SSSTree v2.0: Búsqueda por Similitud en Espacios Métricos con Solapamiento de Planos*

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

Abstract. *Sparse Spatial Selection* is a new pivot-based structure for similarity search in metric spaces. It is a array-type structure which has shown a good search performance when compared with other selection methods.
This work considers *SSS* as a general method for pivot or center selection and describes the building of a new clustering-based metric structure organized as a tree built using *SSS*. Experimental results show that it presents a better performance, in terms of the number of evaluations of the distance function, than other well-known structures.
Keywords: Databases, data structures, algorithms, metric spaces, similarity queries.

Resumen *Sparse Spatial Selection* es una nueva estructura basada en pivotes para búsqueda por similitud en espacios métricos. Esta estructura es del tipo arreglo y ha demostrado un buen rendimiento durante la búsqueda comparado con otros métodos de selección.
El presente trabajo considera al *SSS* como un método general para la selección de centros o pivotes y describe la construcción de una nueva estructura métrica, basada en clustering y del tipo árbol la que es construida usando *SSS*. Los resultados experimentales demuestran que tiene mejor desempeño, en términos de evaluaciones de distancia, que muchas otras estructuras conocidas.
Palabras claves: bases de datos, estructuras de datos, algoritmos, espacios métricos, consultas por similitud.

1. Introducción

1.1. Antecedentes

Uno de los problemas de gran interés en ciencias de la computación es el de "búsqueda por similitud", es decir, encontrar los elementos de un conjunto más similares a una muestra. Esta búsqueda es necesaria en múltiples aplicaciones, como ser en reconocimiento de voz e imagen, compresión de video, genética, minería de datos, recuperación de información, etc. En casi todas las aplicaciones la evaluación de la similitud entre dos elementos es cara, por lo que usualmente se trata como medida del costo de la búsqueda la cantidad de similitudes que se evalúan.

Interesa el caso donde la similitud describe un espacio métrico, es decir, está modelada por una función de distancia que respeta la desigualdad triangular. En este caso, el problema más común y difícil es en aquellos espacios de "alta dimensión" donde el histograma de distancias es concentrado, es decir, todos los objetos están más o menos a la misma distancia unos de otros.

* Parcialmente financiado por programa de investigación PR-F1-02IC-08, Universidad de Magallanes, Chile y proyecto de investigación 29/C035/1, Unidad Académica de Río Turbio, Universidad Nacional de la Patagonia Austral, Argentina.

El aumento de tamaño de las bases de datos y la aparición de nuevos tipos de datos sobre los cuales no interesa realizar búsquedas exactas, crean la necesidad de plantear nuevas estructuras para búsqueda por similitud o búsqueda aproximada, esto en busca de superar los problemas de las estruturas existentes hasta ahora. Asimismo, se necesita que dichas estructuras sean dinámicas, es decir, que permitan agregar o eliminar elementos sin necesidad de crearlas nuevamente. Así también, las aplicaciones reales requieren que dichas estructuras permitan ser almacenadas en memoria secundaria eficientemente, como también que posean métodos optimizados para reducir los costos de accesos a disco. Adicionalmente, el método elegido en las estructuras para la elección de centros y pivotes resulta relevante al momento de descartar elementos durante la búsqueda.

1.2. Marco Teórico

La similitud, en muchos casos, es modelada a través de un *espacio métrico* y la búsqueda de objetos más similares bajo una función conveniente de similitud, a través de una búsqueda por rango o vecinos más cercanos.

Definición 1 (*Espacios Métricos*): Un espacio métrico es un conjunto $\mathbb{X}$ con una función de distancia $d : \mathbb{X}^2 \to \mathbb{R}$, tal que $\forall x, y, z \in \mathbb{X}$,

1. $d(x, y) \geq 0$ *and* $d(x, y) = 0$ *ssi* $x = y$. *(positividad)*
2. $d(x, y) = d(y, x)$. *(Simetría)*
3. $d(x, y) + d(y, z) \geq (d(x, z)$. *(Desigualdad Triangular)*

Definición 2 (*Consulta por Rango*): Sea un espacio métrico ($\mathbb{X}$,d), un conjunto de datos finito $\mathbb{Y} \subseteq \mathbb{X}$, 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 $d(x, y) \leq r$.

