Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

Similarity Search Algorithms

Uwe Draisbach, Dustin Lange, Tobias Vogel,
Prof. Dr. Felix Naumann

Beispiele

- Versandhändler
- Bestellstatus per Telefon erfahren
  - Kundennummer?
  - Name?
  - Adresse?

Beispiele

Anfrage:

Name	PLZ	Ort
Hans Meier	14482	Potsdam

SQL-Query:

```
SELECT *
FROM Kunden
WHERE
    Name = „Hans Meier“
    AND PLZ = „14482“
    AND Ort = „Potsdam“
```

Datenbestand:

Name	PLZ	Ort
Hans Meyer	14482	Potsdam
Hans Meier	14482	Postdam
Hanns Mayer	14483	Potsdam

Ergebnis:

0 rows returned.

Beispiele

Google eiffel tower Search Advanced Search

SafeSearch: Moderate

Web › **Images** Show options... Results **1** - **20** of about **1,960,000** for **eiffel tower**. (**2.86** seconds)

Related searches: eiffel tower **drawing**

Strike forces **Eiffel**
400 x 327 - 33k - jpg
travelhouseuk...
Find similar images

arc-de-triomphe **eiffel**
500 x 666 - 61k - jpg
thetravelpeach.com
Find similar images

pic1.jpg
768 x 1024 - 95k - jpg
plone.scottsdalecc.edu
Find similar images

"We'll see the
1024 x 683 - 106k - jpg
songwriterstory...
Find similar images

Pictures of **Eiffel**
335 x 500 - 48k - jpg
world-city-photos.org
Find similar images

Eiffel Tower in
617 x 896 - 262k - jpg
visitingdc.com
Find similar images

Eiffel Tower
397 x 401 - 22k - jpg
atkielski.com
Find similar images

Eiffel Tower
552 x 700 - 39k - gif
etc.usf.edu
Find similar images

Eiffel Tower
1600 x 1200 - 164k - jpg
tourstravelandhotelb...
Find similar images

Eiffel Tower At
480 x 640 - 50k - jpg
earthinpictures.com
Find similar images

Beispiele

Google Search Advanced Search

SafeSearch: Moderate ▾

Similar images Results **1** - **20** of about **457** for **eiffel tower**. (**0.08** seconds)

« Back to results for **eiffel tower**


500 x 666 - 61k - jpg
neatorama.com
Find similar images


300 x 400 - 27k - jpg
kheimeh.org
Find similar images


298 x 400 - 33k - jpg
nastrorosa.altervista.org
Find similar images


600 x 800 - 73k - jpg
cinerama.unblog.fr
Find similar images


360 x 504 - 81k - jpg
treklens.com
Find similar images


960 x 1280 - 128k - jpg
kaliteliresimler.com
Find similar images


1200 x 1600 - 179k - jpg
everythingspossible...
Find similar images


480 x 640 - 69k - jpg
southafrica.to
Find similar images


450 x 600 - 31k - jpg
sehenswuerdigkeiten-der...
Find similar images


300 x 401 - 46k - jpg
memarblog.com
Find similar images

Gliederung

- Similarity Search

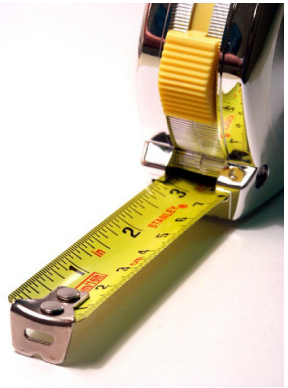
- Seminarinhalt
  - Datensätze
  - Indexierungsalgorithmen

- Organisatorisches

Ähnlichkeitssuche

(1) **Ähnlichkeitsmaß**

- Wie ähnlich sind 2 Objekte?
  Speziell: Wie ähnlich ist das Anfrageobjekt zu einem Objekt im Datenbestand?



(2) **Zugriffsmethoden**

- Wie können schnell alle zu einer Anfrage ähnlichen Objekte gefunden werden?

Ähnlichkeitsmaß

- Wie **ähnlich** sind 2 Objekte?
- Funktion:
  - $similarity : D \times D \rightarrow [0,1]$
  - $distance(a,b) = 1 - similarity(a,b)$
- Beispiele:
  - Strings
    - Edit-distance (Levenshtein, Jaro-Winkler, …)
    - Phonetische Distanz (Soundex, Kölner Phonetik, …)
  - Graphen
    - Tree Edit Distance
  - Text-Dokumente
    - N-gram
  - Bilder
    - Ähnlichkeit der Histogramme

