Recuperación de Datos para el Procesamiento de Datos Masivos

Fernando Kasián, Verónica Ludueña, Franco Merenda,
Marcela Printista, Nora Reyes, Patricia Roggero
LIDIC, Dpto. de Informática, Fac. de Cs. Físico Matemáticas y Naturales, Universidad Nacional de San Luis
{*fkasian, vlud, mprinti, nreyes,proggero*}@unsl.edu.ar, merenda.franco83@gmail.com
Karina Figueroa
Universidad Michoacana de San Nicolás de Hidalgo, México
karina@fismat.umich.mx
Claudia Deco
Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario
deco@fceia.unr.edu.ar

Resumen

La gran cantidad y variedad de información disponible en formato digital, junto con la evolución de las tecnologías de información y comunicación, han llevado al surgimiento de nuevos repositorios no estructurados de información, en los cuales los datos son no estructurados y no se adaptan fácilmente al modelo relacional. A tipos de datos tales como texto libre, imágenes, audio, video, secuencias biológicas de ADN o proteínas, entre otros; no se los puede estructurar fácilmente en claves y registros (tanto manual como computacionalmente), o tal estructuración carece de sentido práctico , y restringe de antemano los tipos de consultas que luego se pueden realizar. Por lo tanto, es cada vez más evidente en la actualidad la necesidad de procesar grandes conjuntos de datos, para obtener información útil a partir de ellos.

Así, dada una base de datos y una consulta, el objetivo de un sistema de recuperación de información es obtener, desde la base de datos, lo que podría ser útil o relevante para el usuario, usando alguna estructura de almacenamiento sobre dichos datos que permita responder a la consulta de manera eficiente. Estas estructuras necesitan ser diseñadas especialmente para ese propósito.

Palabras Claves: *bases de datos masivas, computación de alto desempeño, recuperación de información.*

1. Contexto

Esta línea de investigación se encuentra enmarcada dentro del Proyecto Consolidado 3-03-2018 de la Universidad Nacional de San Luis (UNSL) y en el Programa de Incentivos (Código 22/F834): "Tecnologías Avanzadas Aplicadas al Procesamiento de Datos Masivos", dentro de la línea "Recuperación de Datos e Información", desarrollada en el Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC) de la UNSL. Este proyecto ha sido aprobado en la UNSL por Resolución del Consejo Superior 126/18 y finaliza en 2021.

En esta línea de investigación se busca desarrollar herramientas eficientes para sistemas de recuperación de información sobre bases de datos masivas, conteniendo datos no estructurados. Por lo tanto, se analizan nuevas técnicas que permitan una buena interacción con el usuario, nuevas estructuras de datos (índices) capaces de manipular eficientemente datos no estructurados y que puedan utilizarse para administrar bases de datos masivas que contienen este tipo de datos.

Por lo tanto, el objetivo principal de esta línea es el diseño y desarrollo de índices que sirvan de apoyo a sistemas de recuperación dedicados a conjuntos de datos no estructurados masivos tales como: datos multimedia, texto, secuencias de ADN, etc. , favoreciendo en ellos la disponibilidad de estructuras de datos eficientes y escalables, para memorias jerárquicas, que además aprovechen, cuando sea necesario, la aplicación de técnicas de computación de alto desempeño (HPC).

2. Introducción y Motivación

En nuestros días, el uso masivo de internet y la disponibilidad de dispositivos electrónicos en diversos ámbitos, ha generado una significativa aceleración tanto en el crecimiento del volumen de datos capturados y almacenados, como la variedad de tipos de datos que aparecen. En este escenario, surge la necesidad de redefinir las técnicas tradicionales para el procesamiento, análisis y obtención de información útil, para formular nuevas metodologías de abordaje y lograr una mayor aplicabilidad.

Como se sabe, aún en la actualidad, los sistemas tradicionales de computación utilizan principalmente información estructurada. Este tipo de información puede organizarse en claves y registros, sobre los cuales las búsquedas tradicionales tienen sentido y donde la estructura misma de los datos puede interpretarse y utilizarse en programas casi directamente. Sin embargo, como se ha mencionado, dado el volumen y variedad de los datos actualmente disponibles, dos de las características que aparecen en los datos en el contexto de problemas de "big data", no es posible restringir las búsquedas sobre datos estructurados a las búsquedas tradicionales, porque obligaría a representar una visión parcial del problema, dejando fuera información que podría ser relevante para la resolución efectiva del mismo. Por lo tanto, en la era de "big data" es necesario administrar eficientemente información no estructurada y considerar tipos de búsqueda mucho más generales, que puedan servir de apoyo, por ejemplo, en la toma de decisiones. Las búsquedas por similitud son un tipo de búsqueda más general, que se sustentan, para lograr eficiencia en las respuestas, sobre *métodos de acceso* o *índices métricos* [5].