Definición 3 (*Los k Vecinos más Cercanos*): Sea un espacio métrico ($\mathbb{X}$,d)un conjunto de datos finito $\mathbb{Y} \subseteq \mathbb{X}$, 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.

La implementación trivial de la búsqueda por similitud consiste en comparar la consulta con todos los objetos de la colección. El problema es que, en general, la evaluación de la función de distancia tiene un coste computacional elevado que hace que la búsqueda secuencial sea muy ineficiente. Esto ha dado lugar a una gran cantidad de trabajos en indexación y búsqueda en espacios métricos, que tratan de hacerla más eficiente y poder así aplicarla en grandes colecciones de datos. El objetivo de los algoritmos de indexación es reducir el número de evaluaciones de la función de distancia durante los procesos de búsqueda, aunque pagando algún costo adicional en el proceso de construcción. Esto se logra manteniendo en el índice información que, dada la consulta, permita descartar una gran cantidad de objetos sin compararlos directamente con la consulta. Para ello se utilizan las propiedades de espacios métricos como la desigualdad triangular.

Existen dos grandes enfoques que determinan los algoritmos de búsqueda en espacios métricos generales:

Algoritmos Basados en Pivotes **:** En los algoritmos basados en pivotes se selecciona un conjunto de k pivotes $\{p_1, p_2, \ldots, p_k\} \in \mathbb{Y}$. En tiempo de indexamiento, para cada objeto x

de la base de datos $\mathbb{Y}$ se calcula y almacena su distancia a los k pivotes $(d(x,p_1),\ldots,d(x,p_k))$. Si para algún pivote p_i no se cumple (1), entonces por desigualdad triangular se conoce que $d(q,x) > r$ y por lo tanto, no es necesario evaluar explicitamente $d(x,q)$. Todos los objetos que no se puedan descartar por esta regla deben ser comparados directamente con la consulta.

$$|d(q,p_i) - d(x,p_i)| \leq r, \forall i = i...k \tag{1}$$

Algoritmos Basados en Clustering **:** Los algoritmos basados en clustering dividen el espacio en áreas, donde cada área tiene un centro o split. Se almacena alguna información sobre el área que permita descartarla completamente mediante sólo comparar la consulta con su centro.

Existen varios criterios para determinar las áreas en las estructuras basadas en clustering:

Criterio de partición de Voronoi: Se selecciona un conjunto de puntos y se coloca cada punto restante dentro de la región con el centro más cercano. Las áreas se delimitan con hiperplanos y las zonas son análogas a las regiones de Voronoi en espacios vectoriales.

- **Definición (Diagrama de Voronoi):** considerese un conjunto de puntos $\{c_1, c_2, ..., c_m\}$ (centros). Se define el diagrama de Voronoi como la subdivisión del plano en m áreas, por cada c_i. La q pertenece al área si y sólo si la distancia euclidiana $d(q,c_i) \leq d(q,c_j)$ para cada c_j, con $j \neq i$.

Durante la búsqueda se evalúa $d(q,c_1), ..., d(q,c_m)$, se elige como centro c_i más cercano y se descartan todas las zonas c_j que cumplan con $d(q,c_j) > d(q,c_i) + 2r$, dado que su área de Voronoi no puede intersectar la *bola de consulta* con centro q y radio r. Se entiende por bola de consulta al conjunto de objetos que está a distancia máxima r de q.

Criterio de radio cobertor: El radio cobertor $rc(c_i)$ es la distancia entre el centro c_i y el objeto más alejado dentro de su zona. Entonces se puede descartar la zona c_i si $d(q,c_i) - r > rc(c_i)$.

Algunas estructuras combinan estas técnicas, como el caso del *GNAT* [1]. Otras sólo utilizan radio cobertor, como los *M-trees* [3], *Lista de Clusters* [2]. Algunas que utilizan hiperplanos son *GHT* y sus variantes [8,6] y los *Vorinoi trees* [4,5].

Por lo general, las estructuras diseñadas para realizar búsquedas por similitud en espacios métricos no definen un criterio para seleccionar centros o pivotes. Sin embargo, una buena elección de un conjunto centros o pivotes puede aumentar el rendimiento en los procesos de búsqueda. En el siguiente trabajo se presenta un estructura métrica diseñada en términos de los centros seleccionados, para ello se utiliza como base un método de selección de centros o pivotes denominado ***SSS***.

