Aportes al Modelo de Bases de Datos Métricas

Jorge Arroyuelo, Susana Esquivel, Alejandro Grosso, Verónica Ludueña, Cintia Martínez, Nora Reyes
Dpto. de Informática, Fac. de Cs. Físico-Matemáticas y Naturales, Universidad Nacional de San Luis
{*bjarroyu, esquivel, agrosso, vlud, nreyes*}@unsl.edu.ar, *cintiavmartinez*@hotmail.com
Edgar Chávez
Centro de Investigación Científica y de Educación Superior de Ensenada, México
elchavez@cicese.mx
Karina Figueroa
Fac. de Cs. Físico-Matemáticas, Universidad Michoacana de San Nicolás de Hidalgo, México
karina@computo.fismat.umich.mx
Gonzalo Navarro
Dpto. de Cs. de la Computación, Universidad de Chile
gnavarro@dcc.uchile.cl
Manuel Hoffhein, Rodrigo Paredes
Dpto. de Cs. de la Computación, Fac. de Ingeniería, Universidad de Talca, Chile
mahahein3@gmail.com, *raparede*@utalca.cl

Resumen

La computación se ha vuelto indispensable en cualquier ámbito de la vida moderna: ciencias, arte, educación, finanzas, diversión, etc., por lo que se hizo prioritario el desarrollo de aplicaciones capaces de manipular casi cualquier tipo de datos. Para lograr un alcance masivo, muchas de estas aplicaciones son cada vez más intuitivas; por ejemplo, es común ingresar una imagen o un trozo de canción a un buscador y esperar que éste muestre imágenes o canciones parecidas a la provista.

Claramente, para lograr la manipulación eficiente de datos como imágenes, audio, video, secuencias de ADN, texto, huellas digitales, etc., es necesario utilizar depósitos especializados y técnicas de búsquedas no exactas sobre ellos, ya que las soluciones tradicionales no permiten hacer frente a tales requerimientos. Las *Bases de Datos Métricas* son uno de los modelos generales en los cuales se pueden utilizar estructuras de datos especializadas que contemplen estos aspectos. Además de proveer una respuesta rápida y adecuada, será necesario un eficiente uso del espacio disponible, y si se consideran bases de datos masivas, dichas estructuras en particular serán *estructuras de datos con I/O eficiente*.

Otro aspecto importante son los lenguajes de consulta, necesarios para la manipulación de una base de datos, que no siempre poseen el poder expresivo necesario para reflejar las consultas consideradas de interés en este modelo. Así, nuestra investigación pretende contribuir a la consolidación de este nuevo modelo de bases de datos.

Palabras Claves: bases de datos no convencionales, lenguajes de consulta, índices, expresividad.

Contexto

El Proyecto Consolidado 330314 (Código 22/F414 en el Programa de Incentivos a la Investigación), *Tecnologías Avanzadas de Bases de Datos* de la Universidad Nacional de San Luis, y en particular la línea *Bases de Datos no Convencionales*, constituyen el marco en el cual se desarrolla el presente trabajo. Éste se centra en la investigación de aspectos relacionados con la administración de bases de datos capaces de manipular todo tipo de datos. Esto incluye la expresividad de los lenguajes de consulta, los operadores necesarios para responder preguntas de interés, y el análisis de aspectos teóricos, empíricos y aplicativos de los mismos; contribuyendo así a distintos campos de aplicación: sistemas de información geográfica, computación móvil, robótica, visión artificial, motores de búsqueda en internet, diseño asistido por computadora, etc.

La colaboración de nuestros integrantes en investigaciones nacionales e internacionales permite contemplar nuevas perspectivas en nuestros estudios. Se mantiene cooperación con: Universidad de Chile, Universidad de Talca (Chile), Universidad Michoacana de San Nicolás de Hidalgo (México), Centro de Investigación Científica y de Educación Superior de Ensenada (México), Universidad de Massey (Nueva Zelanda) y Universidad Nacional de La Matanza.

Introducción

En la actualidad la computación ha alcanzado todos los ámbitos de la vida moderna, esto ha provocado el desarrollo de aplicaciones capaces de adaptarse tanto a estos nuevos entornos como a los diversos usuarios de las mismas. Esto implica el desarrollo de bases de datos capaces de administrar todo tipo de datos y responder consultas sobre los mismos de una manera totalmente diferente a la tradicional, muchas veces más intuitiva. Algunos ejemplos de aplicaciones: comparación de huellas digitales, bases de datos médicas, reconocimiento de voz, reconocimiento facial, reconocimiento de imágenes, recuperación de texto, biología computacional, minería de datos, clasificación y aprendizaje automático, etc.

