

Bases de Datos Espacio-Temporales aplicadas al análisis y seguimiento de focos epidémicos

María Mercedes Guasch y María Rosana Piergallini

Escuela de Informática y Tecnología,
Universidad Nacional del Noroeste de la Provincia de Buenos Aires, Argentina
{mercedes.guasch, mrosanapiergallini}@gmail.com

María Gisela Dorzán, Edilma Olinda Gagliardi y María Teresa Taranilla

Departamento de Informática, Facultad de Ciencias Físico Matemáticas y Naturales,
Universidad Nacional de San Luis, Argentina
{mgdorzan, oli, tarani}@unsl.edu.ar

Abstract. En este artículo se presenta el diseño de un sistema de bases de datos espacio-temporales para el análisis y seguimiento de focos epidémicos. Este diseño forma parte de una herramienta didáctica que permite establecer relaciones entre objetos con características espacio-temporales, y realizar el seguimiento de estos objetos en espacio y tiempo. Particularmente, se utilizó el índice espacio-temporal I+3 R-Tree.

Keywords: Epidemiología, Bases de datos espacio-temporales, Índices ET, Geometría Computacional.

1 Introducción

En medicina, la cura de numerosas enfermedades se ha logrado mediante planes de prevención resultando primordial la detección temprana de focos epidémicos a fin de evitar su propagación. Se hace necesario entonces identificar áreas geográficas donde habitan personas con necesidades de salud insatisfechas. En la definición de epidemia, están implícitos componentes que hacen referencia al tiempo y al espacio, debido a que una epidemia se presenta en uno o varios lugares, en determinado momento bajo ciertas condiciones de diversa naturaleza [1].

Considerar los aspectos relevantes de los focos epidémicos de una región dada, a lo largo de la historia, para la prevención de nuevos focos o expansiones de focos existentes requiere de la administración de un tipo de dato específico. En tal sentido este tipo de datos refiere a un objeto geométrico cuya forma o posición puede cambiar en el tiempo.

El modelo implementado permite establecer relaciones entre objetos con características espacio-temporales y realizar el seguimiento de dichos objetos en espacio y tiempo. La herramienta soporte de este modelo, presentada en [2], cuenta

con una interfaz que permite visualizar la distribución geográfica de los objetos analizados. La herramienta se desarrolló utilizando Bases de Datos Espacio-Temporales y el método de almacenamiento y recuperación implementado fue I+3 R-Tree [3].

Este artículo está organizado de la siguiente manera: en la primera sección se presentan algunos aspectos teóricos referidos a las Bases de Datos Espacio-Temporales y el índice espacio-temporal utilizado. A continuación, se describen aspectos de la implementación del modelo desarrollado para el análisis y seguimiento de focos epidémicos. Finalmente, se exponen las conclusiones y perspectivas de trabajo futuro.

2 Aspectos teóricos

Esta sección tiene como finalidad introducir brevemente algunos conceptos relacionados con Bases de Datos Espacio-Temporales y el índice I+3 R-Tree.

Los requerimientos de aplicaciones actuales de bases de datos promueven la necesidad de representar y operar con nuevos tipos de datos. Existen aplicaciones que requieren almacenar y consultar información histórica y actual, relacionada a los cambios de forma o posición de los objetos estudiados en diferentes escenarios a lo largo del tiempo. Para ello, es necesario disponer de herramientas que permitan modelar estos tipos de datos y realizar operaciones sobre ellos.

Las Bases de Datos Espaciales (BDE) surgen como respuesta a la necesidad de representar objetos espaciales. Para modelar los atributos espaciales de los objetos se utilizan básicamente tres tipos de abstracciones: punto, línea y región. Con el propósito de responder consultas relacionadas con las propiedades espaciales se implementan algoritmos sobre índices espaciales creados a partir de esos objetos [4].

Las Bases de Datos Temporales (BDT) soportan algún tipo de dominio de tiempo y almacenan datos históricos además de datos actuales [5].