2. Sparse Spatial Selection Tree (*SSSTree*)

2.1. *Sparse Spatial Selection* (*SSS*)

Sparse Spatial Selection [7] es un nuevo método de búsqueda basado en pivotes. El principal aporte de este método es la estrategia de selección de pivotes. A diferencia de otros, en los que los pivotes de referencia se eligen de forma aleatoria, éste método selecciona un conjunto dinámico de pivotes bien distribuidos en el espacio, lo que permite descartar más objetos al momento de realizar el proceso de búsqueda. Además el conjunto de pivotes es dinámico, en el sentido de que es capaz de adaptarse al crecimiento de la base de datos sin que su eficiencia quede reducida.

Estrategia de selección de pivotes Sea $(\mathbb{X}, d)$ un espacio métrico, $\mathbb{Y} \subset \mathbb{X}$ y M la distancia máxima entre dos objetos cualesquiera, $M = max\{d(x, y)/x, y \in \mathbb{Y}\}$. Inicialmente el conjunto de pivotes está formado por el primer elemento de la colección. Después, cada elemento $x_i \in \mathbb{Y}$ se selecciona como pivote si y sólo sí está a una distancia mayor o igual a $M * \alpha$ de todos los pivotes ya seleccionados, siendo α una constante cuyos valores óptimos están en torno a 0.4 [7]. Es decir, un objeto de la base de datos se toma como pivote si está situado a más de una fracción de la distancia máxima de todos los pivotes (ver algoritmo 1 y figura 1).

Algoritmo 1 Selección de Pivotes

```
PIVOTES ⇐ {x_1}
for all x_i ∈ 𝕐 do
  if ∀ p ∈ PIVOTES, d(x_i, p) ≥ M * α then
    PIVOTES ⇐ PIVOTES ∪ {x_i}
  end if
end for
```

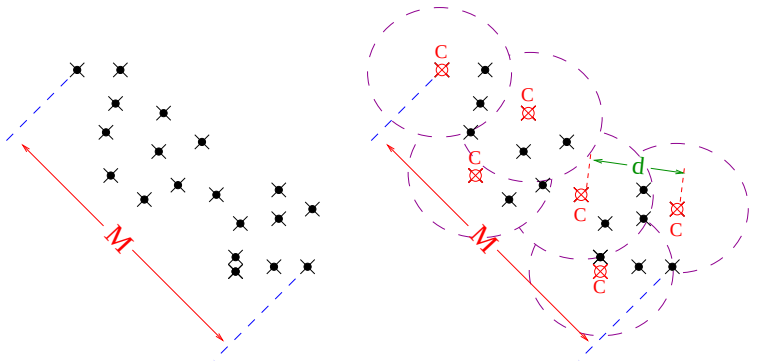


Figura 1. Representación de un espacio métrico. M: distancia entre los dos objetos más alejados de la base de datos, la distancia entre los pivotes (d) es siempre mayor a M * α.

2.2. Sparse Spatial Selection Tree Versión 1.0

SSSTree v1.0 es un árbol basado en clustering, que divide el espacio a través de *Particiones de Voronoi* y utiliza *SSS* para seleccionar los centros para cada nodo. Además utiliza bolsas o buckets auxiliares para poder calcular el valor del parámetro M. El valor de M es necesario para poder dividir el espacio sin que éste quede sobredimensionado y así obtener un mayor rendimiento en los procesos de búsqueda. Las bolsas o bucket también son utilizadas para almacenar los objetos que no son centros y luego reinsertarlos sobre la *Partición de Voronoi* a la que pertenecen.

La principal desventaja de éste método es que al momento de calcular el valor del parámetro M y al reinsertar los objetos restantes sobre la *Particion de Voronoi* a la que pertenecen se realizan una gran cantidad de evaluaciones de distancia haciendo sumamente ineficiente la construcción y/o reinserción de objetos sobre la estructura.

3. *Sparse Spatial Selection Tree* con Solapamiento de Planos

El presente trabajo considera al *SSS* como método general de selección de centros tanto como de pivotes. El artículo presenta al *SSSTree v2.0*, que es una estructura métrica del tipo árbol basada en clustering o particiones compactas en la cual la elección de centros para cada nodo es efectuada por medio de la técnica *SSS* y además utiliza en criterio de radio cobertor para delimitar cada subespacio. Lo interesante de ésta estructura es que no es necesario la utilización de estructuras auxiliares durante el proceso de construcción(a diferencia de la versión 1.0).