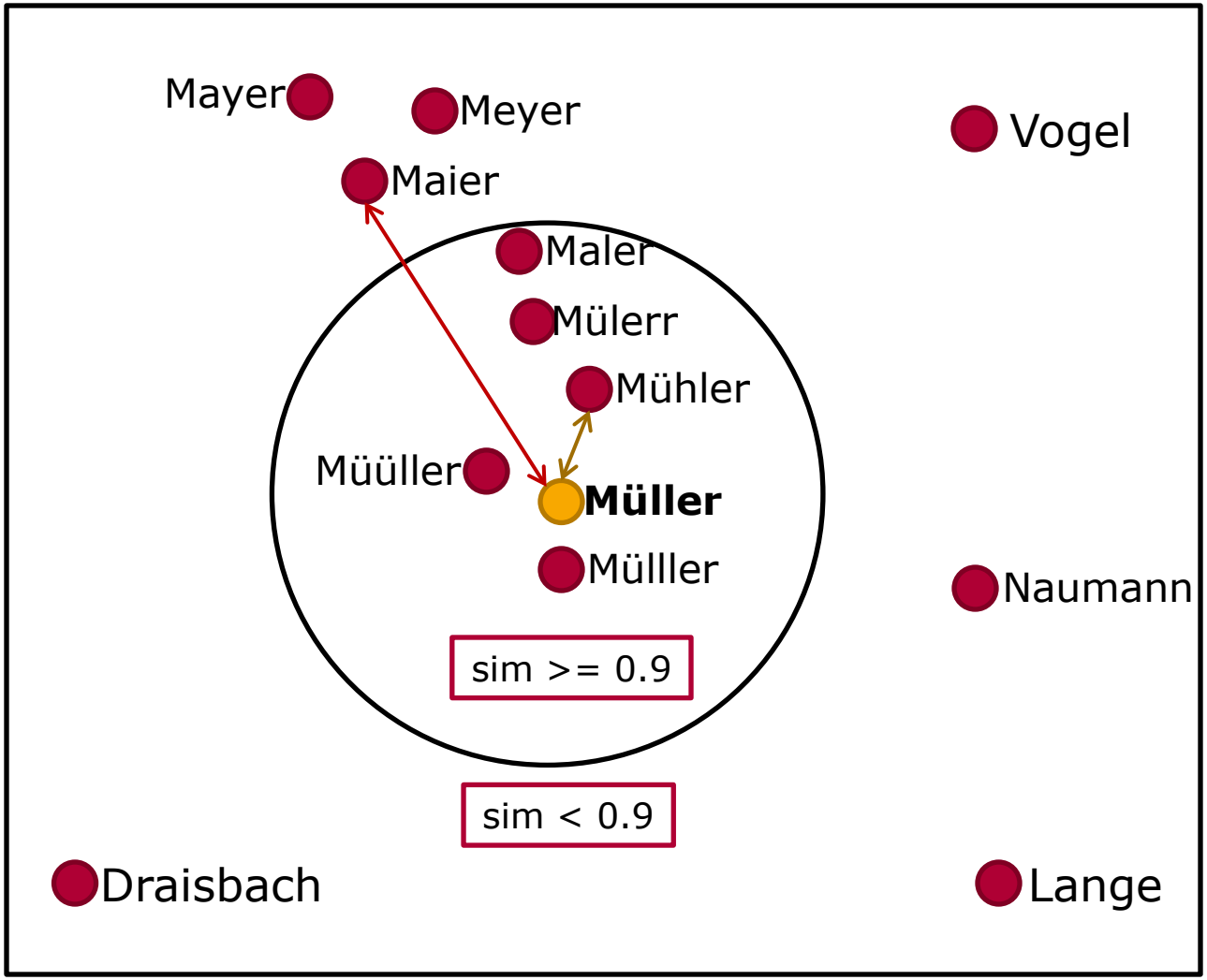
Zugriffsmethoden

- Geg. Datenbestand + Ähnlichkeitsmaß
- Wie finde ich **schnell** die richtigen Treffer zu einer Anfrage?
- Naiv: Vergleiche Anfrage mit allen Objekten im Datenbestand
- Besser: Vermeide möglichst viele Vergleiche
  - Verwende Index

Zugriffsmethoden

- Bekannte Zugriffsmethoden?
  - B-Baum
  - Weitere: B+-Baum, B*-Baum, AVL-Baum, R-Baum, 2-3-4-Baum, Rot-Schwarz-Baum
  - Nur für exakte Suche/Zahlen/Vektoren
- Wie suchen mit Ähnlichkeitsmaß?
- Idee: Partitioning

Ball Partitioning



Spaces

- Ähnlichkeitsmaß kann beliebig komplex werden
- Indexierungsmethoden setzen Grenzen für Ähnlichkeitsmaße

- **Vector Space**
  - Daten liegen als Vektoren vor
  - Ähnlichkeitsmaß häufig L_p-Metrik (z.B. Euklid. Distanz = L_2)
- **Metric Space**

Unser Fokus im Seminar

  - Daten beliebig
  - Ähnlichkeitsmaß muss **Metrik** sein
    - Symmetrie: s(a,b) = s(b,a)
    - Nichtnegativität: s(a,b) >= 0
    - Dreiecksungleichung: s(a,c) <= s(a,b) + s(b,c)