Las Bases de Datos Espacio-Temporales (BDET) se han desarrollado a partir de las BDE y las BDT agrupando las funcionalidades de las mismas, permitiendo capturar la evolución de objetos espaciales en el tiempo. Debido a la componente temporal, las BDET manejan grandes cantidades de datos acumulados en un largo período de tiempo. Cuando se realiza una consulta sobre estos datos, examinar todos los objetos recorriendo la estructura para retornar solamente aquellos que forman parte de la respuesta, resulta ineficiente sobre todo cuando este proceso se realiza reiteradas veces. Una posible solución consiste en explorar una porción menor de la base de datos mediante el uso de índices específicos.

El I+3 R-Tree, es un índice espacio-temporal orientado a la recuperación eficiente de información histórica y actual de objetos que se mueven en espacios libres. Está compuesto por dos índices espacio-temporales vinculados por una estructura denominada Índice I que almacena la información actual de los objetos espacio-temporales y otra denominada 3D R-Tree [6] que almacena la información histórica. Ambos índices se vinculan a través de enlaces representativos del último tramo de la trayectoria correspondiente a cada objeto. En el Índice I se almacenan

las posiciones actuales de los objetos espacio-temporales modelados como cubos abiertos considerando que el final del intervalo de tiempo aún no ha sucedido. También se guardan las referencias necesarias a los cubos anteriores que describen la trayectoria del objeto, estableciendo de esta forma el vínculo con el índice 3D R-Tree. La estructura soporte de este índice es una lista secuencial de N elementos, tal que N es la cantidad de objetos considerados. Las nuplas que se almacenan en la estructura tienen la siguiente forma $(Oid, mbr, t, p3D, pa, ps)$ donde Oid es identificador del objeto, mbr es la región aproximada que ocupa actualmente el objeto, t es el tiempo de llegada del objeto a su ubicación actual, $p3D$ es un puntero al cubo anterior correspondiente al mismo Oid utilizado para mantener un historial de trayectoria, pa es el puntero al objeto insertado en el instante de tiempo inmediatamente anterior y ps es puntero al objeto insertado en el instante de tiempo siguiente. Los elementos de esta lista pueden ser accedidos en forma directa utilizando el identificador de cada objeto. Los punteros pa y ps enlazan objetos ordenándolos según los tiempos en que arribaron a sus posiciones, en orden creciente.

El índice I+3 R-Tree modifica la estructura del árbol 3D R-Tree resolver la consulta Trayectoria. Esta modificación consiste en mantener una lista para cada objeto enlazando en el índice 3D R-Tree, los cubos correspondientes a las diferentes posiciones por las cuales ha pasado el objeto. El acceso a estas listas se realiza a través del Índice I , desde la posición actual de cada objeto, mediante un direccionamiento directo.

En el 3D R-Tree se almacenan los cubos cerrados, cada uno de los cuales representa la estadía de un objeto en una posición durante un intervalo definido de tiempo. Las nuplas almacenadas en el 3D R-Tree son del tipo $(Oid, mbr_{3D}, p_{tray})$ donde Oid es el identificador del objeto, mbr_{3D} es la región tridimensional cuya altura representa el intervalo temporal durante el cual el objeto se mantuvo en la posición espacial definida por su base y p_{tray} es un puntero al cubo anterior correspondiente al mismo Oid , utilizado para mantener un historial de la trayectoria.

El índice espacio-temporal $I+3$ R-Tree admite cuatro tipos de consultas:

- *Intervalo*: Recupera todos los objetos existentes en una cierta región en el intervalo de tiempo $[t_i, t_f]$.
- *Instante*: Recupera todos los objetos presentes en una cierta región en un instante dado t .
- *Evento*: Recupera todos los objetos que entraron/salieron en una determinada región en un instante dado t .
- *Trayectoria*: Recupera el camino que ha seguido un objeto en un intervalo de tiempo $[t_i, t_f]$.

En la Figura 1 se muestra un ejemplo de la estructura del índice espacio-temporal I+3 R-Tree con los componentes de las nuplas del índice I y del índice espacio-temporal 3D R-Tree.

