CC52V
Bases de datos multimedia

Prof. Benjamin Bustos

Capítulo 4
Índices multidimensionales

9.1 Modelo de costo para el R-tree

- Rendimiento de índices multidimensionales empeora al aumentar la dimensión
  - *Maldición de la dimensionalidad* ("*curse of the dimensionality*")
  - Investigación de las causas con ayuda de un modelo de costo
  - ¿Cómo optimizar los índices y los algoritmos de búsqueda?

9.1 Modelo de costo para el R-tree

- Modelo de costo [BBK+97,Böh00]
  - Objetivo: estimar el número de páginas a accesar durante una búsqueda
    - Para R-trees e índices relacionados
    - Distintos tipos de búsqueda
      - Consultas por rango
      - Consultas por vecino más cercano y *k* vecinos más cercanos
      - Distancia euclidiana y del máximo

3

9.1 Modelo de costo para el R-tree

- Modelo de costo
  - Limitaciones del modelo
    - Índice idealizado
      - Sin traslapes
      - Regiones son “casi” cuadradas (lo más posible)
    - Espacio es el hipercubo unitario $[0,1]^d$
      - Volumen del espacio = 1 (independiente de *d*)
    - Puntos y consultas están distribuidos uniformemente en el espacio

4

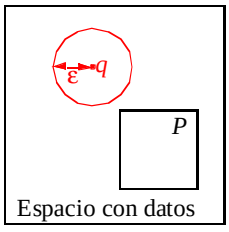
9.1.1 Modelo de costo para consultas por rango

- Datos conocidos
  - Radio de tolerancia ε
  - Extensión de las regiones
- Más tarde, ambos datos serán estimados
- Datos desconocidos
  - Posición relativa de la región y centro de la consulta (punto de consulta)
  - Ambos se considerarán uniformemente distribuidos

5

9.1.1 Modelo de costo para consultas por rango

- Se busca
  - Probabilidad que la consulta *q* debe accesar a una región *P*
  - Probabilidad (correspondiente) que la bola de consulta intersecta la región



6

9.1.1 Modelo de costo para consultas por rango

- El problema es fácil de resolver para "*point queries*"

$$\text{Probabilidad de acceso} = \frac{\text{Volumen de la región}}{\text{Volumen del espacio}}$$

  - Cálculo de la probabilidad (combinatorial) requiere de "eventos de punto"

7

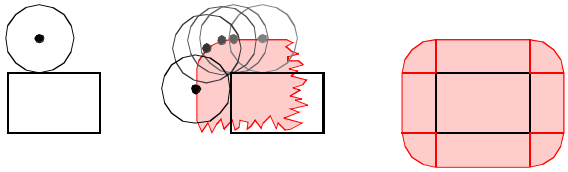
9.1.1 Modelo de costo para consultas por rango

- Truco para obtener "eventos de punto": transformar la consulta por rango en una *point query* equivalente
  - Reducir la consulta por rango a un punto
  - Agrandar en la misma medida las regiones
  - La nueva point query accesará a las regiones agrandadas ssi la consulta por rango accesa a la región original

8

9.1.1 Modelo de costo para consultas por rango

- Gráficamente

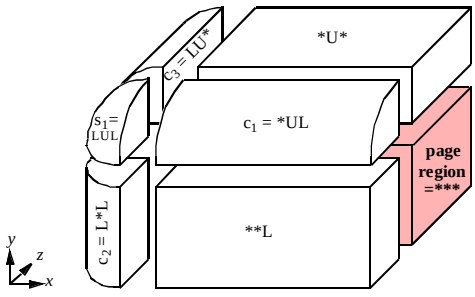


  - El objeto que surge de la consulta y región se denomina *suma de Minkowski*
    - Originalmente rectangular
    - En cada esquina hay una parte de una esfera de radio ε
    - En cada arista hay una parte de un cilindro de radio ε

9

9.1.1 Modelo de costo para consultas por rango

- Ejemplo en 3-D de la suma de Minkowski



• En cada cara: rectángulo con superficie igual a la cara y grosor ε
• En cada arista: ¼ de cilindro, largo como arista, tapas son círculos de radio ε
• En cada esquina: 1/8 de esfera de radio ε

10

9.1.1 Modelo de costo para consultas por rango

- Un hipercubo *d*-dimensional tiene además de esquinas (0-D), aristas (1-D) y caras (2-D), segmentos superficiales en 3-D, 4-D, …, hasta (*d*-1)-D
  - En cada segmento *i*-dimensional se ubica un objeto (“hipercilindro”) que tiene la forma del segmento de superficie al que está conectado en *i* dimensiones y es redondeado en *d-i* dimensiones

