Algorithmen für Geographische Informationssysteme

3. Vorlesung: Map Matching

Alexander Wolff

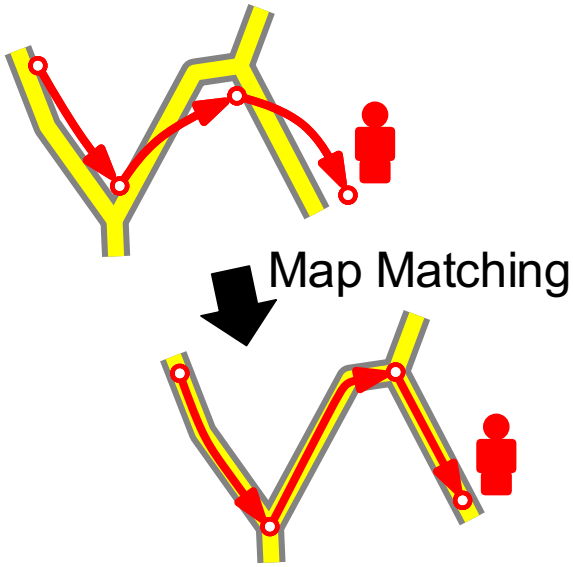
basiert auf Folien von Jan-Henrik Haunert

Map Matching

Problemformulierung

Gegeben:
Das Straßennetz als planar eingebetteter Graph $G = (V, E)$
Die GPS-Trajektorie als Folge $P = (p_1, p_2, \dots, p_n)$ von Punkten

Gesucht:
Weg in G, der **minimale Distanz** zu P hat.

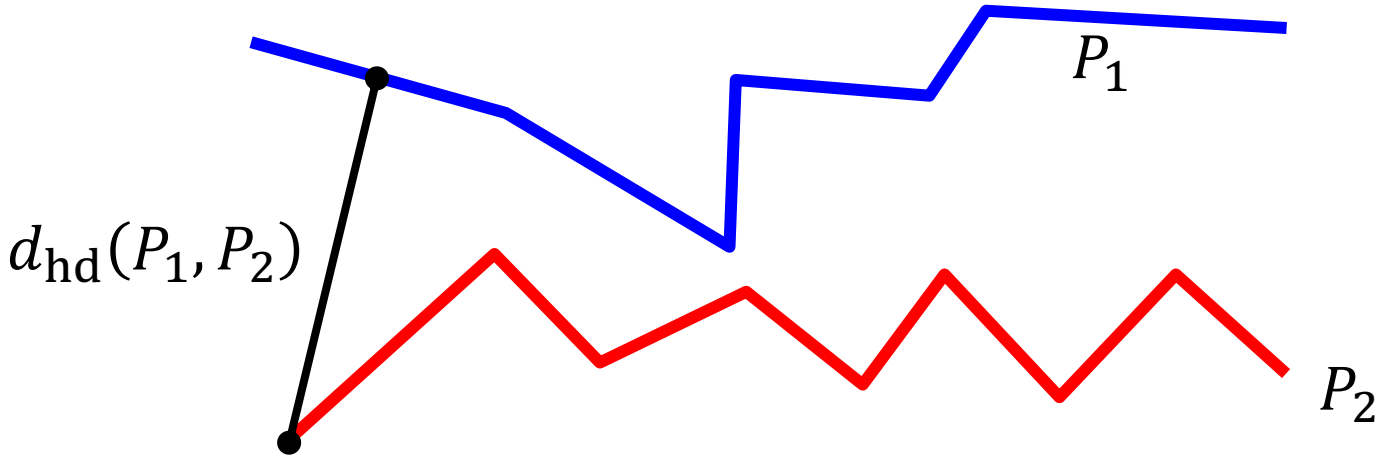


Hausdorff-Distanz

$$d_{\mathrm{hd}}(P_1, P_2) = \max\{\, \vec{d}_{\mathrm{hd}}(P_1, P_2),\, \vec{d}_{\mathrm{hd}}(P_2, P_1) \,\}$$

mit $\vec{d}_{\mathrm{hd}}(P_1, P_2) = \max\{\, d(p_1, P_2) \mid p_1 \in P_1 \}$

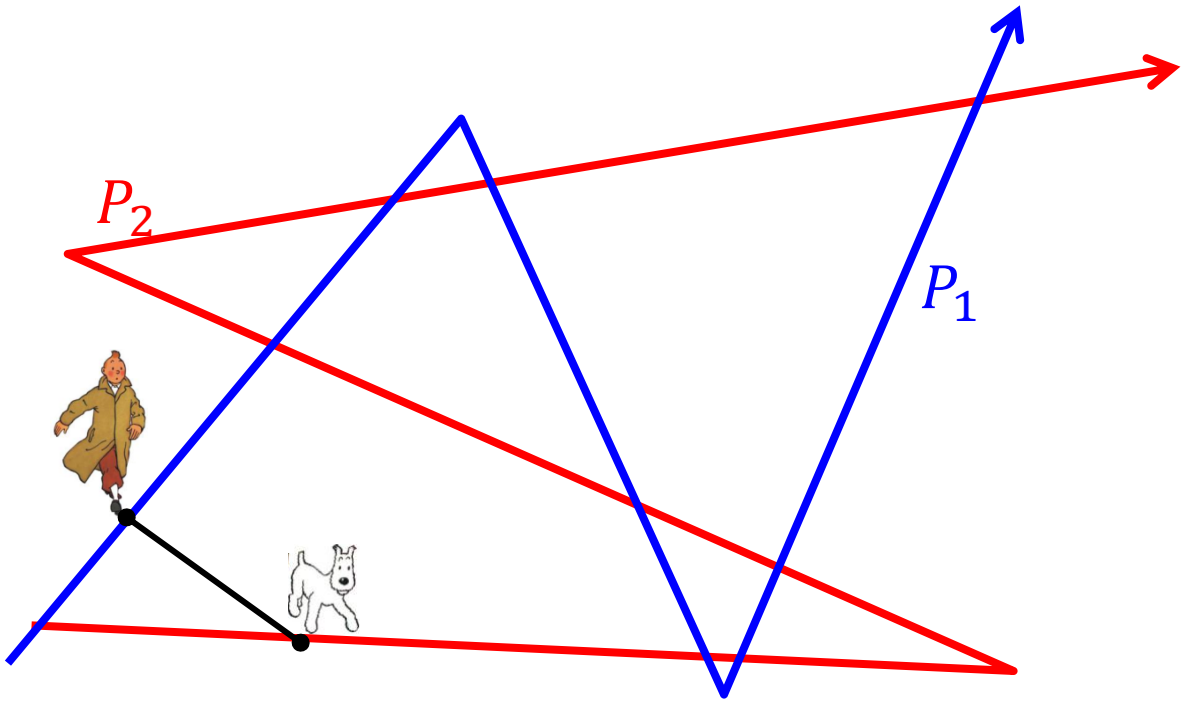
und $d(p_1, P_2) = \min\{\, d(p_1, p_2) \mid p_2 \in P_2 \,\}$



Fréchet-Distanz

Definition: $$d_{\text{fréchet}} = \min_{\substack{\alpha:[0,1]\to[0,m-1] \\ \beta:[0,1]\to[0,n-1]}} \max_{t\in[0,1]} \left\{ d\left(p_1(\alpha(t)), p_2(\beta(t))\right) \right\}$$

wobei α und β **monoton wachsende** und **stetige Funktionen** sind und $\alpha(0) = \beta(0) = 1, \alpha(1) = m - 1, \beta(1) = n - 1.$



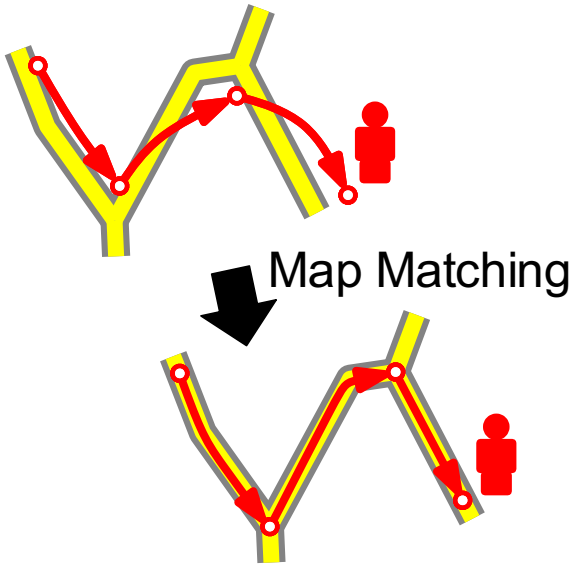
Map Matching

Problemformulierung

Gegeben:

- Das Straßennetz als planar eingebetteter Graph $G = (V, E)$
- Die GPS-Trajektorie als Folge $P = (p_1, p_2, \dots, p_n)$ von Punkten

Gesucht:
Weg in G, der
minimale Distanz
zu P hat.



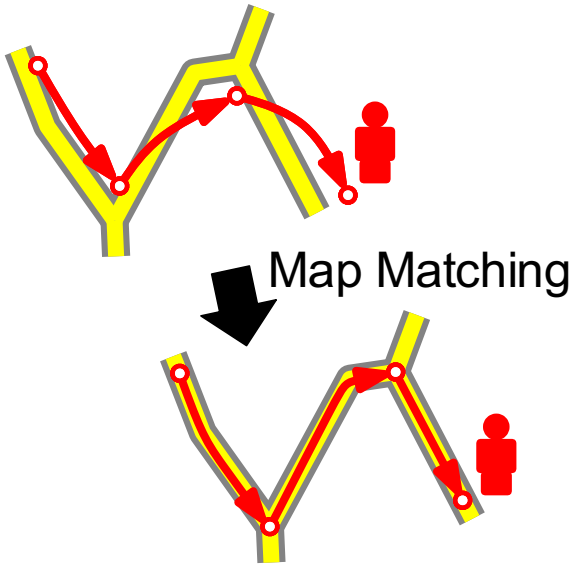
Map Matching

Problemformulierung

Gegeben:

- Das Straßennetz als planar eingebetteter Graph $G = (V, E)$,
- Die GPS-Trajektorie als Folge $P = (p_1, p_2, \dots, p_n)$ von Punkten.

Gesucht:
Weg in G, der
minimale Fréchet-Distanz
zu P hat.



