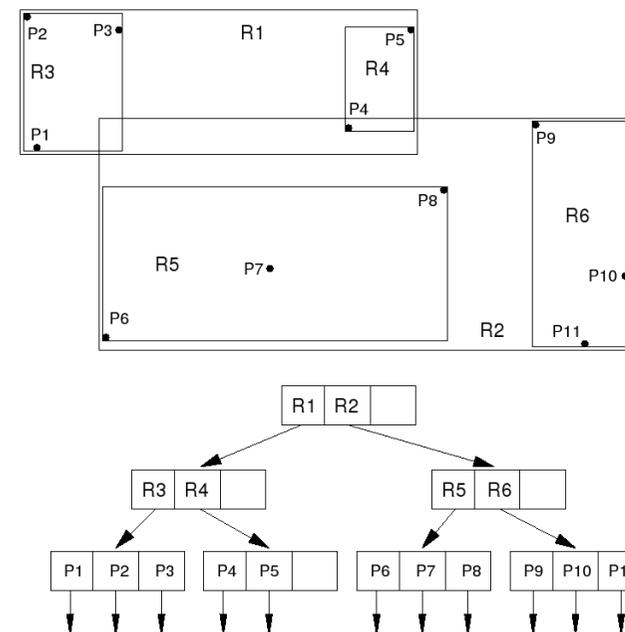


- R-Baum
- R⁺-Baum
- X-Baum
- M-Baum

- ♦ R-Baum: Guttman 1984
- ♦ Erweiterung B-Baum um mehrere Dimensionen
- ♦ Standardbaum zur Indexierung im niedrigdimensionalen Raum → Einsatz etwa in GIS
- ♦ Feature-Objekte können beliebige Ausdehnung haben
- ♦ Cluster-Bildung: lokale Gruppierung durch MBRs (Überlappungen erlaubt)
- ♦ balanciert und Feature-Objekte in den Blättern

Struktur:

- ♦ Mehrwegbaum
- ♦ Anzahl Kindknoten pro Knoten durch vordefinierten Minimal- bzw. Maximalwert eingeschränkt (Anpassung an Seitengröße)
- ♦ MBR eines Knoten umfasst minimal alle MBR der Kindknoten
- ♦ Blätter enthalten pro Feature-Objekt entsprechende Pointer und MBR
- ♦ Suchaufwand wächst mit Grad der Überlappung
→ Ziel: minimale Überlappung bei Baumkonstruktion



1. Finden eines geeigneten Blatts
2. Suche beginnt an der Wurzel
3. Navigation zum Kindknoten, dessen erforderliches Erweiterungsvolumen minimal ist
4. wenn Auswahlkriterium nicht eindeutig, dann Kindknoten mit minimalen Volumen bevorzugen
5. Blatt gefunden und Objekt eingefügt
→ Anpassung der MBRs der Vaterknoten

wenn Knoten zu viele Einträge besitzt

Zerlegung MBR in zwei MBRs

Variante 1:

- ♦ Ziel: Minimierung der Volumensumme der neuen MBR
→ Minimierung der Überlappungswahrscheinlichkeit

Variante 2:

- ♦ Zerlegung nach Idee aus R^{*}-Baum-Ansatz:
 - Zerlegung in genau einer Dimension
 - Aufwand zum Finden bester Zerlegung:
erschöpfende Suche
Punktzahl * Dimensionsanzahl

HS- und RKV-Algorithmus einsetzbar

bei Bereichsanfragen:

Navigation in jeden Unterbaum, dessen MBR den Suchbaum schneidet

NN-Suche ist effizient wenn Anzahl Dimensionen klein (etwa < 10)

bei vielen Dimensionen → viele MBR-Überlappungen

- ♦ Ausschließen von Teilbäumen von Suche wird extrem unwahrscheinlich
- ♦ Suchaufwand höher als bei sequentieller Suche

viele Varianten des R-Baums versuchen Problem zu lösen, können es aber nur verringern („curse of dimensionality“)