3.1. Construcción

Inicialmente, se extráe un objeto de la base de datos, posteriormente se crea un nodo vacío, dicho objeto es considerado como primer centro del nodo y se almacena el valor $M * \alpha$ como radio cobertor del subespacio al cual representa dicho objeto. Luego, para cada objeto de la base de datos se calcula su distancia a cada uno de los centros, si la distancia a cada uno de los centros es mayor a $M * \alpha$ el objeto es considerado como un nuevo centro para el nodo anteriomente creado, de lo contrario, se inserta en el primer cluster cuya distancia al centro sea menor o igual a $M * \alpha$, respetando el orden de inserción de centros en dicho nodo.

Finalmente, este proceso se realiza recursivamente para cada subespacio hasta que todos objetos de la base de datos se han insertado en algún nodo.

Esto puede causar principalmente que un subespacio tenga concentrado una gran cantidad de objetos en comparación a los subespacios contiguos, como se puede apreciar en la Figura 2.

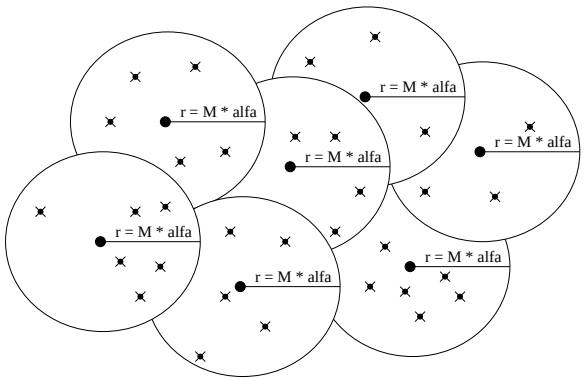


Figura 2. Solapamiento de Subespacios.

3.2. Estimación del Cálculo de M para los Subespacios

Como M es la distancia entre los dos objetos más alejados del espacio, se cumple que $M \leq 2 * RC$, donde el radio cobertor RC es la distancia entre el centro y el objeto más alejado a éste en el mismo subespacio. Esta desigualdad es útil porque permite eliminar el cálculo de M (Pruebas de esto se realizaron sobre el *egnat* con excelente resultados, no publicado).

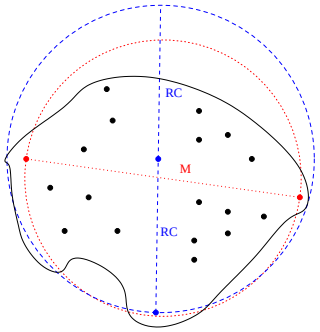


Figura 3. Optimización para el cálculo de M.

Como se puede apreciar en la figura anterior, si se utiliza $2 * RC$ como diámetro de la región, también se logra abarcar los mismos objetos como al utilizar M como diámetro. Como el valor de $2*RC$ supera o iguala al valor M, cubre una región más amplia, esto puede afectar la distribución de los centros lo cual puede perjudicar la búsqueda, sin embargo, disminuye los costos de construcción.

3.3. Búsqueda

Para descartar áreas durante la búsqueda, el *SSSTree* utiliza el criterio de radio cobertor:

El radio cobertor $rc(c_i)$ es la distancia entre el centro c_i y el objeto más alejado dentro de su zona. Entonces, se puede descartar completamente la zona c_i si $d(q, c_i) - r > rc(c_i)$.

La figura 6 muestra los resultados experimentales para la búsqueda en ambos espacios. En el espacio de Gauss, los valores graficados corresponden a los rangos que permiten recuperar el 0,01, el 0,1 y el 1 %.

De los gráficos de búsqueda se puede ver claramente que el nuevo método aventaja a las estructuras *MTree*, *GNAT* y *EGNAT*. En el espacio de Gauss el *SSSTree* tiene un desempeño similar o mejor al *EGNAT* para menores radios de búsqueda, pero decrece con el aumento de los radios de búsqueda, es decir, en el espacio de palabras, cuya dimensión intrínseca es elevada, la nueva estructura se comporta mejor que para dimensiones más bajas, logrando diferencias extremadamente notorias. La figura 7 muestra los resultados comparativos con el *EGNAT* para dos espacios de imágenes.