Search

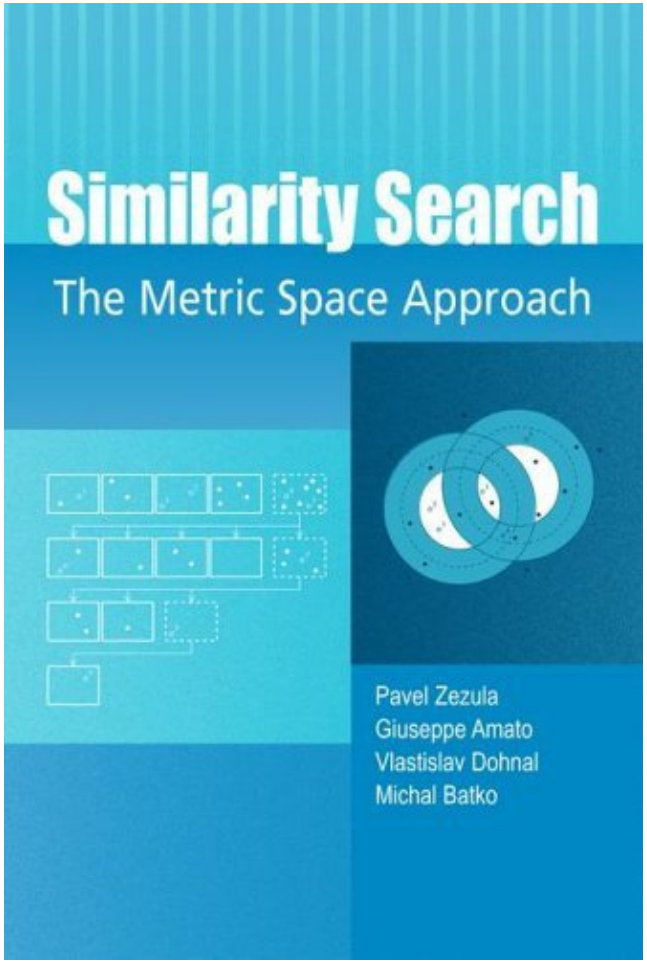
- **Range Search**
  - Für Query *q* und maximale Distanz *r*:
  - Finde alle Objekte *o* im Datenbestand mit *distance(q,o) <= r*

- **k Nearest Neighbors Search**
  - Für Query *q* und Anzahl gesuchter Elemente *k*:
  - Finde die *k* Objekte im Datenbestand mit kürzester Distanz zu *q*

Mehr davon?

Unbedingt lesen:
Similarity Search –
The Metric Space Approach

Eingescannte Seiten für
Seminarteilnehmer



Gliederung

- Similarity Search
- Seminarinhalt
  - Datensätze
  - Indexierungsalgorithmen
- Organisatorisches

Seminarinhalt

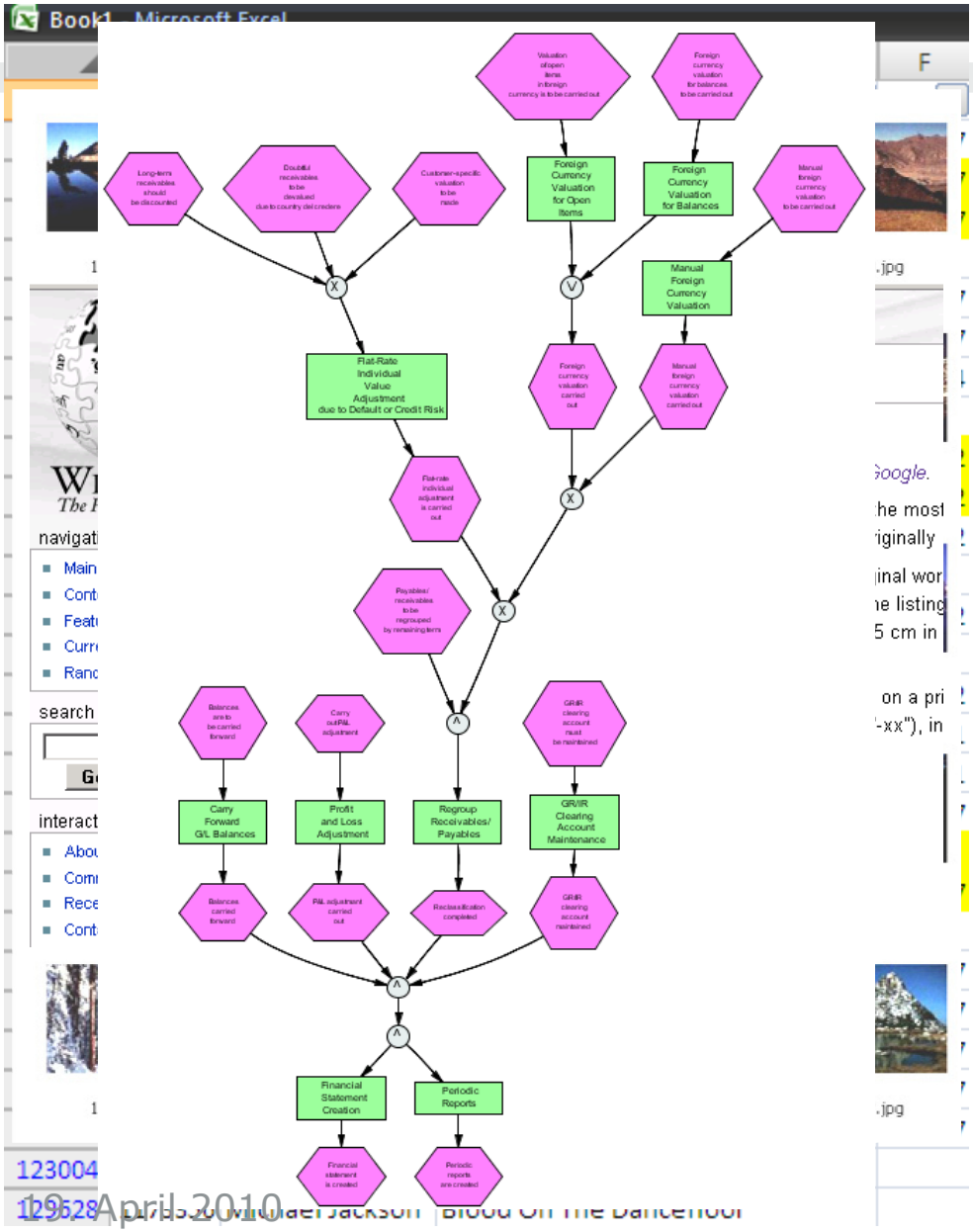
- Implementierung von 2 Ähnlichkeitsmaßen für Testdatensätze
  - 1 Maß für ein relationales Schema
  - 1 Maß für ein anderes Format
- Implementierung und Evaluierung einer Indexstruktur
  - Vorgegebenes Framework (Java)
  - Vergleichbarkeit
- Dokumentation im Trac-Wiki