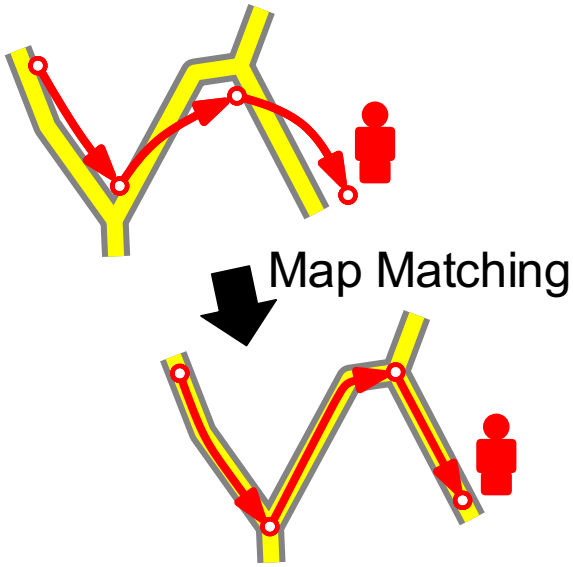
Map Matching

Entscheidungsproblem

Gegeben:

- Das Straßennetz als planar eingebetteter Graph $G = (V, E)$,
- Die GPS-Trajektorie als Folge $P = (p_1, p_2, \dots, p_n)$ von Punkten,
- Zulässige Distanz ϵ.

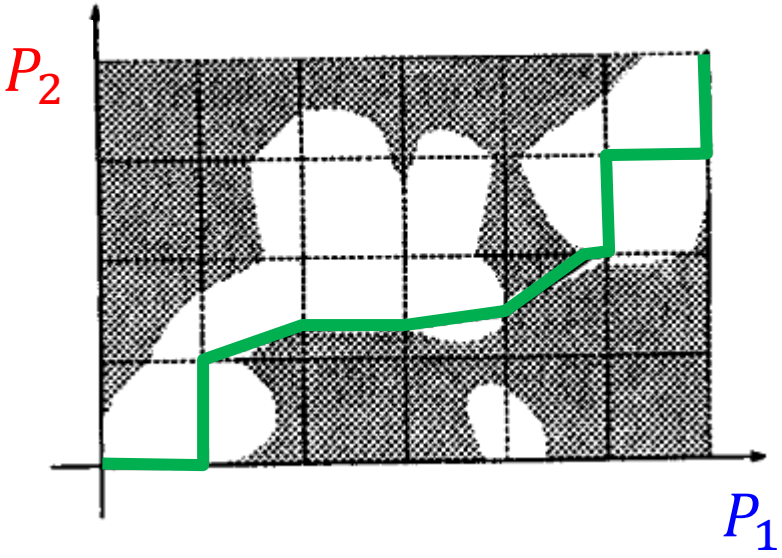
Gibt es einen Weg in G, dessen
Fréchet-Distanz
zu P kleiner ist als ϵ?



Map Matching

Entscheidungsproblem

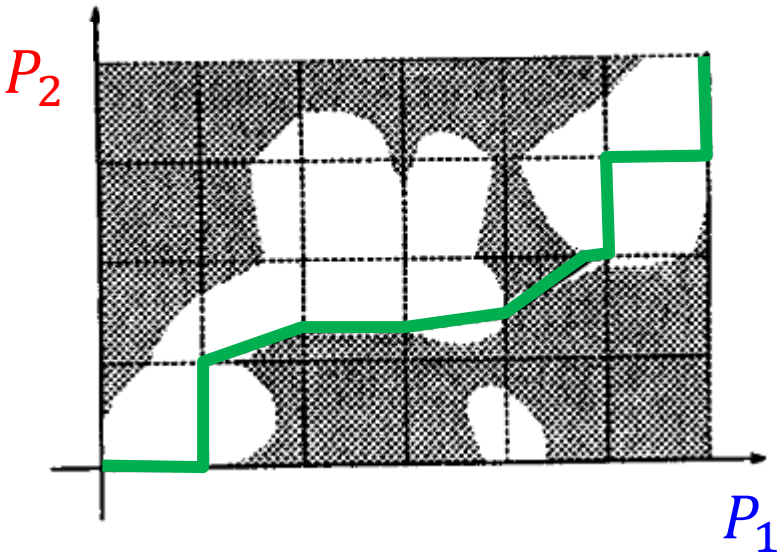
Freiraumdiagramm für zwei Polygonzüge



Map Matching

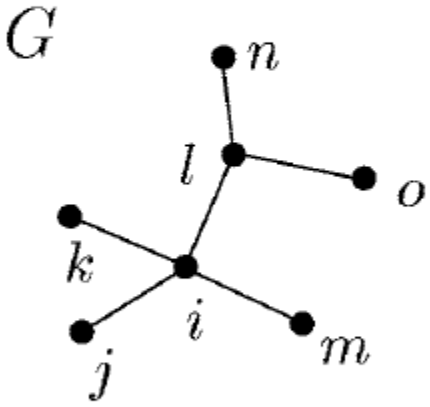
Entscheidungsproblem

Freiraumdiagramm für zwei Polygonzüge



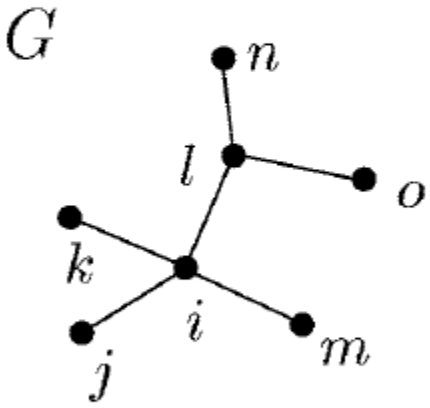
Idee: Freiraumdiagramm für Polygonzug und Graph