3.4. Datos Experimentales

Para validar el comportamiento y eficiencia de la estructura formulada en este trabajo, se han realizado diversas pruebas, con distintas colecciones de objetos. A continuación se describirán las colecciones de datos utilizadas en las distintas pruebas.

Colección de palabras: En este caso se trata de un espacio métrico real, una colección de 86.061 palabras extraídas del diccionario español.

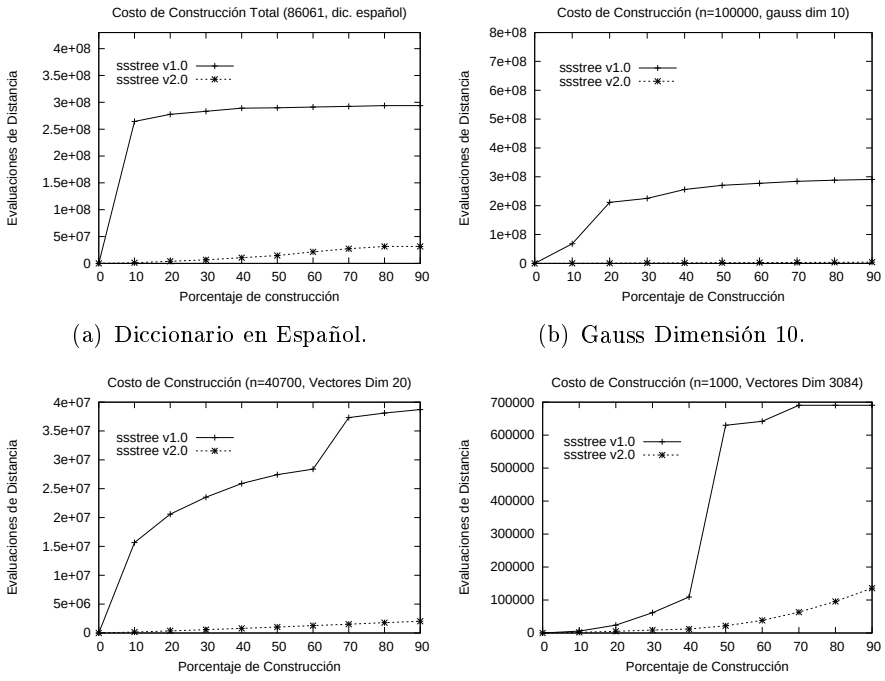
Espacios vectoriales sintéticos: Colección de 100.000 vectores de dimensión 10, creados sintéticamente, con distribución gaussiana.

Colecciones de imágenes: La primera es una colección de 40.700 imagenes extraídas de los archivos de imágenes y videos de la NASA. La segunda colección corresponde a 1.000 imágenes representadas como vectores de dimensión 3.084.

Los experimentos fueron realizados para cada versión de la estructura con cada espacio descrito anteriormente. Además, los procesos de costrucción son realizados utilizando el 90 % del total de los objetos de cada base de datos y los procesos de búsqueda son realizados utilizando como consultas el 10 % restante.

3.5. Resultados Experimentales Preliminares

Resultados de Construcción En esta sección se presentan resultados de pruebas preliminares que comparan el rendimiento de la construcción del *SSSTree v2.0* en comparación a su versión anterior.



(a) Diccionario en Español.

(b) Gauss Dimensión 10.

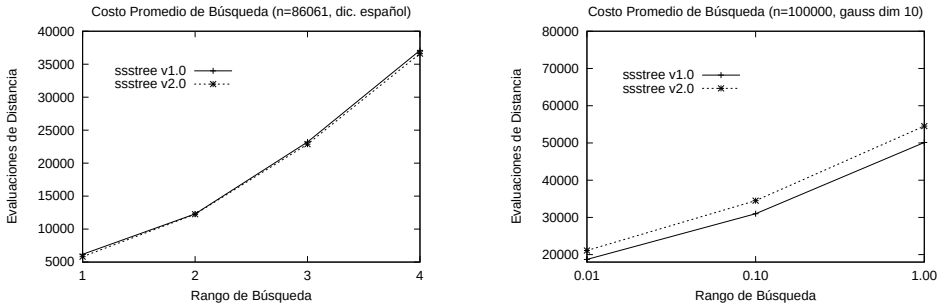
(c) Vectores Dimensión 20.

