Ludwig Maximilians Universität München
Institut für Informatik
Lehr- und Forschungseinheit für Datenbanksysteme

Skript zur Vorlesung

Datenbanksysteme II

Sommersemester 2005

Kapitel 8: Hochdimensionale Räume

Vorlesung: Christian Böhm
Übungen: Elke Achtert, Peter Kunath

http://www.dbs.informatik.uni-muenchen.de/Lehre/DBSII

Datenbanksysteme II
Kapitel 8: Hochdimensionale Räume

Inhalt

Einführung

- Anfrageleistung von Indexstrukturen verschlechtert sich mit zunehmender Dimension
  "Curse of Dimensionality"
  → Häufig haben scanbasierte Methoden bessere Anfrage-Performanz als z.B. R*-Bäume
- In diesem Kapitel:
  - Ermittlung der Ursachen mit Hilfe eines Kostenmodells
  - Optimierung der Indexstrukturen sowie der Algorithmen zur Anfragebearbeitung
  - Entwicklung neuer Indexstrukturen, die besonders an die Problemstellung hochdimensionaler Datenräume angepasst sind

Inhalt

Kostenmodell für R-Bäume

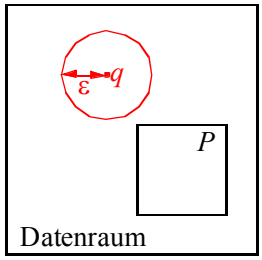
([Berchtold S., Böhm C., Keim D., Kriegel H.-P.: *A Cost Model for Nearest Neighbor Search in High-Dimensional Data Spaces,* PODS 1997])

Ziel: Schätzung der zu erwartenden Anzahl der Seitenzugriffe bei der Anfragebearbeitung

- für R-Bäume und verwandte Indexstrukturen
- verschiedene Anfragetypen:
  - Bereichsanfragen
  - nächste-Nachbar-Anfragen sowie *k*-nächste-Nachbar-Anfragen
  - verschiedene Metriken, hier nur euklidische und Maximums-Metrik (Ellipsoid-Anfragen sind schwierig zu modellieren)
- Einschränkungen:
  - idealisierte Indexstruktur:
    überlappungsfrei
    Seitenregionen sind annähernd quadratisch (“so quadratisch wie möglich”)
  - Datenraum ist der Einheits-Hypercube $[0..1]^d$
  - *zunächst: Punkte und Anfragen folgen einer unabhängigen Gleichverteilung*
  - *später: Beschreibung der Datenverteilung durch fraktale Dimension*
    (genauere Darstellungsmethoden der Datenverteilung wie z.B. Histogramme sind im Hochdimensionalen schwierig)

Kostenmodell Bereichsanfragen (1)

- Bekannt:
  - Radius ε der Anfrage
  - Ausdehung der Seitenregion
    → später wird beides geschätzt werden
- Unbekannt:
  - relative Lage von Seitenregion und Zentrum der Anfrage (*Anfragepunkt*)
  - beides wird als unabhängig gleichverteilt angenommen, d.h. jede Position von Anfragepunkt und Seitenregion ist gleich wahrscheinlich
- Gesucht:
  - Wahrscheinlichkeit, mit der die Query *q* auf die Seite *P* zugreift (Zugriffswahrscheinlichkeit)
  - entspricht Wahrscheinlichkeit, mit der sich der Kreis mit der Seitenregion schneidet



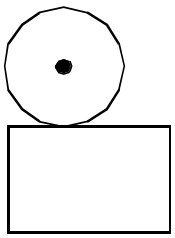
Kostenmodell Bereichsanfragen (2)

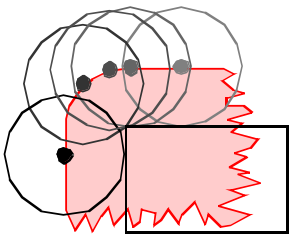
- Das Problem ist leicht zu lösen, wenn z.B. die Anfrage punktförmig ist:

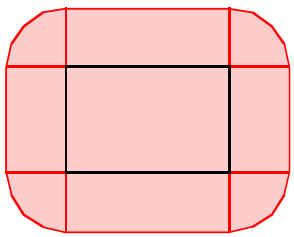
$$\text{Zugriffswahrscheinlichkeit} = \frac{\text{Volumen Seitenregion}}{\text{Volumen Datenraum}}$$