Por ello, si se consideran conjuntos masivos de datos no estructurados, dada la gran cantidad de datos con los que se trabaja, se pueden utilizar estos índices para lograr eficiencia en la respuesta cuando se presentan al sistema consultas de recuperación de información . Dichos índices pueden tener distintas características que los hacen adecuados para aplicaciones reales: eficientes, dinámicos, escalables, resistentes a la *maldición de la dimensión*, entre otras.

Un enfoque útil para sistemas de recuperación usando búsqueda por similitud es "la búsqueda basada en contenidos". Dicho tipo de búsqueda usa el dato no estructurado en sí mismo para describir lo que se busca. La similitud entre dos objetos se calcula mediante una función de distancia, la cual intenta describir realmente la disimilitud entre ellos.

El modelo habitual para las búsquedas por similitud es el de espacios métricos. Este modelo, además de brindar un marco formal, es independiente del dominio de la aplicación. Un espacio métrico está compuesto por un *universo* $\mathcal{U}$ de objetos y una *función de distancia* $d : \mathcal{U} \times \mathcal{U} \longrightarrow \mathbb{R}^+$, la cual cumple con las propiedades de una métrica. Sobre una *base de datos* $\mathcal{S} \subseteq \mathcal{U}$, se pueden considerar dos tipos básicos de búsqueda por similitud: la *búsqueda por rango* y la *búsqueda de los k vecinos más cercanos*. La función de distancia permite medir el mínimo esfuerzo (costo) necesario para transformar un objeto en otro. Sin embargo, para algunos de los tipos de datos no estructurados, el cálculo de la distancia puede ser muy costoso. Por lo tanto, un objetivo de todo método de acceso es ahorrar cálculos de distancia y en su gran mayoría para lograrlo aprovechan que la función de distancia satisface la desigualdad triangular.

Si se considera que la base de datos $\mathcal{S}$ posee n objetos, cualquier consulta puede responderse de manera trivial realizando n evaluaciones de distancia. Sin embargo, en la mayoría de las aplicaciones sobre grandes volúmenes de datos y siendo las distancias costosas de computar (por ej.: comparación de huellas digitales), no es factible aplicar la solución trivial. Por lo tanto, si el objetivo es responder consultas realizando la menor cantidad de cálculos de distancia posibles, se debe construir un índice a través del preprocesamiento de la base de datos. En algunos casos particulares, es probable que la base de datos, el índice, o ambos, no puedan almacenarse en memoria principal y deban hacer uso de niveles más bajos de la jerarquía de memorias, como es la memoria secundaria. Pero ello acarrea altos costos en las operaciones de E/S. Por lo tanto, para lograr eficiencia, se debe minimizar el número de operaciones de E/S, considerar el nivel de la jerarquía de memorias sobre la que se trabaja y en algunos casos admitir respuestas no exactas; utilizando, cuando sea posible, técnicas de computación paralela.

En este contexto, se considera como objetivo principal obtener herramientas de recuperación de información para procesar conjuntos masivos de datos, desarrollando nuevas técnicas y aplicaciones que soporten la interacción con el usuario, diseñando índices que permitan la manipulación eficiente de grandes volúmenes de datos no estructurados y faciliten la realización de diferentes tipos de consultas. De esta manera, se espera contribuir al desarrollo de aplicaciones reales para problemas de big data.

3. Líneas de Investigación

Dado que en esta investigación se pretende contribuir a distintos aspectos de los sistemas de recuperación de información sobre grandes volúmenes de datos no estructurados, se ha considerado el diseño de nuevas medidas de similitud, de nuevos índices y la resolución de distintas consultas sobre estos tipos de bases de datos y cómo lograr eficiencia y escalabilidad para grandes volúmenes de datos.

