric spaces. In: the 23st International Conference on VLDB. (1997) 426–435
4. Navarro, G.: Searching in metric spaces by spatial approximation. The Very Large Databases Journal (VLDBJ) **11**(1) (2002) 28–46
5. Traina, C., Traina, A., Seeger, B., Faloutsos, C.: Slim-trees: High performance metric trees minimizing overlap between nodes. In: VII International Conference on Extending Database Technology. (2000) 51–61
6. Uribe-Paredes, R.: Manipulación de estructuras métricas en memoria secundaria. Master's thesis, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Santiago, Chile (Abril 2005)
7. Pedreira, O., Brisaboa, N.R.: Spatial selection of sparse pivots for similarity search in metric spaces. In: SOFSEM 2007: 33rd Conference on Current Trends in Theory and Practice of Computer Science. Volume 4362 of Lecture Notes in Computer Science., Harrachov, Czech Republic, Springer (January, 20-26 2007) 434–445
8. Chavéz, E., Navarro, G.: An effective clustering algorithm to index high dimensional metric spaces. In: The 7th International Symposium on String Processing and Information Retrieval (SPIRE'2000), IEEE CS Press (2000) 75–86
9. Bustos, B., Navarro, G., Chávez, E.: Pivot selection techniques for proximity search in metric spaces. In: XXI Conference of the Chilean Computer Science Society, SCCC, IEEE Computer Science Press (2001) 33–40
10. Micó, L., Oncina, J., Vidal, E.: A new version of the nearest-neighbor approximating and eliminating search (aesa) with linear preprocessing-time and memory requirements. Pattern Recognition Letters **15** (1994) 9–17
11. Chavéz, E., Marroquín, J., Navarro, G.: Fixed queries array: A fast and economical data structure for proximity searching. Multimedia Tools and Applications **14**(2) (2001) 113–135
12. Mamede., M.: Recursive lists of clusters: A dynamic data structure for range queries in metric spaces. In: Computer and Information Sciences - ISCIS 2005. Volume 3733 of Lecture Notes in Computer Science. (2008)