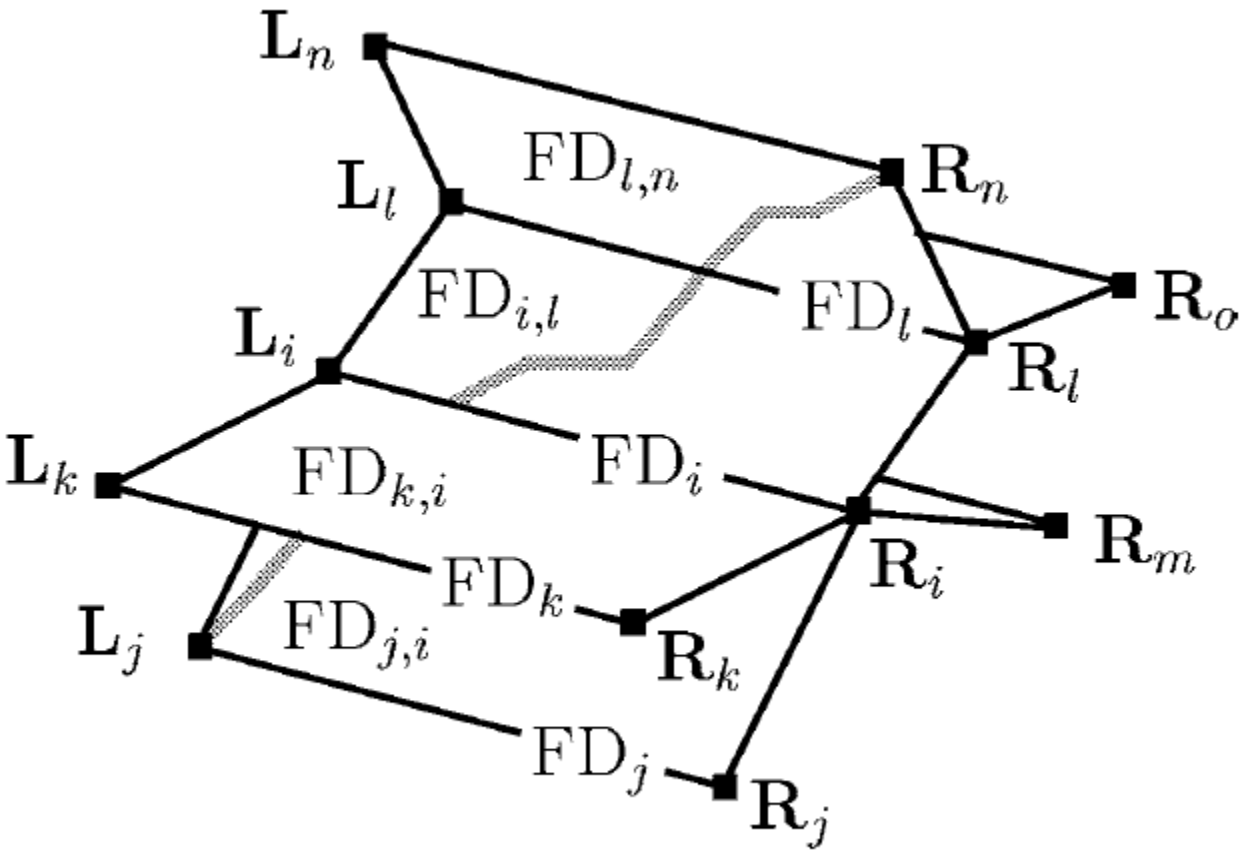
Map Matching



Idee: Freiraumdiagramm für Polygonzug und Graph

Map Matching

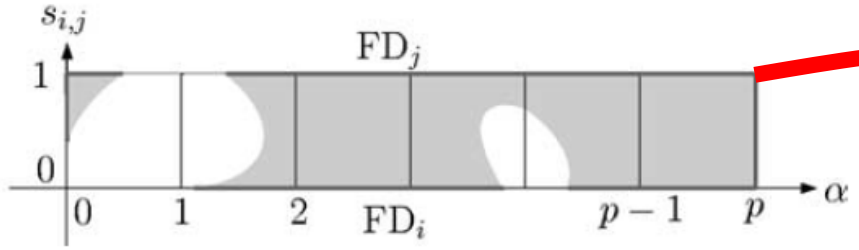


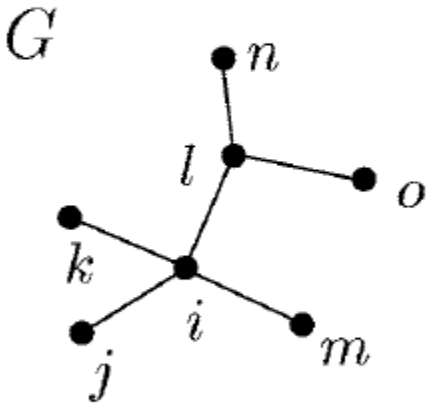


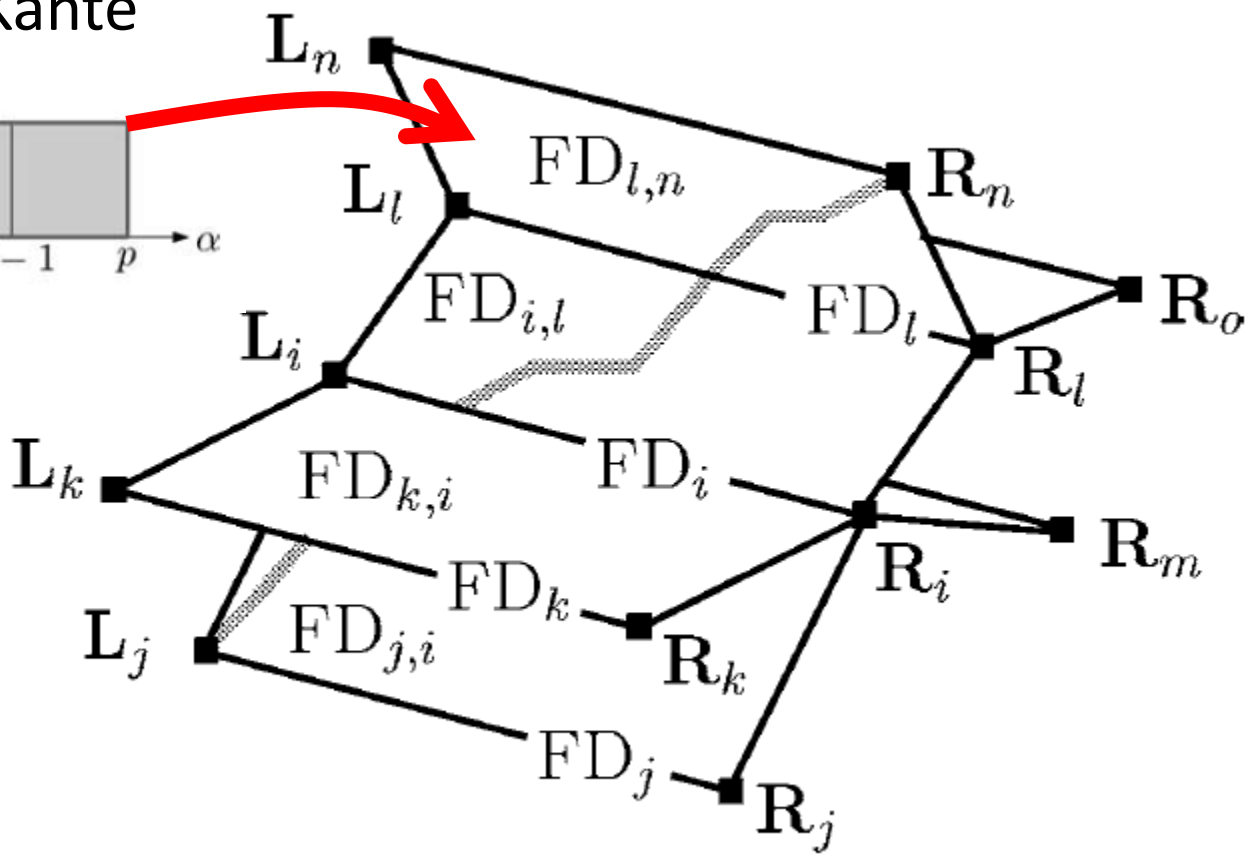
Idee: Freiraumdiagramm für Polygonzug und Graph

Map Matching

Freiraumdiagramm für
Polygonzug und eine Kante



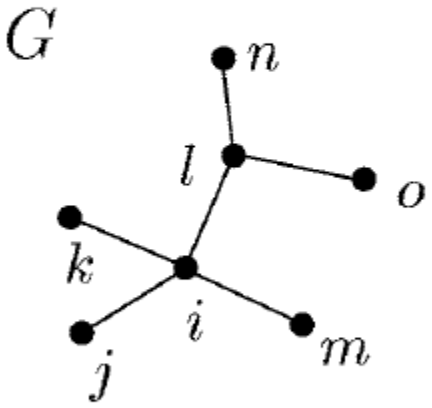


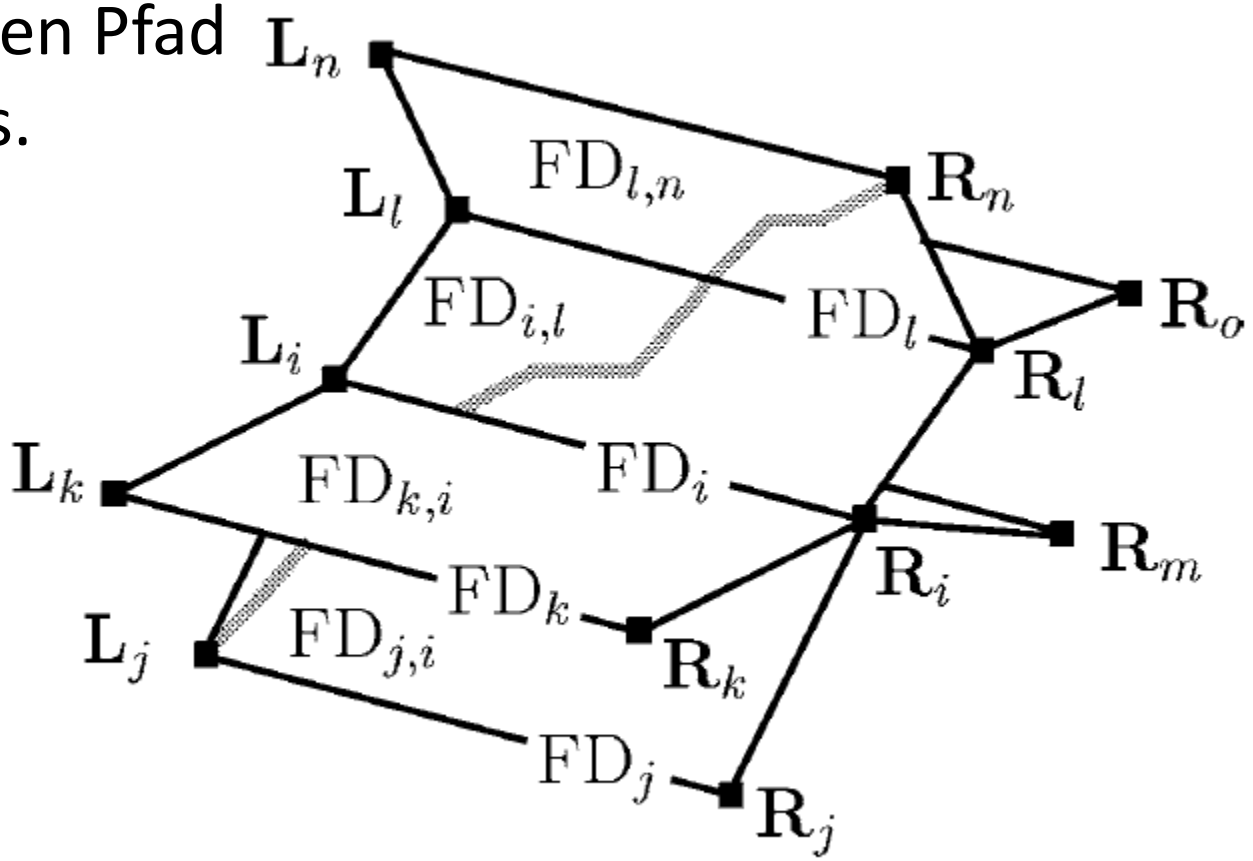


Idee: Freiraumdiagramm für Polygonzug und Graph

Map Matching

Suche einen zulässigen Pfad von links nach rechts.



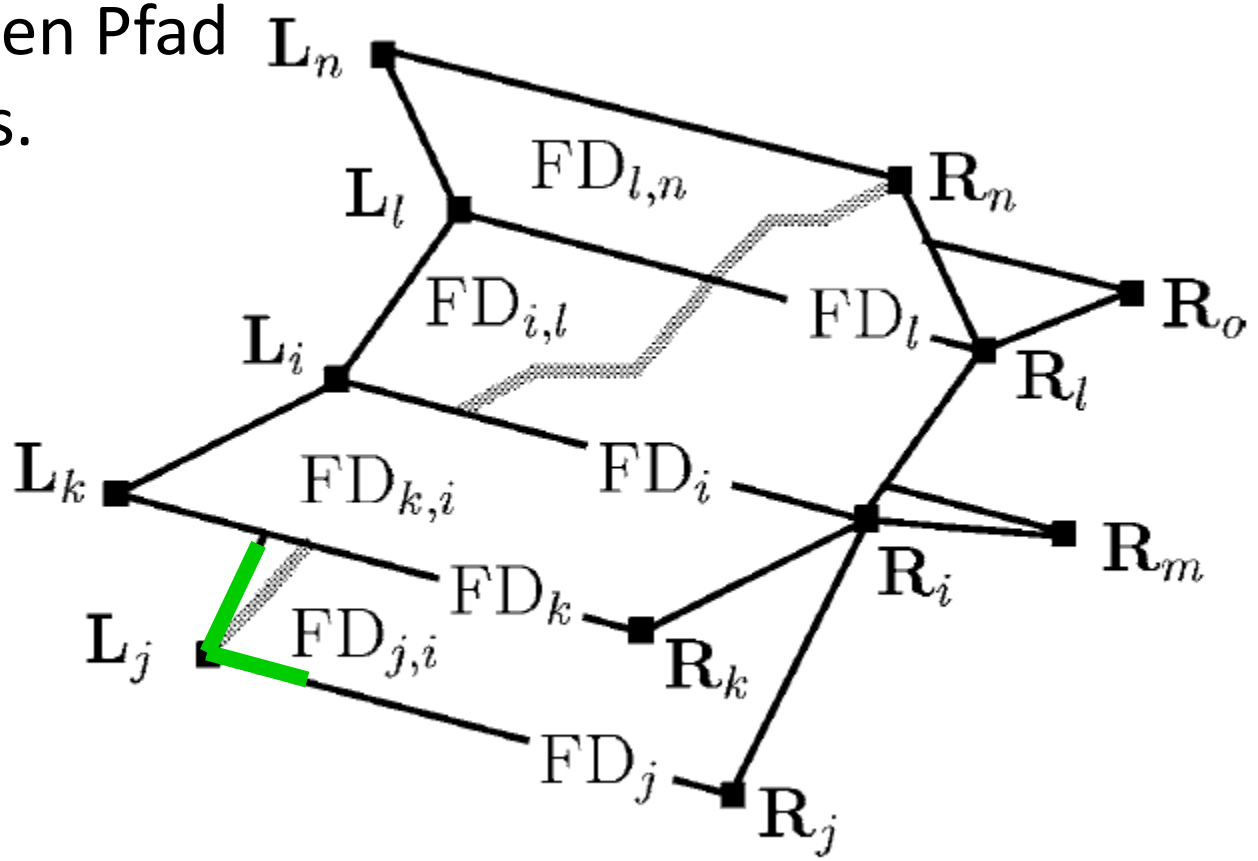


Idee: Freiraumdiagramm für Polygonzug und Graph

Map Matching

Entscheidbar in $O(|E|\,|P|\log|E|)$ Zeit.