Sellig/Roussopolous/Faloutsos 1987

Grundidee: Verbot von Überlappungen

→ neuer Einfügealgorithmus

Forderung ist allerdings nur schwer erfüllbar

→ Einfügung kann Anpassung mehrerer Blätter erfordern

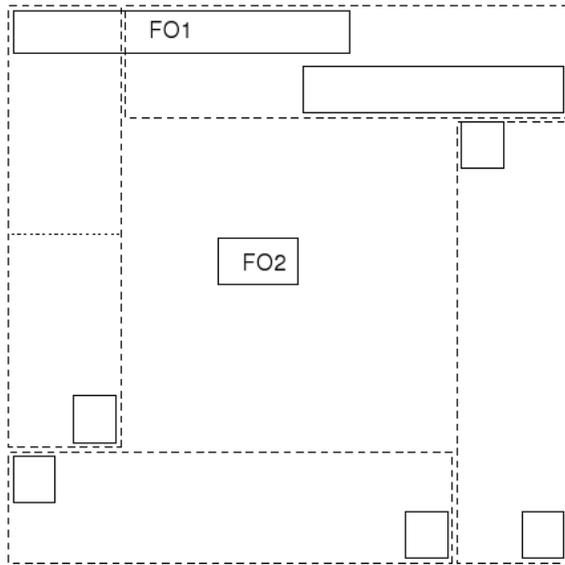
Anpassung kann Zerlegung in kleinere MBR ohne vorherigem Überlauf bedeuten

→ Knoten mit geringer Auslastung

→ viele Knoten (Entartung)

Feature-Objekte mit räumlicher Ausdehnung: u.U. kann kein umfassender MBR gefunden werden

→ Mehrfacheinträge nötig



Brechthold, Keim, Kriegel 1996

Idee basiert auf zwei Beobachtungen:

1. Effizienzprobleme im hochdimensionalen Raum aufgrund steigender Überlappungen
→ sequentieller Durchlauf ist effizienter als Baumdurchlauf
2. Zerlegung sollte an einer bestimmten Dimension erfolgen (auf Kosten der Balance!)

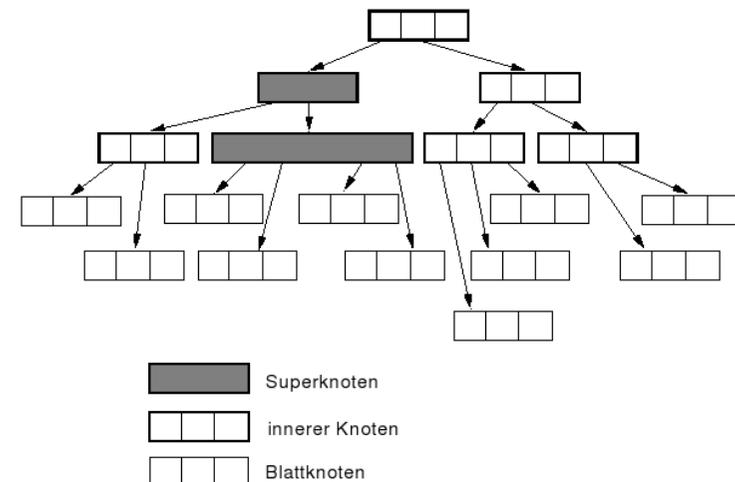
Einführung von Superknoten

Superknoten umfasst beliebig viele Datenbankseiten
(keine Einschränkung bzgl. Anzahl)

Suche im Superknoten erfolgt sequentiell

Superknoten werden dynamisch angelegt, wenn Grad an Überlappung zu hoch

Fazit: X-Baum als dynamische Hybridstruktur zwischen eindimensionalem Array und R-Baum



Split-Historie wird für jeden MBR verwaltet
enthält alle bereits verwendeten Zerlegungsdimensionen
bei Überlauf

1. Anwendung der herkömmlichen, topologischen Zerlegung
2. wenn vordefinierter Überlappungsgrad überschritten
→ Auswahl der Zerlegungsdimension anhand Split-Historie
3. wenn Verletzung der Balance
→ Erzeuge Superknoten

Ciaccia, Patella, Zezula 1997

Annahme bei Bäumen bis jetzt: Feature-Objekte sind Elemente
des euklidischen Vektorraums
→ Problem: Annahme gilt nicht immer

M-Baum setzt nur Metrik voraus:

- ♦ Menge von Feature-Objekten
(müssen keine Vektoren sein!)
- ♦ Distanzfunktionen
(muss nicht eukl. Distanz sein, z.B. Editdistanz zwischen
Wörtern)

M-Baum nutzt Dreiecksungleichung zum Ausschluss von
Teilbäumen von der Suche

Charakteristik:

