Karlsruher Institut für Technologie

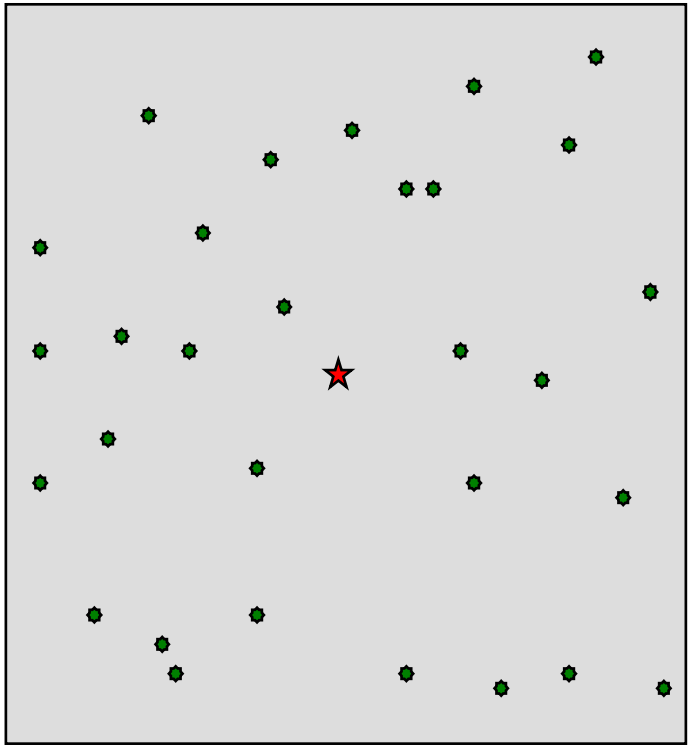
Datenbanksysteme

Kapitel 2: Clustering und Finden von Ausreißern

Lehrstuhl für Systeme der Informationsverwaltung, Fakultät für Informatik

photo by Robert S. Donovan

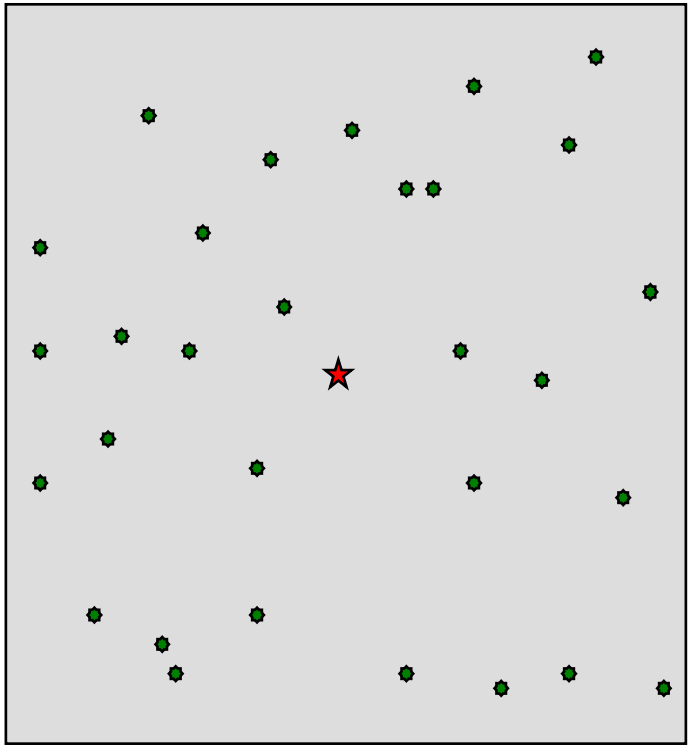
Räumliche Indexstrukturen (1)



Datenraum

- Motivation:
  - Grüne Punkte: Bars, die Ihr bevorzugtes Bier ausschenken.
  - Punkte enthalten in Relation Bar(X, Y, Name).
  - Stern: Ihr Standort.
  - Welche Bar ist am nächsten?
- Offensichtliche Lösung: Relation scannen, Abstand jedes Tupels berechnen.

Räumliche Indexstrukturen (2)



Datenraum

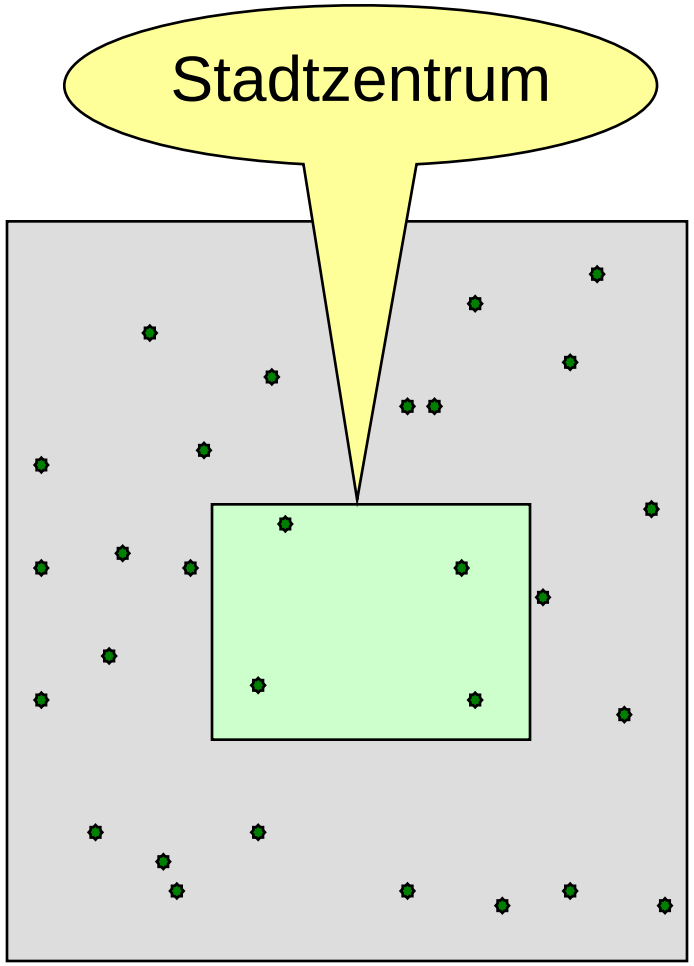
Bar

X	Y	Name

Anfragetypen

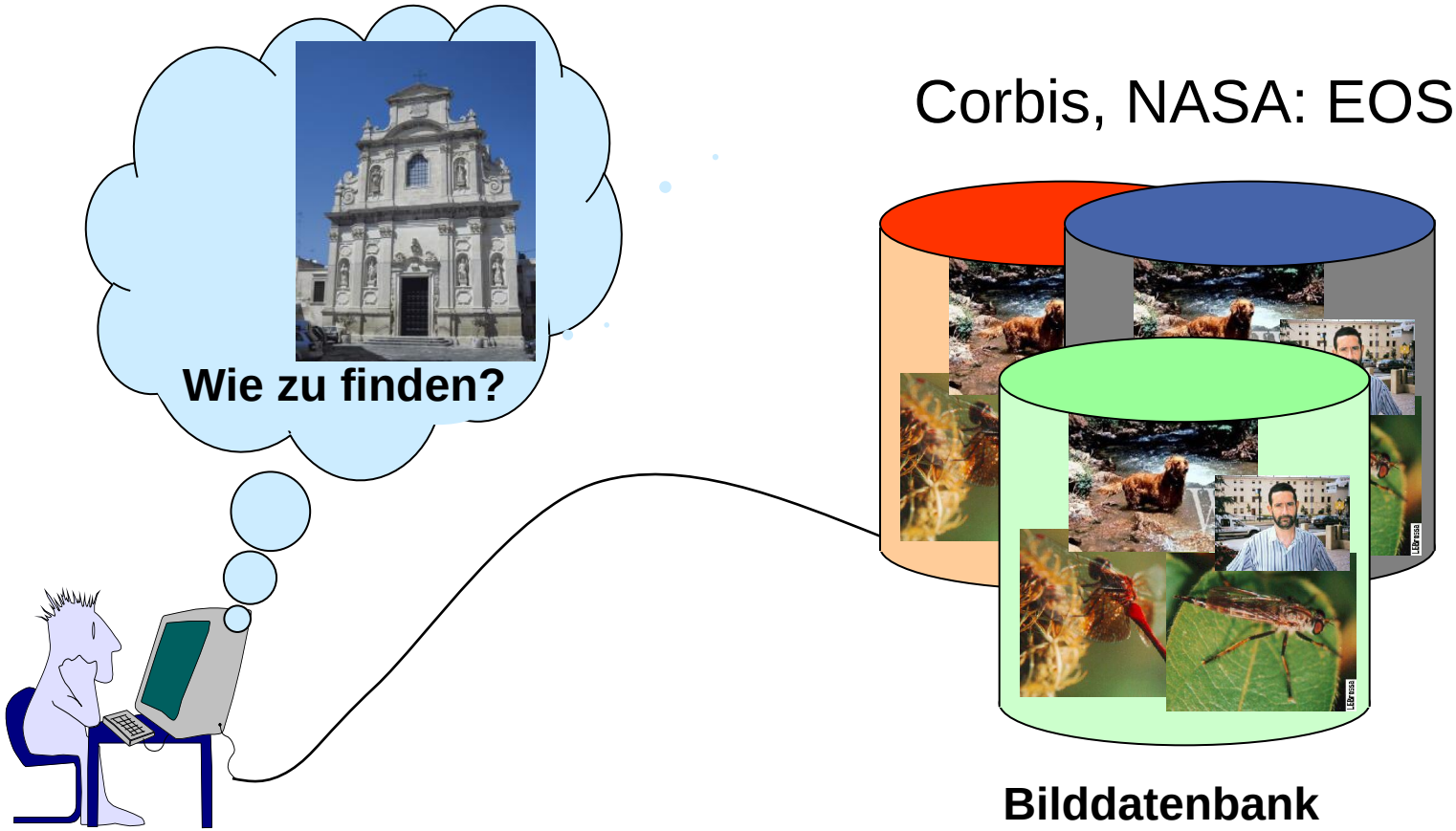
- Unterschiedliche Anfragetypen möglich.
- Hier: Bereichsanfragen.

- Bereichsanfrage:
  - Anfrage über einen Bereich im Datenraum.
  - Wie viele/welche Restaurants gibt es im Stadtzentrum?



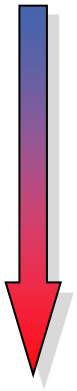
Multimedia-Retrieval – Veranschaulichung

- Input: Anfragebild,
- Ergebnis: Menge ähnlicher Bilder

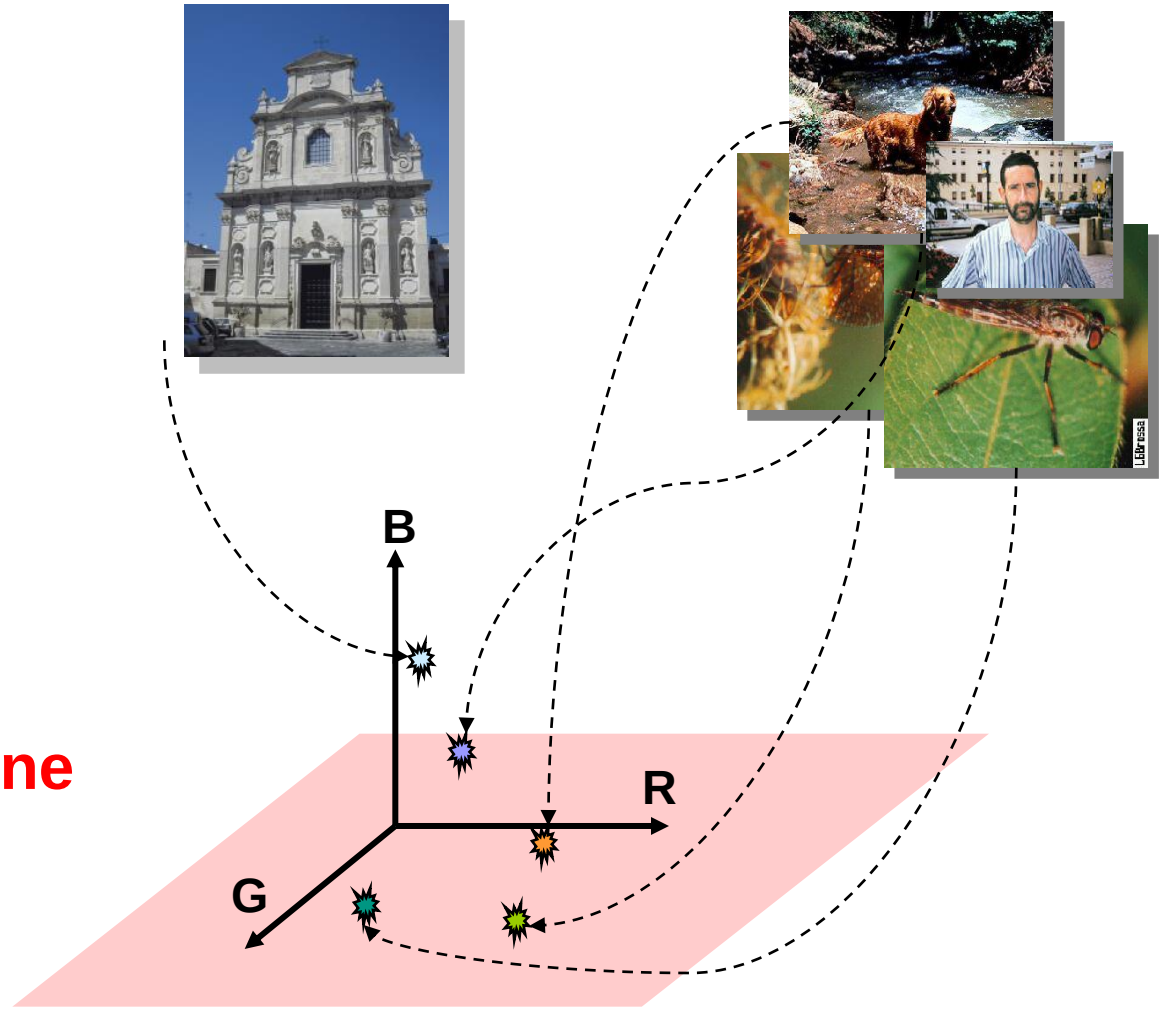


Ähnlichkeitssuche →
Suche nach dem nächsten Nachbarn

Bild-Ebene



Feature-Ebene



Motivation

Index

kd-Baum,
k-NN

Outlier

Outlier
– Verfahren

DBSCAN

Anomalien

Teilräume
Motivation

HiCS

Distanz im Merkmalsraum: Maß der Unähnlichkeit.

Ähnlichkeitssuche → Suche nach dem nächsten Nachbarn (1)

- Dimensionalität der Merkmalsvektoren kann hoch/sehr hoch sein.
  - Z. B. 25 Dimensionen (unterschiedliche Schadstoffarten) bei Luftmessungen des KIT in Peking
  - 16 bei Pendigits (aus UCI Repository).
  - 279 bei Arrhythmia (ebenda).