Suche einen zulässigen Pfad von links nach rechts.

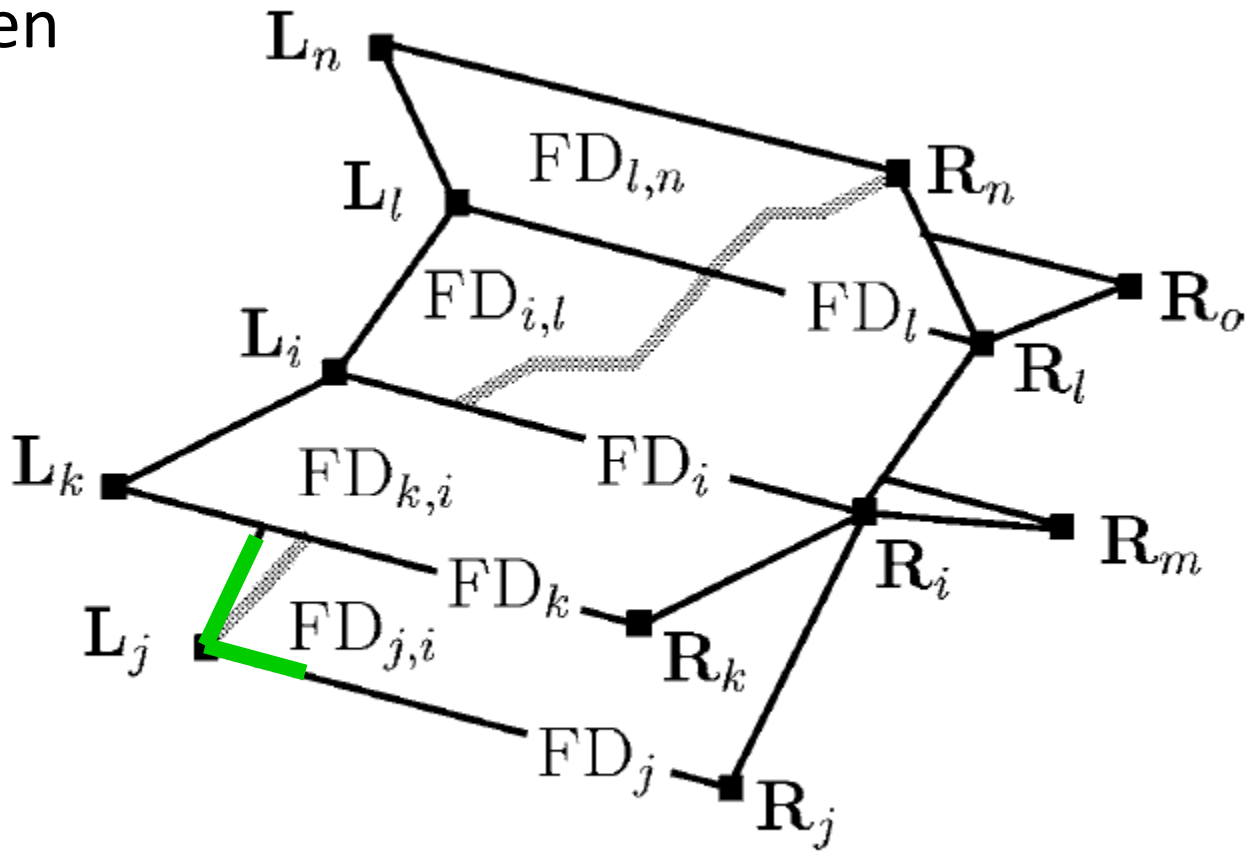


Alt, Efrat, Rote, Wenk:
Matching Planar Maps,
J. Algorithm, 2003.

Map Matching

Lösbar in $O(|E|\,|P|\,\log|E|\,\log|E|\,|P|)$ Zeit.

Finde einen zulässigen Pfad mit **minimaler** Fréchet-Distanz.



Alt, Efrat, Rote, Wenk:
Matching Planar Maps,
J. Algorithm, 2003.

Map Matching

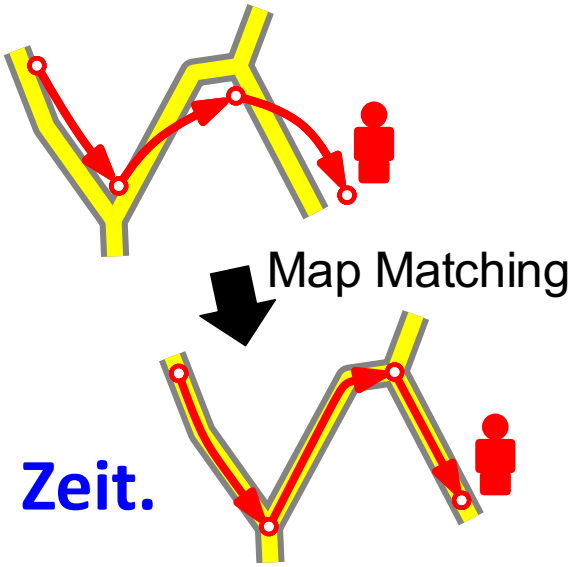
Problemformulierung

Gegeben:

- Das Straßennetz als planar eingebetteter Graph $G = (V, E)$
- Die GPS-Trajektorie als Folge $P = (p_1, p_2, \dots, p_n)$ von Punkten

VIDEO

Gesucht:
Weg in G, der
minimale Fréchet-Distanz
zu P hat.

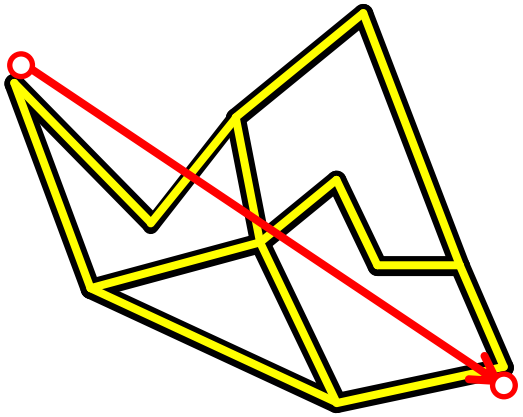


Lösbar in $O(\,|E|\,|P|\log|E|\log|E|\,|P|\,)$ Zeit.

Map Matching

Problem:

GPS-Punkte der Trajektorie weisen einen relativ großen Abstand zueinander auf.

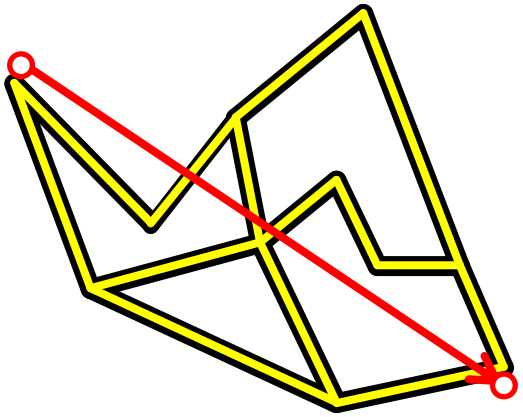


Map Matching

Problem:

GPS-Punkte der Trajektorie weisen einen relativ großen Abstand zueinander auf.

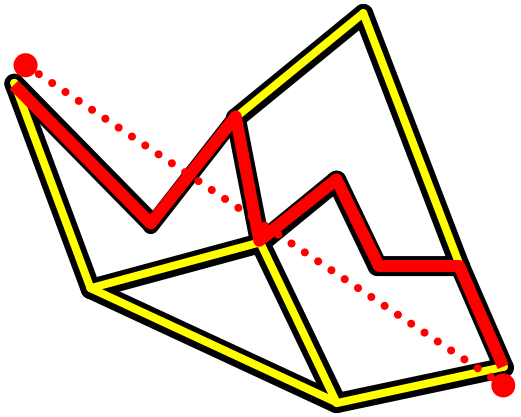
→ Direkte Verbindung zwischen Punkten ist schlechte Näherung



Map Matching

Problem:

GPS-Punkte der Trajektorie weisen einen relativ großen Abstand zueinander auf.



Ergebnis mit minimaler Fréchet-Distanz

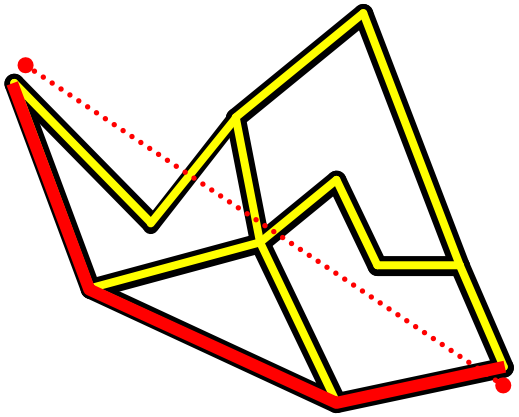
Idee:

Fahrer wählen bevorzugt **kürzeste Wege** im Straßennetz.

Map Matching

Problem:

GPS-Punkte der Trajektorie weisen einen relativ großen Abstand zueinander auf.



kürzeste Route

Idee:

Fahrer wählen bevorzugt **kürzeste Wege** im Straßennetz.

Map Matching

Ansatz über kürzeste Wege

Lou et al.:
Map-Matching for Low-Sampling-Rate GPS Trajectories,
Proc. ACM GIS 2009.

Ein Punkt etwa alle zwei Minuten (z.B. zur Aufzeichnung & Analyse von Taxirouten)

P. Newson und J. Krum:
Hidden Markov Map Matching Through Noise and Sparseness,
Proc. ACM GIS 2009

Eisner et al.:
Algorithms for Matching and Predicting Trajectories,
Proc. ALENEX 2011

Map Matching

Ansatz über kürzeste Wege

Lou et al.:
Map-Matching for Low-Sampling-Rate GPS Trajectories,
Proc. ACM GIS 2009.

