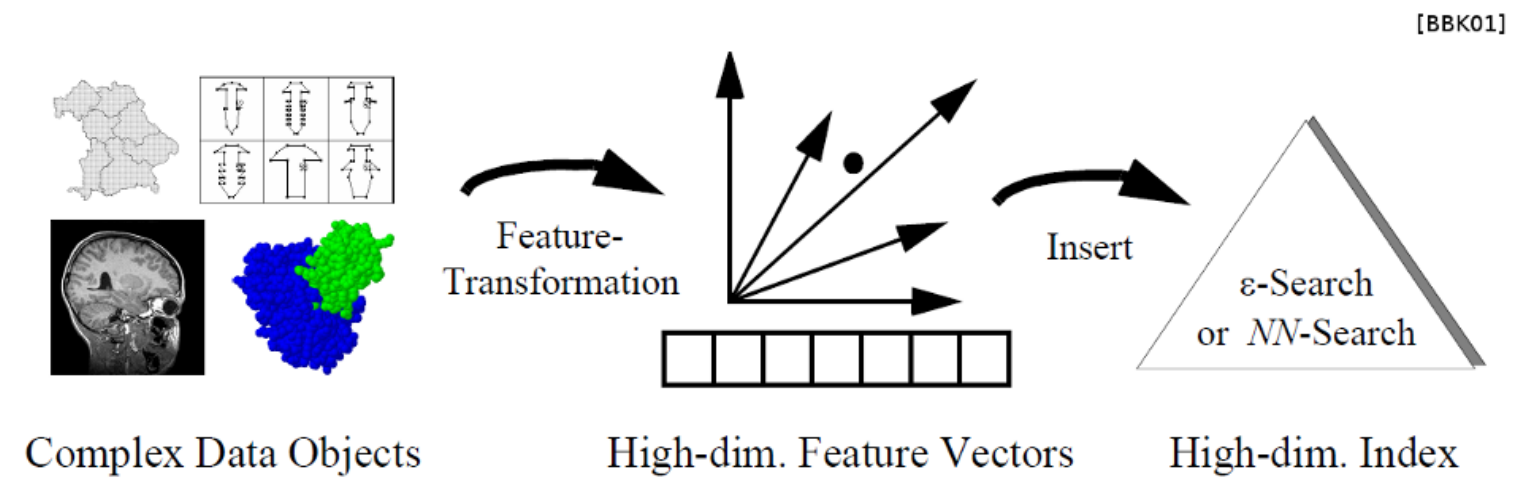
HOCHDIMENSIONALE DATEN

Leipzig, 03.07.2015

Paul Jähne

HOCHDIMENSIONALE DATEN

- ab wann? > 100 typisch
- Beispiele: Geoinformationssystem, CAD, Molekular-Biologie, Gen-Sequenzierung, psychoogische Profile, Data-Mining
- Ähnlichkeitsanfragen werden auf Nachbarschaftsanfragen in Feature-Räumen abgebildet



HOCHDIMENSIONALE FEATURE-VEKTOREN

- Modell für wichtige Eigenschaften eines Objekts
- Vektor mit Zahlenwert für feste Anzahl an Dimension
- bilden Vektorraum
- orthogonale Dimensionen
- Reduktion auf lineare Algebra
- Beispiele: Farb-Histogramm, Fourier-Vektor