- ebenso bei punktförmiger Seitenregion
  → Wahrscheinlichkeitsrechnung (Kombinatorik) benötigt *punktförmige* Ereignisse
- Trick um punktförmige Ereignisse zu erhalten:
  Transformiere Bereichsanfrage in eine äquivalente Punktanfrage:
  - Verkleinere die Bereichsanfrage zum Punkt
  - Vergrößere in gleichem Maß die Seitenregion
  - so dass die neue Punktanfrage auf die vergrößerte Seitenregion zugreift gdw. die Bereichsanfrage auf die ursprüngliche Seitenregion zugreift

Kostenmodell Bereichsanfragen (3)



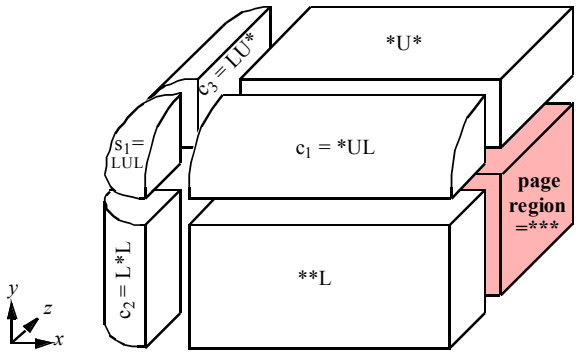




- Das entstehende Objekt heißt *Minkowski-Summe* von Anfrage- und Seitenregion:
  - ursprüngliches Rechteck
  - an jeder Kante ist ein Quader der Breite ε angehängt
  - an jeder Ecke ist ein Viertelkreis mit Radius ε angehängt

Kostenmodell Bereichsanfragen (4)

- Im dreidimensionalen Fall (Grafik unvollständig):



  - ursprünglicher Quader: 3-dimensional rechteckig, 0-dimensional rund
  - an jeder Oberfläche: Quader mit Grundfläche wie Oberfläche, Dicke ε
  - an jeder Kante: ¼ Zylinder: Länge wie Kante, Grundfläche ist Kreis mit Radius ε
  - an jeder Ecke: ⅛ Kugel mit Radius ε

Kostenmodell Bereichsanfragen (5)

- Ein *d*-dimensionaler *Hypercube* hat neben Ecken (0-dimensional), Kanten (1d) und Flächen (2d) auch noch 3-dimensionale, 4-dimensionale .... (*d* - 1)-dimensionale "Oberflächen"-Segmente (engl. *faces*)
  - an jedem *i*-dimensionalen Segment hängt ein Objekt ("*Hyperzylinder*"), das in *i* Dimensionen würfelförmig ist und in (*d* - *i*) Dimensionen kugelförmig ist (genau genommen der 2^{d-i}-te Teil einer solchen Hyperkugel)
- Wie viele Ecken, Flächen, ... *i*-dimensionale Segmente hat ein *d*-dimensionaler Würfel?
  Hierzu führen wir eine Notation ein, die, jedem Oberflächensegment (incl. dem urspr. Hypercube) ein *d*-Tupel über dem Alphabet aus den drei Symbolen L, U, und * zuordnet.
  Hierbei bedeutet:
  - L die untere Grenze (lower bound) in einer Dimension
  - U die obere Grenze (upper bound) in einer Dimension
  - * den gesamten Bereich zwischen der unteren Grenze und der oberen Grenze

Kostenmodell Bereichsanfragen (6)

- Beispiel:
  - Enthält ein Tupel kein *, so bezeichnet es eine *Ecke* (z.B. LUL die linke, obere, vordere, Ecke des dreidimensionalen Würfels)
  - Enthält ein Tupel genau ein *, so bezeichnet es eine *Kante* (z.B. LU* die Kante links oben, von vorne nach hinten)
  - Ein Tupel mit *i* Sternchen bezeichnet ein *i*-dimensionale Oberflächensegment
  - Das Tupel, das *nur* aus *d* Sternchen besteht, bezeichnet den originalen Hypercube
- Anzahl der Tupel mit *i* Sternchen:

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

$\binom{d}{i}$: verteile *i* Sternchen über *d* Positionen

2^{d-i}: fülle die verbleibenden *d-i* Positionen mit L oder U

Kostenmodell Bereichsanfragen (7)

- Gesamte Formel für das Volumen der Minkowski-Summe aus Hypercube mit Seitenlänge *a* und Kugel mit Radius ε:

$$V_{Mink}(a,\varepsilon) = \sum_{0 \le i \le d} \binom{d}{i} \cdot 2^{d-i} \cdot a^i \cdot \frac{V_{(d-i)-\dim.Kugel}(\varepsilon)}{2^{d-i}} = \sum_{0 \le i \le d} a^i \cdot V_{(d-i)-\dim.Kugel}(\varepsilon)$$

- Das Volumen einer *j*-dimensionalen Kugel lässt sich folgendermaßen ermitteln:

$$V_{j-\dim.Kugel} = \frac{\pi^{j/2} \cdot r^j}{\Gamma(j/2+1)}$$

wobei Γ die Gamma-Funktion (Erweiterung der Fakultät in reelle Zahlen) darstellt, mit:

$$\Gamma(x+1) = x \cdot \Gamma(x) \qquad \Gamma(1) = 1 \qquad \Gamma(1/2) = \sqrt{\pi}$$

- Mit V_{Mink} kann die Zugriffswahrscheinlichkeit einer einzelnen bekannten Seite bereits ermittelt werden. Dies werden wir später bei einer Optimierungstechnik anwenden.
- Im allgemeinen interessieren die Kosten für den gesamten Index; Summation der Zugriffswahrscheinlichkeiten aller Seitenregionen zu teuer

Kostenmodell Bereichsanfragen (7)

Schätzung der Seitenlänge des Hypercube

Benutze eine durchschnittliche Seite anstatt der konkreten Seiten.

Für jede Indexebene i läßt sich die Anzahl der Seiten n_i ermitteln, sofern die durchschnittliche Speicherauslastung (su_{eff}) bekannt ist (aus Data Dictionary):

Sei $C_{\text{eff}} := C \cdot su_{\text{eff}}$ *die effektive Kapazität der Seiten (durchschnittliche Anzahl in einer Seite gespeicherter Einträge),* N die Gesamtzahl von Featurevektoren

- $n_0 := N/C_{\text{eff,data}}$ (Anzahl der Datenseiten)
- $n_i := n_{i\text{-}1}/C_{\text{eff,dir}}$ (Anzahl der Seiten auf Directory-Ebene i)

Annahmen:

- Seitenregionen haben Volumen $1/n_i$ (1 ist das Volumen des Datenraums $[0..1]^d$)
- Seitenregionen sind annähernd (hyper-) würfelförmig.

Schätzwert für die Kantenlänge a_i einer Seitenregion auf Indexebene i: $a_i = \sqrt[d]{1/n_i}$

Kosten durch Addition der Zugriffswahrscheinlichkeiten aller Seiten auf allen Ebenen: $\#\text{Zugriffe}(\varepsilon) = \sum_i n_i \cdot V_{Mink}(\sqrt[d]{1/n_i}, \varepsilon)$

Kostenmodell NN-Anfragen (1)

Annahme:

- Die Anfragen werden mit dem Prioritätsalgorithmus nach Hjaltason und Samet [HS 95] bearbeitet.
- Dies ermöglicht Ausnutzung der Optimalität der Anfragebearbeitung
- Die Performanz der anderen NN-Algorithmen hängt u.a. davon ab, welcher Pfad als erstes verfolgt wird und ist deswegen schwerer zu schätzen.

Konsequenz aus dem Optimalitätsbeweis (Lemma 3):

- Der Algorithmus lädt genau die Seiten, die die Nearest-Neighbor-Kugel schneiden
- Der Algorithmus ist äquivalent zu Algorithmus für Bereichsanfragen, wobei ε durch die Nearest-Neighbor-Distanz ersetzt wird.
- Es reicht also prinzipiell, die NN-Distanz zu schätzen.