P. Newson und J. Krum:
Hidden Markov Map Matching Through Noise and Sparseness,
Proc. ACM GIS 2009

= Maximum likelihood estimation + Markov Chains + Hidden Markov Models

Eisner et al.:
Algorithms for Matching and Predicting Trajectories,
Proc. ALENEX 2011

Maximum-Likelihood Estimation

Probability versus Likelihood

- Flipping a coin:
  - $P[heads] = p$
  - $P[tails] = 1 - p$
- Do three independent flips.
- The coin is fair (**p = ½**).
- What is the **probability** of getting outcome **x=(H,H,T)**?

Model

Parameter value

Outcome?

p **is not a random variable!**

$$P[x = (H,H,T) \,;\, p = {}^1/_2] \quad \cancel{P[x = (H,H,T) \mid p = {}^1/_2]} = {}^1/_2 \cdot {}^1/_2 \cdot (1 - {}^1/_2) = 0.125$$

Maximum-Likelihood Estimation

Probability versus **Likelihood**

- Flipping a coin:
  - $P[heads] = p$
  - $P[tails] = 1 - p$
- Did three independent flips.
- The result was **x=(H,H,T)**.
- What is the **likelihood** that the coin is fair (**p = ½**)?

Model

Parameter value?

Outcome

Maximum-Likelihood Estimation

Likelihood Functions

- Given the outcome x,
  how "likely" is a certain parameter value θ?
- Assuming parameter value θ,
  what is the probability of outcome x?
- Define *likelihood function* $\mathcal{L}(\theta|x) = P[x;\theta]$

Not a probability!

- The result was **(H,H,T)**.
- What is the **likelihood** that the coin is fair (**p = ½**)?
- $\mathcal{L}[\ \boldsymbol{p = {}^1/_2}\ |\ x = (H,H,T)\ ] = {}^1/_2 \cdot {}^1/_2 \cdot {}^1/_2 = 0.125$
- $\mathcal{L}[\ \boldsymbol{p = 0.1}\ |\ x = (H,H,T)\ ] = 0.1 \cdot 0.1 \cdot 0.9 = 0.009$
- $\mathcal{L}[\ \boldsymbol{p = {}^2/_3}\ |\ x = (H,H,T)] = {}^2/_3 \cdot {}^2/_3 \cdot {}^1/_3 = 0.\overline{148}$

Maximum-Likelihood Estimation

Predictions versus Estimates

- *Most probable outcome*
  - Given parameter $p = {}^{2}/_{3}$…
  - … the most probably outcome is (H, H, H).

- *Maximum-Likelihood Estimate* (“MLE“ or “ML method“)
  - Given outcome (T, H, T)…
  - …maximum-likelihood estimate is $p = {}^{1}/_{3}$.
  - This value of p “best explains the observed outcome.“
  - This value of p makes the outcome ”least surprising.”

Maximum-Likelihood Estimation

Maximum-Likelihood Map Matching?

- **Model? Parameter? Outcome?**

- *Most probable outcome*
  - Given a path in a road network …
  - … what is the most probable GPS trajectory to observe? (Noisy.)

- *Maximum-Likelihood Estimate* (MLE)
  - Given the observed GPS trajectory (noisy) …
  - … what is the most likely path through the road network?
  - "Which path through the network best explains the observed GPS trajectory?"

Пафну́тий Чебышёв
1821 - 1894

Ма́рков Chains

Андре́й Ма́рков
1856 - 1922

Георгій Вороний
1868 - 1908

Бори́с Делоне́
1890 - 1980

Wacław Sierpiński
1882 - 1969

Markov Chains

Chebyshev
1821 - 1894

Markov
1856 - 1922

Voronoi
1856 - 1922

Delaunay
1890 - 1980

Sierpinski
1882 - 1969

Markov Chains

- Sequence of random variables $X_1, X_2, X_3, \ldots$
- **Markov Property**:

$$\Pr[\, X_{n+1} \mid X_n = x_n, X_{n-1} = x_{n-1}, \ldots, X_1 = x_1 \,] = \Pr[\, X_{n+1} \mid X_n = x_n \,]$$

- "Given the present,
  the future is independent of the past."
- Represent as $\Pr[X_1]$, $\Pr[X_2|X_1]$, $\Pr[X_3|X_2]$, …

Markov Chains

Example

- Possible states $Z = \{Sonne, Regen\}$

- $\Pr[\, X_1 = Regen \,] = 0.3$
- $\Pr[\, X_1 = Sonne \,] = 0.7$

- $\Pr[\, X_{n+1} = Regen \mid X_n = Regen \,] = 0.7$
- $\Pr[\, X_{n+1} = Sonne \mid X_n = Regen \,] = 0.3$
- $\Pr[\, X_{n+1} = Regen \mid X_n = Sonne \,] = 0.2$
- $\Pr[\, X_{n+1} = Sonne \mid X_n = Sonne \,] = 0.8$

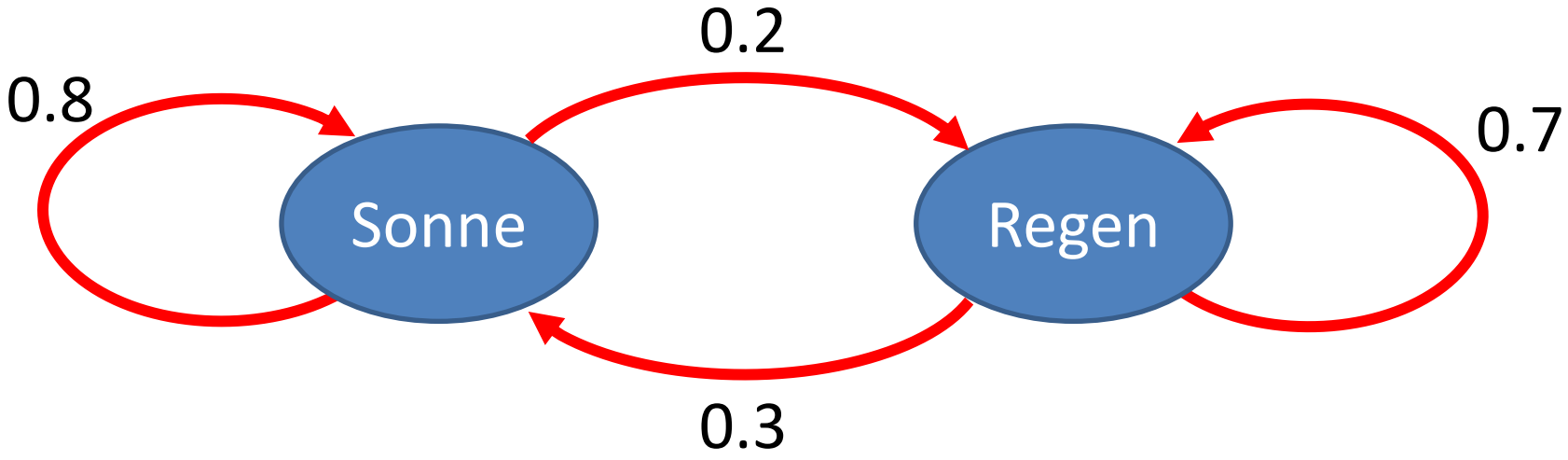
Markov Chains

Example

- Possible states $Z = \{Sonne, Regen\}$

- $\Pr[\, X_1 = Regen \,] = 0.3$
- $\Pr[\, X_1 = Sonne \,] = 0.7$

$\Pr[\, X_2 = Sonne \,]$?



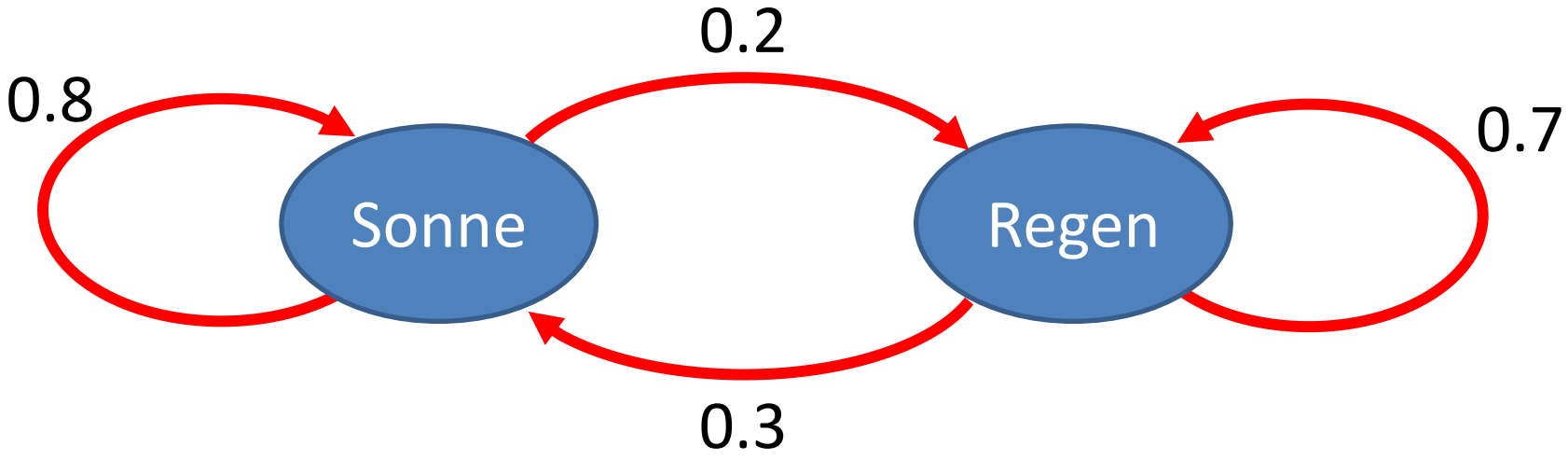
Markov Chains

Example

- Possible states $Z = \{Sonne, Regen\}$

- $\Pr[\, X_2 = Regen \,] = 0.35$
- $\Pr[\, X_2 = Sonne \,] = 0.65$

$\Pr[\, X_3 = Regen \,]?$



Hidden Markov Models

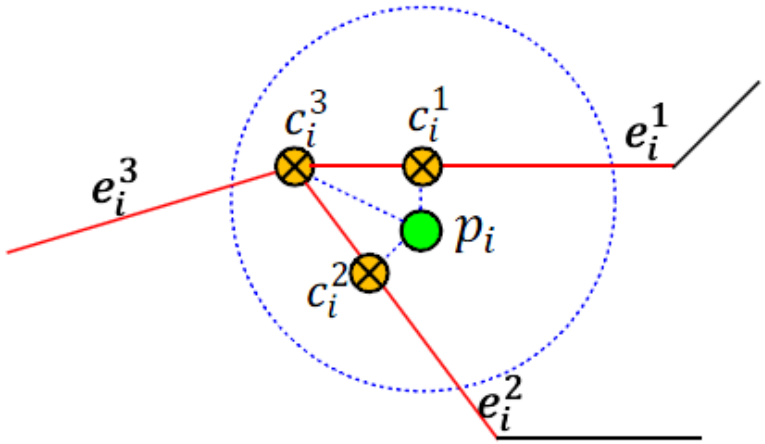
- Markov Chain
  - At every time step, it has a certain ***state***
- But the states are ***hidden***
  - We cannot "see" its state
- Emission ("output")
  - At every time step, get an ***observation***
  - Probability distribution depends on state
  - Emission at each step is independent

Map Matching!