11

9.1.1 Modelo de costo para consultas por rango

- ¿Cuántos segmentos *i*-dimensionales tiene un cubo *d*-dimensional?
  - Notación para denotar cada segmento
    - Una *d*-tupla del alfabeto de tres símbolos L, U, y *
      - L es la cota inferior en una dimensión
      - U es la cota superior en una dimensión
      - * es toda la zona entre la cota inferior y superior

12

9.1.1 Modelo de costo para consultas por rango

- ¿Cuántos segmentos *i*-dimensionales tiene un cubo *d*-dimensional?
  - Ejemplos
    - Si una tupla no tiene *, es una esquina
      - LUL: esquina anterior superior izquierda del cubo 3-D
    - Si una tupla tiene sólo una *, es una arista
      - LU*: arista superior izquierda
    - Una tupla con *i* estrellas denota un segmento *i*-dimensional
    - La tupla que sólo contiene estrellas representa el hipercubo original

13

9.1.1 Modelo de costo para consultas por rango

- ¿Cuántos segmentos *i*-dimensionales tiene un cubo *d*-dimensional?
  - Número de tuplas con *i* estrellas

$$\binom{d}{i} \cdot 2^{d-i}$$

    - Distribuir *i* estrellas en *d* posiciones
    - Llenar posiciones restantes con L o U

14

9.1.1 Modelo de costo para consultas por rango

- Fórmula para el volumen de la suma de Minkowski de un hipercubo de lado *a* y una esfera de radio ε

$$V_{\text{Mink}}(a,\varepsilon) = \sum_{i=0}^{d} \binom{d}{i} \cdot 2^{d-i} \cdot a^i \cdot \frac{V_{\text{esfera } (d-i)\text{–dim}}(\varepsilon)}{2^{d-i}} = \sum_{i=0}^{d} \binom{d}{i} \cdot a^i \cdot V_{\text{esfera } (d-i)\text{–dim}}(\varepsilon)$$

- Volumen de una esfera *d*-dimensional

$$V_{\text{esfera } d\text{–dim}}(r) = \frac{\sqrt{\pi^d} \cdot r^d}{\Gamma\left(\frac{d}{2}+1\right)},\ \Gamma(x{+}1) = x \cdot \Gamma(x),\ \Gamma(1) = 1,\ \Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$$

15

9.1.1 Modelo de costo para consultas por rango

- Con V_{Mink} se puede calcular la probabilidad de acceso de una página
- En general interesa el número de acceso para todo el índice
  - Calcular la suma de las probabilidades de acceso resulta costoso

16

9.1.1 Modelo de costo para consultas por rango

- Estimación de la extensión de un hipercubo
  - Tamaño promedio en vez del tamaño real de las páginas
  - Para cada nivel *i* del índice se puede determinar el número n_i de páginas, si es que se conoce la utilización promedio
  - Sea C_{eff} la capacidad efectiva de una página (promedio de entradas almacenadas en una página), y N el número de puntos indexados. Se define:
    - $n_0 = N/C_{eff,data}$ (número de páginas de datos)
    - $n_i = n_{i-1}/C_{eff,dir}$ (número de páginas en el nivel *i*)

9.1.1 Modelo de costo para consultas por rango

- Estimación de la extensión de un hipercubo
  - Supuestos
    - Regiones tienen volumen $1/n_i$
    - Regiones son aproximadamente hipercubos
  - Valor estimado del largo a_i de una región en el nivel *i* del índice:

$$a_i = \sqrt[d]{\frac{1}{n_i}}$$

  - Número de accesos:

$$\#\ \text{accesos}(\varepsilon) = \sum_i n_i \cdot V_{\text{Mink}}\left(\sqrt[d]{\frac{1}{n_i}}, \varepsilon\right)$$

9.1.2 Modelo de costo para consultas por vecino más cercano

- Supuestos
  - Las búsquedas se realizarán con el algoritmo de Hjaltason y Samet [HS95]
  - Permite utilizar la propiedad de optimalidad de este algoritmo
  - El rendimiento de otros algoritmos de búsqueda del NN dependen del camino que siguen según la heurística utilizada
    - Difícil de estimar

19

9.1.2 Modelo de costo para consultas por vecino más cercano

- Consecuencias de la optimalidad de HS
  - El algoritmo accesa aquellas páginas que intersectan la bola definida por el NN
  - El algoritmo es equivalente a una búsqueda por rango con ε = distancia al NN
  - En principio basta con estimar la distancia al NN y ocupar este valor con el modelo estudiado para consultas por rango

20

9.1.2 Modelo de costo para consultas por vecino más cercano

- Método fácil para estimar la distancia al NN
  - El volumen de una esfera de radio ε multiplicado por *N* corresponde al valor esperado de puntos contenidos en ella
  - Estimar el radio de la esfera de modo que el valor esperado sea 1 (o *k* para el caso de *k*-NN)

$$\varepsilon \approx \sqrt[d]{\frac{k \cdot \Gamma\left(\frac{d}{2}+1\right)}{N}} \cdot \frac{1}{\sqrt{\pi}}$$