Todas estas aplicaciones tienen características comunes, que pueden englobarse en el modelo de *espacios métricos*. Formalmente un espacio métrico consiste de un universo de objetos $\mathbb{U}$ y una función de distancia definida entre ellos $d : \mathbb{U} \times \mathbb{U} \longmapsto \mathbb{R}^+$ que mide la disimilitud entre los objetos. En este escenario las búsquedas exactas carecen de sentido, si se introduce un trozo de melodía en un buscador, se espera obtener aquellas que sean similares a ésta. Entonces es importante la elección de este modelo por las *búsquedas por similitud*, más naturales sobre estos tipos de datos.

Para evitar la examinación secuencial de los datos al responder a este tipo de búsquedas, se utilizan los *Métodos de Acceso Métricos* (MAMs). La mayoría de estos métodos no admiten dinamismo, no están diseñados para operaciones de búsqueda complejas, ni para soportar conjuntos masivos de datos, esto permite analizar distintas maneras de optimizarlos. El trabajo con bases de datos masivas, o con aquellas que almacenan objetos muy grandes, da lugar también a líneas de investigación que, considerando el cambio del modelo de costo utilizado, diseñan MAMs más eficientes (en espacio, en I/O, etc.) para memorias jerárquicas. Otra área de investigación explorada es la expresividad de los lenguajes de consulta utilizados, tratando de incrementar la misma para expresar consultas más precisas y caracterizar la clase de consultas computables.

Líneas de Investigación y Desarrollo

Bases de Datos Métricas

Los espacios métricos generales sirven como modelo para las bases de datos que manipulan datos no convencionales (imágenes, videos, texto libre, secuencias de ADN, audio, etc.). En este modelo la complejidad se mide como el número de cálculos de distancias realizados al crear el índice, o al realizar búsquedas, dado el costo que implica su cálculo. Por ello, y dado que los MAMs son necesarios al momento de responder las diversas consultas a una base de datos, se analizan aquellos que han mostrado buen desempeño en las búsquedas. El objetivo es lograr optimizarlos reduciendo su complejidad y considerando, cuando sea necesario, la jerarquía de memorias. En general, dada una base de datos $X \subseteq \mathbb{U}$ y una consulta $q \in \mathbb{U}$ las consultas son de dos tipos: por *rango* o de *k-vecinos más cercanos*, aunque existen otras operaciones de interés [15].

Métodos de Acceso Métricos

A partir del *Árbol de Aproximación Espacial* [9], un índice que mostró un muy buen desempeño en espacios de mediana a alta dimensión, pero totalmente estático, se desarrolló uno de los pocos índices completamente dinámicos: el *Árbol de Aproximación Espacial Dinámico (DSAT)* [10] que permite realizar inserciones y eliminaciones, conservando su buen desempeño en las búsquedas.

Si en una base de datos métrica, ya sea por ser masiva o porque sus objetos son muy grandes, el índice no cabe en memoria principal, entonces surge la necesidad de hacer uso de la memoria secundaria. Esto requiere diseñar índices especialmente para memoria secundaria. Así, en [11] se presentaron versiones preliminares del *DSAT* (*DSAT+* y *DSAT**), que sólo admiten inserciones y búsquedas eficientes. Sin embargo, numerosas aplicaciones necesitan total dinamismo; es decir, que también puedan realizarse eliminaciones. Así, se ha diseñado un nuevo índice dinámico para memoria secundaria, basado en la *Lista de Clusters* [3], que tiene buen desempeño en espacios de alta dimensión, es completamente dinámico, con buena ocupación de página y sus operaciones son eficientes tanto en cálculos de distancia como en operaciones de I/O [12]. Sin embargo, las búsquedas en este índice deben recorrer completamente la lista de centros de los clusters, lo cual produce costos significativos. Así, combinando esta nueva estructura con lo bueno del *DSAT*, se ha diseñado un nuevo índice que mantiene los centros de los clusters en un *DSAT* y los clusters mismos en memoria secundaria. De esta manera, se han mejorado los costos de las operaciones en cuanto a cálculos de distancia, manteniendo los bajos costos de acceso a disco que se tenían.

Sin embargo, existen otras maneras posibles de lograr un índice totalmente dinámico a partir de la

Lista de Clusters. Por ello, se ha diseñando otro índice que, combinando *algoritmos de pivotes* [3] con "clusters" cuyo tamaño depende del tamaño de página de disco, logra operaciones eficientes.