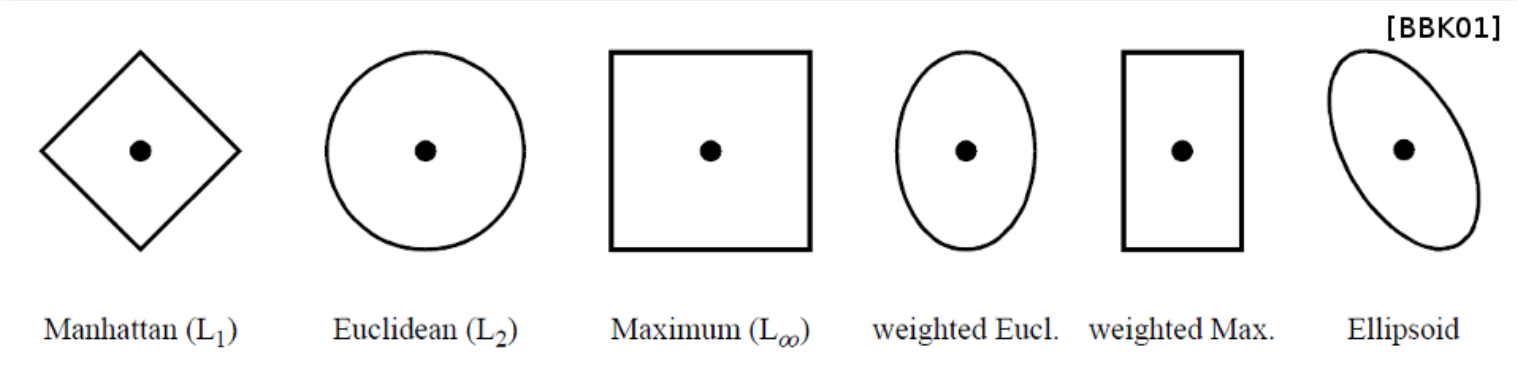
OPERATIONEN AUF FEATURE-VEKTOREN

- Nächste-Nachbarsuche (NN)
- k-Nächste-Nachbarsuche (kNN)
- Approximative Nächste-Nachbarsuche (aNN)
- Reverse-Nächste-Nachbarsuche (rNN)
- Bereichssuche
- Punktsuche
- Partial-Match-Suche
- Ähnlichkeitsverbund

METRIKEN FÜR ABSTÄNDE

Ähnlichkeit:

Unähnlichkeit: ,



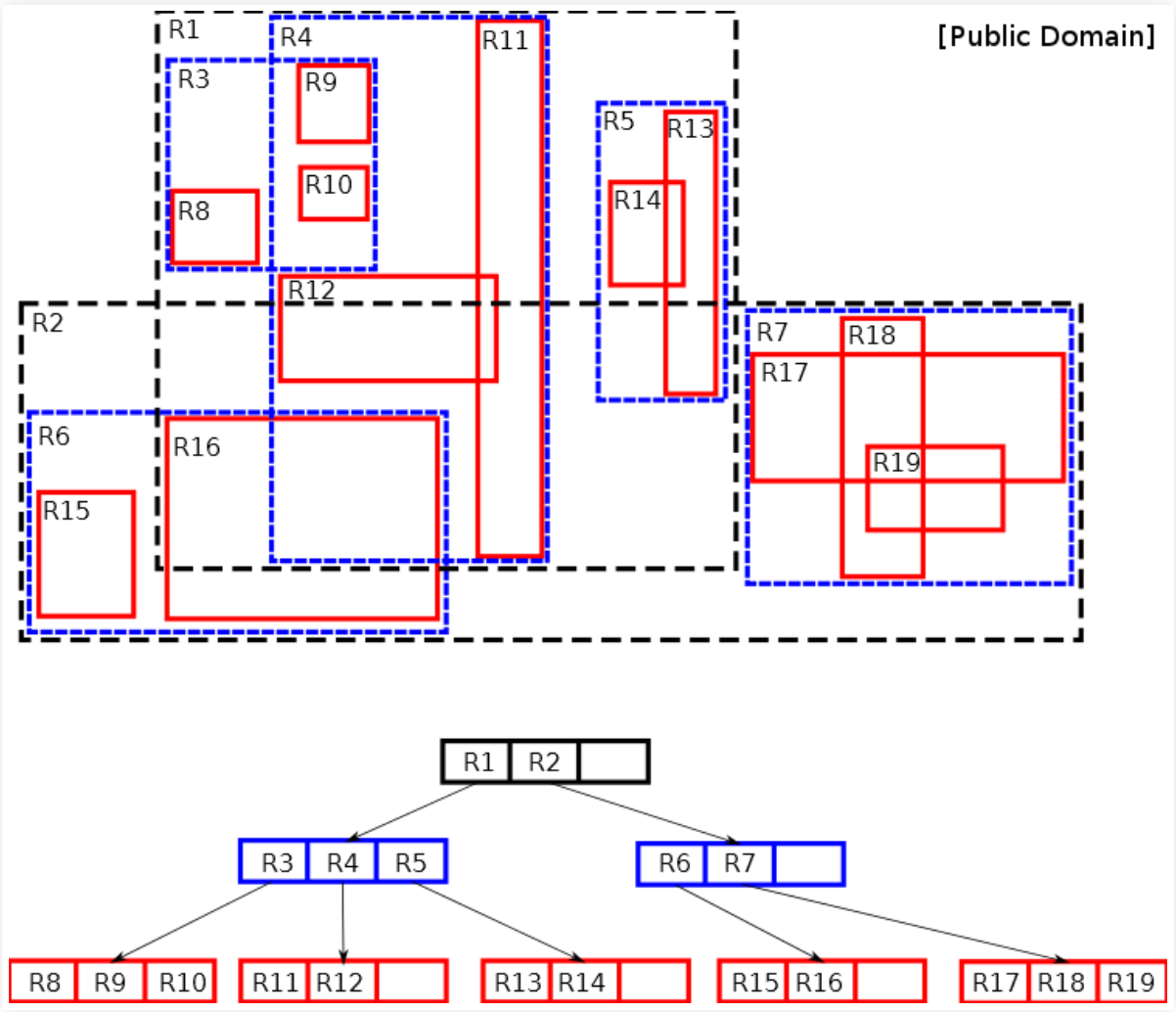
INDEXSTRUKTUREN

- B-Baum ungenügend
  - nur nach einer Dimension indexiert
  - exakte Suche

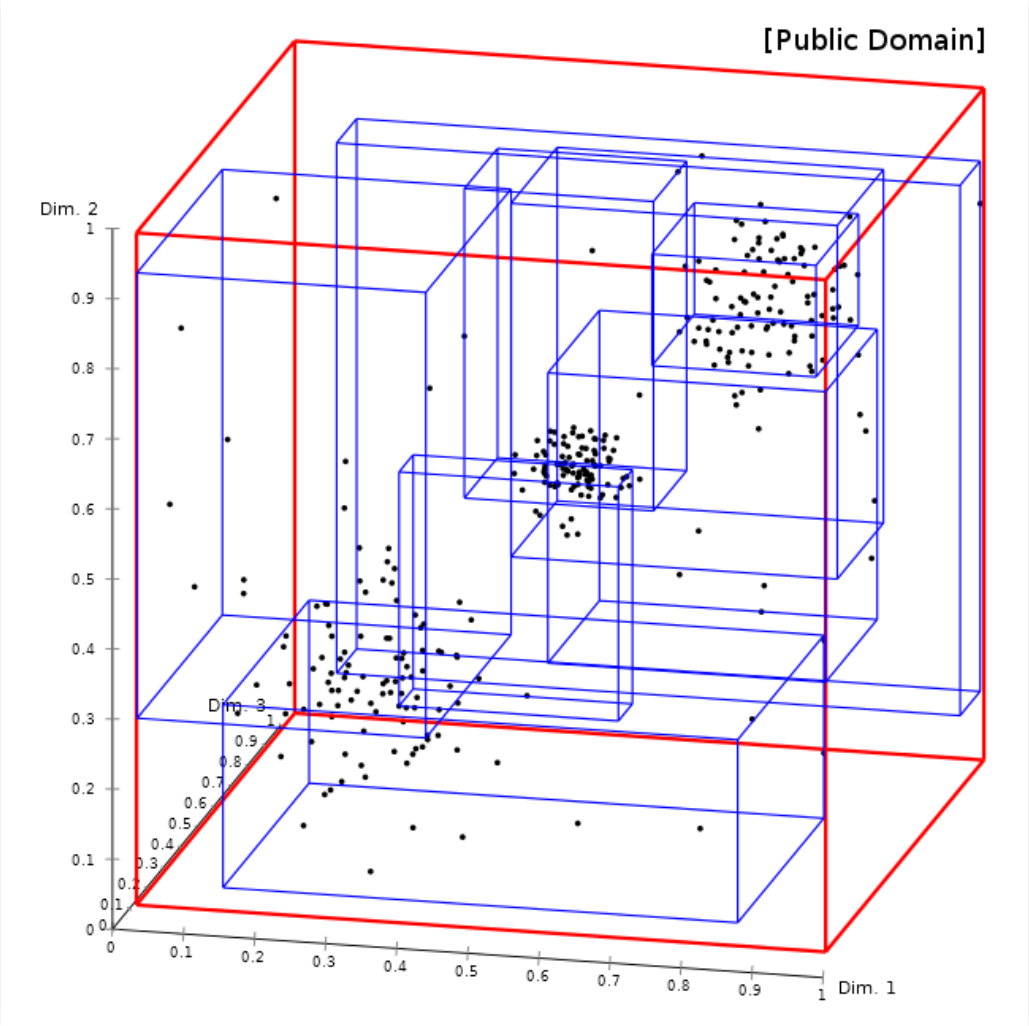
R-BAUM

- Rectangle
- B-Baum mit mehreren Dimensionen
- Clusterbildung mit umschließenden Rechtecken
- jeder Elternknoten umfasst mindestens alle rechtecke der Kindknoten
- für effiziente Suche minimale Überlappung benötigt

Hochdimensionale Daten



Hochdimensionale Daten



R-BAUM EINFÜGEN

- Suche von Knoten mit geringstem Erweiterungsvolumen
- falls gleich, dann kleinerer Knoten
- Blatt gefunden, dann einfügen und Vaterknoten anpassen
- Überlauf
  - Zerlegung in zwei Knoten
  - Aufteilung mit geringster Volumensumme wählen

R-BAUM SUCHE

- löst Nächster-Nachbar-Problem
- RKV-Algorithmus
  - Branch and Bound
  - Tiefensuche
  - kein getNext möglich
- HS-Algorithmus
  - Breitensuche
  - getNext möglich
  - benötigt viel Speicher