(d) Gauss Dimensión 3084.

Figura 4. ***SSSTree***: Gráficos comparativos para la construcción (SSSTree v2.0 y SSSTree v1.0).

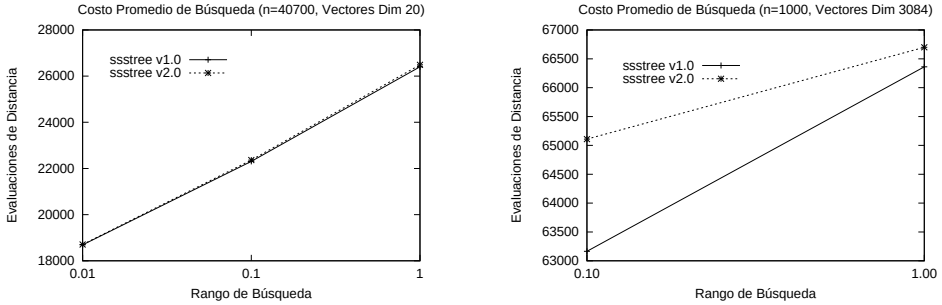
Como se puede apreciar en la figura 4, la nueva versión logra reducir considerablemente los cálculos de distancia durante los procesos de construcción en comparación a la versión anterior.

Resultados de Búsqueda En esta sección se presentan resultados de pruebas preliminares que comparan el rendimiento de la búsqueda del *SSSTree v2.0* en comparación a su su versión anterior.



(a) Diccionario en Español.

(b) Gauss Dimensión 10.



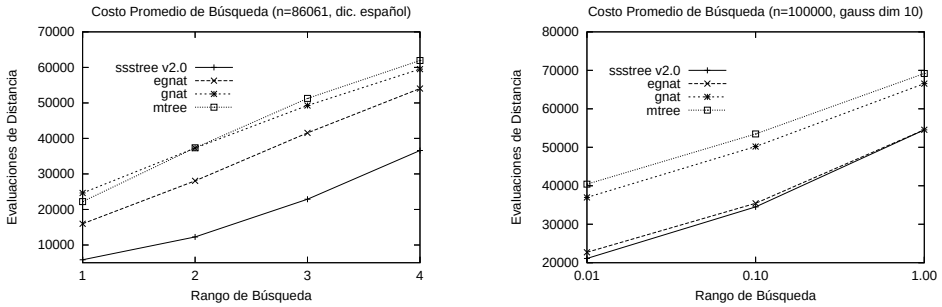
(c) Vectores Dimensión 20.

(d) Gauss Dimensión 3084.

Figura 5. ***SSSTree***: Gráficos comparativos para la búsqueda (SSSTree v2.0 y SSSTree v1.0).

Como se puede apreciar en los gráficos 5(a), 5(b) y 5(c), la cantidad de evaluaciones de distancia es similar a su versión anterior, en cambio en el gráfico 5(d) correspondiente a las imágenes de dimensión 3084 la diferencia es mayor, esto debido probablemente a que los objetos de este espacio están más concentrados y al estimar el valor de M cada subespacio queda ampliamente sobredimensionado.

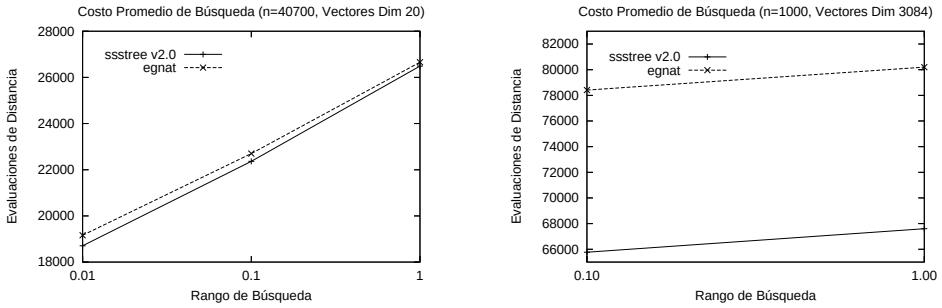
A continuación se presentan resultados de pruebas preliminares que comparan el rendimiento del SSSTree con solapamiento en el proceso de búsqueda con otras estructuras basadas en clustering conocidas.