Using maximum likelihood estimation

Map Matching

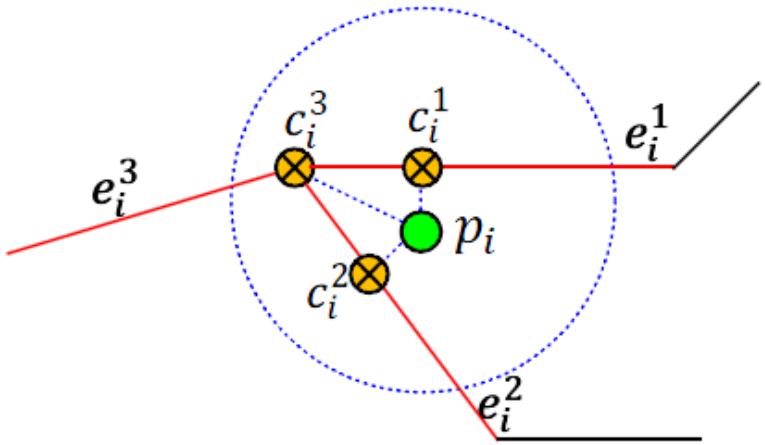
1. Für jeden GPS-Punkt p_i suche Menge von Kandidaten***kanten*** $\{e_i^1, \ldots, e_i^k\}$, die (teilweise) innerhalb eines Kreises mit Radius r um p_i liegen.



Map Matching

1. Für jeden GPS-Punkt p_i suche Menge von Kandidaten***kanten*** $\{e_i^1, \dots, e_i^k\}$, die (teilweise) innerhalb eines Kreises mit Radius r um p_i liegen.

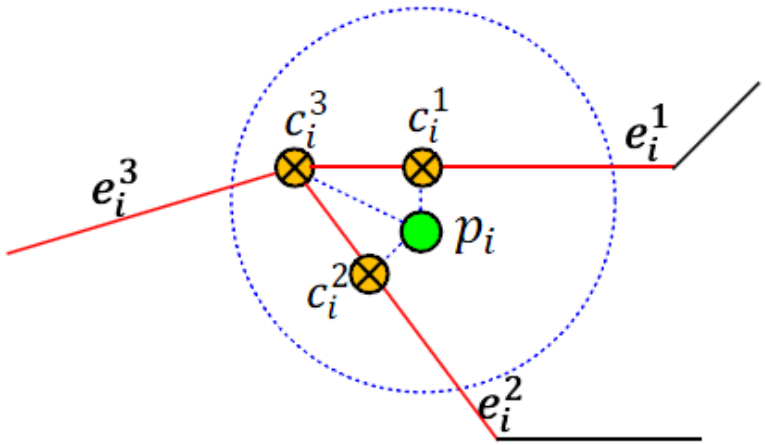
2. Suche Menge von Kandidaten***punkten*** $\{c_i^1, \dots, c_i^k\}$, wobei c_i^j der nächste Punkt auf e_i^j zu p_i ist.



Map Matching

1. Für jeden GPS-Punkt p_i suche Menge von Kandidaten***kanten*** $\{e_i^1, \dots, e_i^k\}$, die (teilweise) innerhalb eines Kreises mit Radius r um p_i liegen.

2. Suche Menge von Kandidaten***punkten*** $\{c_i^1, \dots, c_i^k\}$, wobei c_i^j der nächste Punkt auf e_i^j zu p_i ist.



3. Bewerte jeden Kandidatenpunkt c_i^j mit einer Wahrscheinlichkeit $N(c_i^j)$

$$N(c_i^j) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d(p_i, c_i^j)^2}{2\sigma^2}}$$ **(Normalverteilung)**

$$\sigma = 20 \text{ m}$$

Map Matching

1. Für jeden GPS-Punkt p_i suche Menge von Kandidaten***kanten*** $\{e_i^1, \dots, e_i^k\}$, die (teilweise) innerhalb eines Kreises mit Radius r um p_i liegen.

2. Suche Menge von Kandidaten***punkten*** $\{c_i^1, \dots, c_i^k\}$, wobei c_i^j der nächste Punkt auf e_i^j zu p_i ist.

3. Bewerte jeden Kandidatenpunkt c_i^j mit einer Wahrscheinlichkeit $N(c_i^j)$

4. Bewerte jedes Paar aufeinander folgender Kandidatenpunkte mit einer Übergangswahrscheinlichkeit

$$T(c_{i-1}^r, c_i^s) = \frac{d(p_{i-1}, p_i)}{d_{\text{shortest-path}}(c_{i-1}^r, c_i^s)}$$

Länge des kürzesten Weges von c_{i-1}^r nach c_i^s im Straßennetz

Map Matching

1. Für jeden GPS-Punkt p_i suche Menge von Kandidaten***kanten*** $\{e_i^1, \dots, e_i^k\}$, die (teilweise) innerhalb eines Kreises mit Radius r um p_i liegen.

2. Suche Menge von Kandidaten***punkten*** $\{c_i^1, \dots, c_i^k\}$, wobei c_i^j der nächste Punkt auf e_i^j zu p_i ist.

3. Bewerte jeden Kandidatenpunkt c_i^j mit einer Wahrscheinlichkeit $N(c_i^j)$

4. Bewerte jedes Paar aufeinander folgender Kandidatenpunkte mit einer Übergangswahrscheinlichkeit $T(c_{i-1}^r, c_i^s)$

Map Matching

1. Für jeden GPS-Punkt p_i suche Menge von Kandidaten***kanten*** $\{e_i^1, \dots, e_i^k\}$, die (teilweise) innerhalb eines Kreises mit Radius r um p_i liegen.

2. Suche Menge von Kandidaten***punkten*** $\{c_i^1, \dots, c_i^k\}$, wobei c_i^j der nächste Punkt auf e_i^j zu p_i ist.

3. Bewerte jeden Kandidatenpunkt c_i^j mit einer Wahrscheinlichkeit $N(c_i^j)$

4. Bewerte jedes Paar aufeinander folgender Kandidatenpunkte mit einer Übergangswahrscheinlichkeit $T(c_{i-1}^r, c_i^s)$

Ziel:
Wähle für jeden Punkt p_i einen Kandidatenpunkt c_i aus der Menge $\{c_i^1, \dots, c_i^k\}$, so dass

$$N(c_1) \cdot T(c_1, c_2) \cdot N(c_2) \cdot T(c_2, c_3) \cdot \dots \cdot N(c_n)$$

maximal ist.

Map Matching

Ziel:

Wähle für jeden Punkt p_i einen Kandidaten-punkt c_i aus der Menge $\{c_i^1, \dots, c_i^k\}$, so dass

$$
\begin{aligned}
&\log\big(N(c_1) \cdot T(c_1, c_2)\big) \\
+\ &\log\big(N(c_2) \cdot T(c_2, c_3)\big) \\
&\dots \\
+\ &\log\big(N(c_{n-1}) \cdot T(c_{n-1}, c_n)\big) \\
+\ &\log\big(N(c_n)\big)
\end{aligned}
$$

maximal ist.

Map Matching

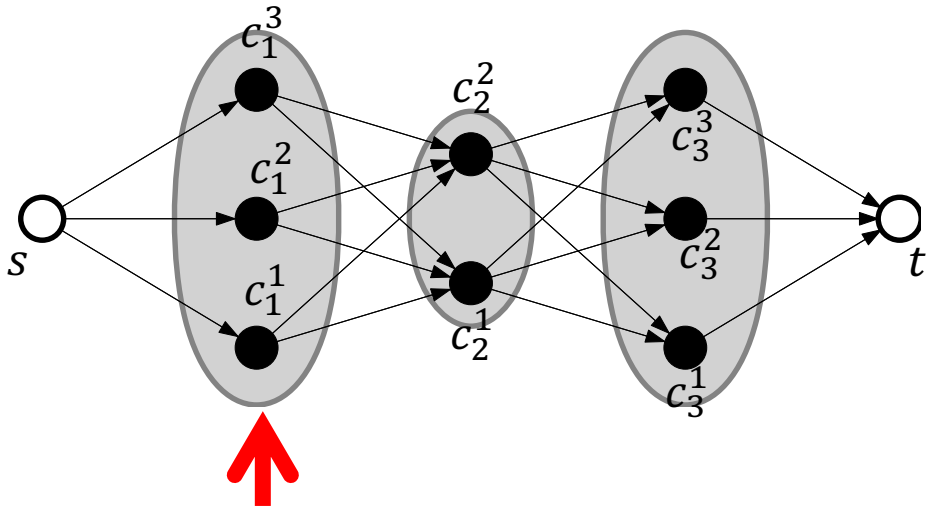
Ziel:

Wähle für jeden Punkt p_i einen Kandidatenpunkt c_i aus der Menge $\{c_i^1, \dots, c_i^k\}$, so dass

$$\begin{aligned} &\log\big(N(c_1)\cdot T(c_1,c_2)\big)\\ &+\log\big(N(c_2)\cdot T(c_2,c_3)\big)\\ &\quad\dots\\ &+\log\big(N(c_{n-1})\cdot T(c_{n-1},c_n)\big)\\ &+\log\big(N(c_n)\big) \end{aligned}$$

maximal ist.