PPG
amazon.com
photo by thisisbossi

Ähnlichkeitssuche → Suche nach dem nächsten Nachbarn (2)

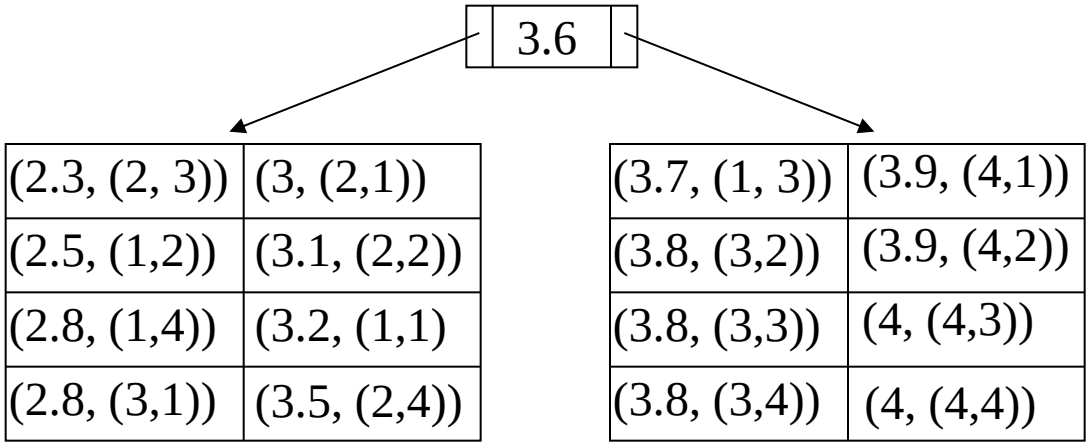
- Anderes Beispiel:
  - Kundenprofile bei Amazon.
  - Jede Produktkategorie (z. B. Informatikbücher, Fotozubehör) ist eine Dimension des Merkmalsraums.
  - Ein Merkmalsvektor pro Kunde.
  - Wert einer Komponente beispielsweise: Wert der Einkäufe dieses Kunden in der jeweiligen Kategorie.
  - Sinnvolle Informationsbedürfnisse:
    - „Suche den Kunden, der mir am ähnlichsten ist.“ (NN-Anfrage)
    - „Wie viele Kunden gibt es, die sowohl nennenswert Infobücher als auch Fotozubehör gekauft haben?“ (Bereichsanfrage)

Index – Motivation

- Seitenweise Anordnung der Daten.
- Daten müssen im Hauptspeicher vorliegen, damit Selektion etc. durchgeführt werden kann.
- Seiten – Einheiten des Zugriffs.
- Laden einer Seite in den Hauptspeicher ist teuer, *Zugriffslücke*.
- Zahl der zu ladenden Seiten möglichst minimieren.

Index – Illustration

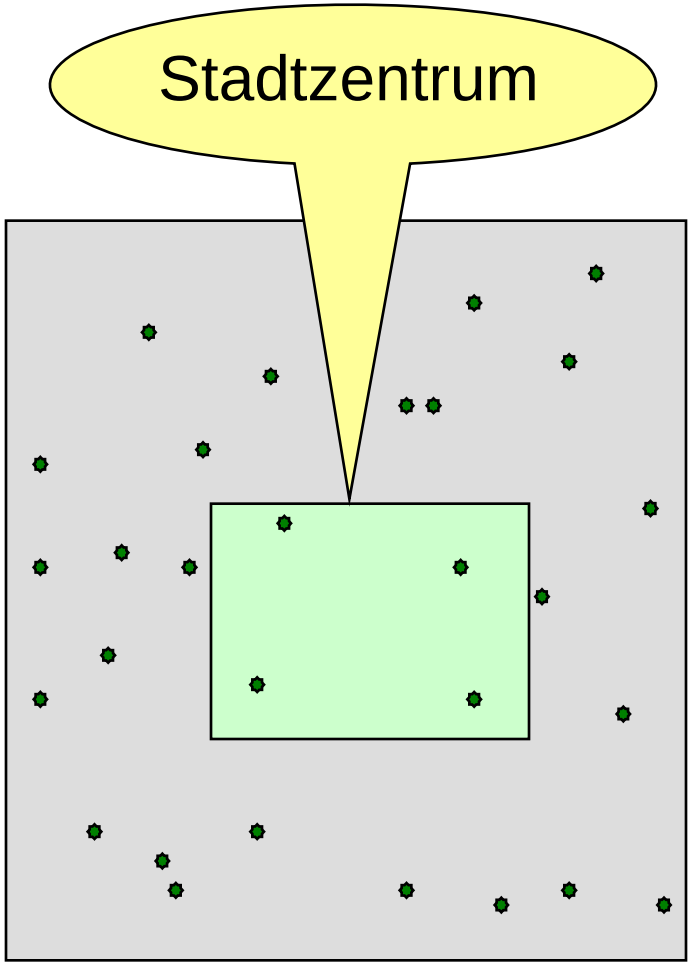
- Student(name, age, gpa, major); t(Student) = 16.
- Non-clustered primary B+-tree für Attribut gpa.



Tom, 20, 3.2, EE	Mary, 24, 3, ECE	Lam, 22, 2.8, ME	Chris, 22, 3.9, CS
Chang, 18, 2.5, CS	James, 24, 3.1, ME	Kathy, 18, 3.8, LS	Vera, 17, 3.9, EE
Bob, 21, 3.7, CS	Chad, 28, 2.3, LS	Kane, 19, 3.8, ME	Louis, 32, 4, LS
Pat, 19, 2.8, EE	Leila, 20, 3.5, LS	Martha, 29, 3.8, CS	Shideh, 16, 4, CS

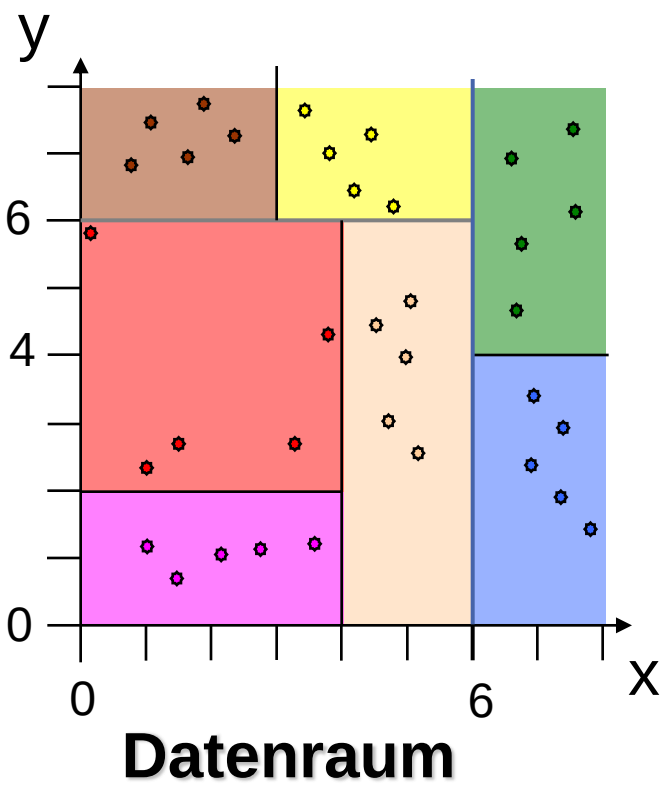
Index – Illustration (3)

- B-Baum löst unser Bar-Problem nicht wirklich.

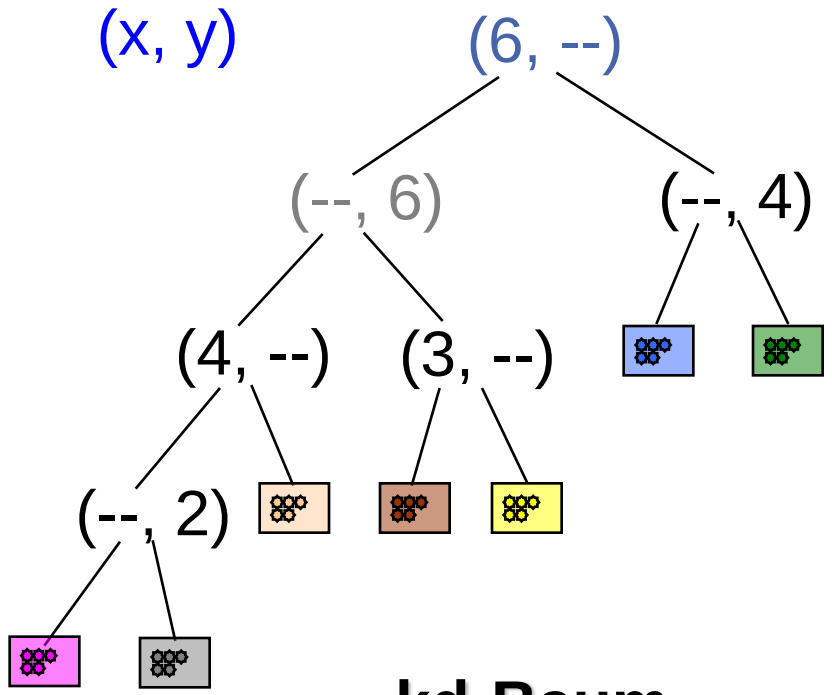


Datenraum

kd-Baum (1)



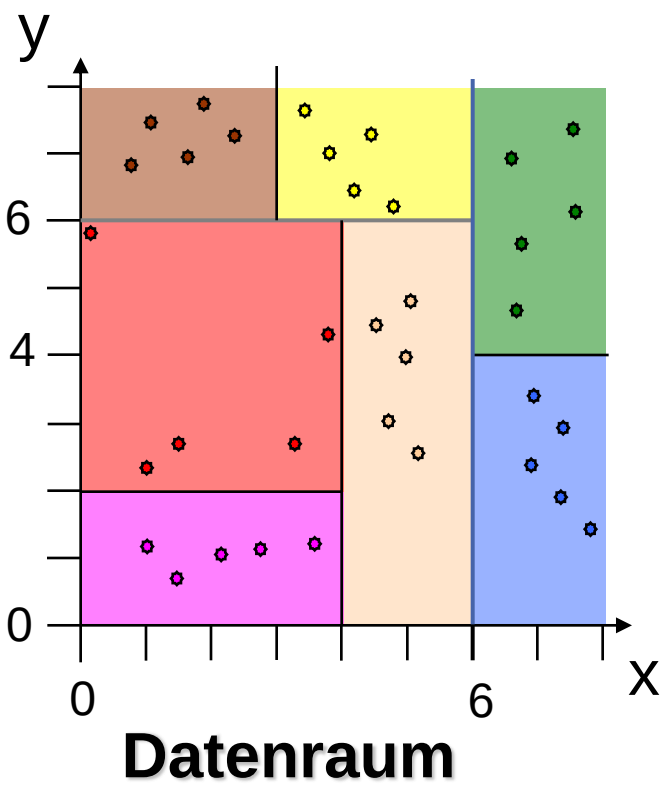
Datenraum



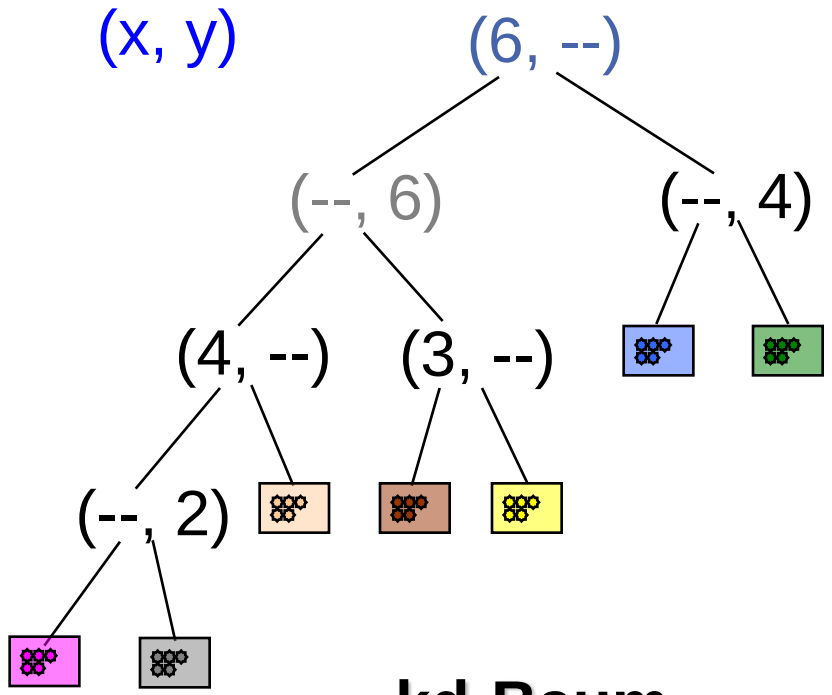
kd-Baum

- Split-Richtung: Eine Dimension nach der anderen, dann wieder von vorne.
- Anzahl Split-Dimensionen: Vier im Beispiel.

kd-Baum (1)



Datenraum

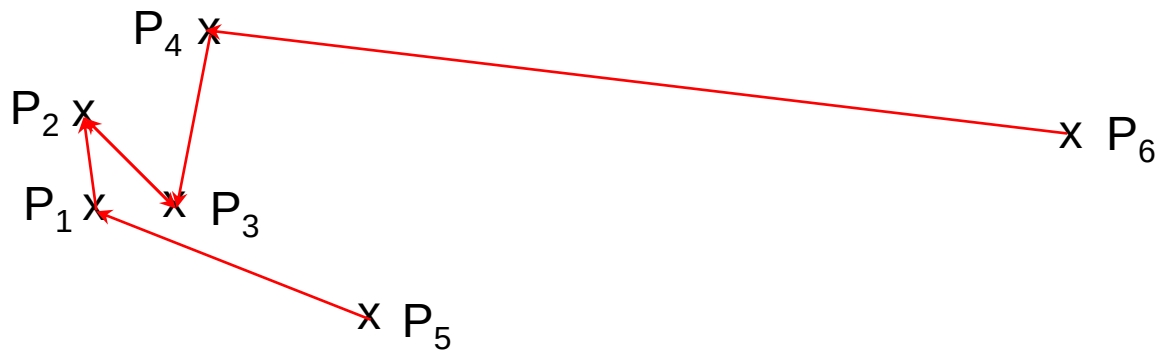


kd-Baum

- Einfügen in den Baum:
  Baum ist nicht balanciert und ‘wächst nach unten’.
- Illustration: Einfügen eines Objekts in lilafarbene Zone.

k-NN

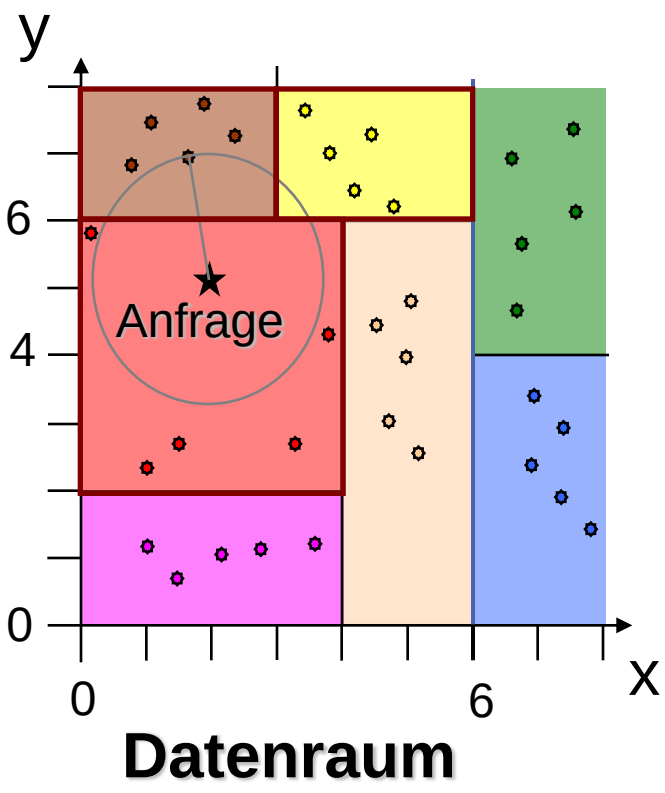
- Illustration für k=2:



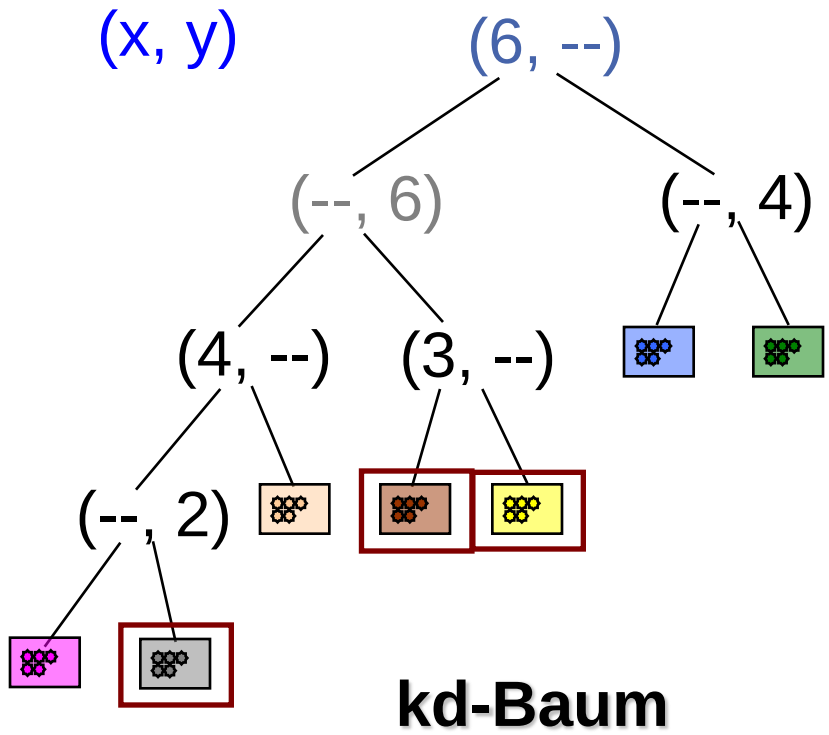
- *k-Abstand* (oder *k-NN Abstand*)
  := Abstand des k-nächsten Nachbarn.