21

9.1.2 Modelo de costo para consultas por vecino más cercano

- Problemas con este método
  - No es correcto desde el punto de vista estocástico
    - Operación de construir un valor esperado no es invertible
  - Incluso bajo condiciones ideales no es muy exacto para *k* pequeño
- Ventajas
  - Fácil de calcular
  - Para *k*>10 puede ser suficientemente exacto

22

9.1.2 Modelo de costo para consultas por vecino más cercano

- Estimación exacta de la distancia al NN
  - Determinación de una función de distribución de la distancia al NN
  - Determinar la función de densidad probabilística correspondiente
  - Derivar el valor esperado mediante integración

23

9.1.2 Modelo de costo para consultas por vecino más cercano

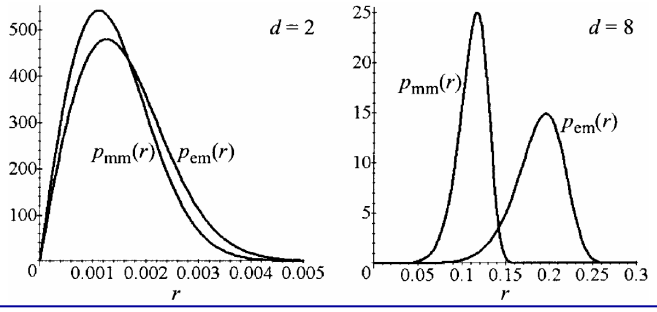
- Función de distribución *P*(*r*)
  - ¿Cuál es la probabilidad que la distancia al NN sea menor o igual que una variable estocástica *r*?
  - Ssi probabilidad que una esfera de radio *r* contenga al menos 1 punto
  - Ssi 1 – probabilidad que ninguno de los *N* puntos esté contenido en la esfera
  - Ssi 1– probabilidad que todos los puntos estén fuera de la esfera
  - Ssi $P(r) = 1 - \left(1 - V_{\text{esfera}}(r)\right)^N$

24

9.1.2 Modelo de costo para consultas por vecino más cercano

- Función de densidad de probabilidad

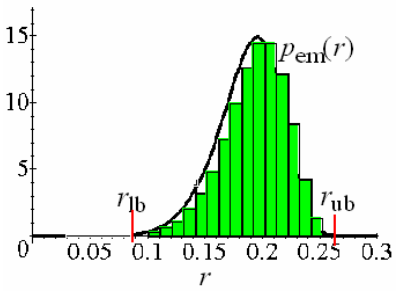
$$p(r) = \frac{\partial P(r)}{\partial r} = \frac{d \cdot N}{r} \cdot \left(1 - \frac{\sqrt{\pi}}{\Gamma\left(\frac{d}{2}+1\right)} \cdot r^d\right)^{N-1} \cdot \frac{\sqrt{\pi}}{\Gamma\left(\frac{d}{2}+1\right)} \cdot r^d$$



25

9.1.2 Modelo de costo para consultas por vecino más cercano

- Valor esperado: $E[\varepsilon] = \int_0^\infty r \cdot p(r) \partial r$
- Analíticamente difícil de calcular
- Integración numérica a través de histogramas
  - Trapezoides
  - Regla de Simpson



26

9.1.2 Modelo de costo para consultas por vecino más cercano

- Importante: determinar cotas donde la función de densidad probabilística tenga valores substancialmente mayores que 0
  - Ejemplo:
    - $P(r_{lb})$=0,001
    - $P(r_{ul})$=0,999
- Valor esperado aproximado

$$E[\varepsilon] = \int_{r_{lb}}^{r_{ub}} r \cdot p(r) \partial r = \frac{r_{ub} - r_{lb}}{i_{max}} \cdot \sum_{i=0}^{i_{max}-1} \left(\frac{r_{ub} - r_{lb}}{i_{max}} \cdot i + r_{lb} \right) \cdot p \left(\frac{r_{ub} - r_{lb}}{i_{max}} \cdot i + r_{lb} \right)$$

27

9.1.2 Modelo de costo para consultas por vecino más cercano

- Para obtener error relativo menor que 1%, basta con utilizar i_{max}=5 rectángulos
- Valor esperado de páginas accesadas
  - Variante simple:

$$\#\ \text{accesos} = \sum_i n_i \cdot V_{\text{Mink}} \left(\sqrt[d]{\frac{1}{n_i}}, E[\varepsilon] \right)$$

  - Variante exacta:

$$E[\#\ \text{accesos}] = \int_0^{\infty} \#\ \text{accesos}(r) \cdot p(r) \partial r$$