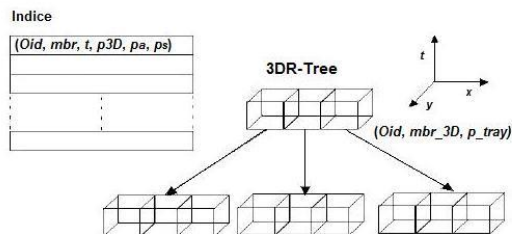


Fig. 1. Ejemplo de la estructura del índice espacio-temporal $I+3 R-Tree$

3 Modelo para el análisis y seguimiento de focos epidémicos

3.1 Descripción general de la aplicación

La aplicación tiene como objetivo principal visualizar el resultado de las consultas realizadas sobre capas seleccionadas libremente por el usuario. Con el término *capa* se referencia a cada uno de los posibles conjuntos de datos sobre los cuales se podrán realizar las consultas. La herramienta permite al iniciar una sesión de trabajo seleccionar las capas, la base de datos sobre la que se realizaran las consultas y fijar el intervalo de tiempo. Una vez iniciada la sesión se selecciona el mapa, una o más capas, el tipo y los parámetros de la consulta a ejecutar. El resultado de la consulta se muestra en modo gráfico y en modo texto, el cual puede guardarse para ser posteriormente recuperado para su visualización. La herramienta se encuentra disponible en el sitio web de la línea de investigación "Geometría Computacional y Bases de Datos" del Proyecto "Tecnologías avanzadas de Bases de datos" en la siguiente dirección:

<http://www.dirinfo.unsl.edu.ar/bd2/GeometriaComp/software/software.html>.

3.2 Aspectos de implementación

La geometría del tipo de dato espacio-temporal implementado en la herramienta es un *disco* el cual está determinado por un *punto* y un *radio*. Para las aplicaciones consideradas, el *punto* es el lugar dónde se produce el evento, por ejemplo la aparición de casos de una enfermedad o la estadía de una USAM (Unidad Sanitaria Móvil) y *radio* tiene una semántica asociada dependiendo del tipo de evento. En el ámbito de aplicación de la herramienta, el disco puede representar diferentes conceptos dependiendo el objeto que se represente:

- *Enfermedades*: punto (x, y) indica el lugar en el que aparece una enfermedad y radio indica cantidad de casos que aparecen de la enfermedad en cuestión. En la Figura 2 se pueden observar dos discos concéntricos que representan la aparición de casos de una enfermedad en distintos momentos, donde la

variación del tamaño de los discos está asociada a una variación en la cantidad de casos de la enfermedad. El radio r_1 representa la cantidad de casos de la enfermedad que aparecieron en el tiempo t_1 y r_2 los casos correspondientes al tiempo t_2 . Se puede deducir así que la cantidad de casos de la enfermedad estudiada ha aumentado en el período de tiempo transcurrido entre t_1 y t_2 .

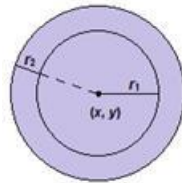


Fig. 2. Variación de tamaño de los discos.

- *USAM*: punto (x, y) indica el lugar en el que permanece la *USAM* y radio indica la cantidad de prestaciones realizadas. La Figura 3 representa los lugares en los que permaneció una *USAM* en los tiempos t_0 y t_1 y las prestaciones realizadas un cada lugar. Donde (x_0, y_0) e (x_1, y_1) son los lugares en los que permaneció en los tiempos t_0 y t_1 respectivamente. Se puede deducir que la cantidad de prestaciones realizadas en el tiempo t_0 fue mayor que las realizadas en el tiempo t_1 .

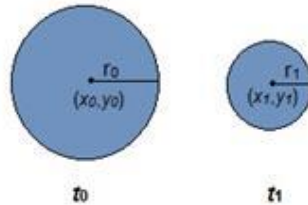


Fig. 3. Cambio de posición y de tamaño de los discos.

En la implementación del modelo propuesto se realizaron modificaciones en la estructura del índice espacio-temporal I+3 R-Tree y en los algoritmos de consulta [7, 8]. A la estructura del índice espacio-temporal I+3 R-Tree se incorporaron los atributos *tipo* y *radio*, donde *tipo* es un atributo que permite identificar la capa y *radio* tiene una semántica asociada dependiendo de la capa seleccionada. El atributo *radio* es utilizado para representar gráficamente el resultado de las consultas. Los atributos *tipo* y *radio* fueron agregados a la estructura del Índice *I* y a las nuplas del 3D R-Tree, donde el *mbr_3D* posee cinco dimensiones.

