LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Institut für Informatik

Discovering Popular Routes from Trajectories

MediaQ - Hauptseminar SS 2015

Markus Götz
Muhibullah Nasari
Sandro Kurpiers

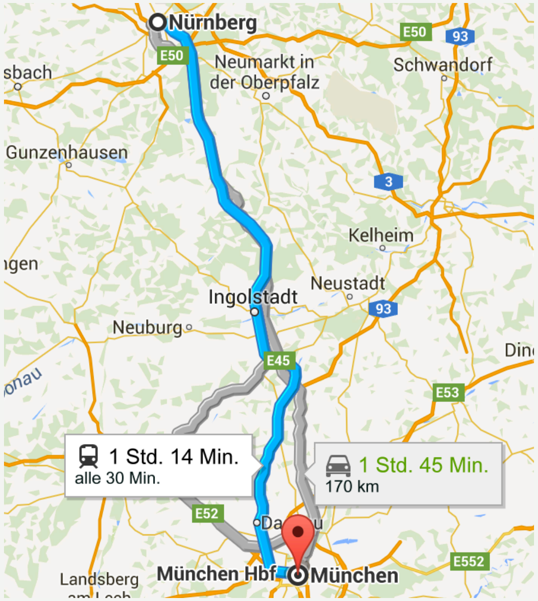
22.05.2015

Agenda

1. Einführung in das Paper
2. Coherence Expanding
3. Absorbing Markov Chain
4. Maximum Probability Product
5. Idee für ein MediaQ Projekt

Einführung in die Thematik des Paper

Meistens gibt es Auskunft über die kürzeste / schnellste Route zwischen zwei Orten



Wie aber kann die beliebteste Route (MPR) zwischen zwei Orten gefunden werden?

Einführung in die Thematik des Paper

Warum die beliebteste Route?

- Nützlich für Leute die an fremde Orte reisen
  ➔ Touristen die viel sehen möchten

- Hilfreich für LKW-Fahrer um qualitativ hochwertigere Straßen zu finden

Einführung in die Thematik des Paper

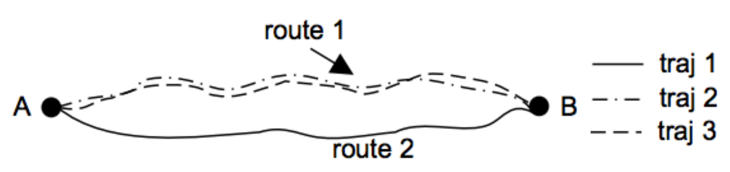
Wie erhält man die beliebteste Route?

- Anhand des Start- und Zielorts können alle Routen ausgemacht werden die diese beiden Orten verbinden
- Die Anzahl der Trajektorien für jede dieser Routen wird gezählt

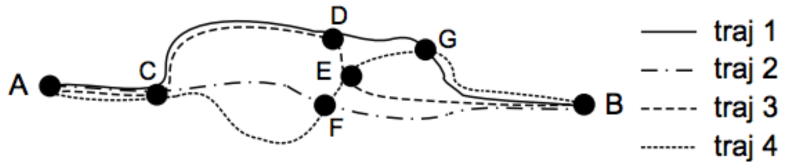
→ Route mit größter Anzahl gilt als beliebteste

Einführung in die Thematik des Paper

Beispiele



(a)



(b)