En algunos casos, aunque la estructura sea eficiente, con el fin de lograr una respuesta más rápida, se intercambia precisión por velocidad en la respuesta. Es decir, se admite que ante una consulta se devuelvan sólo algunos objetos relevantes, siempre que dicha respuesta se encuentre disponible mucho más rápido. Estos tipos de búsquedas se denominan *aproximadas*. Un algoritmo muy eficiente para este tipo de consultas es el llamado *algoritmo basado en permutaciones* [2]. Por lo tanto, se está diseñando un nuevo índice que combine las ideas de [12], pero que agrupe por distancia entre las permutaciones de los objetos, en lugar de por distancia entre objetos. Esto permitiría obtener un índice al que se le pueda indicar el número máximo de cálculos de distancia y/o el número máximo de operaciones de I/O, que se está dispuesto a utilizar, para obtener una respuesta rápida, aunque menos precisa.

Para analizar cuán buenos son los agrupamientos que logran estas estructuras, se pueden utilizar estrategias de optimización basadas en heurísticas bioinspiradas, que sirven para detección de clusters.

Búsqueda Aproximada de los *All-k*-NN

Algunas de las aplicaciones incluídas en el modelo de *espacios métricos* son la clasificación y aprendizaje automático: un nuevo elemento debe ser clasificado de acuerdo a sus vecinos más cercanos; la cuantificación y compresión de imágenes: sólo algunos vectores pueden ser representados y aquellos que no pueden serlo, deben ser codificados como su punto representable más cercano; la predicción de funciones: se desea buscar el comportamiento más similar de una función en el pasado para predecir su comportamiento futuro probable, entre otras.

Como se ha mencionado, el cálculo de la función de distancia d es muy costoso en la mayoría de los casos y por ello se la utiliza como medida de complejidad en ese modelo. Entonces, para tratar de reducir estos costos han surgido varias técnicas para resolver el problema de consultas por similitud en un número sublineal de cálculos de distancia, generalmente basadas en el *preprocesamiento* de los datos.

Entre las primitivas básicas de las búsquedas por similitud, se encuentra la recuperación de los *k-vecinos más cercanos* a un elemento dado (k-NN(u)). Esta puede definirse como: sea X un conjunto de elementos y d la función de distancia definida entre ellos, los k-NN(u) son los k elementos en X-$\{u\}$ que tengan la menor distancia a u de acuerdo con la función d. Una variante menos estudiada de este problema, es la búsqueda de los k-vecinos más cercanos de *todos* los elementos de X, *All-k*-NN. Es decir, si $|X| = n$, obtener los *All-k*-NN es calcular los k-NN(u_i) para cada u_i en X; por supuesto realizando menos de n^2 cálculos de distancia. Algunas soluciones a este problema fueron propuestas y desarrolladas para espacios métricos generales [14, 13], basadas en la construcción del *Grafo de los k-vecinos más cercanos* (kNNG). Éste es un grafo dirigido ponderado que conecta cada elemento del espacio métrico mediante un conjunto de arcos cuyos pesos se calculan de acuerdo a la métrica del espacio en cuestión. El kNNG indexa un espacio métrico, requiriendo una cantidad moderada de memoria, y luego se utiliza en la resolución de las consultas por similitud. El desempeño en las búsquedas por similitud de dicha propuesta supera al obtenido utilizando las técnicas clásicas basadas en pivotes.

Por otro lado, teniendo en la mira la reducción de la cantidad de cálculos de distancias posibles durante una búsqueda, se ha planteado el estudio de un enfoque *aproximado* eficiente para resolver estas consultas por similitud. Este enfoque consiste en permitir una relajación en la precisión de la respuesta a fin de obtener una mejora en la complejidad de la de consulta [16, 3, 17]. El objetivo de la *búsqueda por similitud aproximada* es reducir significativamente los tiempos de búsqueda al permitir algunos errores en el resultado de la consulta. Para ello se provee, además de la consulta, un parámetro de precisión ε que controla cuán lejos queremos el resultado de esta consulta del resultado correcto. Un comportamiento razonable para este tipo de algoritmos es acercarse asintóticamente a la respuesta correcta como ε se acerca a cero. Por lo tanto, el éxito de una técnica de aproximación se basa en la resolución del compromiso calidad/tiempo [4]. Esta alternativa a la búsqueda por similitud "exacta" abarca algoritmos aproximados y probabilísticos.