Kostenmodell NN-Anfragen (2)

Einfache Methode zur Schätzung der NN-Distanz

Das Volumen einer Kugel mit Radius e multipliziert mit N entspricht dem Erwartungswert der eingeschlossenen Datenpunkte. Bestimme die Kugel so daß der Erwartungswert 1 (bzw. k bei k-Nearest-Neighbor Queries) ist:

$$\varepsilon \approx V^{-1}_{d-\dim.Kugel}(k/N) = \sqrt[d]{\frac{k \cdot \Gamma(d/2+1)}{N \cdot \pi^{d/2}}}$$

Probleme:

- Stochastisch Vorgehensweise nicht korrekt (Operation "*Bildung des Erwartungswerts*" ist nicht umkehrbar)
- Auch unter Idealbedingungen (gleichverteilte Daten, niedrigdimensional) nicht sehr exakt für kleines k
- *Für großes* $k > 10$ aber hinreichend genau

Kostenmodell NN-Anfragen (3)

Stochastisch korrekte Ermittlung des Erwartungswerts:

- Ermittlung der Verteilungsfunktion der NN-Distanz
- Daraus: Wahrscheinlichkeitsdichtefunktion durch Differenzieren
- Daraus: Erwartungswert durch Integration

Verteilungsfunktion $P(r)$:

"Wie hoch ist Wahrscheinlichkeit, dass die NN-Distanz ≤ stochastische Variable r ist"

⇔

"Wahrscheinlichkeit, dass in einer Kugel mit Radius r mind. 1 Datenpunkt enthalten ist"

⇔

1 - "Wahrscheinlichkeit, dass *keiner* der N Datenpunkte im Volumen der Kugel liegt"

⇔

1 - "Wahrscheinlichkeit, dass *alle* Datenpunkte im Volumen außerhalb Kugel liegen"

⇔

$$P(r) = 1 - (1 - V_{Kugel}(r))^N$$

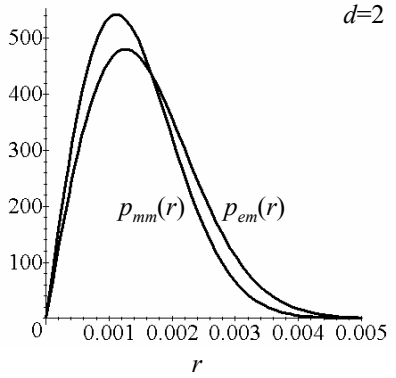
Kostenmodell NN-Anfragen (4)

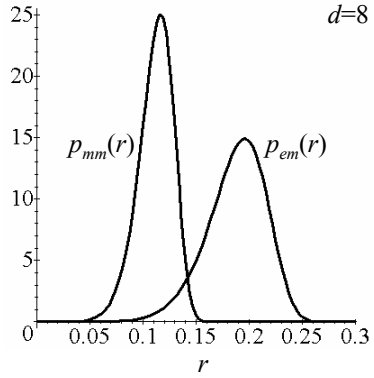
Wahrscheinlichkeitsdichtefunktion $p(r)$:

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

Maximumsmetrik: ähnlich.

Erwartungswert:

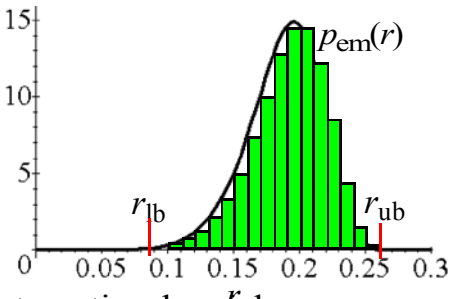




$$\mathrm{E}[\varepsilon] = \int_{-\infty}^{\infty} r \cdot p(r) \cdot \partial r = \int_{0}^{\infty} r \cdot p(r) \cdot \partial r$$

Kostenmodell NN-Anfragen (5)

- Analytisch schwierig zu integrieren
- Numerische Integration z.B. mittels Histogrammen:
  - links, rechts, oder mittig verankerte Rechtecke
  - Trapeze
- Wichtig:
  Erst untere r_{lb} und obere r_{ub} Grenze des Integrationsbereichs ermitteln, etwa so dass
  - $P(r_{\mathrm{lb}}) = 0{,}001$
  - $P(r_{\mathrm{ub}}) = 0{,}999$
- Damit ergibt sich für den Erwartungswert:



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

- Um einen relativen Approximationsfehler von < 1% zu erreichen, sind lediglich $i_{max} = 5$ Rechtecke erforderlich.