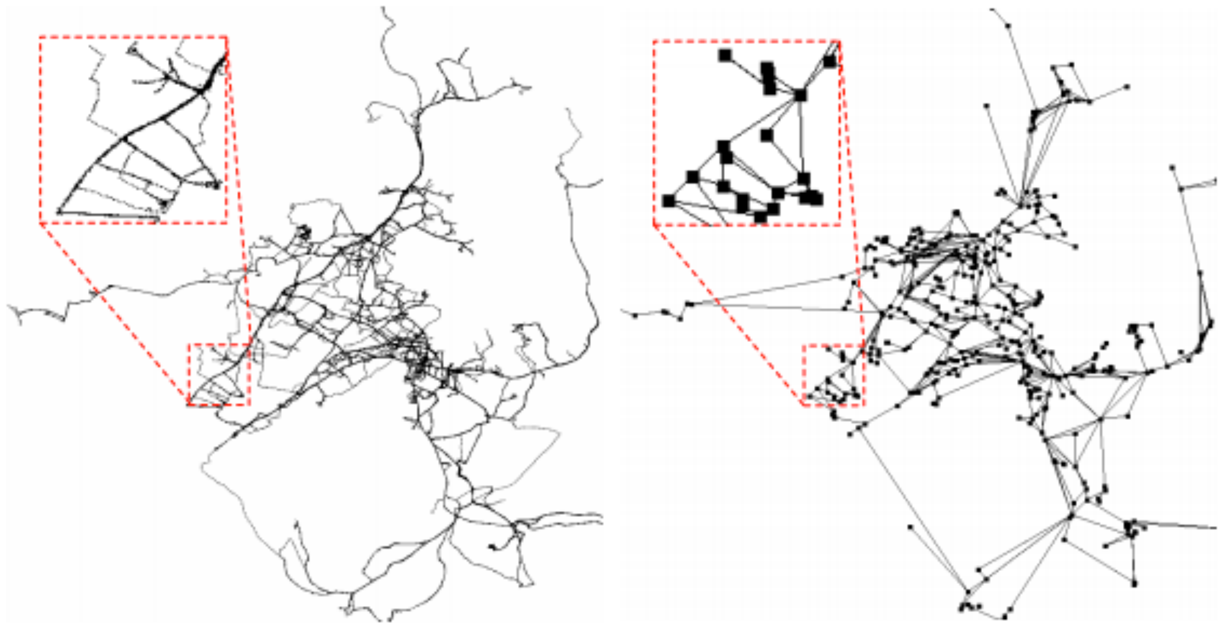
➔ **Funktion zur Berechnung der Popularität einer Route wird benötigt**

Erstellung eines Transfer Networks

- Jeder Knoten im Transfer Network gilt als "significant location" (SL) – jede Trajektorie zwischen zwei Knoten gilt als Transferkante
- Transfer-Wahrscheinlichkeit für jede SL zum Zielort wird errechnet und gilt als Indikator für die Beliebtheit einer Route

➔ Beliebtheit einer Route ist das Produkt aus Transferwahrscheinlichkeiten aller SL auf einer Route

Erstellung eines Transfer Networks



(a) Distribution of Trajectory Points (b) Transfer Network

Coherence Expanding Algorithmus

```
Algorithm 1: Coherence Expanding
input : A set of trajectory points P; Threshold τ, φ;
output: clusters[]
for each point p ∈ P do
    if p.classified=false then
        p.classified ← true;
        cluster = expand(p);
        if cluster.size ≥ φ then
            clusters.add(cluster);
return clusters;
```

```
Algorithm 2: expand(p)
input : A point p
output: A set of points result
Queue seeds ← new Queue();
seeds.add(p);
result.add(p);
while seeds ≠ null do
    seed ← seeds.pop();
    points ← rangeQuery(seed, radius);
    for i=0 ; i < points.size ; i++ do
        pt ← points.get(i);
        if pt.classified=false ∧ coh(seed, pt) ≥ τ then
            seeds.add(pt);
            result.add(pt);
return result;
```

Absorbing Markov Chain

Aufstellung einer transition matrix P

$$P = \begin{array}{c|cccc} & n_1 & n_2 & \cdots & n_m \\ \hline n_1 & P(1,1) & P(1,2) & \cdots & P(1,m) \\ n_2 & P(2,1) & P(2,2) & \cdots & P(2,m) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n_m & P(m,1) & P(m,2) & \cdots & P(m,m) \end{array}$$

$$P(i,j) = \begin{cases} 1 & \text{if } n_i \text{ is an absorbing state \& } i = j \\ Pr_d(n_i \to n_j) & \text{if } n_i \text{ is a transient state \& } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Canonische Form

$$P^t = \begin{array}{c|cc} & \mathrm{TR} & \mathrm{ABS} \\ \hline \mathrm{TR} & Q^t & * \\ \mathrm{ABS} & 0 & \mathrm{I} \end{array}$$

Absorbing State (ABS)
Transition State (TR)