- Im Folgenden verwendete Notation:
  E[k-NN Abstand]

kd-Baum (2)



Datenraum



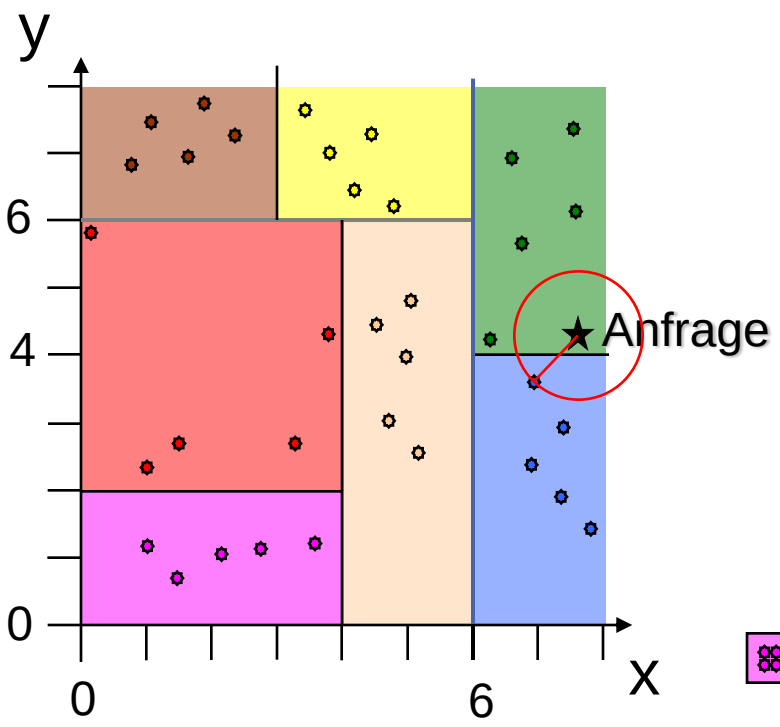
kd-Baum

- *NN-Distanz*, *NN-Sphäre*,
- Einsparung: Nur ein paar Rechtecke inspizieren.

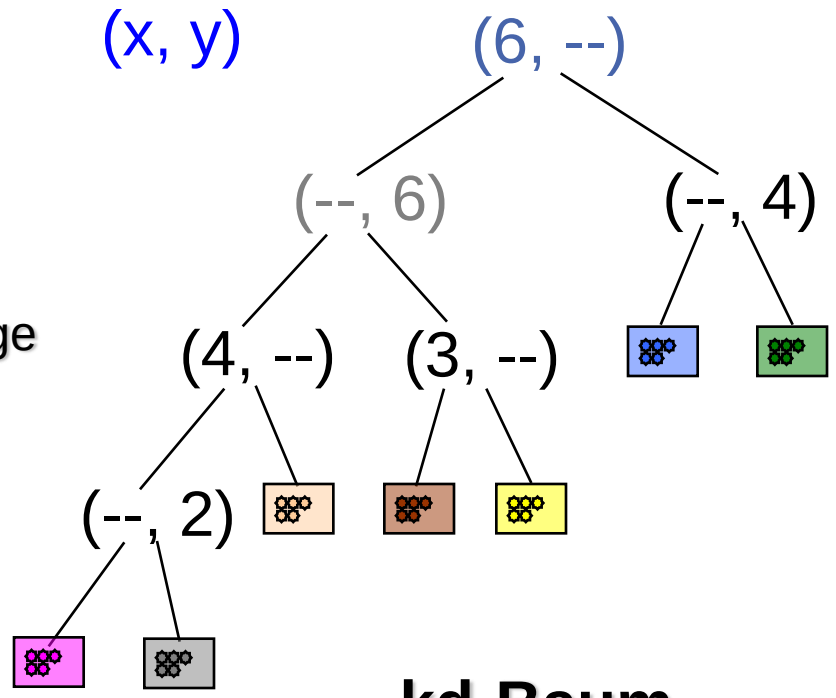
kNN-Suche – Erläuterungen

- Verwendung einer Priority Queue,
  im Beispiel repräsentiert durch [,].
- Objekte in Queue sind entweder Datenobjekte
  oder Knoten des Baums.
- Abstand zum Anfragepunkt
  als Sortierkriterium der Objekte in der Queue.

kd-Baum (3)



Datenraum

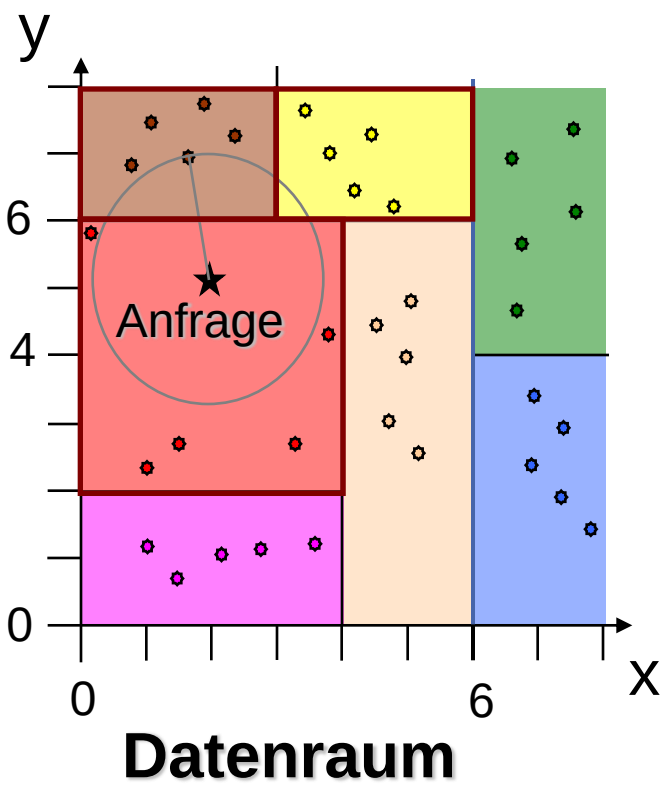


kd-Baum

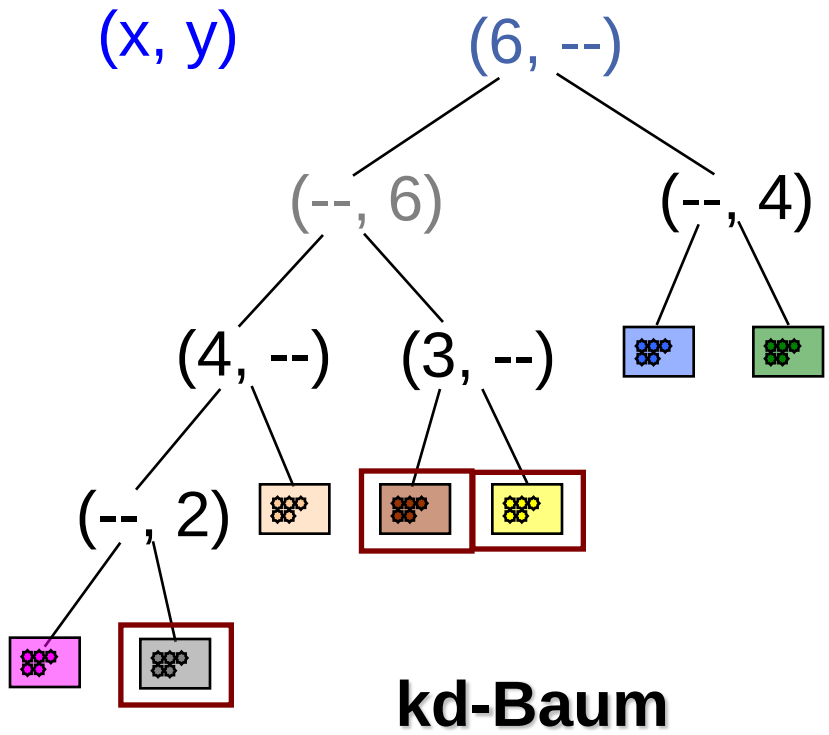
1. [(6, --)]
2. [(--, 4), (--, 6)]
3. [, , (--, 6)]
4. [, , (--, 6), , , ,]
5. [, , …, (--, 6), …]
6. Ende Algorithmus.

kd-Baum (2)

– Folie von vorhin –



Datenraum



kd-Baum

- *NN-Distanz*, *NN-Sphäre*,
- Einsparung: Nur ein paar Rechtecke inspizieren.
  Warum?

nächsterNachbar(kdB-Baum, Anfrage)

```
1    Queue ← neue Prioritätswarteschlange()
2    Region ← Wurzelknoten des kdB-Baums
3    Distance ← Abstand(Region, Anfrage)
4    Einfügen(Queue, Distance, Region)

5    WHILE (true) DO
6      Element ← Head der Queue
         // Dieser Schritt beinhaltet auch ‚pop‘,
         // d. h. Entnahme des Queue-Heads
7      IF (Element ist Datenpunkt) THEN
8        BEGIN; Gib Element zurück; RETURN; END
9      ELSE
10       Traversierung(Element, Queue, Anfrage)
11   END WHILE
```

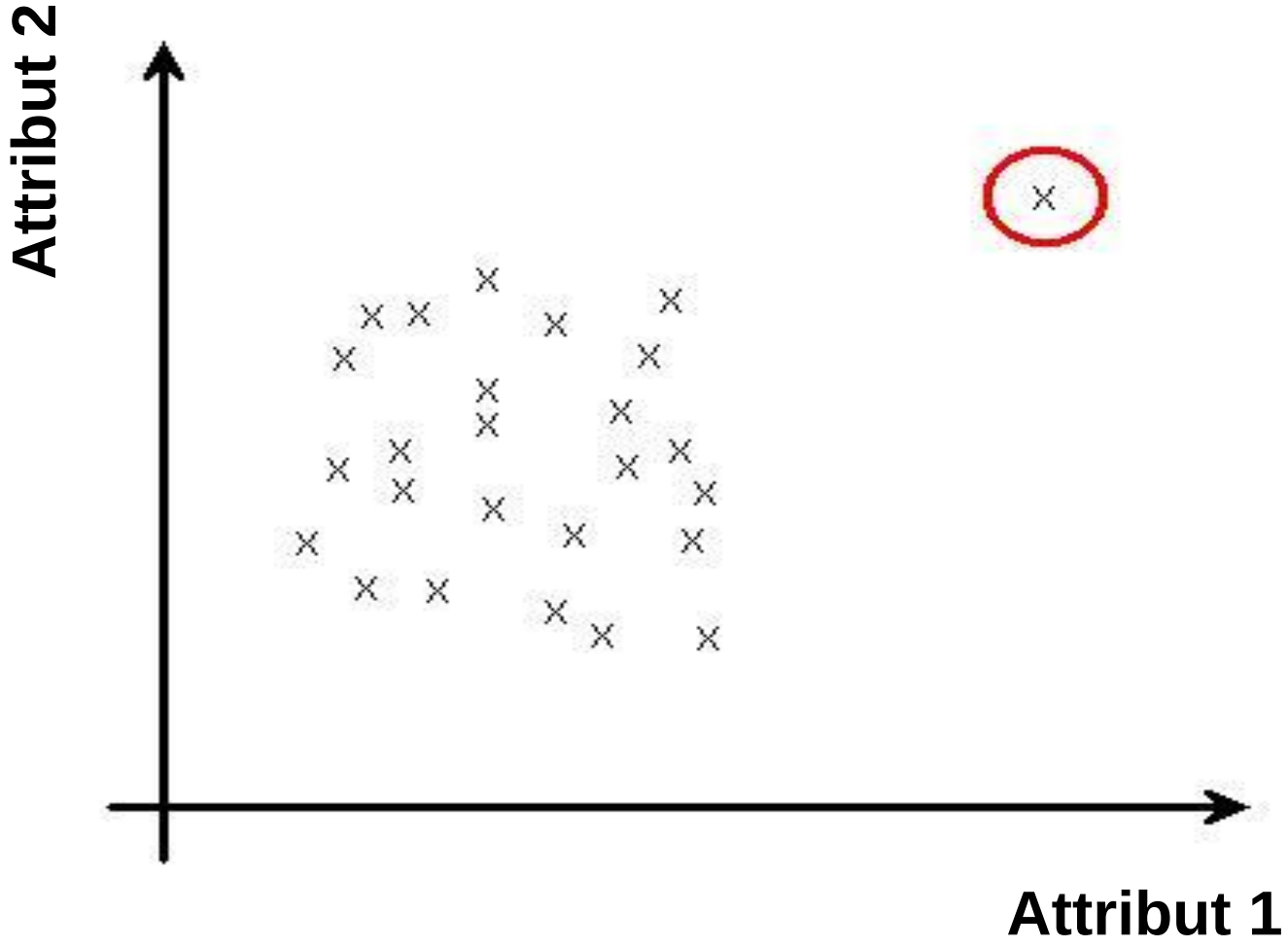
Traversierung(Element, Queue, Anfrage)

```
1   IF (Element ist Blatt) THEN
2     FOR EACH (Datenpunkt in Element) DO
3       Distance ← Abstand(Datenpunkt, Anfrage)
4       Einfügen(Queue, Distance, Datenpunkt)
5     END FOR
6   ELSE
7     FOR EACH (Kind von Element) DO
8       Distance ← Abstand(Kind, Anfrage)
9       Einfügen(Queue, Distance, Kind)
10    END FOR
11  END IF
```

kd-Baum (4)

- Baum ist nicht balanciert; es gibt aber balancierte Bäume für mehrdimensionale Daten. (Die kommen in der Wirklichkeit dann auch zur Anwendung.)

Outlier – Illustration



Was ist ein Outlier?

- Intuitive, einfache Definition:

  „*Outlier ist definiert als Element des Datenbestands, das in bestimmter Hinsicht vom restlichen Datenbestand erheblich abweicht.*“
- Unterschiedliche Techniken zum Ermitteln von Outliern existieren.

Anwendungsszenarien – Beispiele

- Video- und Sicherheitsüberwachung,
- Erkennen von Netzwerkfehlern,
- E-Commerce,
- Finanzen,
- Marketing.

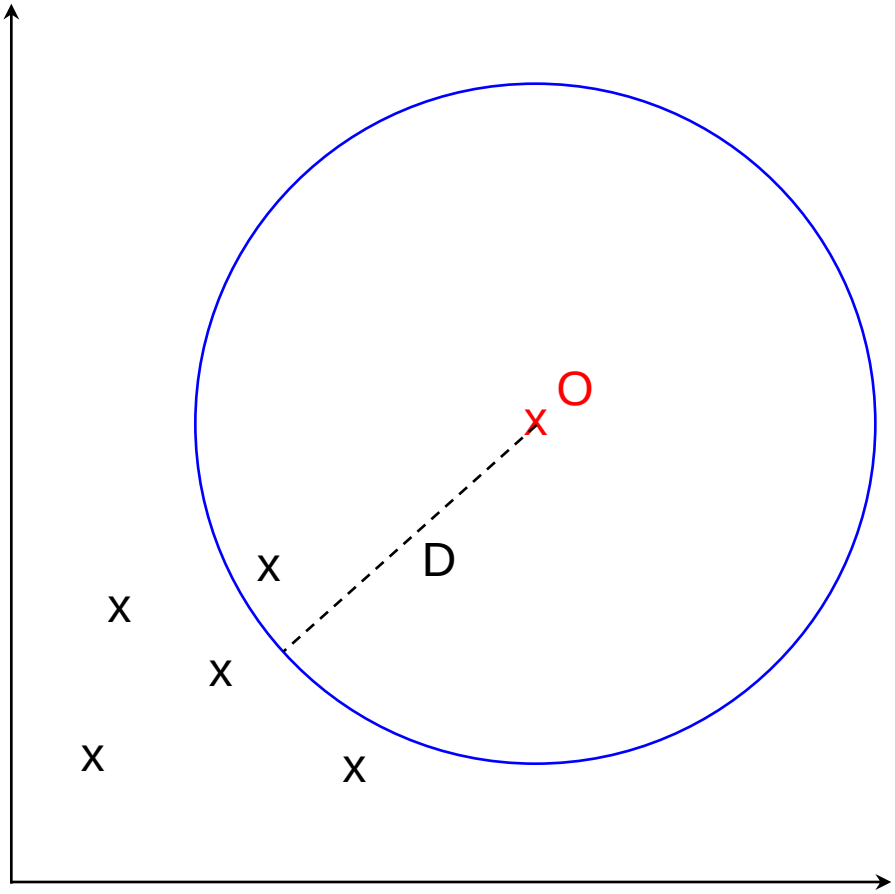
- Data Cleaning.