Índices

Los índices métricos resultan apropiados para realizar búsquedas sobre bases de datos conteniendo datos no estructurados [5]. Todos ellos aprovechan que la función de distancia debe cumplir la propiedad de desigualdad triangular para ahorrar algunos cálculos de distancia y, de esta manera, ahorrar tiempo. La desigualdad triangular permite estimar la distancia entre cualquier objeto de consulta q y los objetos de la base de datos, si se mantienen algunas distancias precalculadas entre los elementos de la base de datos y elementos distinguidos. Los dos enfoques más comunes se diferencian en si esos objetos distinguidos son *pivotes* o *centros*. Si son pivotes se almacenan las distancias de todos los objetos de la base de datos a ellos y si por el contrario son centros se particiona el espacio en zonas denominadas *particiones compactas*, por cercanía a los centros, almacenando un radio de cobertura que determina la zona de cada centro.

Los aspectos que se consideran de interés al diseñar índices incluyen: dinamismo, en qué nivel de la jerarquía de memorias deben almacenarse, si pueden aplicar técnicas de computación de alto desempeño para mejorar los tiempos de respuesta, si deben proporcionar una respuesta exacta o basta con una respuesta aproximada y la dimensionalidad del espacio métrico considerado.

Como los conjuntos de datos masivos de interés en este caso son aquellos que contienen datos no estructurados, los volúmenes de información con los que se debe trabajar (por ejemplo, millones de imágenes en la Web) hacen necesario que los índices sean almacenados en memoria secundaria. En este caso, para lograr eficiencia, no sólo se debe considerar que las búsquedas realicen el menor número de cálculos de distancia sino también, dado el costo de las operaciones de E/S, que efectúen la menor cantidad posible de operaciones sobre el disco. Por ello, esta línea se dedica a diseñar índices especialmente adaptados para trabajar en memoria secundaria, cuyo desempeño en las búsquedas sea bueno. En particular, se ha diseñado e implementado una versión paralela del *Conjunto Dinámico de Clusters* (DSC) [13]. Este índice, basado en la *Lista de Clusters* (LC) [4], está especialmente diseñado para memoria secundaria y es completamente dinámico, admite inserciones y eliminaciones y tiene un buen desempeño en las búsquedas, principalmente en la cantidad de operaciones de E/S. DSC ha demostrado ser muy competitivo frente a otras de las buenas estructuras del estado del arte. Por lo tanto, se buscará aplicar y comparar distintas estrategias de paralelización con el fin de determinar la más adecuada.

El *Árbol de Aproximación Espacial Distal* (DiSAT), basado en el *Árbol de Aproxiación Espacial* [11], es un índice estático que no necesita sintonizar ningún parámetro y es muy eficiente gracias a definir una partiición de hiperplanos con muy buenas características [3]. La raíz elegida para el DiSAT define una partición sobre el espacio, donde las zonas que se obtienen son muy compactas y los hiperplanos que las definen permiten diferenciarlas muy bien. Por ello, se busca aprovechar la información que brindan distintas particiones sobre el espacio para clasificar los elementos de acuerdo a las zonas en las que cada elemento cae en las distintas particiones consideradas. En este caso, a cada elemento se le asigna una secuencia de bits, denominada "sketch", donde cada bit indica de qué lado del hiperplano considerado se encuentra el elemento. Este conjunto de "sketches" constituye el índice en sí mismo. Cuando se considera una consulta, se calcula el sketch del elemento de consulta q y se lo compara con los sketches de todos los elementos de la base de datos, sin calcular realmente distancias entre objetos sino entre sketches y se revisan luego los objetos más prometedores primero. Se espera que un elemento similar a q estará en una partición similar en el espacio. En este caso, se puede limitar de antemano el número de distancias reales que se permite calcular, logrando una respuesta aproximada a la consulta por similitud con poco costo.

Por otra parte, se está estudiando cómo aprovechar los índices para búsquedas por similitud sobre conjuntos masivos de datos no estructurados, para solucionar un problema de estacionamiento de vehículos, usando en este caso los índices como herramienta de apoyo en un sistema de recuperación de datos e información.

Nuevas Medidas de Similitud

Existem numerosos algoritmos que permiten resolver eficientemente las búsquedas cuando se consideran espacios de baja dimensión. Sin embargo, su desempeño empeora a medida que la dimensionalidad intrínseca del espacio crece [5, 12]. Más aún, en bases de datos cuya dimensionalidad intrínseca es alta, el desempeño se puede degradar de tal forma que sea equivalente al de haber realizado una búsqueda exhaustiva sobre la base de datos completa [5]. Por lo tanto, el desafío se encuentra en esa clase

de base de datos, donde el histograma de las distancias entre los objetos es muy concentrado; es decir, donde todas las distancias entre pares de objetos de la base de datos son casi iguales.