$Pr_d(n_i \to n_j)$ = Turning probability

Absorbing Markov Chain

Transferwahrscheinlichkeit $n\downarrow i \rightarrow d$ in t Schritten:

$$\begin{aligned} Pr^t(n_i \rightarrow d) = &\quad \sum_{j=1}^{t} p^j_{n_i,d} \\ = &\quad \sum_{j=1}^{t} \sum_{n_k \in \mathrm{TR}} \left(P^{j-1}(i,k) \cdot P(k,d)\right) \end{aligned}$$

Vektor mit allen Transferwahrscheinlichkeiten zu einem best. Ziel:

$$V = \left[Pr^t(n_1 \rightarrow d), Pr^t(n_2 \rightarrow d), \cdots, Pr^t(n_l \rightarrow d)\right]^T$$

Absorbing Markov Chain

Algorithm 3: Deriving Transfer Probability

input : A transfer network $G(N, E)$
output: A vector V for each node $\in N$

for *each transfer node* $n_i \in N$ **do**
    set n_i as the destination;
    construct the transition matrix P by Equation 5;
    re-organize P in a canonical form;
    acquire Q, S from P;
    derive V by Equation 10;
    store V;

Maximum Probability Product

„Transfer probability“ oder „popularity indicator“
Wahrscheinlichkeit, dass ein Knoten zum gewünschten Ziel (d) führt

$$n_i.popularity(d) = Pr^t(n_i \rightarrow d)$$

Route
Sequenz von Transfer Nodes $n_1 \rightarrow n_2 \rightarrow \cdots n_i$

Maximum Probability Product

Route Popularity (p(R))
Produkt aller popularity indicators von jedem Transfer Node

$$\rho(R) = \prod_{j=1}^{i} n_j.popularity(d)$$

Most Popular Route (MPR)
Route mit maximalem p(R)

Maximum Probability Product

Algorithm 4: Maximum Probability Product

input : A transfer network $G(N, E)$,
$N = \{n_1, n_2, \cdots, n_m\}$;
Start node s; Destination node d

output: The most popular route MPR

For all $n_i \in N$, label $L(n_i) \leftarrow 0$;
$L(s) \leftarrow 1$;
Priority Queue $PQ \leftarrow null$;
Scanned Nodes $SN \leftarrow null$;
PQ.enqueue(s);
while $PQ \neq null$ **do**
  $u \leftarrow PQ$.extractMax();
  if $u = d$ **then**
    return MPR;
  SN.add(u);
  for *each* $v \in u.adjacentNodes$ **do**
    if $L(v) < L(u) \times v.popularity(d)$ **then**
      $L(v) \leftarrow L(u) \times v.popularity(d)$;
      v.predecessor $\leftarrow u$;
      PQ.add(v);

$L(n_i)$
Maximale Wahrscheinlichkeit der Route
$S \rightarrow n_i$

Gibt es eine populärere Route nach v über u ?

MPR : d $\rightarrow n_{d-1} \rightarrow n_{d-2} \rightarrow \ldots \rightarrow s$

Maximum Probability Product

Problem:
Anwender startet auf keinem Transfer Knoten sondern auf einer Transfer Kante → 2 MPR von beiden Randpunkten