Lenguajes de Consulta

Si se piensa en una base de datos simplemente como una estructura finita, se pueden utilizar las lógicas para expresar consultas sobre éstas. El empleo de lógicas para expresar consultas (o problemas) da origen a la complejidad descriptiva, que no clasifica a los problemas por la utilización de recursos como el tiempo y el espacio, sino que lo hace según el uso de recursos lógicos tales como el número de va-

riables, cuantificadores, operadores, etc. Existe una relación estrecha entre estos dos tipos de complejidades para clases que se identifican con la computación factible, pero se requiere que el dominio de las estructuras sea ordenado. En ese caso la clase de complejidad P es capturada por FO (*First-Order Logic*) extendida con un operador de punto fijo. Aún así, estas lógicas todavía resultan incompletas, ya que ninguna caracterización lógica de computación factible es conocida para estructuras cuyo dominio no está ordenado. En [6] A. Dawar demuestra que ninguna extensión de la lógica de punto fijo, con un número finito de cuantificadores generalizados, captura la clase de complejidad P; así es importante utilizar diferentes lógicas para separar problemas que, en complejidad clásica son vistos como similares.

A. Dawar [5] también demuestra que ciertos problemas NP completos sobre inequivalencia de autómatas finitos son expresables en el fragmento existencial de la lógica SO^{ω} mientras que el problema NP completo 3-coloreablidad no lo es. Nosotros definimos la lógica SO^{F} [8], una restricción semántica de la SO que exige que las variables de segundo orden se interpreten con relaciones cerradas bajo FO tipos. Demostramos que SO^{ω} está incluida estrictamente en SO^{F}, y también que en el fragmento existencial de SO^{F} podemos expresar con simpleza la propiedad de rigidez que no es posible expresar en SO^{ω}, es decir, rigidez es la propiedad que separa SO^{F} de SO^{ω}. Además, así como el fragmento existencial $\Sigma_1^{1,\omega}$ de la lógica SO^{ω} captura la clase de complejidad NP_r, contenida en la clase NP, nosotros, con el fragmento existencial $\Sigma_1^{1,F}$ de SO^{F}, capturamos la clase de complejidad NP^{F} que posee un problema perteneciente a co-NP. Con lo que conjeturamos que la clase de complejidad NP^{F} no está incluida en la clase de complejidad NP.

Además definimos una variante del juego de Ehrenfeutch-Fraïssé para la lógica SO^{F} monádico [7] y demostramos que dicho juego captura la semántica de SO^{F} monádico. En base a esto se mostró que 2-coloreabilildad no es expresable en SO^{F} monádico, lo que es curioso ya que 2-coloreabilidad pertenece a la clase de complejidad P y sin embargo no es posible expresarlo en una lógica capaz de discernir si dos elementos en una estructura satisfacen las mismas propiedades de FO. Esto implica que hay un automorfismo que conmuta dichos elementos. Como no se sabe que el problema de establecer automorfismos en una estructura esté en P (la solución por fuerza bruta es exponencial), deberíamos concluir que conocer este tipo de propiedades no basta para resolver 2-coloreabilidad eficientemente.

Por otro lado, también mostramos que un problema de SO^{F} monádico como "una estructura posee un único FO tipo para elementos" no es expresable en SO monádico existencial, donde si se puede expresar 3-coloreabilidad que se sabe que NP-completo. Es decir, en una lógica donde se pueden expresar problemas que se presumen que no tienen una solución eficiente, no es posible expresar que dados dos elementos cualesquiera hay un automorfismo que los conmuta.

En otra línea de investigación que continúa con la línea estudiada por Dawar en SO^{ω}, la cual plantea una restricción semántica a la SO, donde la valuación de las variables relacionales para los cuantificadores de segundo orden son cerrados bajo $\equiv_k$, definimos una nueva lógica de tercer orden (TO), la cual hemos llamado TO^{ω}. Ésta intenta caracterizar y estudiar clases de complejidad relacionales (temporales) de lógicas de orden superior. Una relación se dice cerrada bajo $\equiv_k$ si todas las tuplas del dominio sobre el que trabaja, que tienen el mismo tipo (satisfacen las mismas formulas de FO^{k}), están en la relación. Se definió una variación de una máquina relacional no deterministica, que denotamos como 3-NRM, donde permitimos relaciones de tercer orden en el *relational store*; ésta nos permitió asociar TO^{ω} a una clase de complejidad temporal. Esa clase de complejidad fue llamada $\mathrm{NEXPTIME}_{3,r}$, como la clase de máquinas 3-NRM que trabajan en tiempo exponencial de acuerdo al tamaño de la entrada. La clase $\mathrm{NEXPTIME}_{3,r}$ es exactamente caracterizada por el fragmento existencial de TO^{ω} [1].