28

9.1.3 Efectos en espacios de alta dimensión

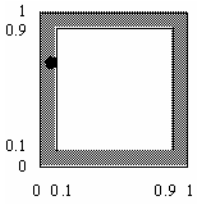
- El modelo de costo presentado supone que tanto la bola definida por el NN como la suma de Minkowski se encuentran completamente dentro del espacio
  - Para espacios de dimensión baja esto es correcto
  - Para espacios de dimensión alta
    - La superficie del espacio es tan grande que cerca de todo punto está el "afuera"
    - Existentes tantas direcciones que casi ningún punto está "adentro"
      - Los puntos se encuentran en la superficie del espacio

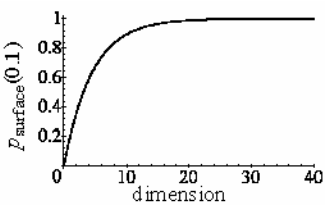
29

9.1.3 Efectos en espacios de alta dimensión

- Para el cubo unitario $[0,1]^d$ como espacio de datos esto significa:
  - La probabilidad que un punto esté a lo más a distancia δ de la superficie del espacio aumenta rápidamente con la dimensión

$$p_{surface}(\delta) = 1 - (1 - 2\delta)^d$$





30

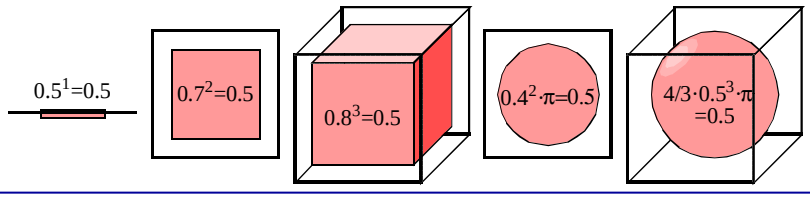
9.1.3 Efectos en espacios de alta dimensión

- Los siguientes valores no son pequeños en comparación con el largo del espacio (=1)
  - Radios típicos para consultas por rango
  - Distancia al *k*-NN
  - El largo de las regiones
- Razón
  - El largo es proporcional a la *d*-ésima raíz del volumen, cuyo valor es cercano a 1 para *d* grande

31

9.1.3 Efectos en espacios de alta dimensión

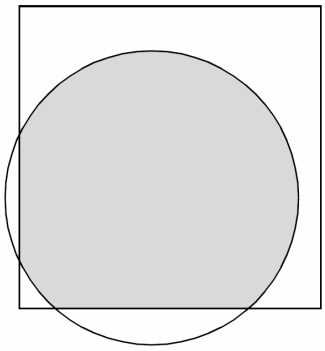
- Algunos valores proporcionales al volumen
  - Esfera *k*-NN: $V \approx \frac{k}{N}$ (espacio de datos normalizado)
  - Región espacial: $V \approx \frac{C_{eff}}{N}$
  - En cierto sentido, las esferas de consultas por rango con resultados significativos, cuando el conjunto retornado es no-trivial (todo o nada)



32

9.1.3 Efectos en espacios de alta dimensión

- Consecuencias para la bola de consulta
  - Con frecuencia su radio es cercano a 1 o sobrepasa a 1 considerablemente
  - La esfera traspasa en una o más partes la superficie del espacio
  - La parte de la esfera que queda fuera del espacio no aporta nada al resultado (*clipping*)



"Efectos de borde"

33

9.1.3 Efectos en espacios de alta dimensión

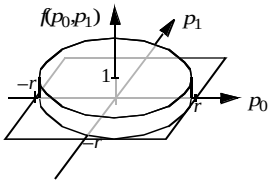
- Consecuencias para la bola de consulta
  - El volumen que queda fuera no se debe evaluar en el modelo de costo
    - Tanto para estimar la distancia al NN
    - Como para la suma de Minkowski
  - Se requiere una función de volumen que corrija el efecto de clipping
  - Como la esfera tiene que ocupar un volumen específico (e.g., k/N), su radio tiene que ser aún más grande para sopesar el efecto de clipping

34

9.1.3 Efectos en espacios de alta dimensión

- Consecuencias para la bola de consulta
  - Idea para determinar la función de volumen corregido
  - Integral del volumen de la esfera

$$V(r) = \int_{-r}^{r} \dots \int_{-r}^{r} \begin{cases} 1 & \text{si } |P| \leq r \\ 0 & \text{si no} \end{cases} \partial p_0 \partial p_1 \dots \partial p_{d-1}$$



  - Esfera en *Q*, "clipeada" en $[0,1]^d$

$$V(r) = \int_{0}^{1} \dots \int_{0}^{1} \begin{cases} 1 & \text{si } |P - Q| \leq r \\ 0 & \text{si no} \end{cases} \partial p_0 \partial p_1 \dots \partial p_{d-1}$$