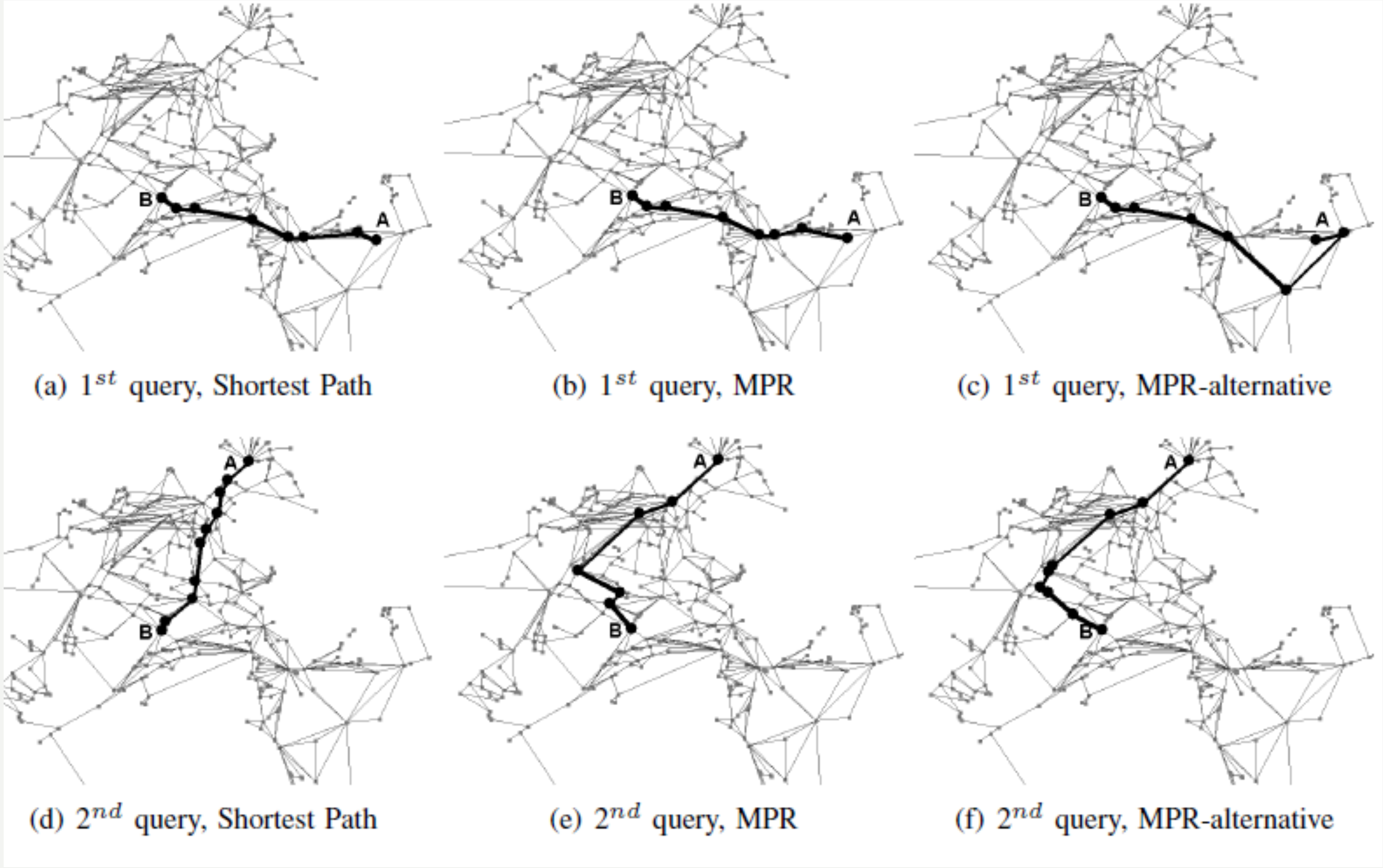
Zukünftig Kantenlänge in Betracht ziehen

MPR-alternative

Popularity indicator = $Pr_d(n_i \rightarrow n_j)$ (Turning Probability)
Problem:
Betrachtet nur Lokalität, keine t Schritte weiter → ggf. falsche Ergebnisse