Una aproximación práctica sobre este tipo de base de datos es resignarse a no obtener la respuesta por similitud exacta para las consultas. En su lugar, es posible conformarse con respuestas aproximadas; lo cual significa que se admite que se pierdan algunos objetos relevantes desde el conjunto de objetos de la respuesta o que se reporten en dicho conjunto algunos elementos que no sean relevantes [6]. Así, el objetivo es diseñar métodos eficientes cuya calidad de la respuesta esté dentro de ciertos límites.

En [1], se presenta un nuevo método aproximado para búsquedas por similitud, cuyo desempeño es insuperable en bases de datos de alta dimensión [10]. Sin embargo, se puede mejorar aún más su desempeño si se considera una medida diferente de similitud entre permutaciones [7].

La medida más utilizada de distancia entre permutaciones es *Spearman Footrule*, la cual se define como $S_F(\Pi_u, \Pi_q) = \sum_{1 \leq i \leq m} |\Pi_u^{-1}(i) - \Pi_q^{-1}(i)|$, donde Π_u y Π_q son las permutaciones de u y q.

Las nuevas medidas de similitud entre permutaciones que en particular, como se demuestra en [7], no cumplen con ser una métrica sino una semimétrica, se basan en particionar las permutaciones en trozos y utilizar en los trozos significativos (el inicial y el final) un factor que permita amplificar las grandes diferencias de posiciones de los permutantes y descartar el trozo medio menos significativo.

DBMS para Bases de Datos Multimedia

A pesar de que las operaciones más comunes sobre bases de datos multimedia son las búsquedas por rango o de k-vecinos más cercanos, existen otras operaciones de interés tales como las distintas variantes del *join* por similitud. La operación de join por similitud se considera una de las operaciones que debería brindar típicamente un sistema administrador para bases de datos multimedia [15].

Existen diferentes variantes para el join por similitud, dependiendo del criterio de similitud Φ utilizado, pero ellas tienen en común que se aplican entre dos bases de datos A y B, ambas subconjuntos del mismo universo del espacio métrico $\mathcal{U}$ que modela a la base de datos multimedia. El resultado de cualquiera de las variantes del join por similitud entre A y B obtendrá el conjunto de pares formados por un objeto de A y otro de B, tales que entre ellos se satisface el criterio de similitud Φ considerado. Las variantes más conocidas son: el join por rango, el join de k-vecinos más cercanos y el join de k pares de vecinos más cercanos; entre otras.

Formalmente, dadas $A, B \subseteq \mathcal{U}$, se define el *join por similitud* entre A y B ($A \underset{\Phi}{\bowtie} B$) como el conjunto de todos los pares (x, y), donde $x \in A$ e $y \in B$; es decir, $(x, y) \in A \times B$, tal que $\Phi(x, y)$ es verdadero (se satisface el criterio de similitud Φ entre x e y). Al resolver el join por similitud es posible que ambas, una o ninguna de la bases de datos posean un índice; o que ambas bases de datos se indexen conjuntamente con un índice diseñado para el join. Calcular cualquiera de las variantes del join por similitud de manera exacta es muy costoso [14], por lo tanto vale la pena analizar posibilidades de obtener una respuesta aproximada al join, más rápidamente, aunque siempre buscando buena calidad en la respuesta.

PostgreSQL es el primer sistema de base de datos que permite realizar consultas por similitud sobre algunos atributos, particularmente indexa para búsquedas de k-vecinos más cercanos (índices *KNN-GiST*). Estos índices pueden ser usados sobre texto, comparación de ubicación geoespacial, etc. Sin embargo, los índices *K-NN GiST* proveen plantillas sólo para índices con estructura de *árbol balanceado* (*B-tree*, *R-tree*), pero el "balance" no siempre es bueno para los índices que se utilizan en búsquedas por similitud [2]. Por otro lado, no se dispone de este tipo de consultas para todo tipo de datos métricos. Así, es importante proveer un DBMS para todos los posibles datos métricos y sus operaciones [9].

Más aún, dado que las respuestas a consultas de join suelen ser conjuntos muy grandes de pares de objetos y muchos de esos pares son muy similares entre sí, se planea introducir sobre las operaciones de join la posibilidad de diversificar las respuestas [16]; es decir, un operador de join por similitud que asegure un conjunto más pequeño, más diversificado de respuestas útiles y, de ser posible, más rápido de obtener. Estos desarrollos, entre otros, permitirán tener un DBMS con mayores posibilidades de aplicación en sistemas de información reales.