35

9.1.3 Efectos en espacios de alta dimensión

- Consecuencias para la bola de consulta
  - Volumen promedio para algún punto *Q* (uniformemente distribuido)

$$V(r) = \int_{0}^{1} \dots \int_{0}^{1} \int_{0}^{1} \dots \int_{0}^{1} \begin{cases} 1 & \text{si } |P - Q| \leq r \\ 0 & \text{si no} \end{cases} \partial p_0 \partial p_1 \dots \partial p_{d-1} \partial q_0 \partial q_1 \dots \partial q_{d-1}$$

  - Integración analítica difícil: integración de Monte Carlo:
    - Elegir 1.000.000 de pares de puntos aleatorios
    - Estimar el volumen a partir de las frecuencias relativas obtenidas
  - El volumen se puede precalcular para
    - Todas las dimensiones relevantes
    - Una discretización de los radios relevantes $0 \leq r_i \leq r_{max} = \sqrt{d}$

36

9.1.3 Efectos en espacios de alta dimensión

- Consecuencias para los MBRs
  - Supuesto que las regiones serán “cubos” se torna irrealista
    - Las regiones se originan a partir de splits
    - El espacio se dividirá en 2 con respecto a una dimensión, las cuales contendrán la misma cantidad de puntos
  - Para dimensión suficientemente alta esto no es posible
    - Si cada dimensión se divide una vez se generan 2^d regiones (*d*=20 implica 1 millón de regiones, *d*=30 …)

37

9.1.3 Efectos en espacios de alta dimensión

- Consecuencias para los MBRs
  - Situación típica en espacios de dimensión alta
    - En muchas dimensiones no se realizó ningún split
    - En el resto de las dimensiones sólo se hizo un split, esto es aproximadamente [0..0,5] y [0,5..1]
    - El número de las dimensiones con split depende del número de páginas de datos

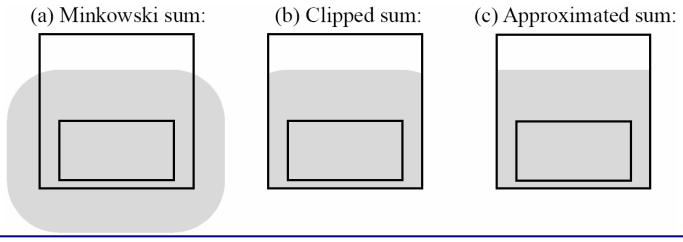
$$d' = \log\left(\frac{N}{C_{eff}}\right)$$

    - Si *d*<*d'* se utiliza el modelo de costo para dimensiones bajas

38

9.1.3 Efectos en espacios de alta dimensión

- Consecuencias para la suma de Minkowski
  - Gran parte de la suma de Minkowski sobresale del espacio de datos, debe ser “clipeada”
  - Si no es “clipeada”, el volumen de la suma de Minkowski puede exceder el volumen del espacio



39

9.1.3 Efectos en espacios de alta dimensión

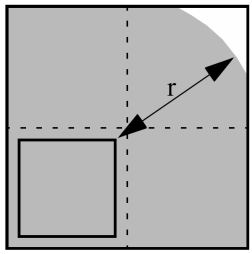
- Consecuencias para la suma de Minkowski
  - Sólo las dimensiones con split de las regiones se agrandarán según la suma de Minkowski, y sólo en un extremo, no en ambos
  - Fracciones de las esferas que son añadidas están sujetas a operaciones especiales de clipping si es que sobrepasan el radio ½

40

9.1.3 Efectos en espacios de alta dimensión

- Consecuencias para la suma de Minkowski
  - En este caso se utilizará también una función especial de volumen
    - Centro en la arista inferior izquierda del espacio de clipping

$$V(r) = \int_0^1 \cdots \int_0^1 \begin{cases} 1 & \text{si } |P| \leq r \\ 0 & \text{si no} \end{cases} \partial p_0 \partial p_1 \ldots \partial p_{d-1}$$



9.1.3 Efectos en espacios de alta dimensión

- Consecuencias para la suma de Minkowski
  - La función de volumen se precalcula y tabula como se mencionó anteriormente
  - Para la suma de Minkowski se obtiene la siguiente formula binomial

$$V_{\text{Mink}}(r) = \sum_{i=0}^{d-1} \binom{d'}{i} \cdot \left(\frac{1}{2}\right)^{d'-i} \cdot V_{\text{esfera clipeada } i\text{–dim}}(r)$$

9.1.3 Efectos en espacios de alta dimensión

- Implicancias de la dimensión alta sobre el rendimiento de los índices
  - Al aumentar la dimensión, aumenta considerablemente la probabilidad de acceso
  - La complejidad es exponencial en la dimensión, hasta que se produce un efecto de saturación (el número de páginas que se pueden leer está restringido)
  - Cuando se alcanza la saturación: búsqueda ya no es logarítmica sino levemente sublineal
    - Búsqueda secuencial puede ser mejor que un índice