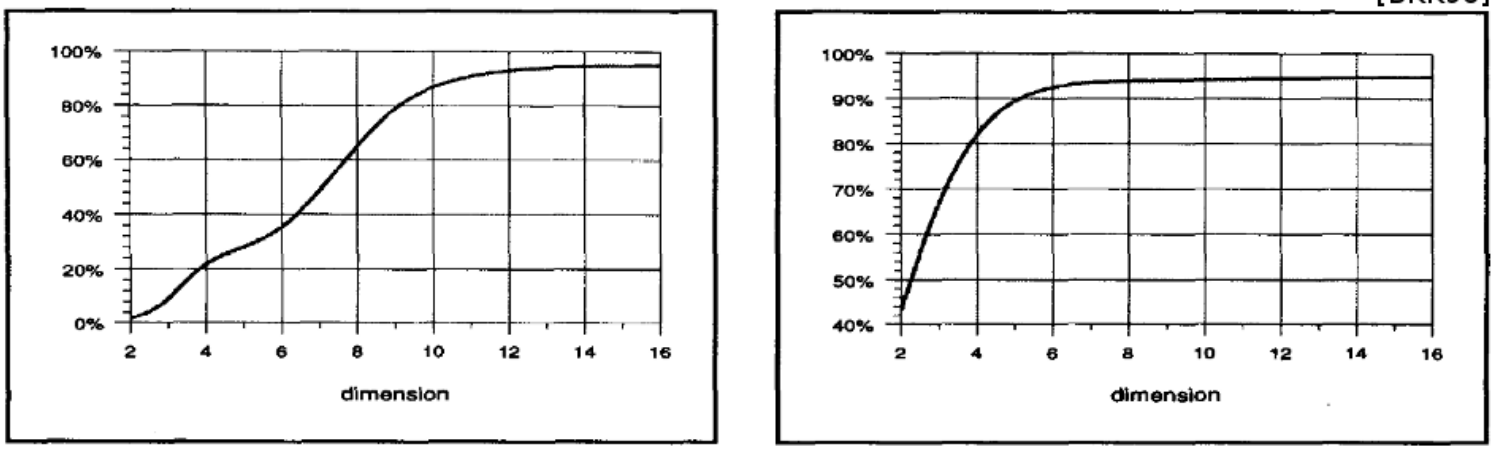
R-BAUM-VARIANTEN

- hohe Dimensionalität ergibt viele Überlappungen
- seltener Teilbäume ausschließbar
- verschiedene Varianten, um dies zu verbessern:
  - R+
    - Überlappungen verboten
    - neuer Einfügealgorithmus
    - dafür geringe Auslastung der Knoten → viele Knoten
  - R*
  - X

X-BAUM

- eXtended
- hybrid aus Array und R-Baum
- R-Baum-Split kann schlechte Strukturen erzeugen
  - viele Überlappungen
  - besonders bei hochdimensionalen Daten
  - viele Pfade müssen untersucht werden
  - gute Split-Strategie: viele Leerknoten

[BKK96]



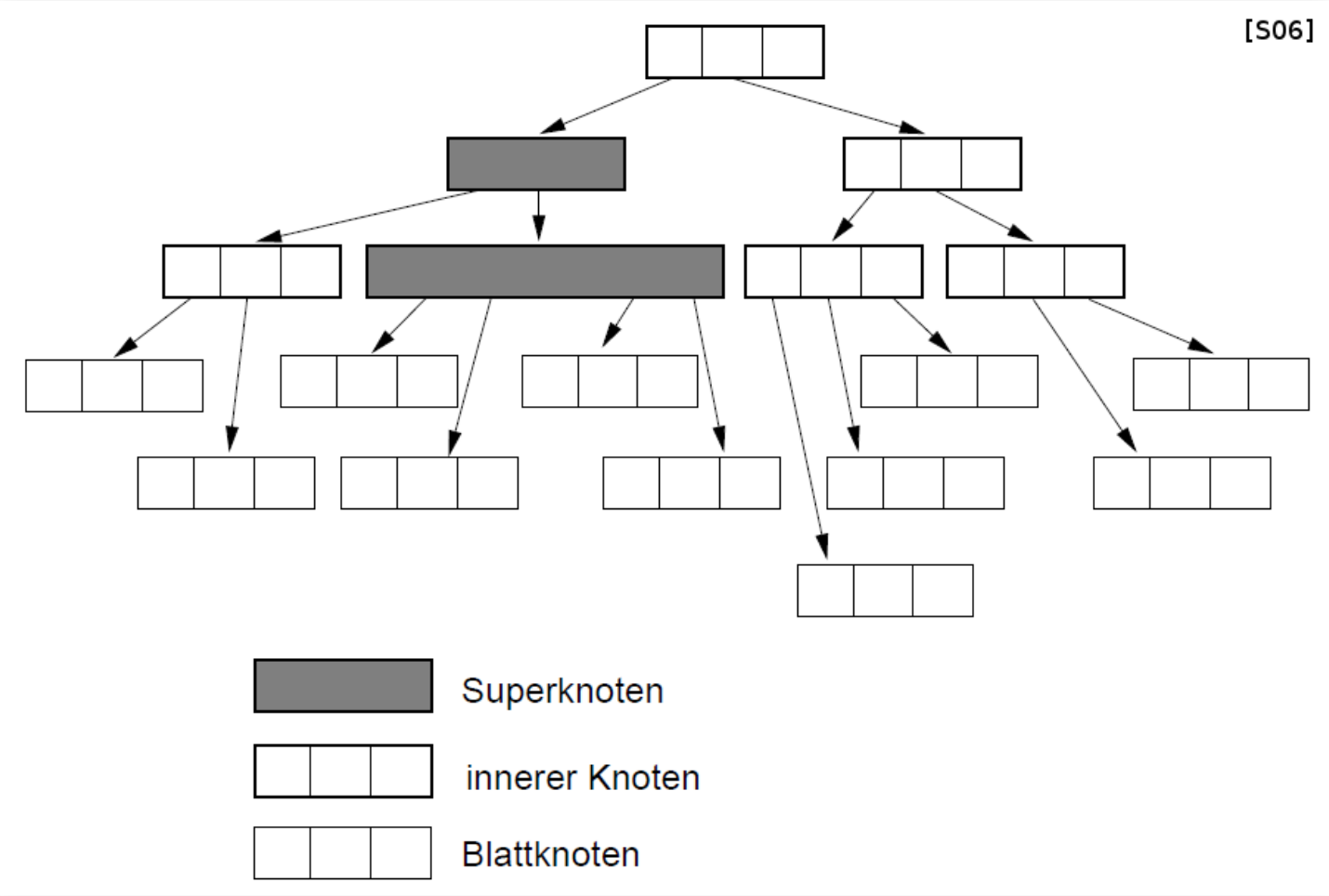
a. Overlap (Uniformly Distributed Data)