Resultados y Objetivos

Estos estudios, sobre espacios métricos y sobre algunas estructuras de datos particulares, permitirán no sólo mejorar el desempeño de las mismas sino también aplicar, eventualmente, muchos de los resultados que se obtengan a otros MAMs. Nos planteamos considerar los distintos aspectos relacionados al diseño de estructuras de datos que, conscientes de la jerarquía de memorias y de las características particulares de los datos a ser indexados, logren ser eficientes en espacio y en tiempo. Por ello, se busca que los índices se adapten mejor al nivel de la jerarquía de memorias donde se almacenarán. Se espera que nuestras propuestas brinden herramientas de administración eficiente al modelo de bases

de datos métricas y así permitan que el desarrollo de dicho modelo se acerque al que tienen los modelos tradicionales de base de datos.

Respecto de los lenguajes de consulta se continuará analizando la expresividad de distintas extensiones y posibles restricciones de SO y TO, para lograr caracterizar la clase de las consultas computables sobre bases de datos no convencionales.

Actividades de Formación

Dentro de esta línea de investigación se forman alumnos y docentes-investigadores en:
Doctorado en Cs. de la Computación: un investigador finalizó su tesis sobre bases de datos métricas. Otro integrante está realizando su tesis sobre la expresividad de la lógica como lenguaje de consulta.
Maestría en Cs. de la Computación: un investigador de la línea desarrolla su tesis sobre búsqueda por similitud aproximada, a finalizar este año.
Maestría en Informática: un alumno de la Universidad Nacional de San Juan está desarrollando su tesis sobre un índice dinámico para búsquedas por similitud aproximadas en memoria secundaria.
Trabajo Final de Ingeniería Civil en Computación: un alumno de la Universidad de Talca finalizó su trabajo de fin de carrera sobre el diseño de un nuevo índice dinámico para memoria secundaria, basado en combinar pivotes con *Lista de Clusters*.

Referencias

[1] J. Arroyuelo and J. Turull Torres. The Existential Fragment of Third Order Logic and Third Order Relational Machines. XX Congreso Argentino de Cs. de la Computación: 324–333. 2014.

[2] E. Chávez, K. Figueroa, and G. Navarro. Effective proximity retrieval by ordering permutations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1647–1658, 2008.

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

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

[5] A. Dawar. A Restricted Second Order Logic for Finite Structures. Information and Computation, 143, 154-174, 1998.

[6] A. Dawar. Feasible Computation through Model Theory. Ph.D. thesis, University of Pennsylvania, 1993.

[7] A. L. Grosso. Tesis Doctoral, SO^F : Una lógica donde las variables de relación se interpretan con uniones de FO tipos, Universidad Nacional de San Luis.

[8] A. L. Grosso and J. M. Turull Torres. A Second-Order Logic in which Variables Range over Relations with Complete First-Order Types. XXIX International Conference of the Chilean Computer Science Society. IEEE 270 – 279, 2010.

[9] G. Navarro. Searching in metric spaces by spatial approximation. *The Very Large Databases Journal*, 11(1):28–46, 2002.

[10] G. Navarro and N. Reyes. Dynamic spatial approximation trees. *Journal of Experimental Algorithmics*, 12:1–68, 2008.

[11] G. Navarro and N. Reyes. Dynamic spatial approximation trees for massive data. *Procs. of 2nd International Conference on Similarity Search and Applications*, 81–88. IEEE, 2009.

[12] G. Navarro and N. Reyes. Dynamic list of clusters in secondary memory. *Proc. of 7th International Conference on Similarity Search and Applications*, LNCS 8821, 94–105. 2014.

[13] R. Paredes. *Graphs for Metric Space Searching*. PhD thesis, University of Chile, Chile, 2008.

[14] R. Paredes, E. Chávez, K. Figueroa, and G. Navarro. Practical construction of k-nearest neighbor graphs in metric spaces. In *Proc. 5th Workshop on Efficient and Experimental Algorithms*, LNCS 4007, 85–97. 2006.

[15] R. Paredes and N. Reyes. Solving similarity joins and range queries in metric spaces with the list of twin clusters. *Journal of Discrete Algorithms*, 7(1):18–35, 2009.

[16] H. Samet. *Foundations of Multidimensional and Metric Data Structures (The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling)*. Morgan Kaufmann Publishers Inc., 2006.

[17] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search: The Metric Space Approach (Advances in Database Systems)*. Springer-Verlag, 2005.