Im Metric Space:
Immer kompatibel

Datensätze

- Relationale Datensätze
  - Adressdaten (1 Mio.)
  - CD-Daten (1,8 Mio.)
- Andere Datensätze
  - Bilder (10000)
  - DBpedia-Abstracts (466000)
  - Prozessmodelle (800)

- Allgemein gilt
  - nicht weitergeben
  - nicht entpackt einchecken
  - Beschreibung: https://www.hpi.uni-potsdam.de/naumann/trac/SimSearch2010/wiki/Daten

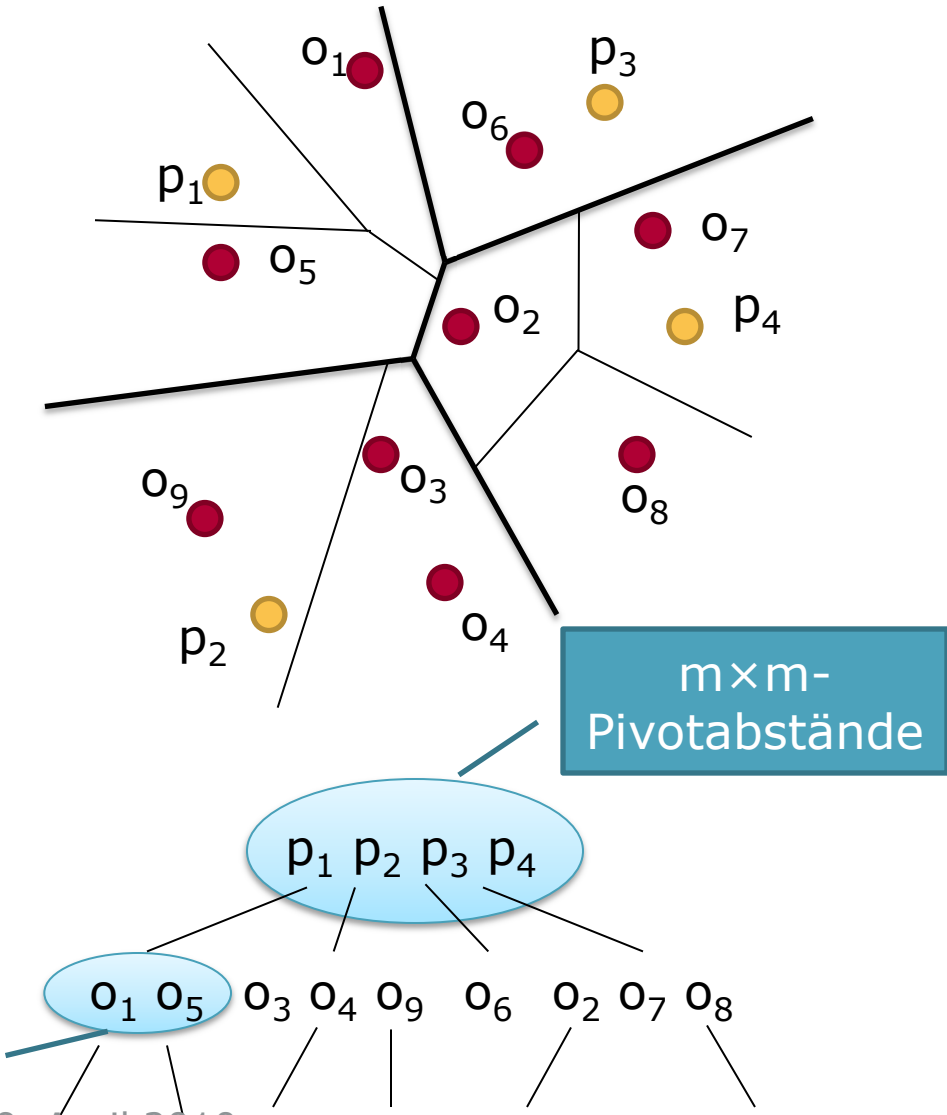


Indexierungsalgorithmen

- Ball Partitioning
  - BKT (Burkhard-Keller Tree)
  - **VPT** (Vantage Point Tree)
- Generalized Hyperplane Partitioning
  - GHT (Generalized Hyperplane Tree)
  - **GNAT** (Geometric Near-neighbor Access Tree)
- Excluded Middle Partitioning
  - VP Forest (Excluded Middle Vantage Point Forest)
  - **D-Index** (Distance Index)
- Hybride
  - **MVPT** (Multi Vantage Point Tree)
  - **M-Tree** (Metric Tree)
- Matrix
  - AESA (Approximating and Eliminating Search Algorithm)
  - **LAESA** (Linear AESA)