43

9.2 Índices multidimensionales para espacios de dimensión alta

- Diseñados considerando las características de los espacios de dimensión alta
- Estudiaremos
  - X-tree [BKK96]
  - SS-tree [WJ96]
  - SR-tree [KS97]
  - TV-tree [LJF94]
  - VA-file [WSB98]

44

9.2.1 X-tree

- eXtended node tree [BKK96]
  - Adaptación del R*-tree para espacios de dimensión alta
  - El algoritmo de split del R-tree (R*-tree) conduce a un traslapamiento alto de las regiones en nodos del directorio en dimensión alta
    - Razón: existen pocas opciones (o sólo una) de split adecuadas en las páginas de directorio
      - En muchas dimensiones no han habido splits
      - Se utilizará una dimensión donde todos los hijos hayan sido divididos

45

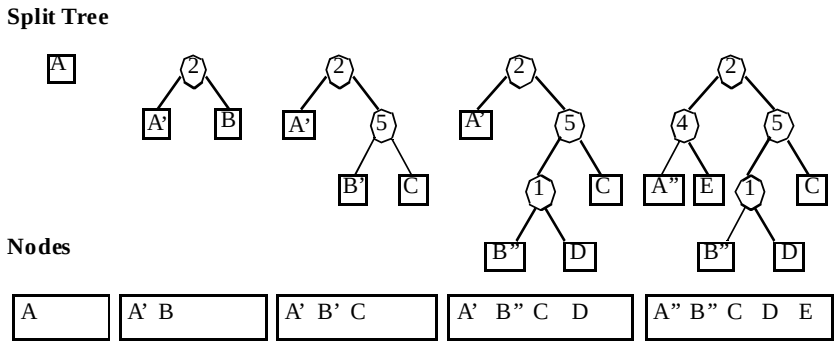
9.2.1 X-tree

- Concepto: split-tree de un nodo de directorio
  - Idea
    - Cada hoja del split-tree corresponde a un MBR del nodo
    - La jerarquía de operaciones de split se puede representar con un árbol binario
    - Los nodos internos del árbol se etiquetan con la dimensión que se utilizó para el split
    - Si d_i es un nodo ancestro de la hoja X, entonces X fue dividido en la dimensión d_i

46

9.2.1 X-tree

- Ejemplo de split-tree



9.2.1 X-tree

- Sólo cuando en un nivel del split-tree todas las dimensiones de split son idénticas, se puede dividir el nodo directorio en esa dimensión
  - Esto ocurre en general sólo en la raíz del árbol
- Ejemplo
  - Split en dim. 1: A'', C y E producirán traslape
  - Split en dim. 2: A'' y E en un grupo, B'', C y D en el otro grupo, no hay traslape

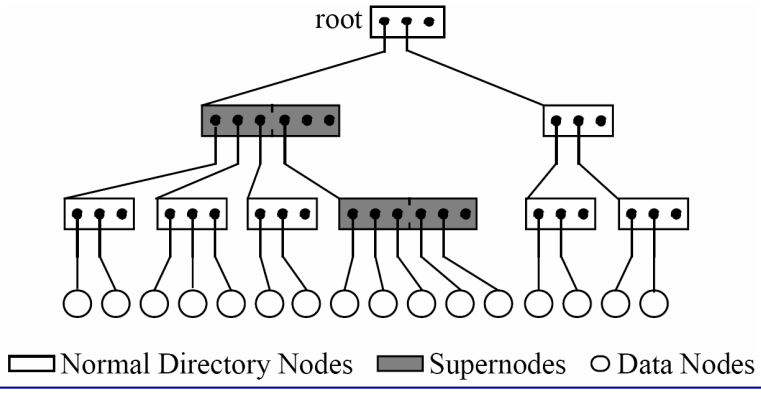
9.2.1 X-tree

- Problema adicional
  - Split-tree puede ser desbalanceado, lo que puede provocar un split desbalanceado
- Solución: supernodos
  - En este caso, se decide no realizar el split, sino que se agranda el nodo directorio
  - Mientras más alta la dimensión, es más probable que se creen supernodos