Cluster-Bildung: lokale Gruppierung

Cluster können sich überlappen

Balance: M-Baum ist balanciert

Objektspeicherung: Verweise auf Feature-Objekte in den
Blättern

Geometrie: festgelegt durch Feature-Objekt (Zentrum) und
Distanz (Radius)
(entspricht Kugel im euklidischen Raum)

innerer Knoten:

durch vordefinierte Maximalanzahl an Seitengröße
angepasst

jeder Eintrag hat folgende Datenstruktur:

- ♦ Zeiger zum Kindknoten
- ♦ routing objekt o : Feature-Objekt als Kugelzentrum des
entspr. Kind-Clusters
- ♦ Radius r : maximal erlaubte Distanz vom routing object
zu Feature-Objekten des Kind-Clusters
- ♦ Distanz zum Vaterknoten: Distanz zwischen o und
routing object des Vaters

Blattknoten:

Verweis auf Feature-Objekt

Distanz des Feature-Objekts zum routing object des Vaterknotens

Bereichssuche und Nächste-Nachbarsuche entsprechen den eingeführten Algorithmen

Ausschlussbedingung anhand zweier Distanzen, welche minimale Distanz eines Clusters C mit routing object O , routing object des Vaters O_p und Radius r zu Feature-Objekt Q berechnen

1. angenäherte minimale Distanz (kann sehr schnell ermittelt werden):

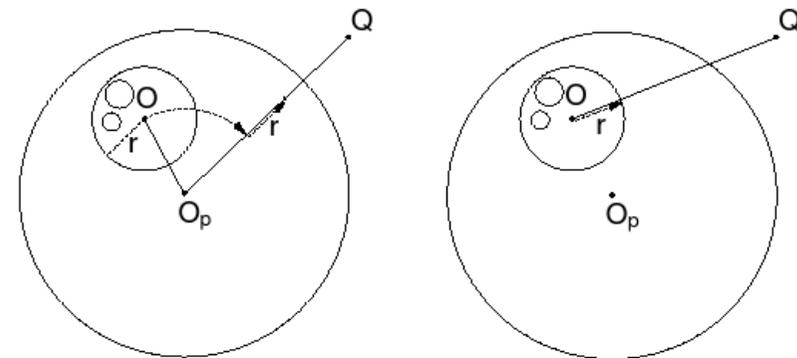
$$d_{min-approx}(Q, C) = \max(d(Q, O_p) - d(O_p, O) - r, 0)$$

2. minimale Distanz:

$$d_{min}(Q, C) = \max(d(Q, O) - r, 0)$$

3. maximale Distanz:

$$d_{max}(Q, C) = d(Q, O) + r$$



es gilt aufgrund Dreiecksungleichung:

$$d(Q, O_p) - d(O_p, O) - r \leq d(Q, O) - r$$

$$d(Q, O_p) \leq d(Q, O) + d(O_p, O)$$

Verwendung der angenäherten minimalen Distanz als schneller Filter

Ausschlussbedingung für Cluster: Ausnutzung der minimalen Distanz und der Dreiecksungleichung (siehe Zeilen 18 und 19 des Branch-and-Bound-Algorithmus)

Beginn bei der Wurzel

Navigation zum geeigneten Blatt anhand folgender Regeln:

- ♦ Vermeidung der Vergrößerung der Radien
- ♦ wenn mehrere Kinder ohne erforderliche Vergrößerung: Auswahl des nächsten routing objects
- ♦ falls alle Kinder Vergrößerung benötigen: Auswahl Kind mit minimaler Vergrößerung

nach Einfügung im Blatt: Anpassung der Radien anhand Pfad zur Wurzel

- ♦ Zerlegung des Clusters in zwei neue Cluster
- ♦ zwei neue routing objects müssen gefunden werden
- ♦ grundsätzlich sind routing objects immer Feature-Objekte
- ♦ verschiedene Strategien zum Finden neuer routing objects
 - ♦ Ziel: Minimierung Clustervolumen und Überlappungsvolumen
- ♦ nachdem neue routing objects gefunden, abwechselnde Zuordnung der jeweils nächsten Feature-Objekte

Fünf Strategien zum Finden neuer routing objects:

1. erschöpfende Suche und Radiensumme:
Suche Objektpaar mit minimaler Radiensumme
2. erschöpfende Suche und Radienmaxima:
Suche Objektpaar mit minimalen Maximalradien
3. am weitesten entfernte Objekte
4. nichtdeterministische Strategie (Zufall)
5. sampling: Auswahl Objektpaar mit minimalen Maximalradien aus Stichprobe