Geometric Near-neighbor Access Tree (GNAT)

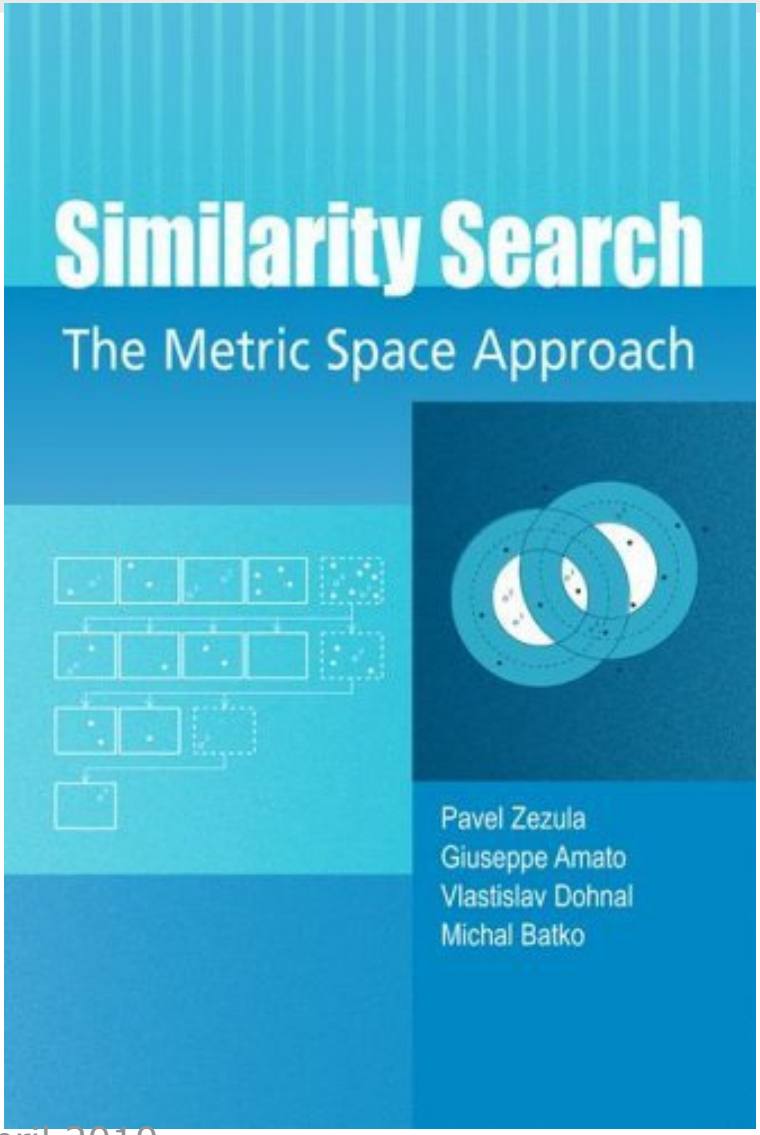
- Sergey Brin, 1995: „Near neighbor search in large metric spaces“
- Features
  - Generalized Hyperplane Partitioning
  - Verbesserung des Generalized Hyperplane Tree (GHT)
  - m Pivots pro Ebene
  - viele Meta-Informationen
    ➔ Pruning leicht möglich