Experimente

- 276 Lastwagen in Athen --> 292.394 Wegpunkte



(a) 1^{st} query, Shortest Path

(b) 1^{st} query, MPR

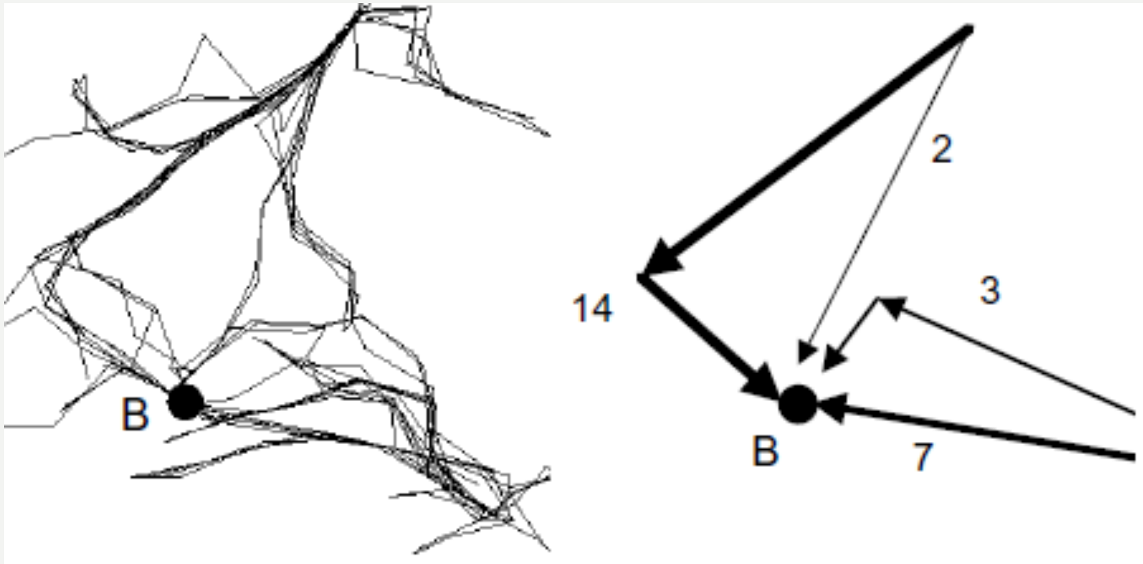
(c) 1^{st} query, MPR-alternative

(d) 2^{nd} query, Shortest Path

(e) 2^{nd} query, MPR

(f) 2^{nd} query, MPR-alternative

Experimente



(a) Trajectories (b) Summary

→ MPR nicht immer der kürzeste Weg

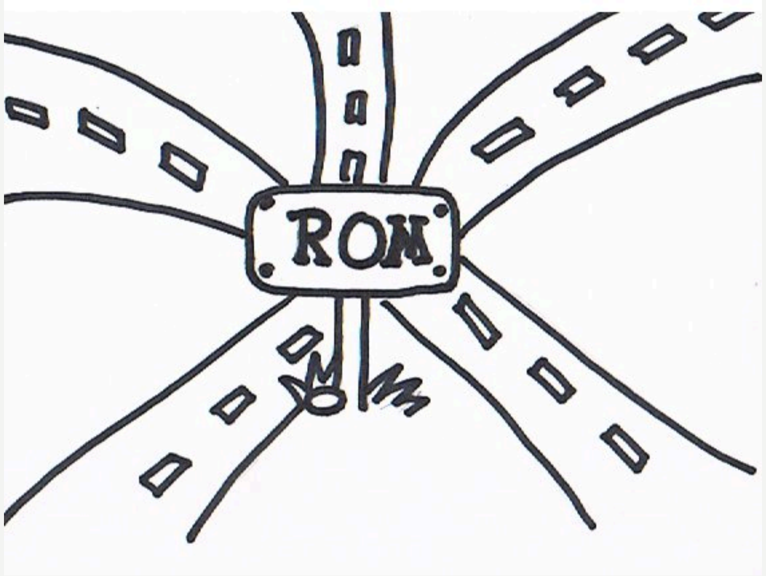
Der kreative Teil

Idee 1: Berechnung eines Panoramavideos

Problem: Zu wenige Daten/Videos

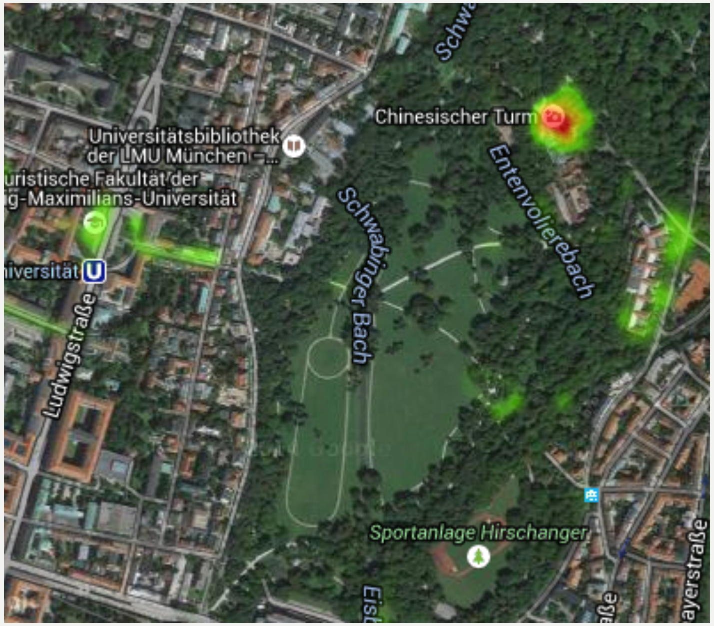
Idee 2: Most Popular Route