Abstandsbasiertes Ermitteln der Outlier – eine mögliche Definition

- Objekt O, das in Datenbestand T enthalten ist, ist ein *DB(p,D)-Outlier*, wenn der Abstand von O zu mindestens p Prozent der Objekte in T größer ist als D.

- Beispiel:
  - 1000 Objekte, p=99.
  - D. h. höchstens neun Objekte mit Abstand D oder weniger.

Illustration



- O ist Outlier, wenn *p=0.6*.
  Outlier, wenn Abstand zu p% der Objekte größer als …
- O ist kein Outlier, wenn *p=0.99*.

Outlier – Erläuterung der Illustration

- Objekt ist Outlier, wenn mehr als 60 Prozent (99 Prozent) der Datenobjekte außerhalb liegen.
- Ersteres ist gegeben, Letzteres nicht.

Überprüfung, ob Punkt Outlier

- Wenn k-Abstand < D, dann ist Punkt kein Outlier.
- $k = N \cdot (1 - p) - 1$
- Erklärung:
  - *p* – Anteil der Objekte ‚außerhalb der Kugel'. $p < 1$.
  - Parameter *k* hingegen: Anzahl Objekte innerhalb bzw. genau auf der Kugel.
  - D. h. wenn *p* sehr nahe bei 1, ist k klein.
  - Faktor *N*: Umrechnung in absoluten Wert.
- Für entsprechende Überprüfung räumliche Datenstruktur verwendbar.
- Allerdings Überprüfung nur für einen Punkt, nicht ‚Finden *aller* Outlier'.

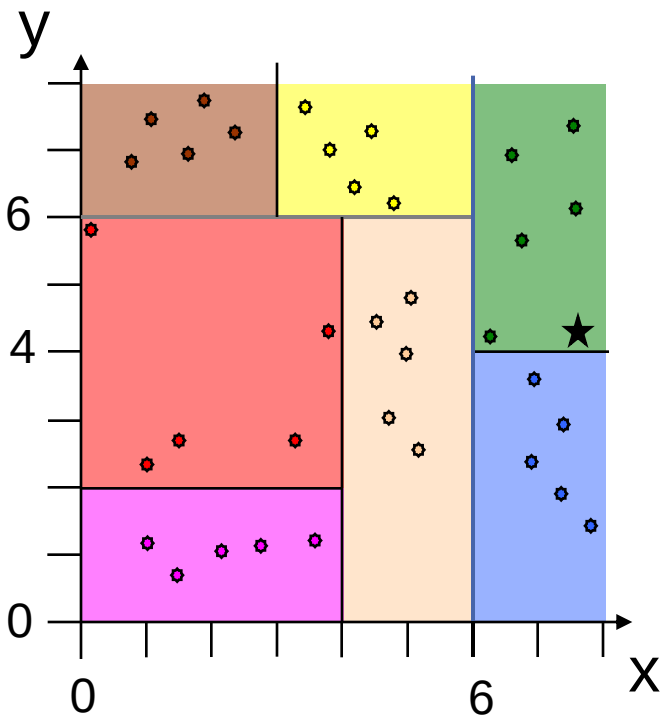
Abstandsbasiertes Ermitteln der Outlier – Index-basierter Algorithmus

- k-NN Query für jeden Punkt,
- Stop, sobald k-NN Distanz kleiner als *D*.

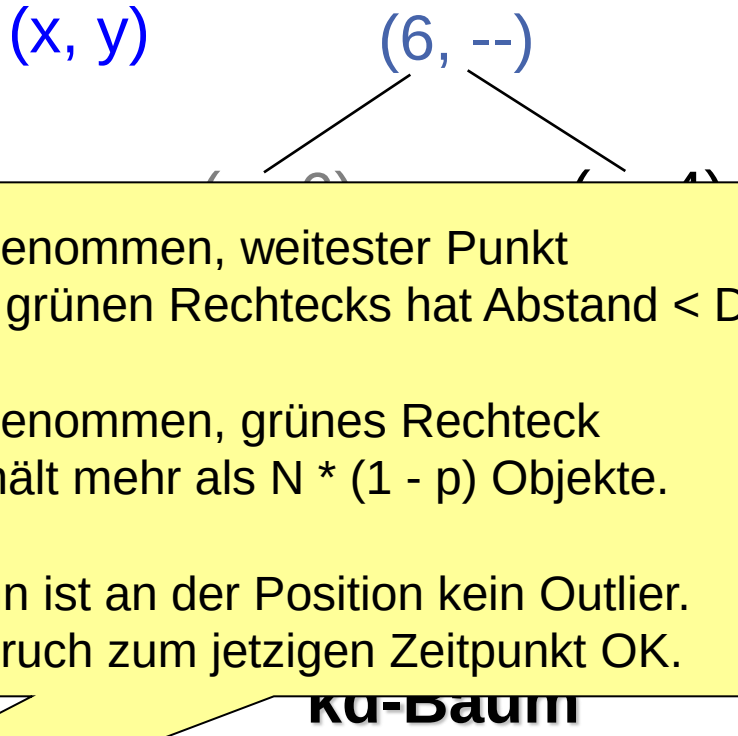
- Ansatz insbesondere dann nicht so teuer, wenn Index bereits existiert und nicht mehr aufgebaut werden muss.

- Man kann k-NN Suche i. Allg. deutlich früher abbrechen. Voraussetzung: Jeder Knoten speichert außerdem Anzahl der entsprechenden Datenobjekte.
  Illustration auf folgender Folie.

kd-Baum (3)



Datenraum



kd-Baum

Angenommen, weitester Punkt des grünen Rechtecks hat Abstand < D.

Angenommen, grünes Rechteck enthält mehr als N * (1 - p) Objekte.

Dann ist an der Position kein Outlier. Abbruch zum jetzigen Zeitpunkt OK.

1. [(6, --)]
2. [(--, 4), (--, 6)]
3. [■, ■, (--, 6)]
4. [■, •, (--, 6), •, •, •, •]
5. [•, •, …, (--, 6)]
6. Ende Algorithmus.

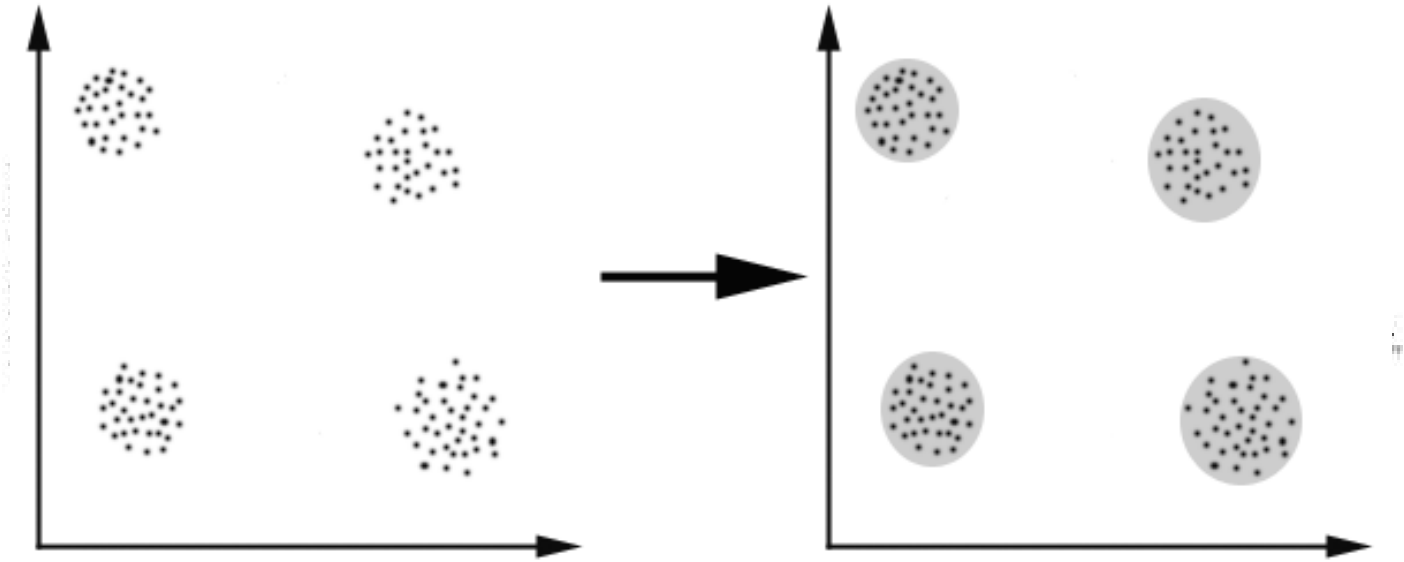
Vielfalt der Ansätze

1. Basierend auf Verteilung,
2. basierend auf Clustering
   (gemeint ist: als ‚Nebenprodukt' eines Clustering-Algorithmus);
3. Abstandsbasiert,
4. Dichte-basiert.

- Es gibt grosse Vielfalt von Verfahren zur Outlier-Erkennung.
- Es gibt auch andere Definitionen von ‚Outlier'.
  Solche mit Ranking i. Allg. bevorzugt.

Beispiel für Clustering: Customer Segmentation

- Gegeben: Große Datenbank mit Kundendaten, die Eigenschaften und Käufe der Kunden in der Vergangenheit enthält.
- Ziel: Gruppen von Kunden mit ähnlichem Verhalten finden.



http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/

DBSCAN (1)

- *Dichte* := Anzahl wasauchimmer pro Volumeneinheit.
- *Objekt ist dicht* := mindestens minPts andere Objekte in Kugel um Objekt mit Radius ε (rote Punkte in der Abbildung mit minPts = 3).

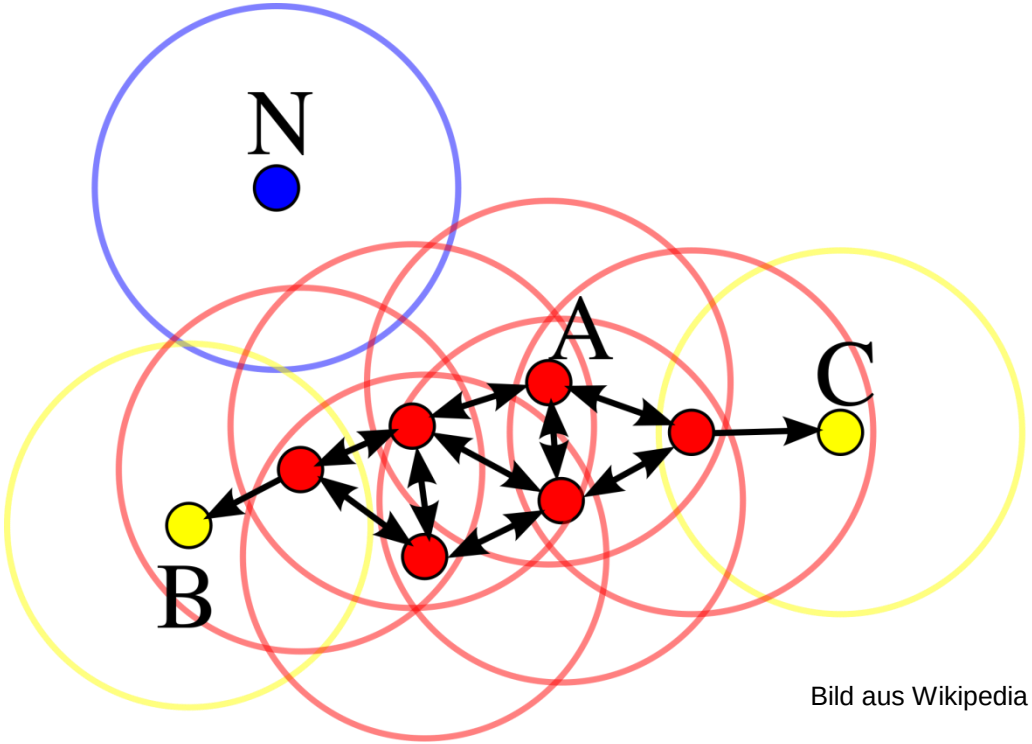


Bild aus Wikipedia

DBSCAN (2)

- *Dichte-erreichbares Objekt* :=
  Objekt in ε-Umgebung eines dichten Objekts,
  das selbst nicht dicht ist. (‚Rand des Clusters‘)
- Zuordnung dichter Punkte zu Cluster ist deterministisch,
  die Dichte-erreichbarer Punkte ist nichtdeterministisch.
  (Gelbes Objekt kann unterschiedlichen Clustern
  zugeordnet werden.)

DBSCAN Pseudocode (1)

```
DBSCAN(D, ε, MinPts)
  C = 0
  for each unvisited point P in dataset D
    mark P as visited
    N = getNeighbors(P, ε)
    if sizeof(N) < MinPts
      mark P as NOISE
    else
      C = next cluster
      add P to cluster C
      for P' in N
        if P' is not yet member of any cluster
          recursiveExpandCluster(P', C, ε, MinPts)
```

DBSCAN Pseudocode (2)

```
recursiveExpandCluster(P, C, ε, MinPts)
  add P to cluster C
  if P is not visited
    mark P as visited
    N = getNeighbors(P, ε)
    if sizeof(N) >= MinPts
      for P' in N
        if P' is not yet member of any cluster
          recursiveExpandCluster(P', C, ε, MinPts)
```

DBSCAN – Diskussion

- Lineare Komplexität,
  wenn ε-Umgebungen vorberechnet wurden
  (bzw. mit räumlichem Index mit konstantem Aufwand herausgesucht werden können).
- D. h. Verwendung mehrdimensionaler Indexstruktur
  i. d. R. sehr vorteilhaft.
- Noise – möglicherweise Outlier.
  D. h. DBSCAN erstellt Vorauswahl möglicher Outlier.

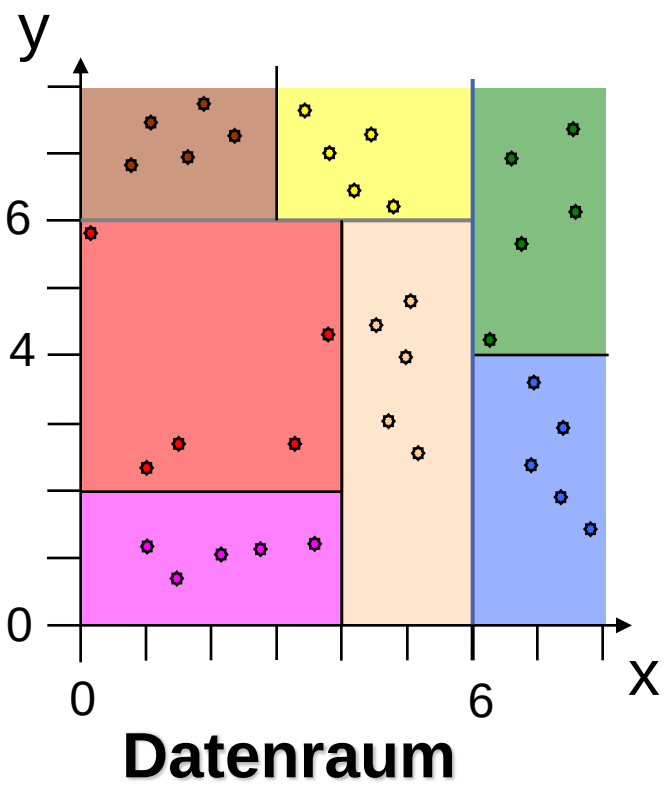
DBSCAN – Einordnung

- DBSCAN – eines von vielen Verfahren, um Cluster zu finden.
- Alternativen beispielsweise: k-Means, BIRCH, LOF. (LOF wird weiter hinten in Experimenten verwendet.)