min/max entferntestes/nächstes Element

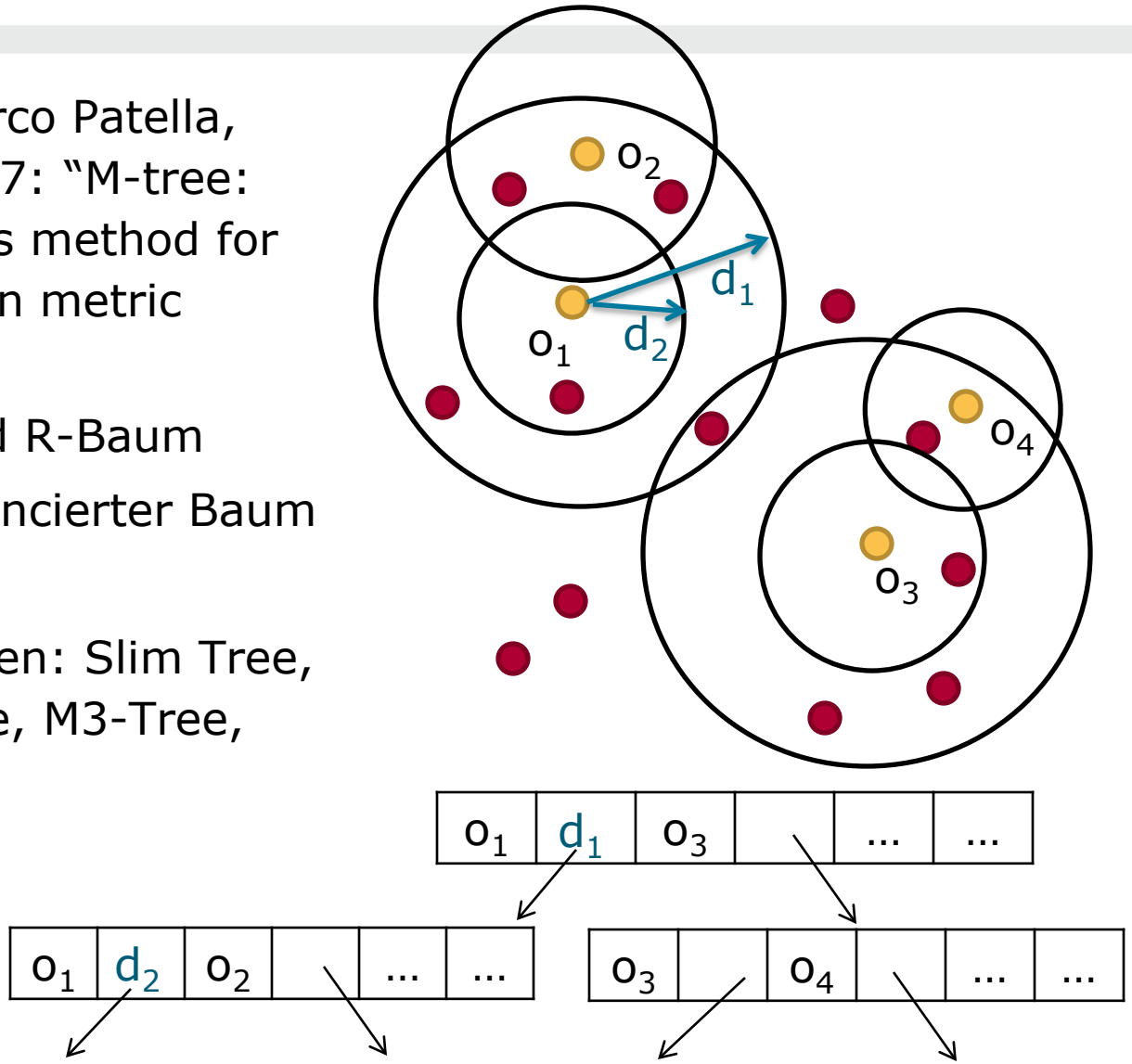
Distance Index (D-Index)

- Vlastislav Dohnal, Claudio Gennaro, Pasquale Savino, Pavel Zezula, 2003: „D-Index: Distance Searching Index for Metric Data Sets“
- Features
  - Pivots
  - Hash-basiert (O(1))
  - Excluded Middle Partitioning



M-tree

- Paolo Ciaccia, Marco Patella, Pavel Zezula, 1997: "M-tree: An efficient access method for similarity search in metric spaces"
- Basiert auf B- und R-Baum
- Dynamischer balancierter Baum
- Ball Partitioning
- Viele Erweiterungen: Slim Tree, M+-Tree, M2-Tree, M3-Tree, Pivoting M-Tree



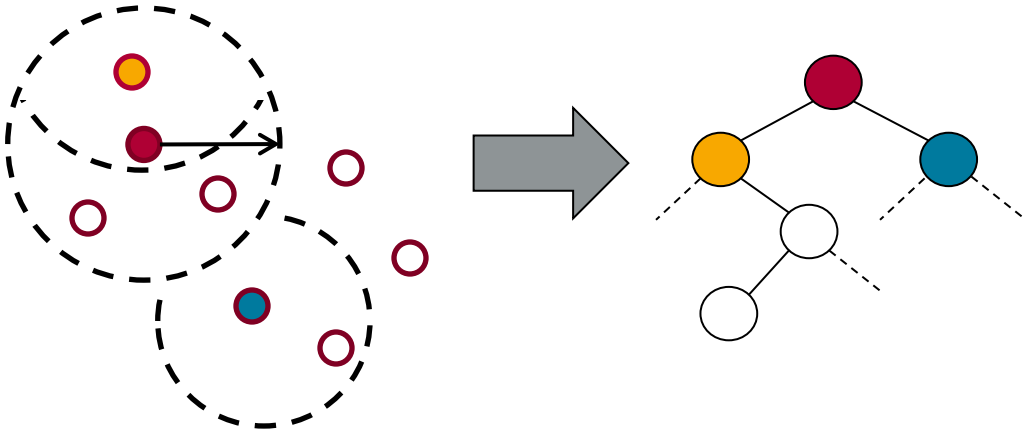
Linear AESA (Approximating and Eliminating Search Algorithm)