Modellierung als Graphproblem:



Map Matching

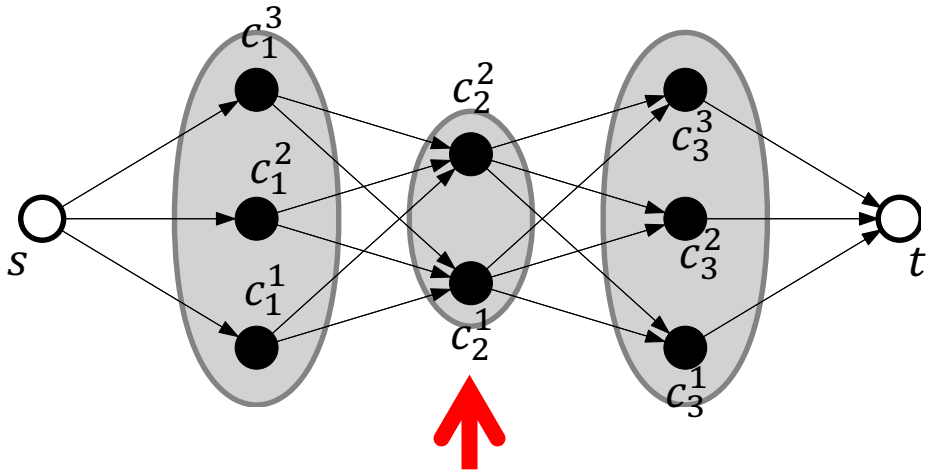
Ziel:

Wähle für jeden Punkt p_i einen Kandidatenpunkt c_i aus der Menge $\{c_i^1, \dots, c_i^k\}$, so dass

$$\begin{aligned} &\log\big(N(c_1)\cdot T(c_1,c_2)\big)\\ &+\log\big(N(c_2)\cdot T(c_2,c_3)\big)\\ &\quad\dots\\ &+\log\big(N(c_{n-1})\cdot T(c_{n-1},c_n)\big)\\ &+\log\big(N(c_n)\big) \end{aligned}$$

maximal ist.

Modellierung als Graphproblem:



Kandidatenmenge für p_2

Map Matching

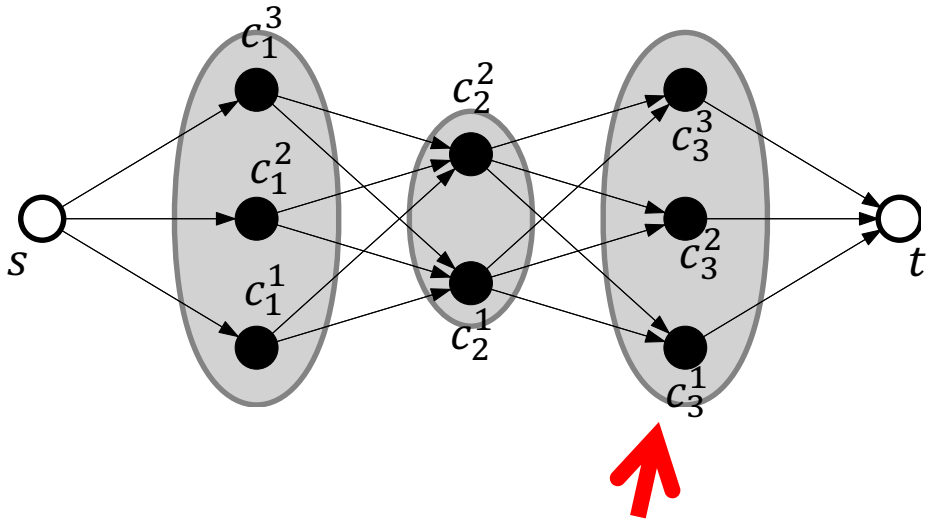
Ziel:

Wähle für jeden Punkt p_i einen Kandidatenpunkt c_i aus der Menge $\{c_i^1, \dots, c_i^k\}$, so dass

$$\begin{aligned} &\log\big(N(c_1)\cdot T(c_1,c_2)\big)\\ &+\log\big(N(c_2)\cdot T(c_2,c_3)\big)\\ &\quad\dots\\ &+\log\big(N(c_{n-1})\cdot T(c_{n-1},c_n)\big)\\ &+\log\big(N(c_n)\big) \end{aligned}$$

maximal ist.

Modellierung als Graphproblem:



Kandidatenmenge für p_3

Map Matching

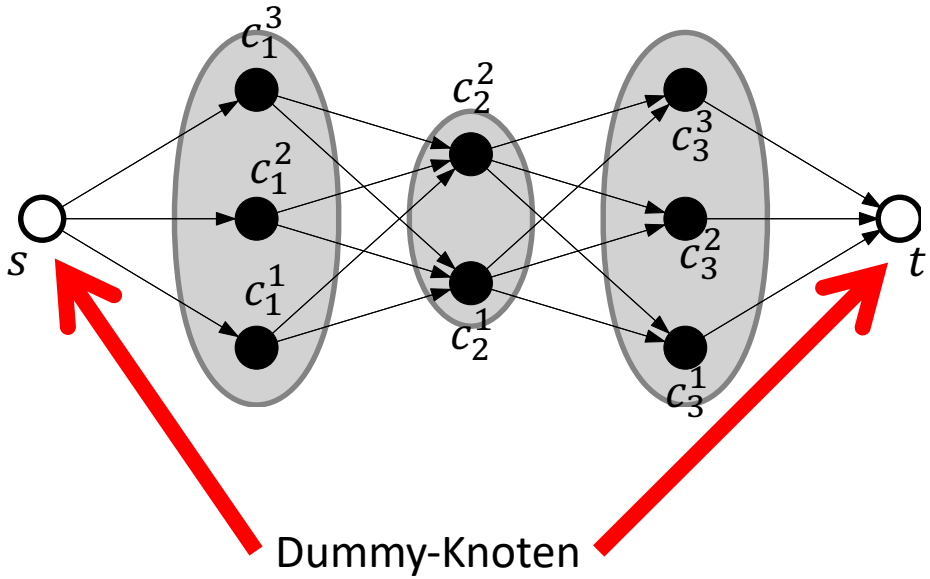
Ziel:

Wähle für jeden Punkt p_i einen Kandidatenpunkt c_i aus der Menge $\{c_i^1, \dots, c_i^k\}$, so dass

$$\begin{aligned} &\log\big(N(c_1)\cdot T(c_1,c_2)\big)\\ +\ &\log\big(N(c_2)\cdot T(c_2,c_3)\big)\\ &\dots\\ +\ &\log\big(N(c_{n-1})\cdot T(c_{n-1},c_n)\big)\\ +\ &\log\big(N(c_n)\big) \end{aligned}$$

maximal ist.

Modellierung als Graphproblem:



Map Matching

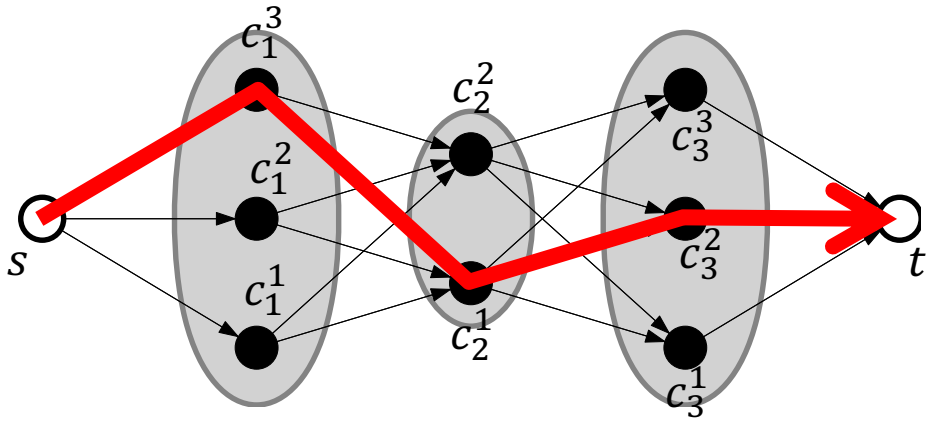
Ziel:

Wähle für jeden Punkt p_i einen Kandidatenpunkt c_i aus der Menge $\{c_i^1, \dots, c_i^k\}$, so dass

$$\begin{aligned} &\log\big(N(c_1)\cdot T(c_1,c_2)\big) \\ &+\log\big(N(c_2)\cdot T(c_2,c_3)\big) \\ &\quad \dots \\ &+\log\big(N(c_{n-1})\cdot T(c_{n-1},c_n)\big) \\ &+\log\big(N(c_n)\big) \end{aligned}$$

maximal ist.

Modellierung als Graphproblem:



Jeder s-t-Pfad steht für eine Lösung des Map-Matching-Problems

Map Matching

Ziel:

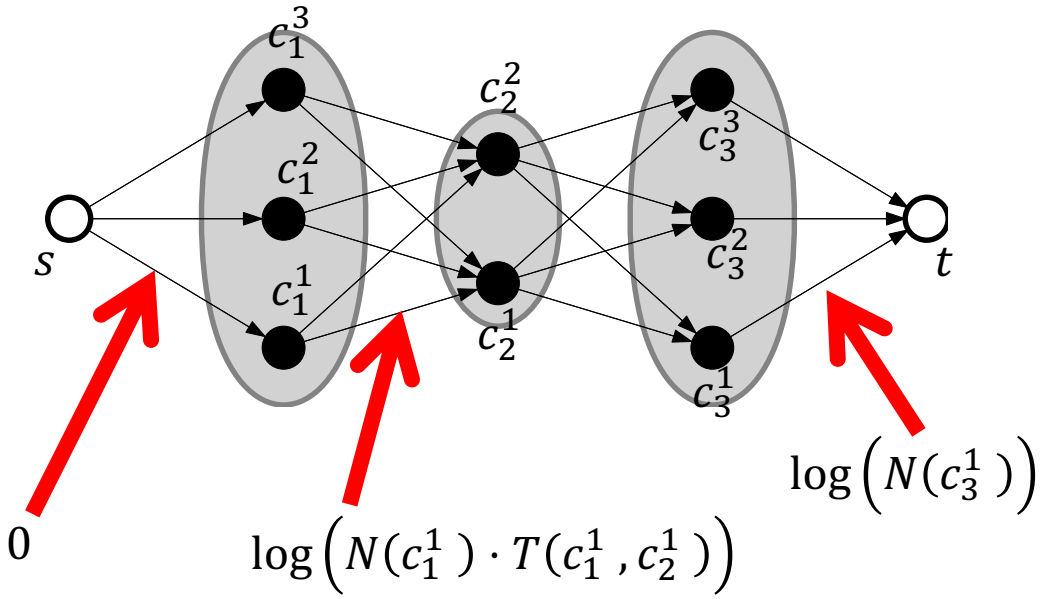
Wähle für jeden Punkt p_i einen Kandidatenpunkt c_i aus der Menge $\{c_i^1, \dots, c_i^k\}$, so dass

$$\begin{aligned} &\log\big(N(c_1)\cdot T(c_1,c_2)\big) \\ &+\log\big(N(c_2)\cdot T(c_2,c_3)\big) \\ &\quad \dots \\ &+\log\big(N(c_{n-1})\cdot T(c_{n-1},c_n)\big) \\ &+\log\big(N(c_n)\big) \end{aligned}$$

maximal ist.