b. Weighted Overlap (Real Data)

Figure 2: Overlap of R*-tree Directory Nodes depending on the Dimensionality

X-BAUM

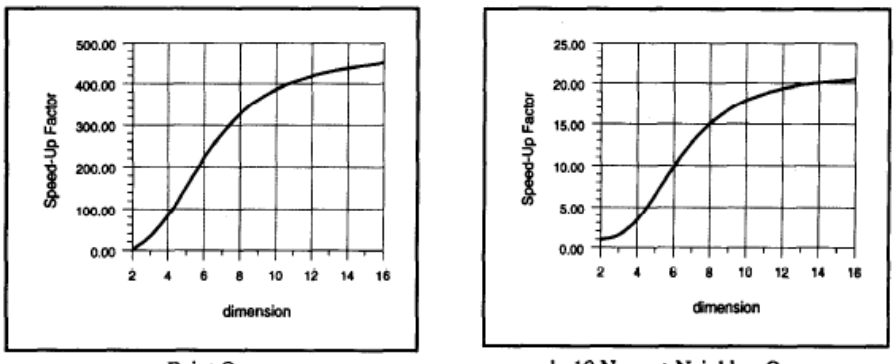
- R*-Erweiterung
- sequentieller Durchlauf bei hoher Überlappung effizienter
- überlappungsfreie Splitstrategie mit Split-Historie
- zusätzliche Superknoten
  - dynamisch angelegt bei hoher Überlappung
  - umfassen beliebig viele Seiten
  - Suche innerhalb sequentiell


[S06]
Superknoten
innerer Knoten
Blattknoten

X-BAUM

- Vorteile:
  - größerer Fanout
  - Scan von Superknoten eventuell schneller als Suche in R-Baum
  - Punktsuche bis zu 450x schneller als R*-Baum
  - NN-Suche bis zu 20x schneller als R*-Baum
  - Einfügen 8x schneler als R*
  - größere Knoten nur dort wo sie benötigt werden

[BKK96]



a. Point Query b. 10 Nearest-Neighbor Query

Figure 11: Speed-Up of X-tree over R*-tree on Real Point Data (70 MBytes)

ZUSAMMENFASSUNG

- hohe Dimensionalität vermeiden
- Modellierung mit Feature-Vektoren
- Anfragen häufig Nächste-Nachbar- oder Bereichssuche
- X-Bäume als Index gut geeignet

QUELLEN

Böhm, Brechtold, Keim: Searching in High-dimensional Spaces - Index Structures for Improving the Performance of Multimedia Databases, 2001

Böhm: Effiziente Indexstrukturen für hochdimensionale Datenräume, 1998

Brechtold, Keim, Kriegel: The X-tree: An Index Structure for High-Dimensional Data, 1996

Wikipedia: R-tree, https://en.wikipedia.org/wiki/R-tree, 2015

Schmitt: Multimedia-Datenbanken - 6 Effziente Algorithmen und Datenstrukturen, http://wwwiti.cs.uni-magdeburg.de/iti_db/lehre/mmdb/skripte/6.pdf, 2009