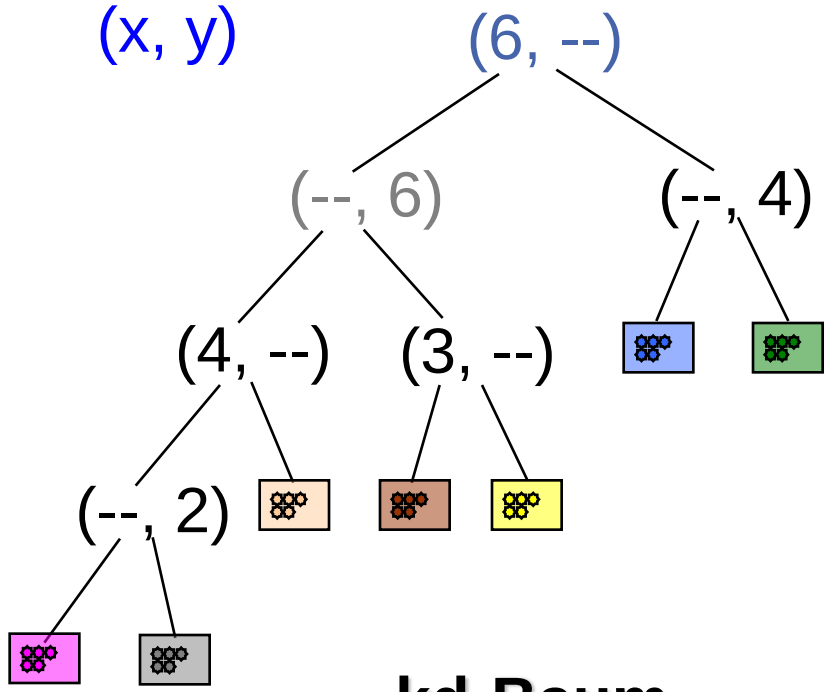
Anomalien
in hochdimensionalen Datenräumen

1. Raum ist mit Punkten nur dünn besetzt (sparsity),
2. hierarchische Datenstrukturen sind nicht effektiv.

kd-Baum



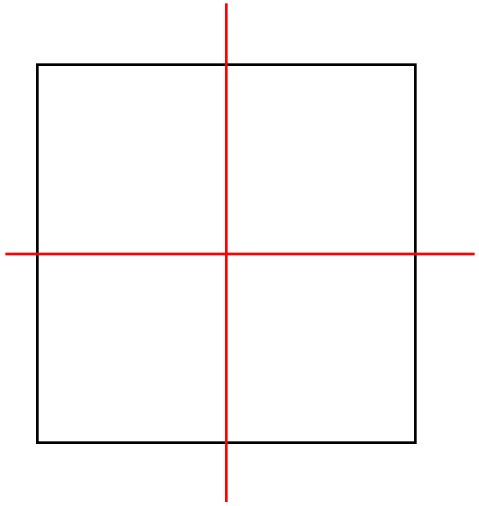
Datenraum



kd-Baum

Sparsity

- Diskretisierung,
  nur zwei Partitionen pro Dimension.
- Illustration:



- Sehr viele Datenobjekte, z. B. N=1,000,000,000.
- Hochdimensionaler Raum, z. B. d=100.
- Wie viele Datenpunkte pro Zelle?
  $N/2^d$ = 0,000000000000000000000789
- Eine zufällige Zelle ist höchstwahrscheinlich leer.

Hierarchische Datenstrukturen sind nicht effektiv (1)

- Annahme im Folgenden: Datenpunkte gleichverteilt. Datenraum $\Omega=[0,1]^d$. D. h. Kantenlänge 1.
- Datenraum hat Volumen 1, für alle *d*.

Hierarchische Datenstrukturen sind nicht effektiv (2)

- Wahrscheinlichkeit, dass NN-Distanz von Punkt Q kleiner ist als r:

$$P[Q,r] = 1 - \left(1 - Volume\left(sphere^d(Q,r) \cap \Omega\right)\right)^N$$

- Erwartete NN-Distanz für Punkt Q:

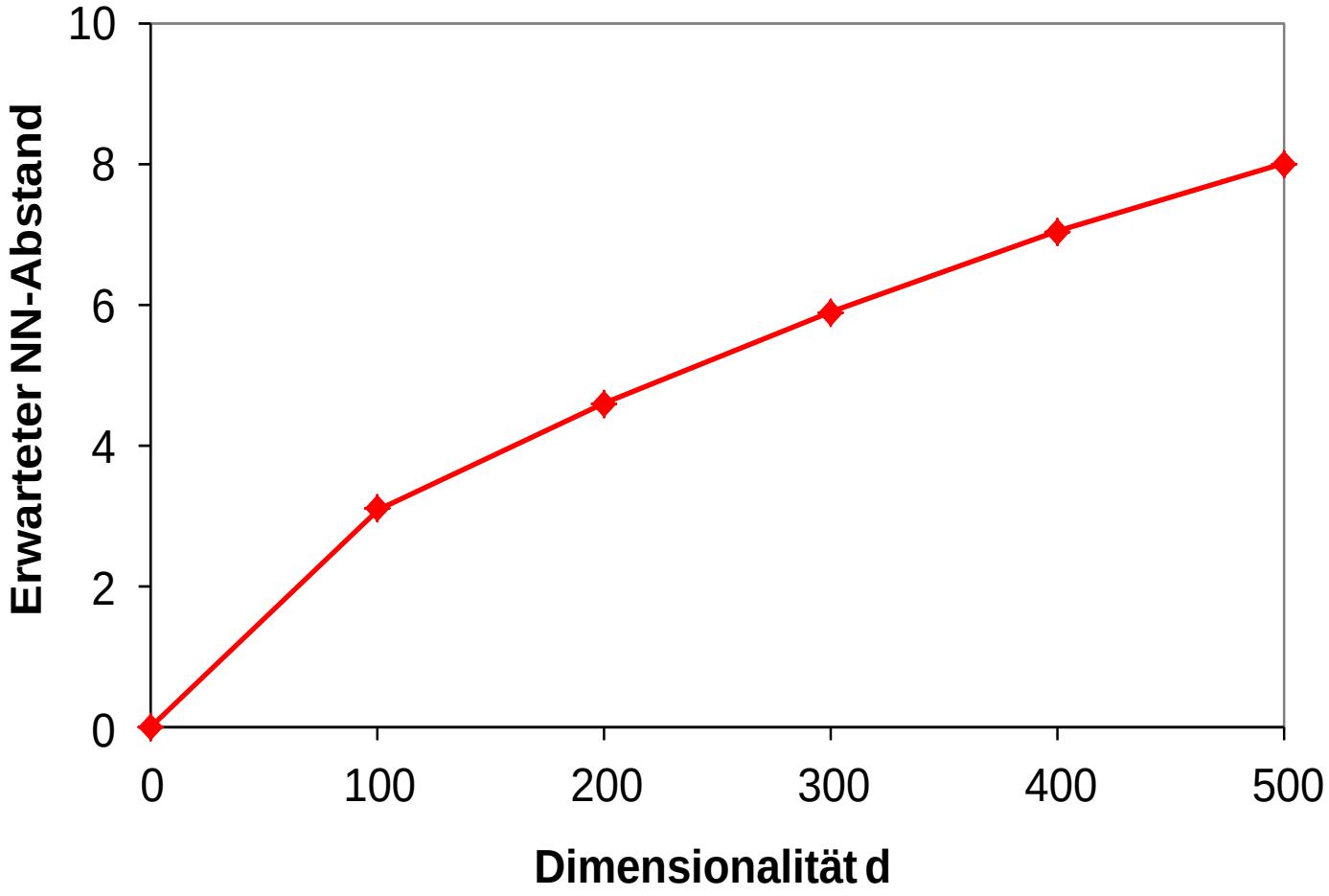
$$E\left[Q, nn^{dist}\right] = \int_0^\infty r \cdot \frac{\partial P[Q,r]}{\partial r} dr$$

- Erwartete NN-Distanz für beliebigen Punkt:

$$E\left[nn^{dist}\right] = \int_{Q \epsilon \Omega} E\left[Q, nn^{dist}\right] dQ$$

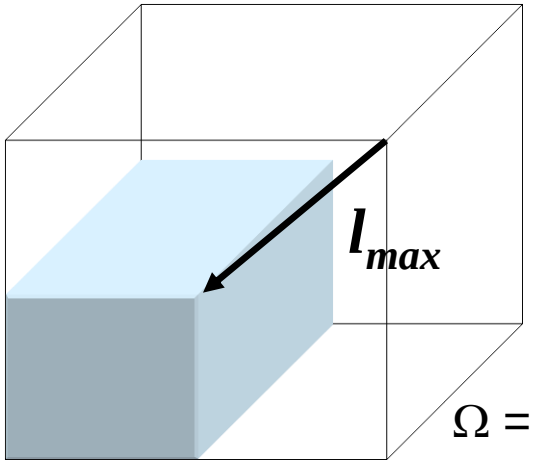
Hierarchische Datenstrukturen sind nicht effektiv (3)

- Formel für E[NN-Abstand] wurde eben hergeleitet:



Hierarchische Datenstrukturen sind nicht effektiv (4)

- Beispiel – kD-Baum:
  Splits nur entlang *d'* Dimensionen.



$\Omega = [0,1]^3$

- Maximalabstand zum Block – zufälliger Punkt, wenn d'<d:

$$l_{\max} = \frac{1}{2} \cdot \sqrt{d'}$$

Hierarchische Datenstrukturen sind nicht effektiv (5)

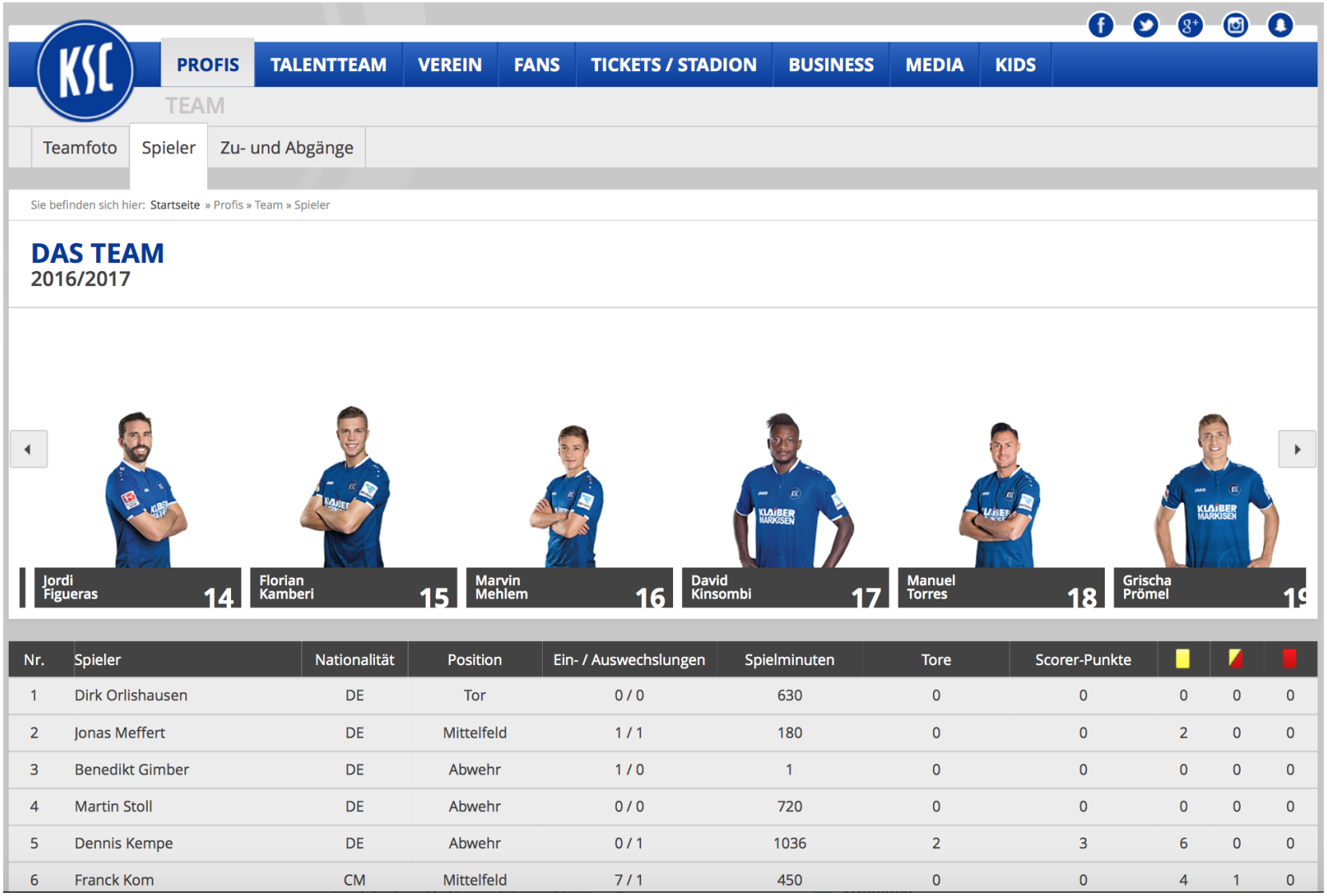
- l_{max} < E[NN-Abstand] für große *d*.
- D. h. jede NN-Kugel schneidet jeden Block.
- D. h. alle Blätter müssen betrachtet werden.
- Bäume helfen nicht.
- *Curse of dimensionality*.

- Läßt sich verallgemeinern
  für Objekte (fast) beliebiger Form,
- gleichverteilte Daten,
  aber – gleicher Effekt mit Realwelt-Daten.

Hierarchische Datenstrukturen sind nicht effektiv (6)

- Verallgemeinerung des gerade Gesagten:
  - Bei sehr, sehr vielen Dimensionen ist Abstand zweier Datenobjekte A, B fast gleich dem zweier anderer Datenobjekte, unter schwachen Annahmen.
  - D. h. es gibt gar keine echten Outlier!
  - Outlier-Algorithmen liefern mehr oder weniger zufälliges Objekt. D. h. Outlier-Suche in hochdimensionalen Merkmalsräumen weitgehend sinnlos.