(a) Diccionario en Español.

(b) Gauss Dimensión 10.

Figura 6. ***SSSTree***: Gráficos comparativos para la búsqueda (MTree v/s GNAT v/s EGNAT v/s SSSTree).



(a) Vectores Dimensión 20. (b) Gauss Dimensión 3084.

Figura 7. ***SSSTree***: Gráficos comparativos para la búsqueda (EGNAT v/s SSSTree).

Como se puede apreciar en los gráficos 6 y 7, esta nueva implementación del *SSSTree* mantiene los buenos resultados obtenidos en comparación a otras estructuras ya conocidas como son el *GNAT*, *EGNAT* y *MTree*.

3.6. Otras Técnicas Aplicadas

Una técnica utilizada en [9] con buenos resultados, fue combinar el uso de la ***Distancia al Padre*** al estilo de estructuras basadas en pivotes. En este caso, cada nodo almacena la distancia al centro del plano al que pertenecen, con el objetivo de discriminar adicionalmente objetos durante la búsqueda utilizando la desigualdad (1).

Una segunda técnica que es posible utilizar en ***SSSTree con Solapamiento de Planos***, es el uso de Propiedades de la ***Lista de Clusters*** [2]. Mediante la utilización de esta técnica es posible discriminar todos los clusters contiguos, siempre y cuando la consulta esté completamente contenida en un único cluster (ver figura 8).

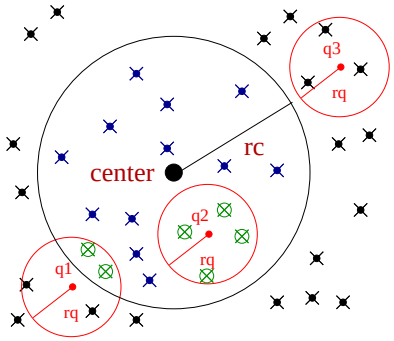
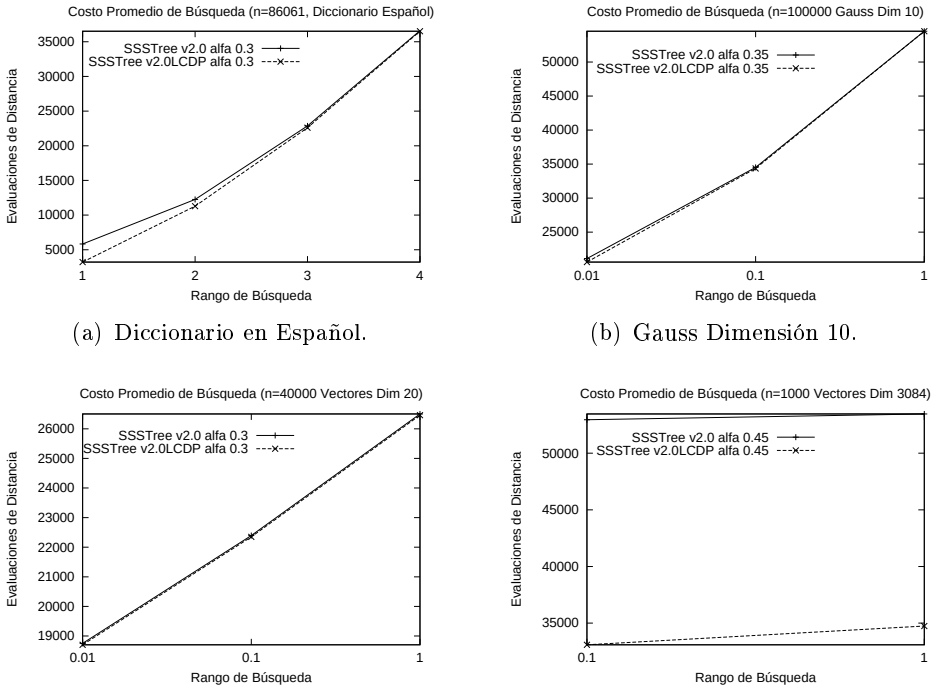


Figura 8. Propiedades de la Lista de Clusters.