Inhalt

X-tree (1)

X-tree (eXtended node)

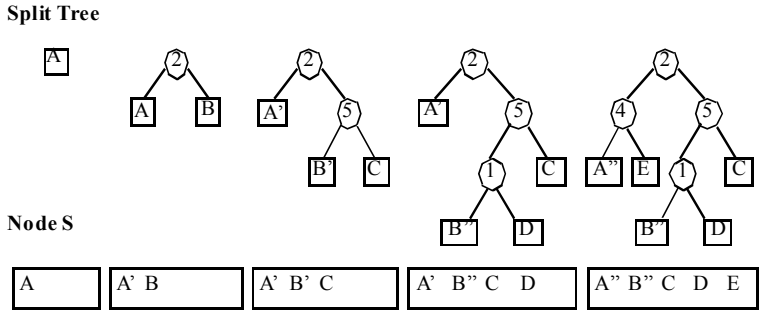
([BKK 96] Berchtold S., Keim D., Kriegel H.-P.: *The X-tree: An Index Structure for High-Dimensional Data*. VLDB 1996, 28-39.)

- Weiterentwicklung des R*-Baums für hochdimensionale Räume
- Der Split-Algorithmus von R-Baum und R*-Baum führt zu hoher Überlappung der Directory-Seitenregionen bei hochdimensionalen Räumen
- Grund: Es gibt nur wenige (meist eine) geeignete Splitebene bei Split der Directoryseite
  - Seiten sind in vielen Dimensionen ungeteilt (Ausdehnung [0..1])
  - Benötigt wird eine Dimension, in der alle Kindseiten geteilt wurden
- Konzept hierfür: Split-Tree eines Directory-Knoten

X-tree (2)

- Idee des Split-Tree:
  - Jede neue Seite entsteht dadurch, daß eine alte Seite aufgeteilt wurde
  - Die Hierarchie von Split-Operationen läßt sich als binärer Baum darstellen, wobei in den inneren Knoten die Split-Dimension vermerkt ist
  - Ist d_i ein Vorgängerknoten eines Blatts *X,* dann wurde *X* in Dimension d_i geteilt
- Nur wenn in einem Level des Split-Tree alle Split-Ebenen identisch sind, kann Directoryknoten in dieser Dimension geteilt werden
  - Dies ist meist nur bei der Wurzel des Splitbaums der Fall

X-tree (3)



- Beispiel: Directoryknoten wird (fälschlicherweise) in Dimension 1 geteilt
  - Die Seite B" kommt in die 1. Nachfolgeseite
  - Die Seite D kommt in die 2. Nachfolgeseite
  - Die Seiten A', C und E sind in Dimension 1 nicht gesplittet. Werden sie einer der beiden Nachfolgeseiten zugeordnet, so überlappt diese die andere Nachfolgerseite vollständig
- Directoryknoten wird (richtig) in Dimension 2 geteilt
  - A' und E kommen in die erste Nachfolgeseite
  - B", C und D kommen in die 2. Nachfolgeseite
  - Die beiden Nachfolgeseiten sind überlappungsfrei

X-tree (4)

- Weiteres Problem:
  Split-Tree kann unbalanciert sein => unbalancierter Split (z.B. nur eine Kindseite in der einen der Nachfolgeseiten)
- Lösung: Supernodes
  Für diesen Fall sieht der X-tree vor, die Directoryseite gar nicht zu splitten, sondern einen Supernode mit der doppelten Länge anzulegen
- Bei hohen Dimensionen entstehen deshalb zunehmend Supernodes

Nachteile des X-tree

- Durch Seiten unterschiedlicher Größe Probleme der Freispeicherverwaltung
- Überlappung entsteht nicht nur bei der Split-Operation sondern z.B. auch beim normalen Einfügen, wenn eine Seitenregion vergrößert werden muß
- Weniger Flexibilität der Struktur, um sich an die Datenverteilung anzupassen
  (meist steht nur eine Splitebene zur Auswahl)
- Konzept der Supernodes wird nur für Directory genutzt, und auch nur um unbalancierten Split zu verhindern.
  ⇒ Später: Wähle Blockgröße so, dass Anfragebearbeitung optimiert wird