4. Resultados

Se ha publicado en [7] una familia de medidas de similitud para permutaciones que permiten mejorar el desempeño de los algoritmos basados en permutaciones [1]. Además, se ha publicado en [8] una nueva estructura para búsquedas aproximadas, especialmente diseñada para trabajar con grandes volúmenes

de datos en memoria secundaria.

Actualmente se está evaluando experimentalmente la versión paralela del índice *DSC*, que trabaja con grandes volúmenes de datos, diseñada especialmente para memoria secundaria, que admite inserciones y eliminaciones de elementos y que permitirá responder eficientemente a lotes de consultas por similitud. Además, se encuentra también en proceso de evaluación la propuesta de sketches basados en el *DiSAT*. Se continúa trabajando en la extensión de *PostgreSQL* para que brinde facilidades de soporte a más tipos de consultas por similitud, sobre distintos tipos de datos y considere opciones de respuesta aproximada, como también la posibilidad de diversificación de respuestas para los joins por similitud.

5. Formación de Recursos

En esta línea se están realizando las siguientes tesis de Maestría en Ciencias de la Computación:

1 - "Estructuras Eficientes sobre Datos Masivos para Búsquedas en Espacios Métricos",

2 - "Cómputo Aproximado del Grafo de Todos los k-Vecinos",

3 - "Sistema Administrador para Bases de Datos Métricas".

Además, está en desarrollo un trabajo final de la Ingeniería en Computación.

Referencias

[1] E. Chávez, K. Figueroa, and G. Navarro. Effective proximity retrieval by ordering permutations. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(9):1647–1658, 2009.

[2] E. Chávez, V. Ludueña, and N. Reyes. Revisiting the VP-forest: Unbalance to improve the performance. In *Proc. de las JCC08*, page 26, 2008.

[3] E. Chávez, V. Ludueña, N. Reyes, and P. Roggero. Faster proximity searching with the distal sat. *Information Systems*, 59:15 – 47, 2016.

[4] E. Chávez and G. Navarro. A compact space decomposition for effective metric indexing. *Pattern Recognition Letters*, 26(9):1363–1376, 2005.

[5] E. Chávez, G. Navarro, R. Baeza-Yates, and J. Marroquín. Searching in metric spaces. *ACM*, 33(3):273–321, September 2001.

[6] P. Ciaccia and M. Patella. Approximate and probabilistic methods. *SIGSPATIAL Special*, 2(2):16–19, 2010.

[7] K. Figueroa, R. Paredes, and N. Reyes. New permutation dissimilarity measures for proximity searching. In *Similarity Search and Applications - 11th International Conference, SISAP 2018*, volume 11223 of *LCNS*, pages 122–133. Springer, 2018.

[8] K. Figueroa, N. Reyes, A. Camarena-Ibarrola, and L. Valero-Elizondo. Improving the list of clustered permutation on metric spaces for similarity searching on secondary memory. In *Pattern Recognition*, pages 82–92, Cham, 2018. Springer.

[9] F. Kasián and N. Reyes. Búsquedas por similitud en PostgreSQL. In *Actas del XVIII CACiC*, pages 1098–1107, Oct. 2012.

[10] B. Naidan, L. Boytsov, and E. Nyberg. Permutation search methods are efficient, yet faster search is possible. *Proc. VLDB Endow.*, 8(12):1618–1629, August 2015.

[11] G. Navarro. Searching in metric spaces by spatial approximation. *VLDBJ*, 11(1):28–46, 2002.

[12] G. Navarro, R. Paredes, N. Reyes, and C. Bustos. An empirical evaluation of intrinsic dimension estimators. *Information Systems*, 64:206 – 218, 2017.

[13] G. Navarro and N. Reyes. New dynamic metric indices for secondary memory. *Information Systems*, 59:48 – 78, 2016.

[14] R. Paredes and N. Reyes. Solving similarity joins and range queries in metric spaces with the list of twin clusters. *JDA*, 7:18–35, March 2009.

[15] C. Rong, C. Lin, Y. N. Silva, J. Wang, W. Lu, and X. Du. Fast and scalable distributed set similarity joins for big data analytics. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 1059–1070, April 2017.

[16] L. Santos, L. Carvalho, W. Oliveira, A. Traina, and C. Jr. Traina. Diversity in similarity joins. In *Similarity Search and Applications*, volume 9371 of *LNCS*, pages 42–53. Springer, 2015.