Los algoritmos de inserción utilizados por I+3 R-Tree, *InsertarIndice* e *Insertar3DRTree* se modificaron debido a la incorporación de los atributos *tipo* y *radio* en el Índice *I* y a las nuplas del 3D R-Tree. Estos algoritmos se describen a continuación.

En el algoritmo *InsertarIndice* los parámetros de entrada son los siguientes: *Oid* es el identificador del objeto, *MBR* es la posición actual del objeto, *radio* es un atributo utilizado para la representación gráfica de la consulta, *tipo* es el tipo al que pertenece el objeto de la consulta y *T* es el tiempo en el que dicho objeto está en esa posición.

Algoritmo InsertarIndice(Oid, MBR, radio, tipo, T)

```

Index.oid ← Oid
Index.mbr ← MBR
Index.radio ← radio
Index.tipoObj ← tipo
Index.T ← T
p3D ← null
EncadenarTiempos(Oid)

```

donde:

EncadenarTiempos(Oid): permite enlazar el puntero *pa* con la nupla que contiene el objeto insertado en el instante de tiempo anterior y el puntero *ps* a la nupla que contiene el objeto insertado en el instante de tiempo siguiente.

En el algoritmo *Insertar3DRTree* los parámetros de entrada son los siguientes: *Oid* es el identificador del objeto, *MBR* es la región del objeto conformada por la posición del mismo y el tiempo que se mantuvo en dicha posición el radio y el tipo de objeto, *P3D* es el puntero que guarda la dirección del nodo donde se encuentra almacenada la posición anterior del objeto.

Algoritmo Insertar3DRTree(Oid, MBR, P3D)

```

L ← BuscarHoja(Oid, Mbr, P3D, radio, tipo) /* L es
un nodo del 3D R-Tree */
Si hay suficiente espacio en L
  Guardar(Oid, MBR, P3D, radio, tipo)
  Ajustar(L)
Sino
  Split(L, Oid, Mbr, P3D)
  Ajustar(L)
  Ajustar(L')
FinSi

```

donde:

BuscarHoja(Oid, Mbr, P3D, radio, tipo): permite seleccionar un nodo hoja adecuado para almacenar la nueva entrada.

Guardar(Oid, MBR, P3D, radio, tipo): guarda la entrada en el nodo seleccionado previamente.

Split(L, Oid, Mbr, P3D): divide el nodo *L* en dos nodos, *L* y *L'*, distribuyendo los datos que se encontraban en *L* en ambos nodos. El split se realiza cuando en el nodo *L* no hay suficiente espacio.

En la implementación de las consultas Intervalo, Instante, Evento y Trayectoria se modificaron los algoritmos de consulta del I+3 R-Tree, para incorporar los cambios realizados en la estructura. A continuación se muestra un detalle de los algoritmos de consulta implementados.

Consulta Intervalo

El algoritmo *Intervalo* tiene como parámetros de entrada: p_1, p_2 los puntos utilizados para armar el rectángulo de consulta, t_i es el límite inferior del intervalo de tiempo y t_f es el límite superior sobre el cual se realiza la consulta, *tipo* es el o los tipos de objetos de la consulta. Retorna como respuesta el conjunto Q de objetos que responden a la consulta.

Algoritmo Intervalo($p_1, p_2, t_i, t_f, tipo$)

```

Si  $t_i > TMax3D$  entonces
   $Q \leftarrow$  BuscarEnIndice( $p_1, p_2, t_i, tipo$ )
Sino
  region  $\leftarrow$  ArmarRegion( $p_1, p_2, t_i, t_f$ )
  Si  $t_i < TMinInd$  entonces
     $Q \leftarrow$  BuscarEn3DR(region, tipo)
  Sino
     $Q \leftarrow$  BuscarEnIndice( $p_1, p_2, t_i, tipo$ )
     $Q \leftarrow Q \cup$  BuscarEn3DR(region, tipo)
  Finsi
Finsi
Retornar  $Q$ 

```

donde:

ArmarRegión(p_1, p_2, t_i, t_f): construye la región de consulta a partir de los puntos que representan los extremos opuestos de la misma y el tiempo de consulta.

BuscarEn3DR (región): recibe como parámetro una región plana de consulta y el o los tipos de objeto de la consulta, y devuelve todos los objetos que se intersecan con esta región de consulta en el tiempo t_i y responden a los tipos buscados.