49

9.2.1 X-tree

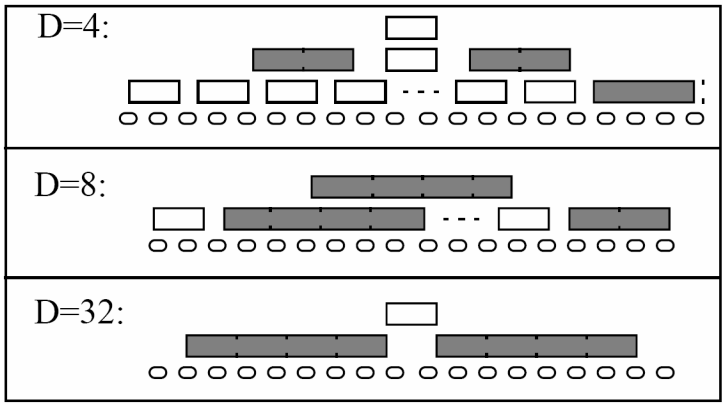
- Ejemplo de supernodos



50

9.2.1 X-tree

- X-tree para dimensiones crecientes



51

9.2.1 X-tree

- Desventajas del X-tree
  - Debido a que los nodos pueden tener ahora distinto tamaño, se complica el manejo del espacio libre en disco
  - Traslape no sólo se produce al realizar un split, sino que también al realizar inserciones
  - Se le quita flexibilidad al índice para adaptarse a la distribución de los puntos (en general sólo hay una dimensión disponible para el split)
  - Concepto de supernodo sólo se utiliza en nodos de directorio y sólo para aliviar el problema del desbalanceo

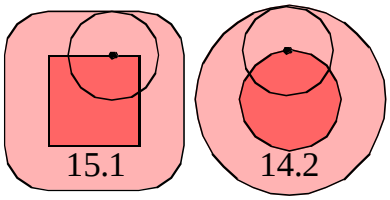
52

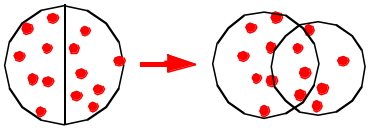
9.2.2 SS-tree

- Similarity search tree [WJ96]
  - En vez de MBRs se utilizan hiperesferas para definir las regiones espaciales
  - En principio, las esferas tienen ventajas con respecto a la probabilidad de acceso de una página, cuando las consultas también forman regiones esféricas
    - Para el mismo volumen de una región, la suma de Minkowski es menor en el caso de la esfera

9.2.2 SS-tree

- Comparación de la suma de Minkowski





Problema: split

9.2.2 SS-tree

- Split
  - Cuando se divide una esfera en dos, no se obtiene como resultado dos esferas
  - Se utiliza la esfera cubridora mínima para encerrar a los puntos de las regiones resultantes
    - Produce traslape alto
  - El centroide (centro de masa) de los puntos en la región se utiliza como centro de la esfera, y se busca el radio mínimo tal que todos los objetos de la región queden cubiertos

9.2.2 SS-tree

- Algoritmo de inserción
  - El objeto se inserta en el nodo hijo cuyo centroide tenga la mínima distancia al nuevo punto
- Manejo de overflow
  - Se reinserta un 30% de los puntos (los más lejanos al centroide)
- Criterio de split
  - La dimensión de split es aquella que tenga la mayor varianza
  - Para la posición de corte se prueban todas las posibilidades que garantizan la utilización mínima de espacio
  - El punto de corte minimiza la suma de las varianzas de ambos nodos resultantes

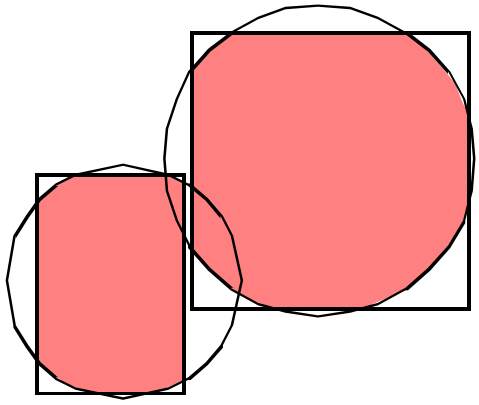
9.2.3 SR-tree

- Sphere-rectangle tree [KS97]
  - Utiliza la combinación (intersección) de un MBR y una esfera como región espacial
  - Busca combinar las ventajas de ambos métodos
    - Menor probabilidad de acceso de una esfera
    - Mejor “particionabilidad” de un MBR
  - Experimentalmente se muestra un mejor rendimiento que el SS-tree

57

9.2.3 SR-tree

- Ejemplo de región espacial en un SR-tree



58

9.2.4 TV-tree

- Telescope vector tree [LJF94]
  - Especialmente diseñado para vectores donde la importancia de las dimensiones está ordenada (e.g., mediante PCA)
  - Características de estos vectores
    - Varianza alta en las primeras dimensiones
    - Varianza baja en las últimas dimensiones
  - Útil para puntos con valores discretos en las coordenadas

9.2.4 TV-tree

- Una página del índice (directorio o datos) diferencia entre 3 tipos de dimensiones
  - Activas
  - Inactivas
  - Otras
- Una dimensión *inactiva* en un página fue *activa* en el pasado (con respecto a la jerarquía del árbol)
- El número α de dimensiones activas por página es constante en todo el índice
  - Parámetro del sistema
  - Experimentalmente se muestra que α=2 es óptimo