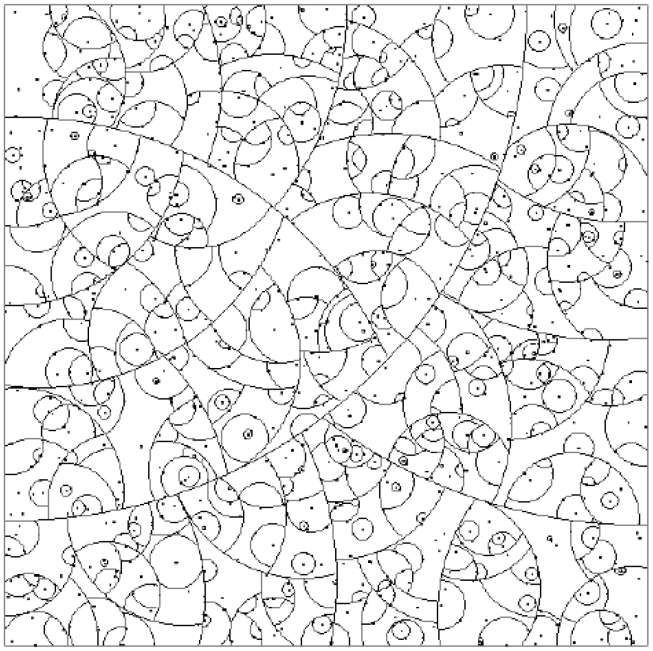
- Mará Luisa Micó, José Oncina, Enrique Vidal, 1994: "A new version of the nearest-neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing time and memory requirements"
- Branch and Bound
- Vorberechnete Ähnlichkeiten
  - AESA: zw. allen Objekten
  - LAESA: zw. ausgewählten Objekten

d	o_1	o_2	o_3	o_4	…
o_1	0				
o_2	0.9	0			
o_3	0.6	0.8	0		
o_4	0.2	0.1	0.5	0	
…					

Vantage Point Tree

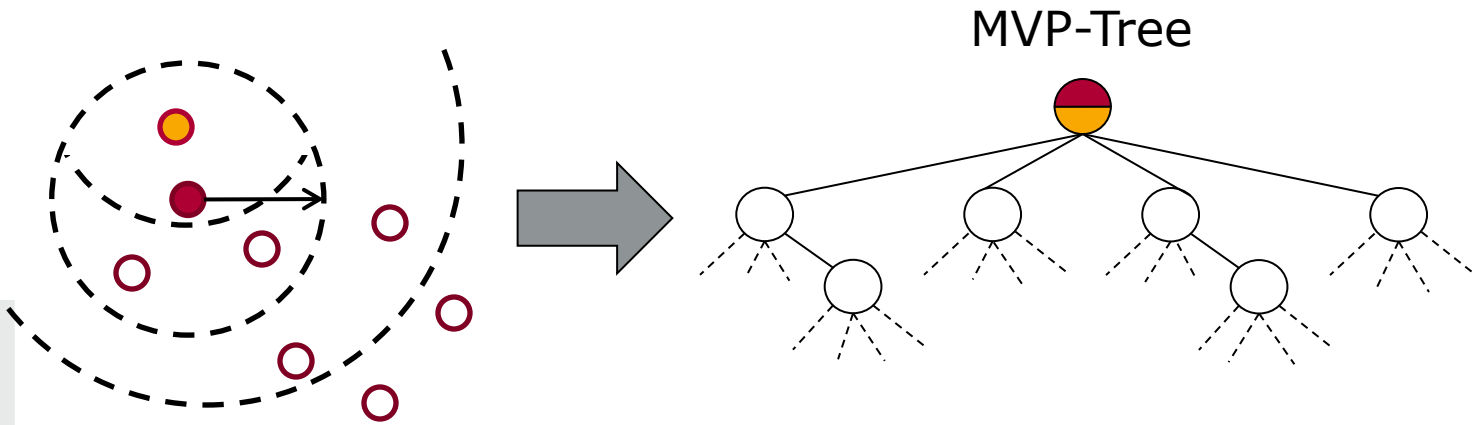
- Peter N. Yianilos, 1993, „Data Structures and Algorithms for Nearest Neighbor Search in general Metric Spaces“
- Ball Partitioning-Verfahren zum Aufbau eines binären Suchbaums
- Auswahl eines Elements und Berechnung des Median der Distanzen zu den anderen Elementen, so dass die Menge der Elemente halbiert wird





Multi Vantage Point Tree

- Tolga Bozkaya, Meral Ozsoyoglu, 1997,
  „Distance-based indexing for high-dimensional metric spaces“
- Weiterentwicklung des Vantage Point Trees
  - Mehrere Vantage Points pro internem Knoten
  - Vantage Points müssen nicht innerhalb einer Region liegen
  - Blattelemente enthalten vorberechnete Distanzen



Gliederung

- Similarity Search
- Seminarinhalt
  - Datensätze
  - Indexierungsalgorithmen
- Organisatorisches

Organisatorisches

- Anmeldung
  - bis zu 12 Teilnehmer
  - Anmeldung alleine oder als 2er-Team möglich
  - bis spätestens 26.04.2010
  - per E-Mail an Dustin Lange
    - priorisierte Liste der Wunschthemen
      (a) Daten + (b) Indexstruktur
- Anforderungen
  - Kenntnisse in Java und ggf. anderen Programmiersprachen
  - Hilfreich sind Vorkenntnisse über Indexierungstechniken für exakte Suche in Datenbanken (bekannt aus DBS II).