BuscarEnIndice($p_1, p_2, t_i, tipo$): recibe como parámetros de entrada dos puntos para armar el rectángulo de consulta, el instante de tiempo sobre el cual se realiza la consulta y el o los tipos de objeto de la consulta. Retorna todos los objetos que responden a la consulta.

Consulta Instante

El algoritmo *Instante* tiene como parámetros de entrada: p_1, p_2 los puntos utilizados para armar el rectángulo de consulta, t_i es el instante de tiempo sobre el cual se realiza la consulta, *tipo* es el tipo o los tipos de objetos de consulta. Retorna como respuesta el conjunto Q de objetos que responden a la consulta.

Algoritmo Instante ($p_1, p_2, t_i, tipo$)

```
Si  $t_i > TMax3D$  entonces
   $Q \leftarrow BuscarEnIndice(p_1, p_2, t_i, tipo)$  /*La respuesta
  está sólo en el índice*/
Sino
  region  $\leftarrow ArmarRegion(p_1, p_2, t_i, t_i)$ 
  Si  $t_i < TMinInd$  entonces
     $Q \leftarrow BuscarEn3DR(region)$ 
  Sino
     $Q \leftarrow BuscarEnIndice(p_1, p_2, t_i, tipo)$ 
     $Q \leftarrow Q \cup BuscarEn3DR(region, tipo)$ 
  FinSi
FinSi
Retornar  $Q$ 
```

En una consulta del tipo Instante la región de consulta es plana, por lo tanto, en *ArmarRegión* la tercera coordenada que representa la dimensión temporal tendrá el mismo valor t_i para todo punto en la frontera de la región.

Consulta Evento

El algoritmo *Evento* tiene como parámetros de entrada: p_1, p_2 los puntos utilizados para armar el rectángulo de consulta, *tipo* es el o los tipos de objetos de la consulta. Retorna como respuesta el conjunto Q de objetos que responden a la consulta.

Algoritmo Evento ($p_1, p_2, t_i, tipo$)

```
Si  $t_i > TMax3D$  entonces
   $Q \leftarrow BuscarEnIndiceEvento(p_1, p_2, t_i, tipo)$ 
Sino
  region  $\leftarrow ArmarRegion(p_1, p_2, t_i, t_i)$ 
  Si  $t_i < TMinInd$  entonces
     $Q \leftarrow BuscarEn3DREvento(region, tipo)$ 
  Sino
     $Q \leftarrow BuscarEnIndiceEvento(region, t_i, tipo)$ 
     $Q \leftarrow Q \cup BuscarEn3DREvento(region, tipo)$ 
  FinSi
FinSi
Retornar  $Q$ 
```

donde:

BuscarEnIndiceEvento($p_1, p_2, t_i, tipo$), recibe como parámetros de entrada dos puntos que son utilizados para armar el rectángulo de consulta, el instante de tiempo sobre el cual se realiza la consulta, el o los tipos de objetos de la consulta. Devuelve como respuesta el conjunto de objetos que responden a la consulta.

Consulta Trayectoria

El algoritmo *Trayectoria* tiene como parámetros de entrada: *oid* el identificador del objeto a consultar, t_i es el límite inferior del intervalo de tiempo y t_k es el límite superior sobre el cual se realiza la consulta. Retorna como respuesta el conjunto Q de objetos que responden a la consulta.

Nupla: objeto de la forma (*Oid*, *pos*, *radio*, *tipoObj*, *tiempo*, *Psig*, *Pant*, *Ptray*) donde *Oid* es el identificador del objeto, *pos* es la posición del objeto, *radio* es un atributo utilizado para la representación gráfica de la consulta, *tipoObj* es el tipo de objeto de la consulta, *tiempo* es el instante en el cual el objeto llegó a *pos*, *Psig* es el puntero a la siguiente nupla ordenada por tiempo, *Pant* es el puntero a la nupla anterior ordenada por tiempo y *Ptray* es puntero a la pos anterior del objeto para mantener la trayectoria.