9.2.4 TV-tree

- Para cada región espacial solo se especifican las α dimensiones activas
  - En estas dimensiones, la región tiene la forma de una esfera L_p (p en {0, 1, máx})
- La región hereda de las dimensiones inactivas sólo la forma del antecesor (valor en esa coordenada es igual al padre)
- Las otras dimensiones no están especificadas

61

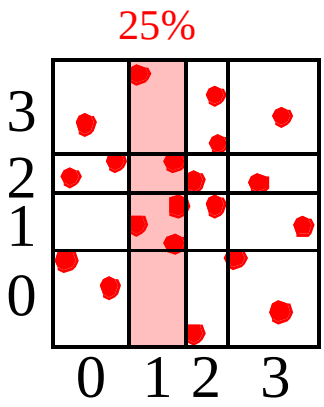
9.2.4 VA-file

- Vector approximation file [WSB98]
  - No es un índice jerárquico, sino una variante de la búsqueda secuencial
  - Todos los puntos son considerados
  - Inserción fácil: añadir al final
  - Los vectores son comprimidos (con pérdida)
    - Superponer a los datos una grilla no regular (basada en cuantiles)
  - Distribuir los puntos en el espacio en 2^{dr} celdas
  - En vez de las coordenadas exactas, almacenar el número de la celda donde queda el punto
  - Reducción del volumen de datos a $r/32$ (para reales de 32 bits)
  - El tiempo de lectura de los datos se reduce al mismo factor

62

9.2.4 VA-file

- Ejemplo

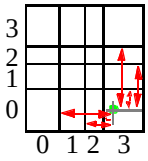


63

9.2.4 VA-file

- El tiempo de CPU para calcular las distancias se reduce con el siguiente truco
  - Las distancias entre consulta y cada línea de partición (por cada dimensión) se precalculan y tabulan
    - Se simplifica el cálculo de la distancia entre consulta y celda

$$\delta^2(p,q) = \sum_{i=1}^{d} (p_i - q_i)^2;\ \delta(p,q)^2_{VA} = \sum_{i=1}^{d} lookup_i[c_i]$$



64

9.2.4 VA-file

- Por cuantización de los vectores
  - Sólo si una celda se encuentra completamente dentro del rango de la consulta se puede garantizar un objeto relevante
  - Casi siempre conjunto de candidatos a NN
- Se necesita paso de refinamiento
  - Leer vector exacto (un acceso aleatorio por candidato), calcular distancia a la consulta
  - Los datos están almacenados en otro archivo con el mismo orden que el VA-file

9.2.4 VA-file

- Consultas por rango: cada celda intersectada debe ser leída y refinada
- *k*-NN: como en el caso de búsqueda en índices se pueden utilizar varias estrategias
- Para resoluciones muy bajas de la grilla aumenta el costo del refinamiento
  - Típicamente 6-8 bits es óptimo, pero depende de la distribución de los datos

9.3 Referencias

- [BKK96] S. Berchtold, D. Keim, and H.-P. Kriegel. The X-tree: An index structure for high-dimensional data. In *Proc. 22nd International Conference on Very Large Databases (VLDB'96)*, pages 28—39. Morgan Kaufmann, 1996
- [BBK+97] S. Berchtold, C. Böhm, D. Keim, and H.-P. Kriegel. A cost model for nearest neighbor search in high-dimensional spaces. In *Proc. ACM International Conference on Principles of Database Systems (PODS'97), pages 78—86*. ACM Press, 1997
- [BBK01] C. Böhm, S. Berchtold, and D. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys*, 33(3):322—373, 2001
- [Böh00] C. Böhm. A cost model for query processing in high dimensional spaces. *ACM Transactions on Database Systems*, 25(2):129—178, 2000
- [HS95] G. Hjaltason and H. Samet. Ranking in Spatial Databases. In *Proc. International Symposium on Large Spatial Databases (SSD'95), LNCS 951*, pages 83—95. Springer-Verlag, 1995
- [KS97] N. Katayama and S. Satoh. The SR-tree: An index structure for high-dimensional nearest neighbor queries. In *Proc. ACM International Conference on Management of Data (SIGMOD'97)*, pages 369—380. ACM Press, 1997
- [LJF94] K.-I. Lin, H. Jagadish, and C. Faloutsos. The TV-tree: An index structure for high-dimensional data. *The VLDB Journal*, 3(4):517—542, 1994
- [WJ96] D. White and R. Jain. Similarity indexing with the SS-tree. In *Proc. 12th International Conference on Data Engineering (ICDE'96)*, pages 516—523. IEEE CS Press, 1996
- [WSB98] R. Weber, H.-J. Scheck, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In Proc. 24th International Conference on Very Large Databases (VLDB'98), pages 194—205. Morgan Kaufmann, 1998