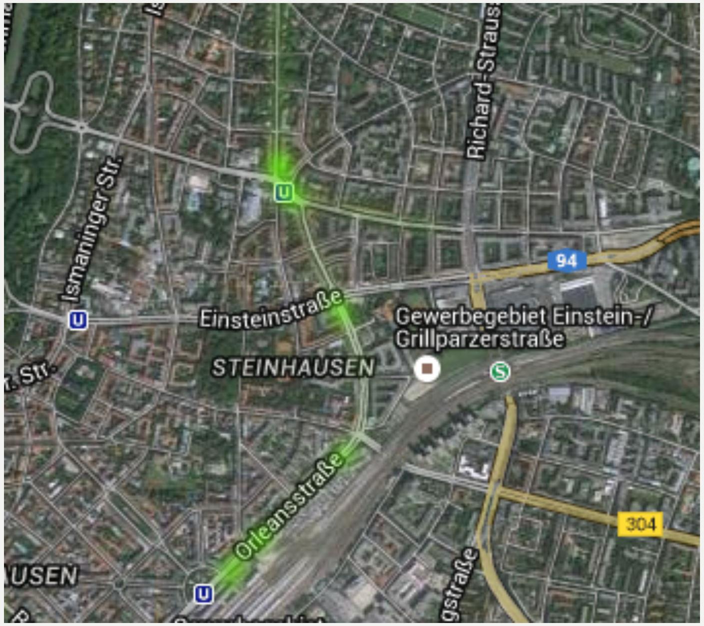
Umsetzung des Konzepts aus dem Paper



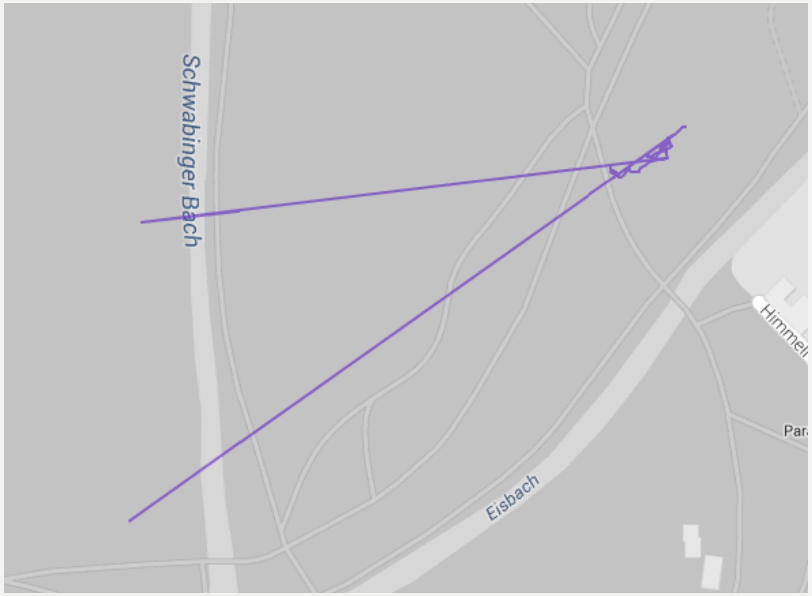
Problem: Zu wenige Daten/Pfade

Idee 3: Location Data Heatmap



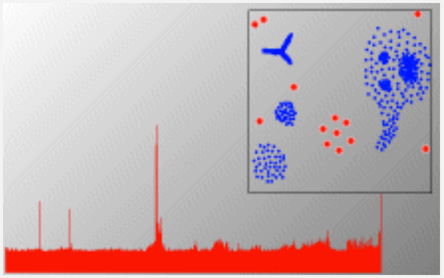


Idee 4: Outlier Detection & Clustering



Konzepte aus KDD:

- DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
- OPTICS (Ordering Points To Identify the Clustering Structure)
- LOF (Local Outlier Factors)



Mehr Infos:
http://www.dbs.informatik.uni-muenchen.de/Forschung/KDD/Clustering/

Fragen & Feedback