Illustration



PROFIS TALENTTEAM VERein FANS TICKETS / STADION BUSINESS MEDIA KIDS

TEAM

Teamfoto | Spieler | Zu- und Abgänge

Sie befinden sich hier: Startseite » Profis » Team » Spieler

DAS TEAM
2016/2017

Jordi Figueras 14 | Florian Kamberi 15 | Marvin Mehlem 16 | David Kinsombi 17 | Manuel Torres 18 | Grischa Prömel 19

Nr.	Spieler	Nationalität	Position	Ein- / Auswechslungen	Spielminuten	Tore	Scorer-Punkte			
1	Dirk Orlishausen	DE	Tor	0 / 0	630	0	0	0	0	0
2	Jonas Meffert	DE	Mittelfeld	1 / 1	180	0	0	2	0	0
3	Benedikt Gimber	DE	Abwehr	1 / 0	1	0	0	0	0	0
4	Martin Stoll	DE	Abwehr	0 / 0	720	0	0	0	0	0
5	Dennis Kempe	DE	Abwehr	0 / 1	1036	2	3	6	0	0
6	Franck Kom	CM	Mittelfeld	7 / 1	450	0	0	4	1	0

Bundesliga-Datenbank – weitere mögliche Attribute

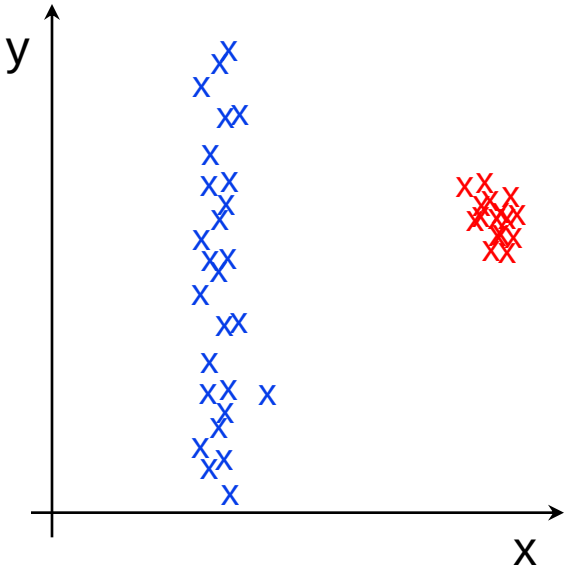
- Tore in der Nachspielzeit
- Tore in den letzten 15 Minuten des Spiels
- Tore mit dem Kopf
- Anteil gewonnene Zweikämpfe
- dto. im eigenen Strafraum
- dto. im gegnerischen Strafraum
- dto. in der eigenen Hälfte
- Angekommene Zuspiele
- dto. über 10 Meter
- dto. in der gegnerischen Hälfte
- …

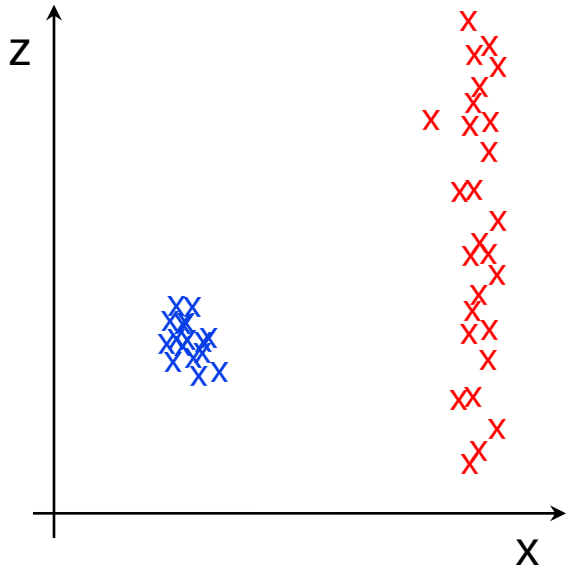
BL-Datenbank – Diskussion

- Kein Spieler ist Ausreißer bezüglich all dieser Kriterien.
- Nimmt man alle Kriterien
  als Grundlage für die Abstandsberechnung,
  ist – unter bestimmten nicht-restriktiven Annahmen
  – der Abstand von Spieler A zu Spieler B
  in etwa gleich dem von A zu Spieler C!
  (A, B, C zufällig gewählt.)
- Spieler kann jedoch Ausreißer
  bezüglich ein paar Kriterien sein.
- Konsequenz:
  Nur erfolgversprechende Teilräume
  nach Ausreißern absuchen.

Probleme mit hochdimensionalen Räumen

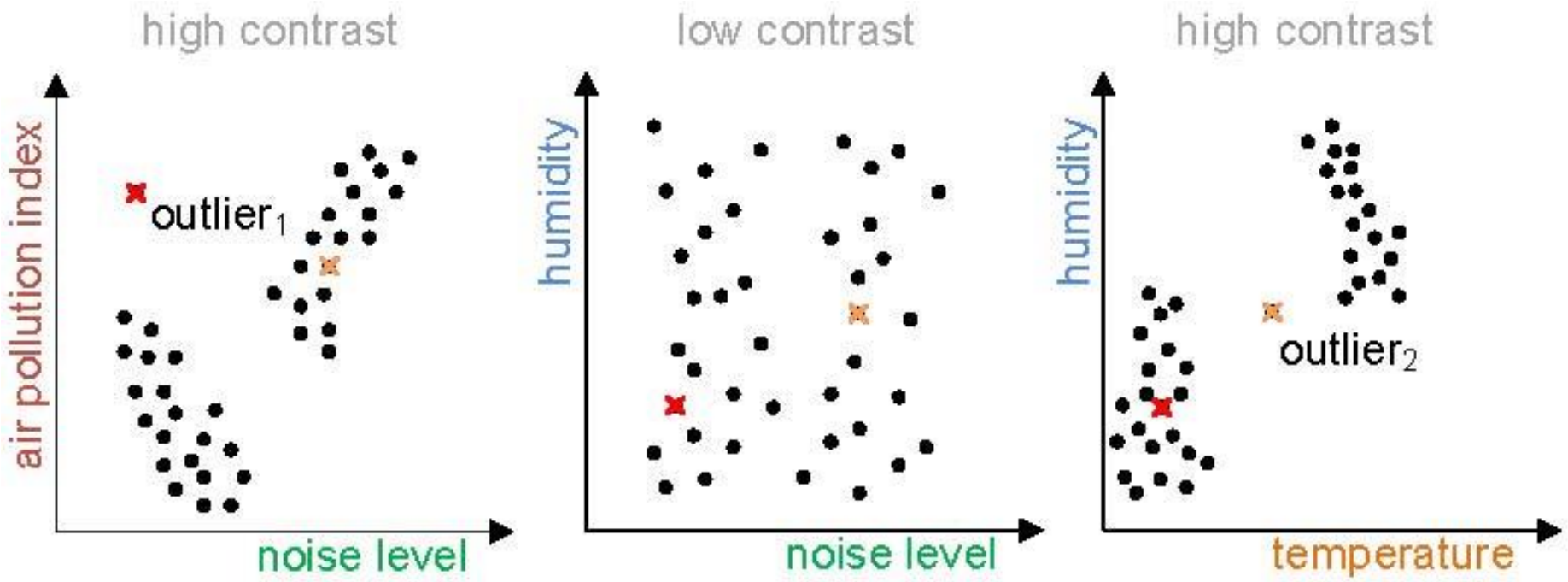
Interessante Cluster sind i. d. R. nicht Cluster in allen Dimensionen.





Illustration

- Relation – Attribute: Grad der Luftverschmutzung, Lärm, Luftfeuchtigkeit, Temperatur, …



Illustration

- Relation – Attribute: Grad der Luftverschmutzung, Lärm, Luftfeuchtigkeit, Temperatur, …

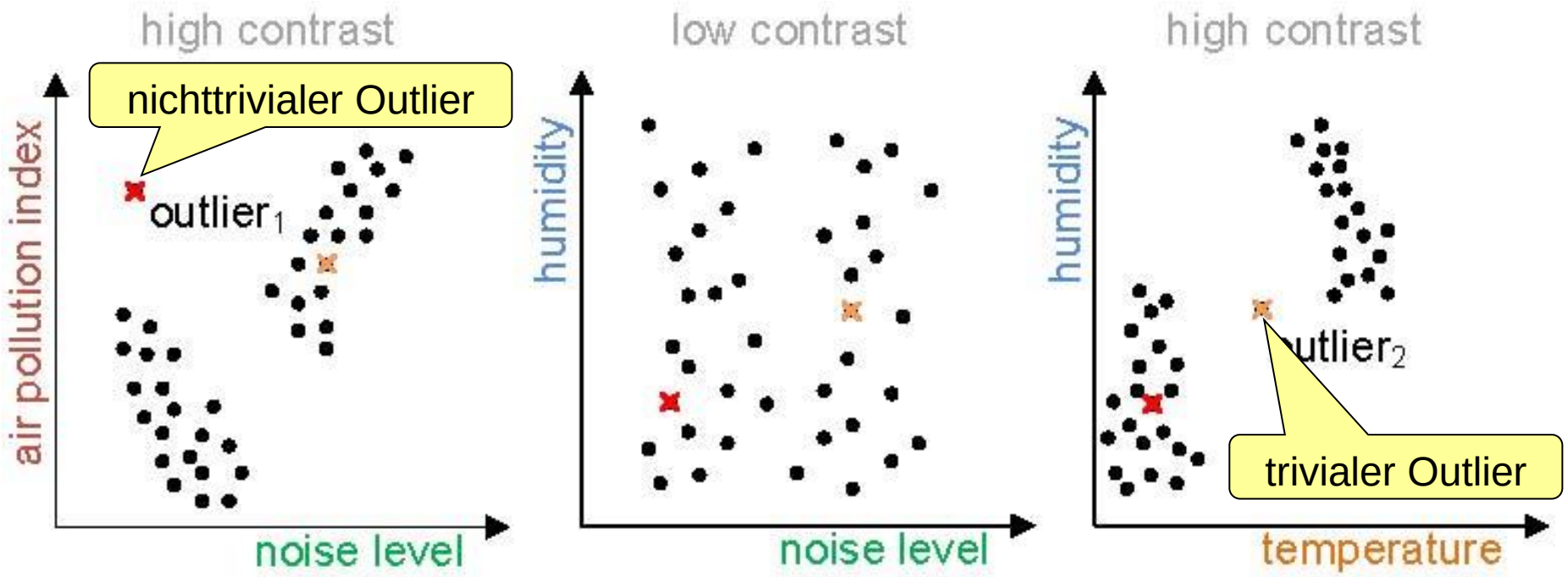


Illustration – Diskussion

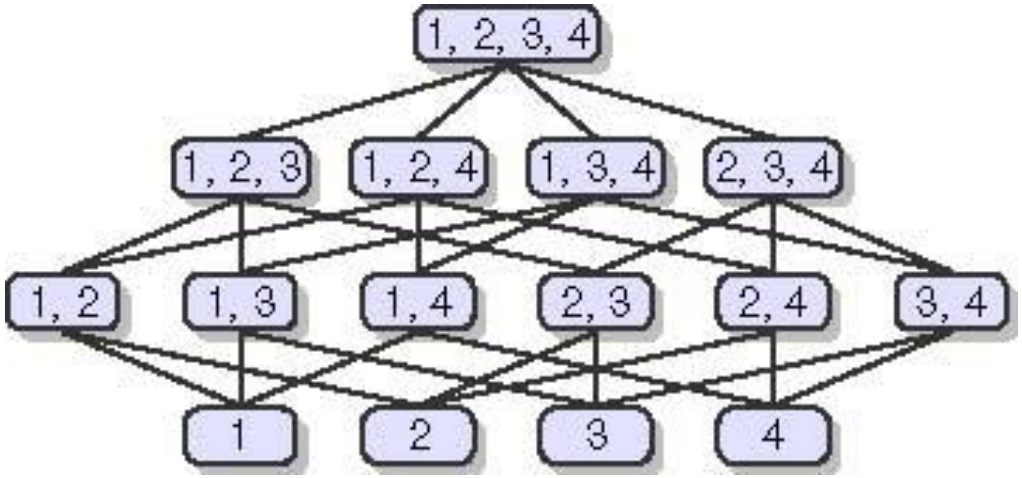
- Outlier erscheinen als solche nur in manchen Teilräumen.
- Manche Teilräume enthalten keine Ausreißer.
- Teilräume unterschiedlicher Dimensionalität enthalten Ausreißer.
- Unterscheidung zwischen *trivialen* und *nichttrivialen Ausreißern*.
  - trivial – Objekt ist bereits in Teilraum Ausreißer.
  - nichttrivial – Gegenteil.

$\Rightarrow$ Maß gesucht dafür, wie interessant/relevant Teilräume sind.

$\Rightarrow$ Wie findet man relevante Teilräume?

Suche nach Teilräumen – Subspace Search

- Exponentiell viele Teilräume $\mathcal{P}(A)$
- Auswahl relevanter Teilräume $RS \subset \mathcal{P}(A)$



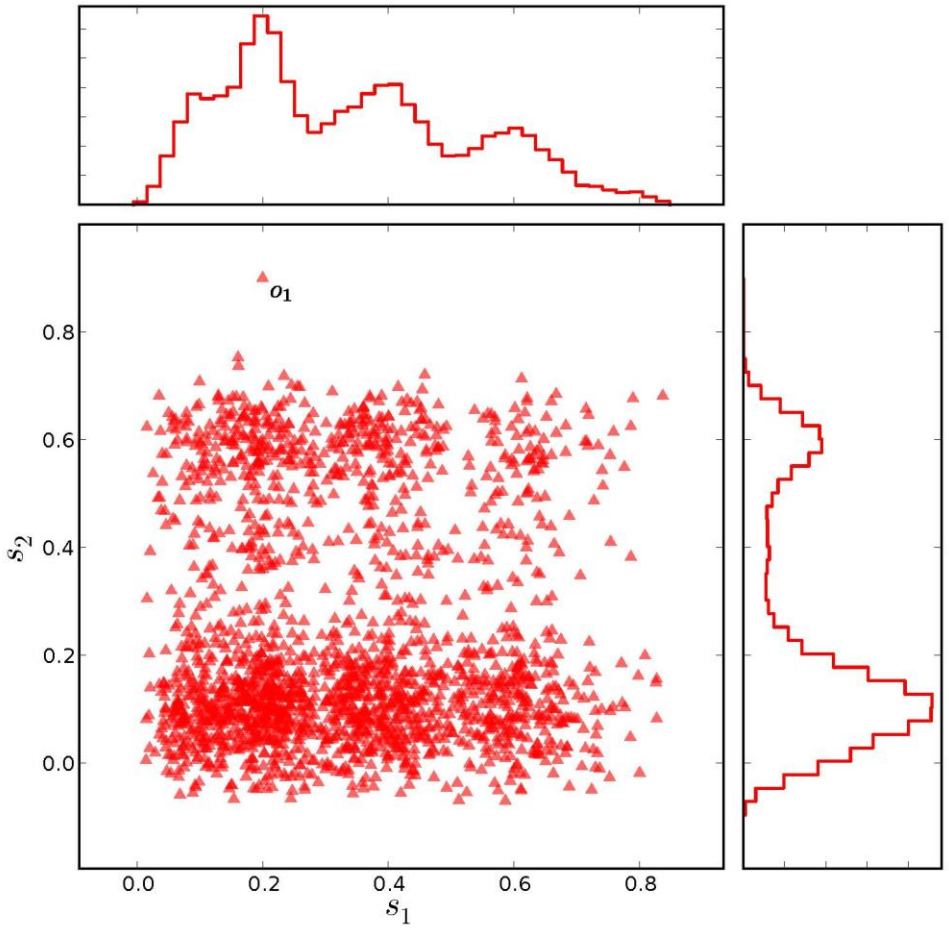
Subspace Search

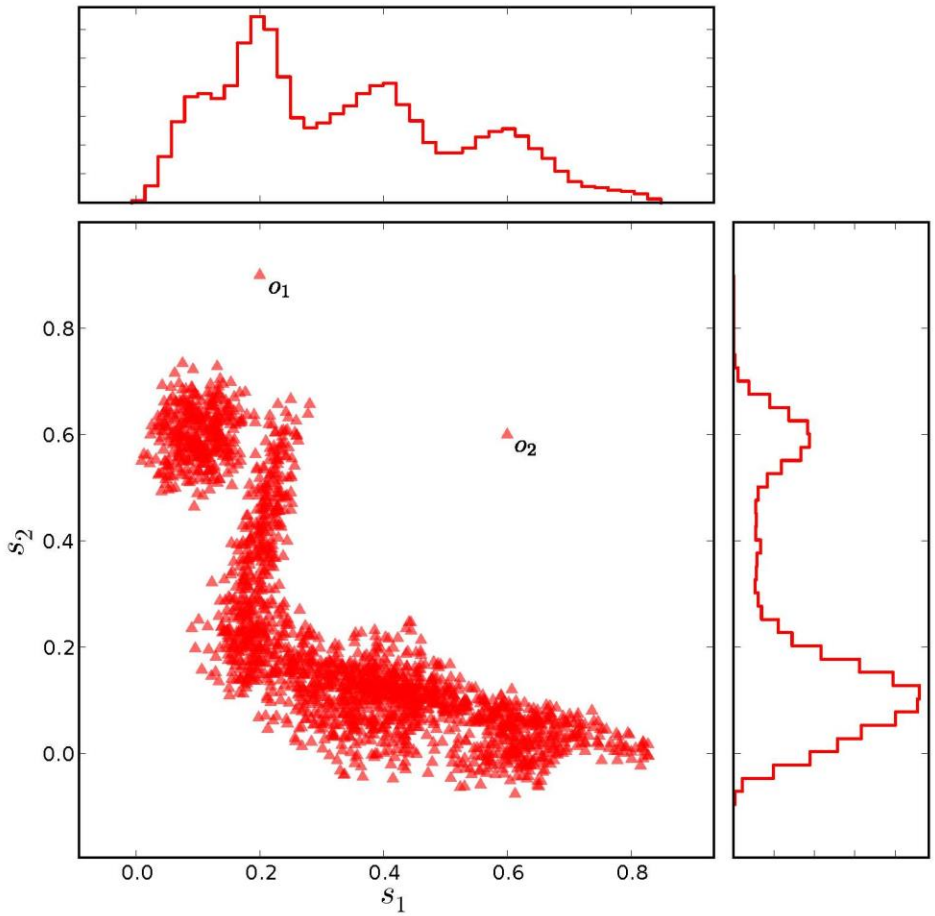
- Vorgehen im Folgenden:
  - Heuristische Suche nach Teilräumen, die wahrscheinlich Ausreißer enthalten. (Subspace Search)
  - Diese (und nur diese) dann mit konventionellen Methoden (z. B. DBSCAN) nach Ausreißern absuchen. [Nicht Thema unserer Forschung.]
- Zentrale Frage: Wie entscheiden, ob Teilraum wahrscheinlich Ausreißer enthält? (Heuristik selbst spreche ich nicht an.)