Algoritmo Trayectoria(*Oid*, t_i , t_k)

```
Nupla ← BuscarEnIndiceTray(Oid);
fin ← false;
Si (Nupla.tiempo >=  $t_i$  and Nupla.tiempo <=  $t_k$ )
  Q ← Q ∪ Nupla.pos
FinSi
Mientras (Nupla.Ptray <> Null and not fin )
  Nupla ← Nupla.Ptray
  Si Nupla.tiempo >=  $t_i$  and Nupla.tiempo <=  $t_k$ 
    Q ← Q ∪ Nupla.pos
  Sino
    Si Nupla.tiempo <  $t_i$ 
      fin ← true
    FinSi
  FinSi
FinMientras
Retornar Q
```

Estos algoritmos nos permiten dar respuestas a interrogantes tales como: ¿aumentaron o disminuyeron los casos de tal o cual enfermedad?, ¿aparecieron casos de nuevas enfermedades o de enfermedades que estaban erradicadas?, ¿existe una relación directa entre el recorrido de las USAM y la erradicación de una enfermedad?, entre otras.

Para analizar, por ejemplo, la evolución o involución de una enfermedad en una región seleccionada y en un intervalo de tiempo determinado se debe ejecutar el algoritmo Intervalo. En el modelo presentado, luego de la consulta se obtienen los focos de la enfermedad, la fecha de aparición y el número de casos en cada foco.

Se puede conocer que enfermedades ocurrieron en una región y en un instante de tiempo determinado utilizando el algoritmo Instante. Como resultado se obtienen todos los focos de las enfermedades que aparecieron en el instante de tiempo y la región seleccionados, indicando la fecha de aparición y el número de casos en cada foco.

En el caso que se necesite conocer que USAM ingreso o salió de una región seleccionada en un momento determinado, se ejecuta el algoritmo Evento obteniendo como resultado todas las USAM que ingresaron o salieron de la región, en ese instante de tiempo junto con la cantidad de prestaciones realizadas y la fecha ingreso o egreso a la región.

Para conocer el recorrido realizado por una USAM en particular en un período de tiempo dado se utiliza el algoritmo Trayectoria. En el modelo presentado se obtiene como resultado los lugares en los que permaneció, la fecha de llegada y la cantidad de prestaciones realizadas en cada uno de los lugares visitados.

4 Conclusiones y visión de futuro

En este artículo se presentó BDET aplicada al análisis y seguimiento de focos epidémicos. Se implementó un modelo utilizando BDET, específicamente el índice espacio-temporal I+3 R-Tree.

Como alcance y visión de futuro, se plantea la utilización de otros índices espacio-temporales reutilizando la interfaz de visualización de consultas, el diseño de algoritmos que permitan responder consultas espacio-temporales de mayor complejidad, como la reunión espacio-temporal y la búsqueda de los vecinos más cercanos; la incorporación de objetos espacio-temporales con otra representación geométrica y el manejo de georeferenciación en la aplicación. Otra propuesta es la implementación de un visualizador didáctico-pedagógico que muestre gráficamente el proceso de generación y carga del índice I+3 R-Tree.

Referencias bibliográficas

1. Vaughan P., Morrow R. H.: Salud comunitaria: manual de epidemiología para la gestión de servicios locales de salud. SILOS. Ed. Latinoamericana. Ochoa H. (1998)
2. Guasch, M., Piergallini, R. Dorzan, M., Gagliardi, E., Palmero, P., Taranilla M.: Una herramienta para el análisis y seguimiento de focos epidémicos, JAIHO 2014. (2014)
3. Carrasco, F., Gagliardi, E. García Sosa, J., Gutierrez, G.: Una propuesta de un método de acceso espacio-temporal: I+3 R-Tree publicado en CACIC 2006. (2006)
4. Manolopoulos Y., Papadopoulos A., Vassilakopoulos M.: Spatial Databases: Technologies, Techniques and Trends. Ideal Group Publishing. (2005)
5. Silberschatz A., Korth H., Sudarshan S.: Fundamentos de Base de Datos. McGraw-Hill (2006)
6. Theodoridis, Y, Vazirgiannis M., Sellis T.:Spatio-temporal indexing for large multimedia applications. In International Conference on Multimedia Computing and Systems (1996)
7. Libspatialindex - Hadjieleftheriou M., <http://libspatialindex.github.io/> (2013)
8. Hadjieleftheriou M., Hoel E., Tsotras Hadjieleftheriou: SaIL A Spatial Index Library for Efficient Application Integration. Geoinformatica, Volume 9, Number 4 (2005)