Como experimentalmente, tanto como la utilización de la *Distancia al Padre*, como las *Propiedades de Lista de Cluster*, permitieron disminuir la cantidad de evaluaciones de distancia en los procesos de búsqueda en comparación a su implementación original, se decidió utilizar estas dos propiedades combinadas para obtener mejores resultados(ver figura 9).



(a) Diccionario en Español.

(b) Gauss Dimensión 10.

(c) Vectores Dimensión 20.

(d) Gauss Dimensión 3084.

Figura 9. ***SSSTree***: Gráficos comparativos para la búsqueda (SSSTree 2.0 v/s SSSTree con Distancia al Padre y Propiedad de la Lista de Clusters).

Como se puede apreciar en la figura 9 los resultados obtenidos en el caso de *Gauss Dimensión 10* y *Vectores de Dimensión 20* no varían gran cantidad en comparación a su implementación original. Pero en el caso del *Diccionario Español* y sobre todo en el espacio de *Vectores de Dimensión 3084* se puede apreciar una diferencia relevante en comparación a su versión anterior. Esto es probablemente debido a que la dimensión intrínseca de éstos dos últimos espacios es más elevada.

4. Conclusiones

4.1. Aspectos Relevantes y Aportes

Una buena elección de pivotes y 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 una alternativa de construcción para el *SSSTree*, la cual permite disminuir los costos durante el preproceso de datos sin un aumento considerable en los costos de búsqueda. La nueva versión de la estructura utiliza de forma natural el método *SSS* durante el proceso de contrucción. Adicionalmente, sobre la nueva versión fueron aplicadas técnicas conocidas en otras estructuras. La primera fue la de mantener la distancia al padre típicamente utilizada en estructuras basadas en pivotes. También fue posible

utilizar una de las propiedades de la estructura *Lista de Clusters*. Todo lo anterior provoca que la nueva versión de la estructura iguala o mejora el desempeño de los procesos de búsqueda en comparación a la versión original como se muestran en los gráficos de la sección 3.6.

Los primeros resultados demuestran lo anterior y proporcionan una visión de las enormes ventajas frente a otras estructuras que son prometedoras.

4.2. Trabajos Futuros

El diseño de esta nueva estructura trae consigo desafios que requieren analisis y pruebas con diferentes espacios métricos de baja como alta dimensión, de análisis de las capacidades dinámicas, del desempeño en memoria secundaria, de las mejores alternativas de distribución de la estructura, considerando su ejecución en paralelo, entre otros trabajos.

4.3. Agradecimientos

Al laboratorio de Base de Datos de la Facultad de Informática de la Universidad de la Coruña, España, en conjunto con quienes fue inicialmente propuesta la estructura.

Referencias

1. Sergei Brin. Near neighbor search in large metric spaces. In *the 21st VLDB Conference*, pages 574–584. Morgan Kaufmann Publishers, 1995.
2. E. Chavéz and G. Navarro. An effective clustering algorithm to index high dimensional metric spaces. In *The 7th International Symposium on String Processing and Information Retrieval (SPIRE'2000)*, pages 75–86. IEEE CS Press, 2000.
3. P. Ciaccia, M. Patella, and P. Zezula. M-tree : An efficient access method for similarity search in metric spaces. In *the 23st International Conference on VLDB*, pages 426–435, 1997.
4. F. Dehne and H. Noltemeier. Voronoi trees and clustering problems. *Informations Systems*, 12(2):171–175, 1987.
5. H. Noltemeier. Voronoi trees and applications. In *International WorkShop on Discrete Algorithms and Complexity*, pages 69–74, Fukuoka, Japan, 1989.
6. H. Noltemeier, K. Verbarg, and C. Zirkelbach. Monotonous bisector* trees - a tool for efficient partitioning of complex schemes of geometric object. In *Data Structures and Efficient Algorithms*, LNCS 594, pages 186–203, Springer-Verlag 1992.
7. Oscar Pedreira and Nieves R. Brisaboa. 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*, pages 434–445, Harrachov, Czech Republic, January, 20-26 2007. Springer.
8. J. Uhlmann. Satisfying general proximity/similarity queries with metric trees. In *Information Processing Letters*, pages 40:175–179, 1991.
9. Roberto Uribe-Paredes. Manipulación de estructuras métricas en memoria secundaria. Master's thesis, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Santiago, Chile, Abril 2005.