Modellierung als Graphproblem

Definiere Kantenlängen:



Map Matching

Ziel:

Wähle für jeden Punkt p_i einen Kandidatenpunkt c_i aus der Menge $\{c_i^1, \dots, c_i^k\}$, so dass

$$\begin{aligned} &\log\big(N(c_1)\cdot T(c_1,c_2)\big) \\ &+\log\big(N(c_2)\cdot T(c_2,c_3)\big) \\ &\quad\dots \\ &+\log\big(N(c_{n-1})\cdot T(c_{n-1},c_n)\big) \\ &+\log\big(N(c_n)\big) \end{aligned}$$

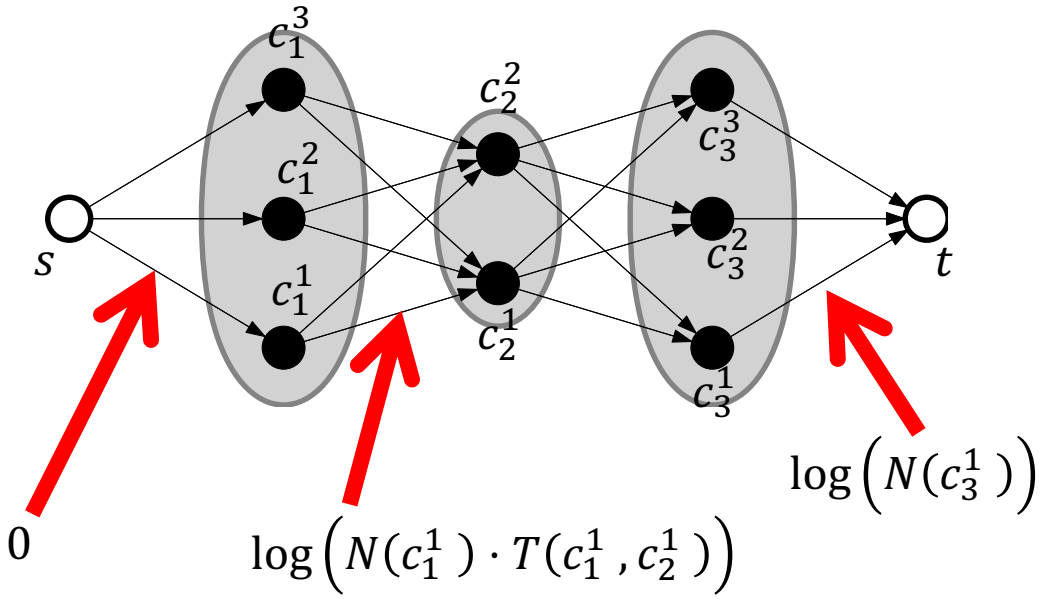
maximal ist.

Lösung:

Längster s-t-Pfad

Modellierung als Graphproblem

Definiere Kantenlängen:



Map Matching

Ziel:

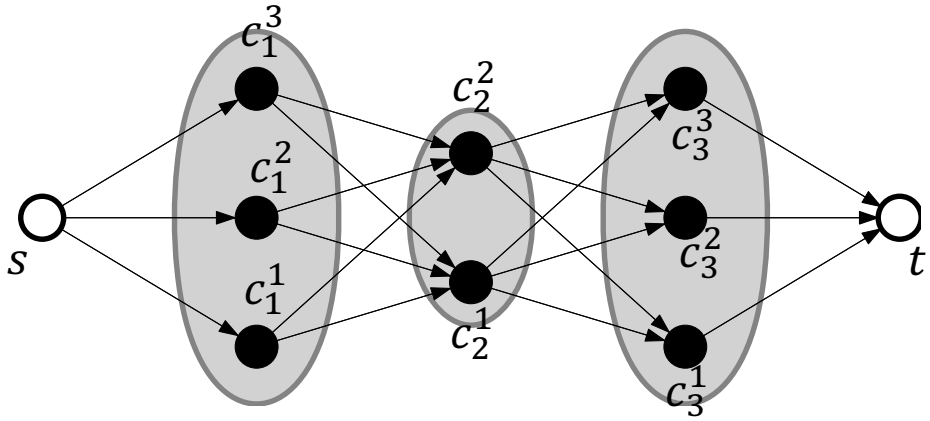
Wähle für jeden Punkt p_i einen Kandidatenpunkt c_i aus der Menge $\{c_i^1, \dots, c_i^k\}$, so dass

$$\begin{aligned} &\log\big(N(c_1)\cdot T(c_1,c_2)\big) \\ +\ &\log\big(N(c_2)\cdot T(c_2,c_3)\big) \\ &\dots \\ +\ &\log\big(N(c_{n-1})\cdot T(c_{n-1},c_n)\big) \\ +\ &\log\big(N(c_n)\big) \end{aligned}$$

maximal ist.

Modellierung als Graphproblem

Definiere Kantenlängen:



Lösung:

Längster s-t-Pfad

Im Allgemeinen ist das Längste-Pfad-Problem NP-schwer!

Map Matching

Ziel:

Wähle für jeden Punkt p_i einen Kandidatenpunkt c_i aus der Menge $\{c_i^1, \dots, c_i^k\}$, so dass

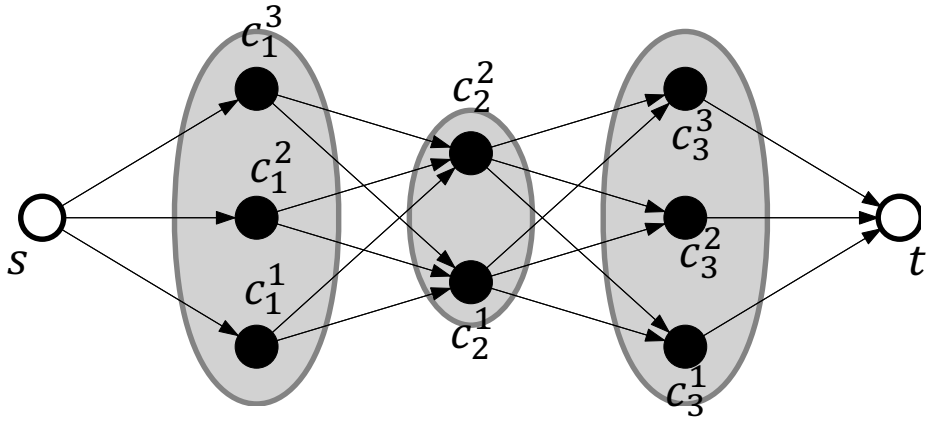
$$\begin{aligned} &\log\big(N(c_1)\cdot T(c_1,c_2)\big)\\ &+\log\big(N(c_2)\cdot T(c_2,c_3)\big)\\ &\quad\dots\\ &+\log\big(N(c_{n-1})\cdot T(c_{n-1},c_n)\big)\\ &+\log\big(N(c_n)\big) \end{aligned}$$

maximal ist.

Lösung:

Modellierung als Graphproblem

Definiere Kantenlängen:



Längster s-t-Pfad

Für gerichtete azyklische Graphen (DAGs) mit m Kanten und n Knoten ist das Längste-Pfad-Problem in $O(m+n)$ Zeit lösbar!

Map Matching

Ziel:

Wähle für jeden Punkt p_i einen Kandidatenpunkt c_i aus der Menge $\{c_i^1, \dots, c_i^k\}$, so dass

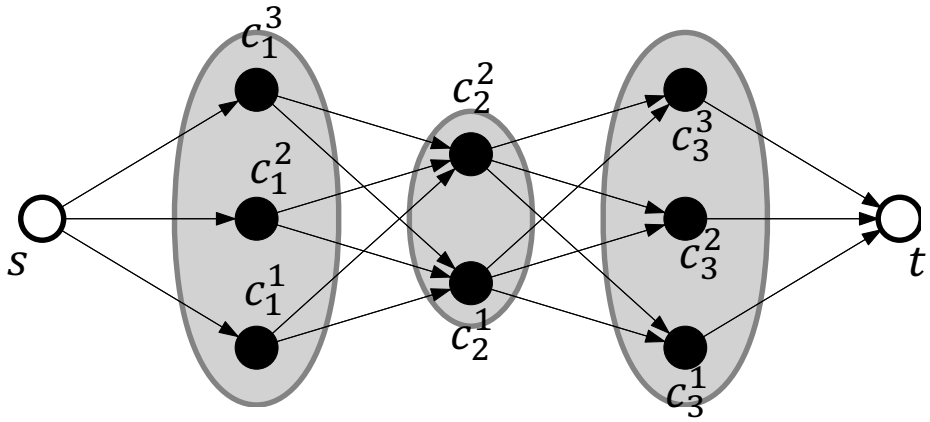
$$\begin{aligned} &\log\big(N(c_1)\cdot T(c_1,c_2)\big) \\ +\ &\log\big(N(c_2)\cdot T(c_2,c_3)\big) \\ &\dots \\ +\ &\log\big(N(c_{n-1})\cdot T(c_{n-1},c_n)\big) \\ +\ &\log\big(N(c_n)\big) \end{aligned}$$

maximal ist.

Lösung:

Modellierung als Graphproblem

Definiere Kantenlängen:



Längster s-t-Pfad

Für gerichtete azyklische Graphen (DAGs) mit m Kanten und n Knoten ist das Längste-Pfad-Problem in $O(m+n)$ Zeit lösbar!

Wie?

Map Matching

Aufgabe:
Finde längsten Pfad in gerichtetem azyklischen Graphen $G = (V, A; \ell)$.

Lösung:
1. Sortiere G topologisch → Reihenfolge $v_1, \ldots, v_n$, so dass für jede Kante (v_i, v_j) gilt $i < j$.
2. Löse das Problem durch dynamische Programmierung:

$d(v_1) = 0$
for $j = 2$ **to** n // Berechne $d(v_j)$ = Länge eines längsten v_1-v_j-Pfades
  $d(v_j) = -\infty$
  foreach v_i: $(v_i, v_j) \in A$
    if $d(v_j) < d(v_i) + \ell(v_i, v_j)$ **then** // $\ell(v_i, v_j)$ = Länge der Kante (v_i, v_j)
      $d(v_j) = d(v_i) + \ell(v_i, v_j)$
      $predecessor(j) = i$