Vergleich mit anderen Ansätzen

	Clustering	Outlier Mining
Fixed Space (Full Space)	● DBSCAN [KDD 1996] ● …	● LOF [SIGMOD 2000] ● LOCI [ICDE 2003] ● …
Coupled Method	● CLIQUE [SIGMOD 1998] ● …	● OUTRES [ICDE 2011] ● …
Decoupled Method	● Enclus [KDD 1995] ● RIS [PKDD 2003]	● RandSubs [KDD 2005]

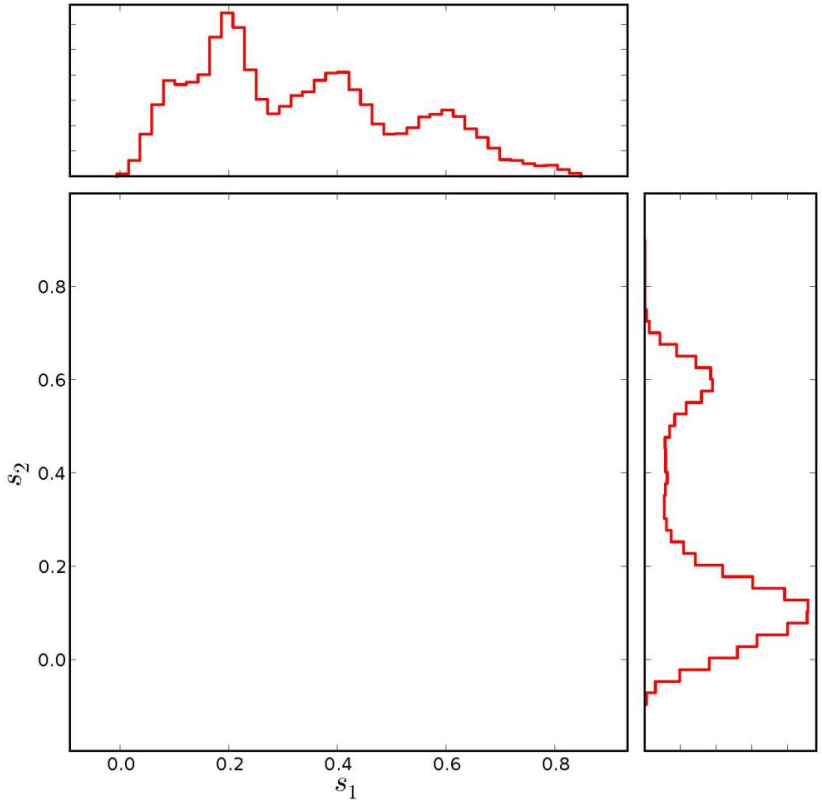
HiCS: Prinzip (1)





- 1 D Ausreißer vs. Echte 2D Ausreißer.

HiCS: Prinzip



Motivation

…

HiCS

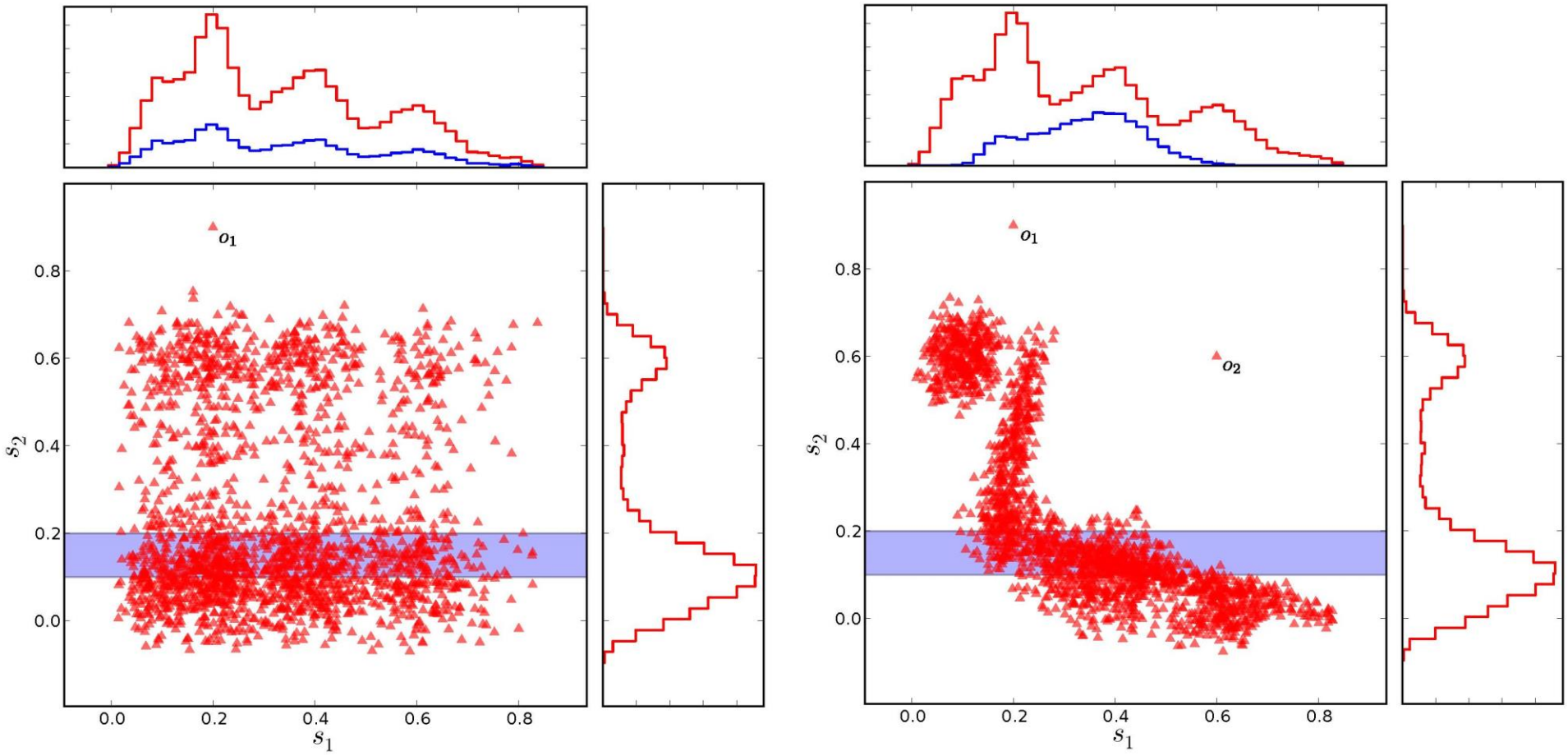
- Motivation
- Prinzip
- Formalis.
- Verfahren
- Evaluation
- Schluss

HiCS: Prinzip (2)

- Wenn Attribute nicht korreliert, sind Ausreißer in diesem Raum tendenziell eher triviale Outlier.
- Idee deshalb: Suche nach Verletzungen statistischer Unabhängigkeit. Wir nennen das *Kontrast*.
- Allgemeiner als Suche nach Korrelationen: Nicht nur
  - lineare,
  - paarweise

  Zusammenhänge.

HiCS: Prinzip (3)



- Vergleich Randverteilung mit bedingter Verteilung.

Kontrast von Teilräumen (1)

- Für Teilräume mit niedrigem Kontrast gilt:

$$\underbrace{p_{s_1|s_2,\ldots,s_d}\left(x_{s_1} \mid x_{s_2},\ldots, x_{s_d}\right)}_{p^{(c)}_{s_i|C_i}} = \frac{p_{s_1,\ldots,s_d}\left(x_{s_1},\ldots, x_{s_d}\right)}{p_{s_2,\ldots,s_d}\left(x_{s_2},\ldots, x_{s_d}\right)} = \underbrace{p_{s_1}\left(x_{s_1}\right)}_{p^{(m)}_{s_i}}$$

- Links:
  - Bedingte Wahrscheinlichkeit (blau)
  - s_2 ist y, s_1 ist x.
- Rechts:
  - Zähler ist Wahrscheinlichkeit im 2D Raum.
  - Nenner ist Wahrscheinlichkeit im 1D Raum (y in unserem Fall).

Kontrast von Teilräumen (2)

- Kontrast – Definition:
  Evaluiere einen Teilraum
  mit Monte Carlo Algorithmus mit M Iterationen:
  - Wähle zufällig isoliertes Attribut s_i.
  - Generiere zufällig Menge von Bedingungen C_i.
  - Überprüfe, ob Gleichung auf voriger Folie verletzt ist.

$$\text{contrast}(S) \equiv \frac{1}{M} \sum_{i}^{M} \text{deviation}\left(p_{s_i}^{(m)}, p_{s_i|C_i}^{(c)}\right)$$

- Wie generiert man die C_i?
- Wie instanziiert man $\text{deviation}\left(p_{s_i}^{(m)}, p_{s_i|C_i}^{(c)}\right)$?

Instanziierung von *deviation*

- Verwendung etablierter Methoden
  für den Vergleich von Stichproben (statistischer Test).
- Null Hypothese H_0:
  „Den Stichproben liegen die gleichen Verteilungen zugrunde.“
- Tests sagen uns, wann wir Hypothese ablehnen sollten.
- Mögliche Instanziierungen:
  - Welch-t-Test
  - Kolmogorov-Smirnov

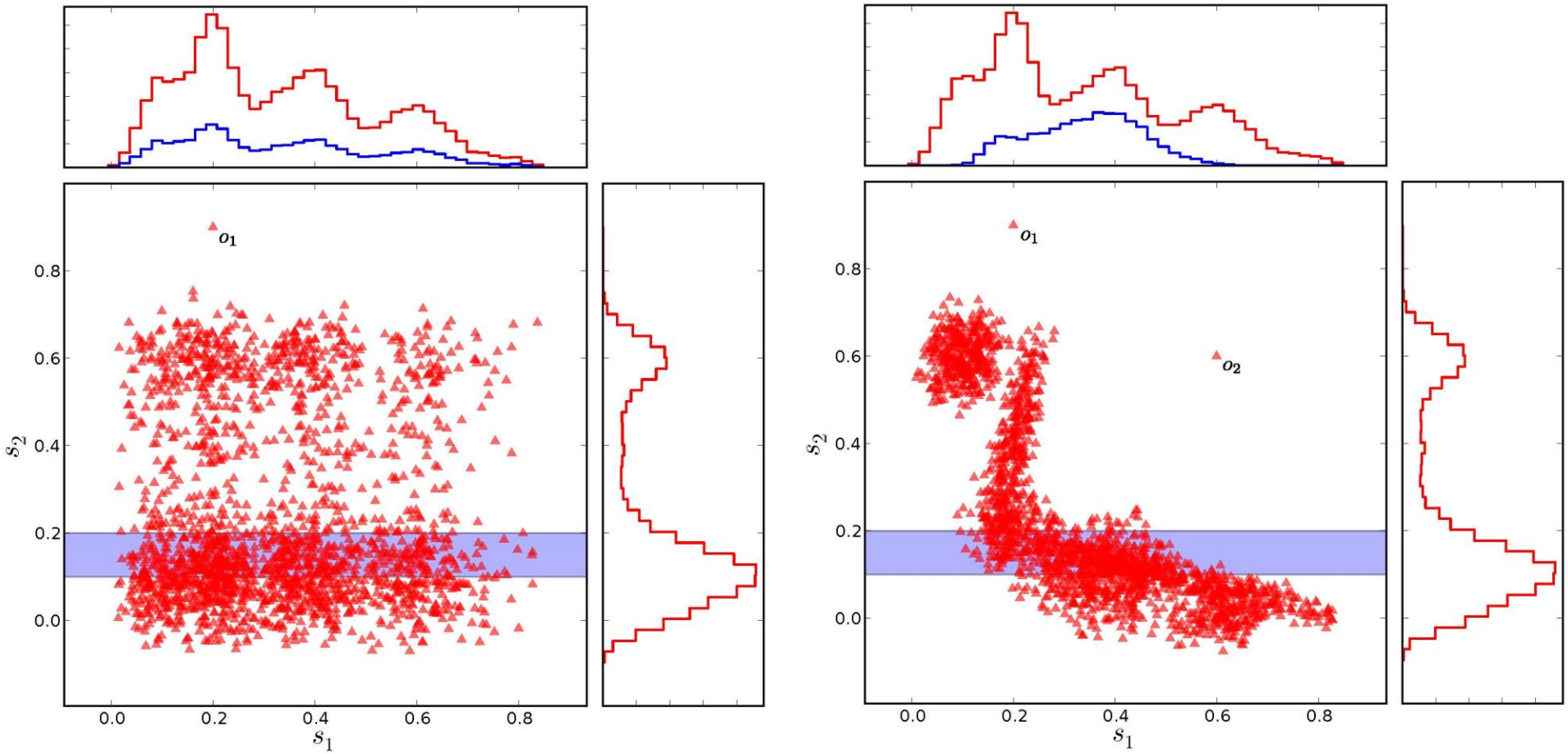
Motivation

…

HiCS

- Motivation
- Prinzip
- Formalis.
- Verfahren
- Evaluation
- Schluss

HiCS: Prinzip (3)



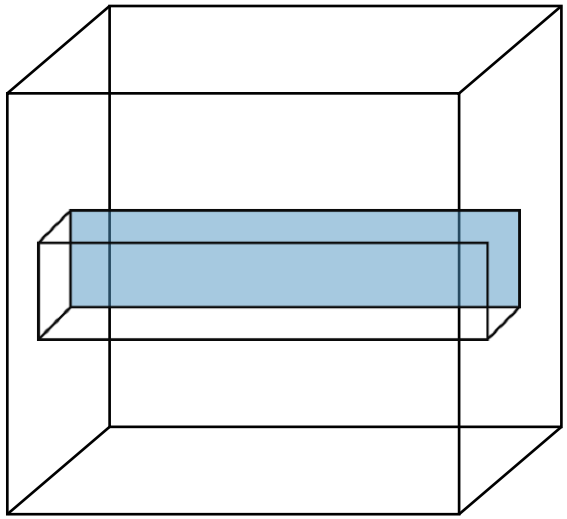
- Vergleich Randverteilung mit bedingter Verteilung.

Generierung der Bedingungen (1)

- Anschaulich gesprochen: Wie breit sollen blaue Streifen sein?
- Feste Anzahl von Objekten in blauem Streifen.
- Genauer: Erwartete Anzahl von Objekten im blauen Streifen gemäß Randverteilung soll fix sein.
- Beispiel:
  - Blauer Streifen soll 1/100 der Datenobjekte enthalten.
  - Angenommen, Randverteilung wäre Gleichverteilung.
    Dann wäre Breite des blauen Streifens 1/100 des Wertebereichs.

Generierung der Bedingungen (2)

- Beispiel:
  - Blauer Streifen soll 1/100 der Datenobjekte enthalten.
  - Angenommen, Randverteilung wäre Gleichverteilung.
  - Angenommen, Bedingung über zwei Dimensionen.
    Dann Breite 1/10 des Wertebereichs pro Dimension.