Leistungserfassungsprozess

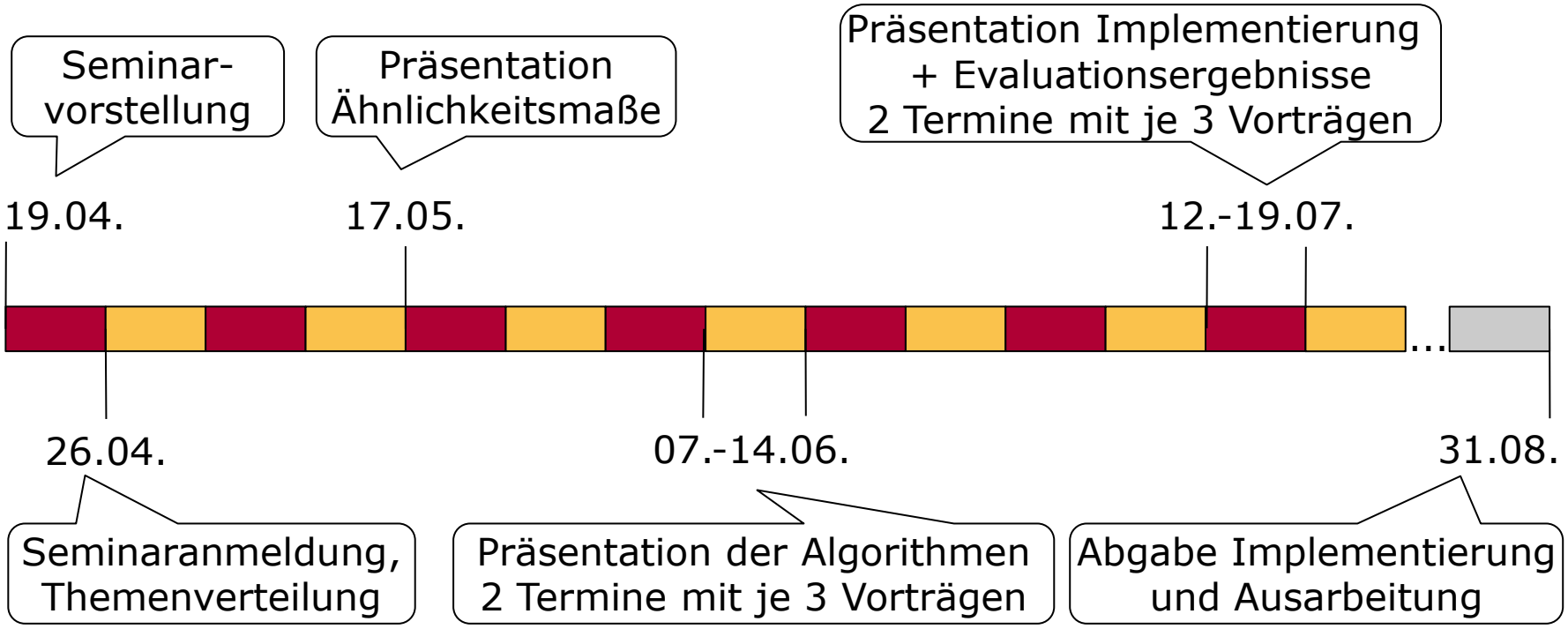
- Teilnahme an allen Seminarterminen
- Implementierung von zwei Ähnlichkeitsfunktionen in vorgegebenen Domänen
- Implementierung eines Algorithmus' zur Erstellung einer Similarity-Search-Indexstruktur sowie zur Suche in diesem Index
- 1. Vortrag: Ähnlichkeitsmaße vorstellen (ca. 10-15 min)
- 2. Vortrag: Indexierungsalgorithmus vorstellen (ca. 20-25 min)
- 3. Vortrag: Implementierung/Evaluierungsergebnisse vorstellen (ca. 20-25 min)
- Ausführliche Dokumentation im Trac-Wiki (5 Druckseiten)

Leistungserfassungsprozess

- Regelmäßige Gespräche mit Betreuer
- Abschlussnote berücksichtigt die folgenden Punkte
  - Vorträge
  - Implementierte Lösung
  - Dokumentation im Trac-Wiki
  - Mündliche Beteiligung
  - Regelmäßige Treffen mit dem Betreuer

Präsentationen	Implementierung	Ausarbeitung	Sonstige Mitarbeit
35%	20%	35%	10%

Zeitlicher Ablauf



- Mind. 1 Woche vor jedem Vortrag muss jede Gruppe ein Betreuungsgespräch führen

Gesamt-Evaluation

- Evaluation der Indexerstellung
  - Anzahl Vergleiche und Laufzeit der Indexerstellung
  - Speicherbedarf der Indexstruktur
- Evaluation der Suche in den Indexstrukturen
  - Durchschnittliche Kosten für eine Menge von Anfragen
- Gesamt-Evaluation erfolgt im August/September durch die Betreuer und wird in einem zusätzlichen Termin vorgestellt

	VP-Tree	GNAT	D-Index	...
CD-Daten	<<comparison count>> <<runtime>>			
Adress-Daten				
DBpedia				
Prozessmodelle				
Bilder				

Danke

- Wir freuen uns auf Eure Anmeldung bis 26.04.2010!