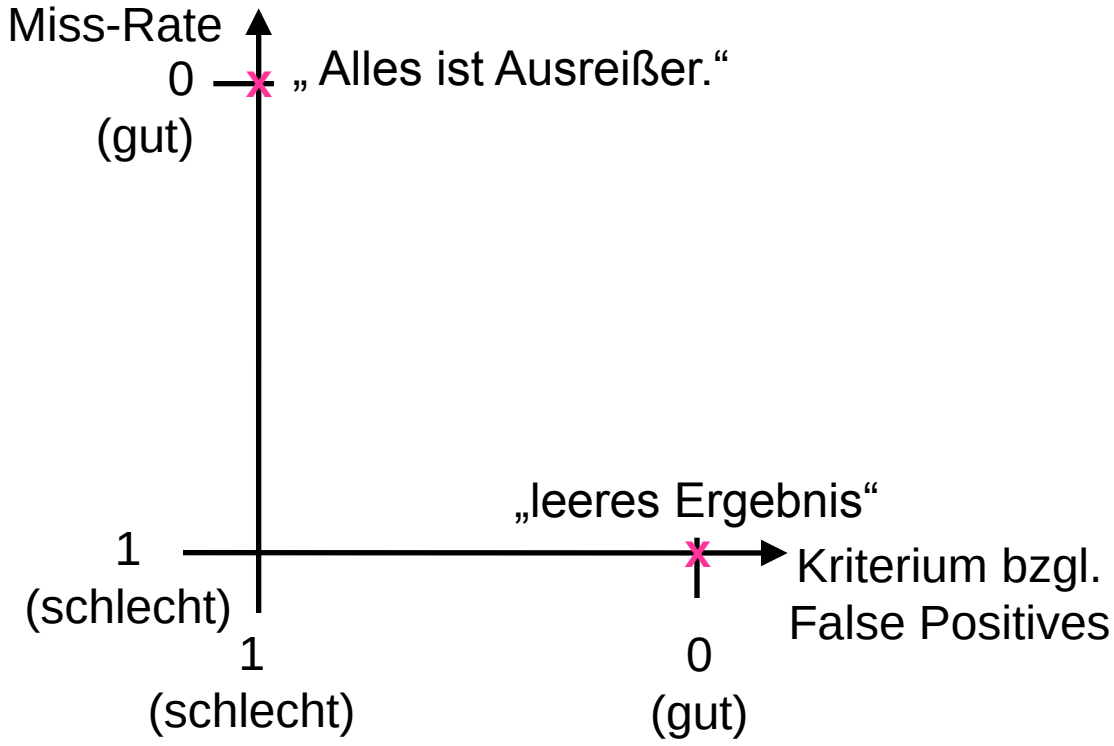


Generierung der Bedingungen (3)

- Sei S ein Teilraum mit d Dimensionen.
  Dann gilt für Anzahl der Bedingungen: $|C| = d - 1$
- Offener Punkt, wie eben illustriert:
  Bedingungen sollten an Dimensionalität
  der Teilräume anpassbar sein.
- Unser Ansatz –
  feste Größe der Stichprobe sicherstellen:
  - Objekt in Mitte des Ausschnitts zufällig wählen.
  - Breite des Ausschnitts wählen gemäß $N \cdot \sqrt[|C|]{\alpha}$
    (α ist gewünschte relative Größe der Stichprobe.)
  - Breite der Streifen berücksichtigt Randverteilung.

Evaluation (1)

- Was kann schiefgehen? Zwei Fehlerarten:
  - Verfahren zur Ausreißererkennung erkennt Ausreißer nicht. (*Miss*)
  - Verfahren sagt, Nicht-Ausreißer sei ein Ausreißer. (*False Positive*)



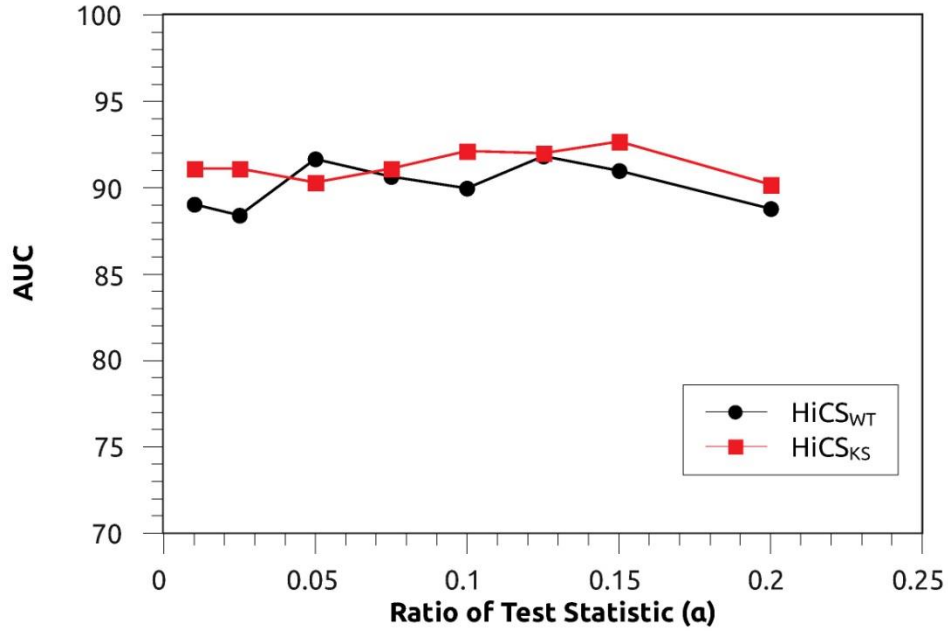
Evaluation (2)

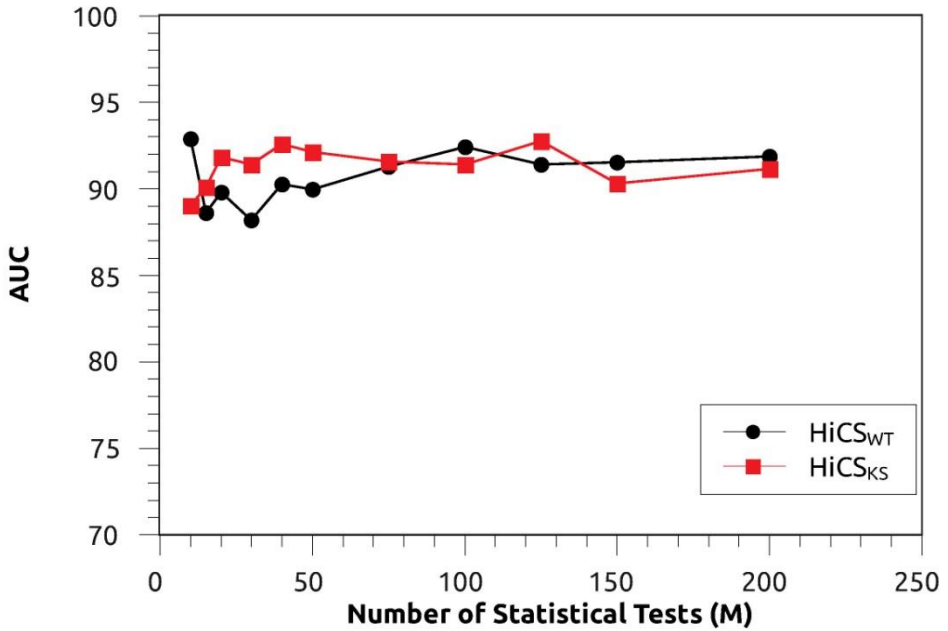
- Wichtigste Größe:
  Qualität, gemäss *Area under Curve* (AUC).

- Methode:
  Local Outlier Factor (LOF) mit fixer Parametrisierung.
- Suche nach Teilräumen:
  - keine
  - Enclus
  - RIS
  - HiCS

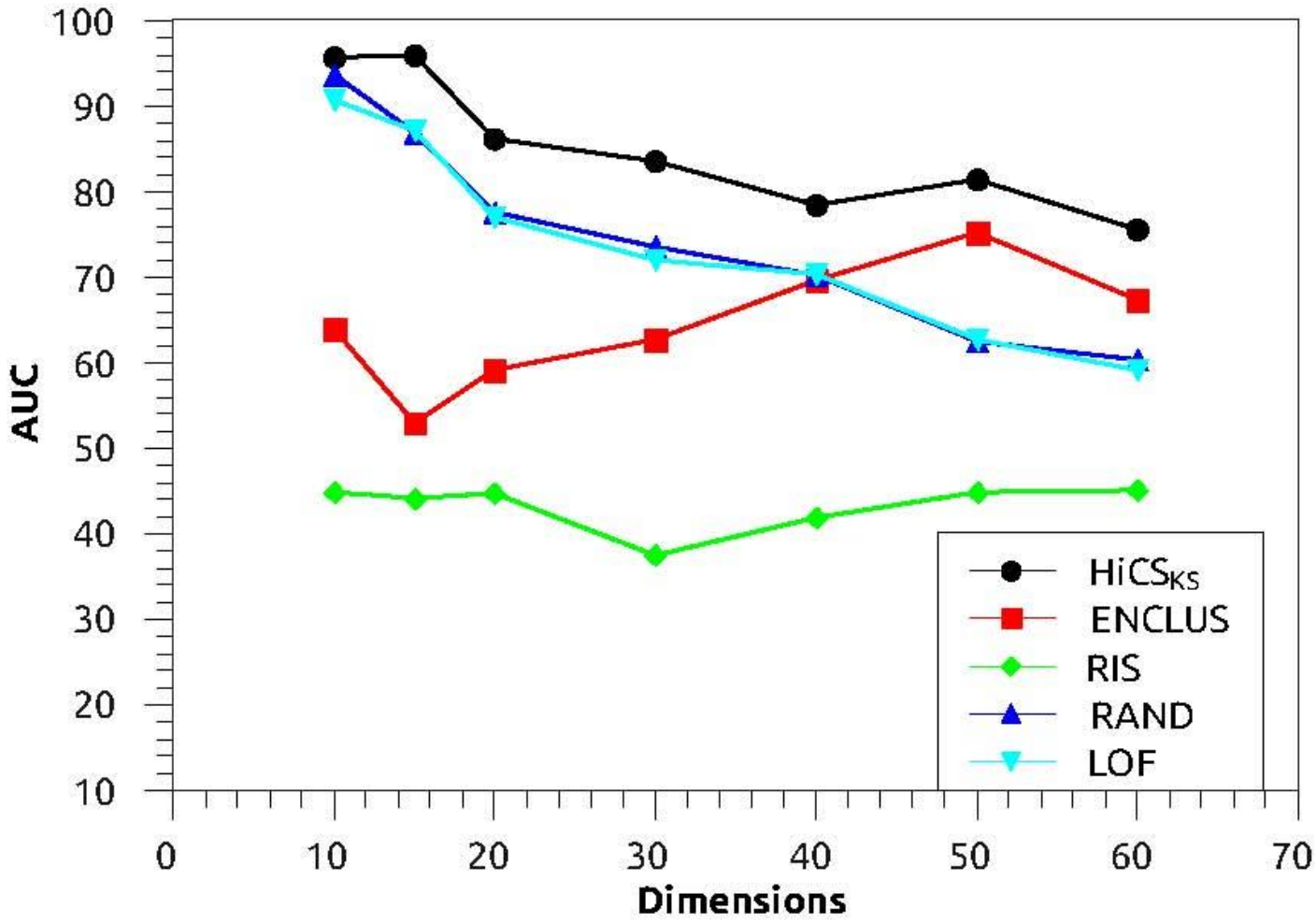
Betrachtung der Parameter

- HiCS ist im Experiment robust hinsichtlich Parametereinstellungen.





Einfluß der Dimensionalität



Motivation

…

HiCS

- Motivation
- Prinzip
- Formalis.
- Verfahren
- Evaluation
- Schluss

Experimente mit Realwelt-Daten (1)

- Acht häufig verwendete Datenbestände.
- Ergebnisqualität und Laufzeit.

Experiment	Laufzeit [sec.]				
	LOF	HiCS	Enclus	RIS	RANDSUB
Ann-Thyroid	7.1	37.2	68.1	574.0	674.0
Arrhythmia	0.5	26.4	7.9	2216.1	48.2
Breast	0.1	2.4	1.5	-	3.5
Breast (diagnostic)	0.3	15.8	11.8	14.3	28.2
Diabetes	0.3	3.3	5.9	4.0	26.2
Glass	0.0	0.2	0.3	0.1	1.7
Ionosphere	0.1	6.1	4.2	668.2	11.0
Pendigits	34.1	1194.5	2195.6	11282.7	3326.2

Experimente mit Realwelt-Daten (2)

Experiment	AUC [%]				
	LOF	HiCS	Enclus	RIS	RANDSUB
Ann-Thyroid	86.16	95.11	94.32	**95.16**	93.32
Arrhythmia	62.92	62.29	62.11	**63.61**	63.52
Breast	56.42	59.31	**59.55**	-	56.98
Breast (diagnostic)	86.94	**94.23**	94.19	90.77	87.07
Diabetes	70.98	**72.47**	71.15	71.63	71.70
Glass	76.86	80.05	79.73	**80.65**	78.48
Ionosphere	77.97	82.34	**82.37**	80.93	79.02
Pendigits	93.54	**95.04**	94.29	90.74	93.22

Motivation

…

HiCS

- Motivation
- Prinzip
- Formalis.
- Verfahren
- Evaluation
- Schluss

Schluss (1)

- Resultate:
  - Suche nach (kontrastreichen) Teilräumen entkoppelt von der nach Ausreißern.
  - Statistische Definition von Kontrast in Teilräumen.
  - Fokussierung auf echte Ausreißer in Teilräumen.

Motivation

…

HiCS

- Motivation
- Prinzip
- Formalis.
- Verfahren
- Evaluation
- Schluss


photo from Wikipedia

Schluss (2)

- Forschung des Lehrstuhls beschäftigt sich u. a. mit Subspace Search und dem Finden von Auffälligkeiten in komplexen Datenstrukturen.
- Forschungsfragen (über das hier Angesprochene hinaus) u. a.:
  - Wie kommt man zu sinnvollen Merkmalen?
  - Betrachtung von Datenströmen.
- Mehrere Anwendungen werden explizit betrachtet:
  - Predictive Maintenance von Maschinen.
  - Fehlerlokalisation in Software.
  - Energieverbrauch (des KIT).
  - Kreditkartenbetrug.

Mögliche Prüfungsfragen (1)

- Warum kann man für räumliche Anfragen nicht ohne weiteres auswerten, wenn man für jede Dimension separat einen B-Baum angelegt hat?
- Wie funktioniert der Algorithmus für die Suche nach den nächsten Nachbarn/ nach den k nächsten Nachbarn mit Bäumen wie dem kd-Baum?
- Warum werden bei der NN-Suche nur genau die Knoten inspiziert, deren Zonen die NN-Kugel überlappen?

Mögliche Prüfungsfragen (2)

- Was ist ein Outlier?
- Was ist ein Zusammenhang zwischen k-NN Suche mit Bäumen wie dem kd-Baum und Outlier-Berechnung?
- Warum ist die Zuordnung Dichte-erreichbarer Punkte mit DBSCAN nichtdeterministisch?
- Warum sind hierarchische Datenstrukturen in hochdimensionalen Merkmalsräumen für die k-NN Suche nicht das Mittel der Wahl? (Im Prinzip die Argumentation aus der Vorlesung wiedergeben können.)
- Was bedeutet Subspace Search?
- Geben Sie die Unterscheidung zwischen trivialen und nichttrivialen Outliern aus der Vorlesung wieder.
- Was genau bedeutet Kontrast im Kontext von HiCS?

Literatur (1)

- Gísli R. Hjaltason, Hanan Samet: Ranking in Spatial Databases. Symposium on Large Spatial Databases, 1995. http://citeseer.ist.psu.edu/hjaltason95ranking.html
- Martin Ester et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. Second International Conference on Knowledge Discovery and Data Mining (KDD), 1996. http://www.dbs.ifi.lmu.de/Publikationen/Papers/KDD-96.final.frame.pdf.
- Roger Weber, Hans-Jörg Schek, Stephen Blott: A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. 24th International Conference on Very Large Data Bases Pages, 1998.
- Kevin S. Beyer et al.: When Is "Nearest Neighbor" Meaningful? 7th International Conference on Database Theory, 1999.

Literatur (2)

- Fabian Keller, Emmanuel Müller, Klemens Böhm: HiCS: High Contrast Subspaces for Density-Based Outlier Ranking. 28th International Conference on Data Engineering (ICDE), 2012.