Maximum-likelihood estimation

Maximum-Likelihood Map Matching? ... Model? Parameter? Outcome?

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

- Maximum-Likelihood Estimate (”MLE”)
  - Given an 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?”

Hidden Markov Model

Gegeben:

- Beobachtungen $O = \langle o_1, \ldots, o_n \rangle$, je eine pro Zeitpunkt
- mögliche Systemzustände Z
- für jeden Systemzustand $z \in Z$ und jede Beobachtung o
  - die Wahrscheinlichkeit $\Pr(o \mid z)$ oder
  - die Wahrscheinlichkeits*dichte* $f(o \mid z)$,

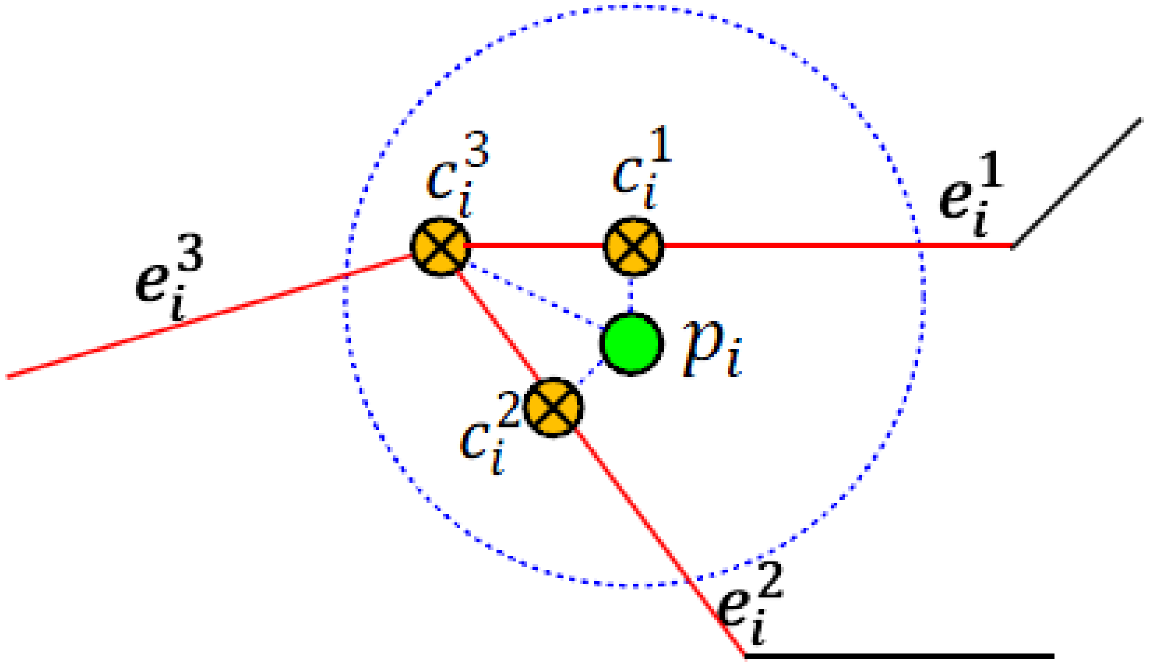
  dass o bei Zustand z beobachtet wird
- für jedes Paar von Zuständen $z_1, z_2 \in Z$ die Wahrscheinlichkeit

  $\Pr(z_2 \mid \text{was } z_1 \text{ before})$,

  also die Wahrscheinlichkeit für z_2 bei Kenntnis, dass zuvor z_1 vorherrschte (*Übergangswahrscheinlichkeit*)
- für jeden Zustand $z \in Z$ die (a-priori-)Wahrscheinlichkeit $\Pr(z)$

Map Matching

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

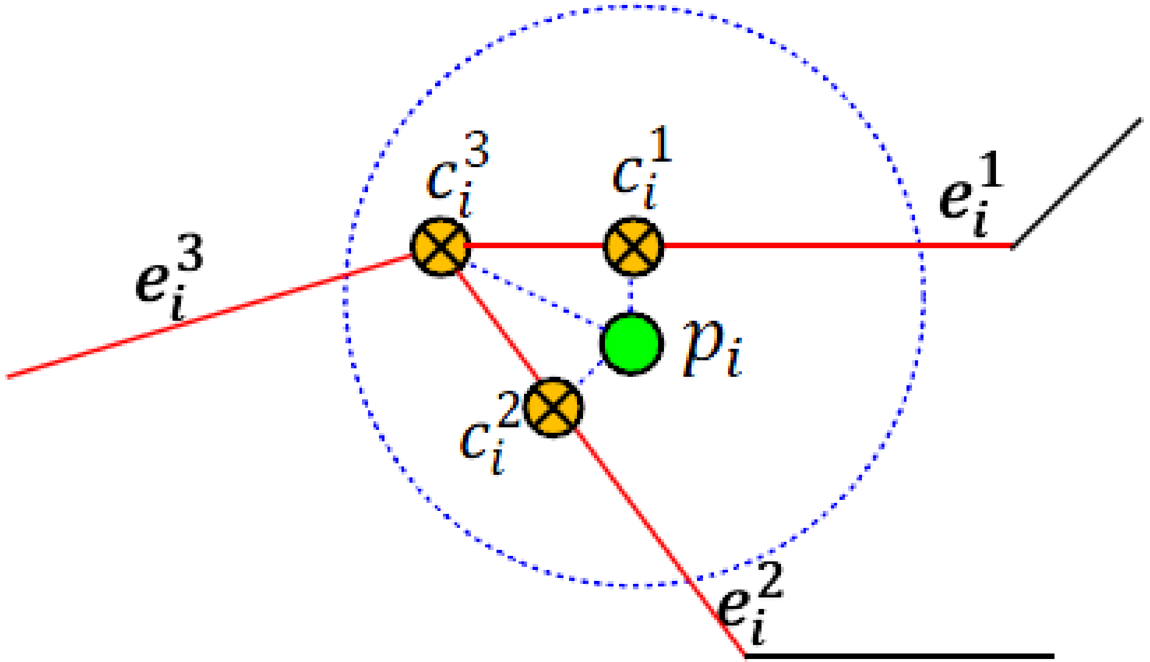
2. Suche Menge von Kandidaten*punkten* $\{c_i^1, \dots, c_i^k\}$, wobei c_i^j der nächste Punkt auf e_i^j zu $\boldsymbol{p_i}$ ist.

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

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

3. Bewerte jeden Kandidatenpunkt $\boldsymbol{c_i^j}$ mit einer Wahrscheinlichkeit $N\left(c_i^j\right)$

4. Bewerte jedes Paar aufeinander folgender Kandidatenpunkte mit einer Übergangswahrscheinlichkeit

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

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

Map Matching

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

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

3. Bewerte jeden Kandidatenpunkt $\boldsymbol{c_i^j}$ mit einer Wahrscheinlichkeit $\boldsymbol{N\left(c_i^j\right)}$

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

Map Matching

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

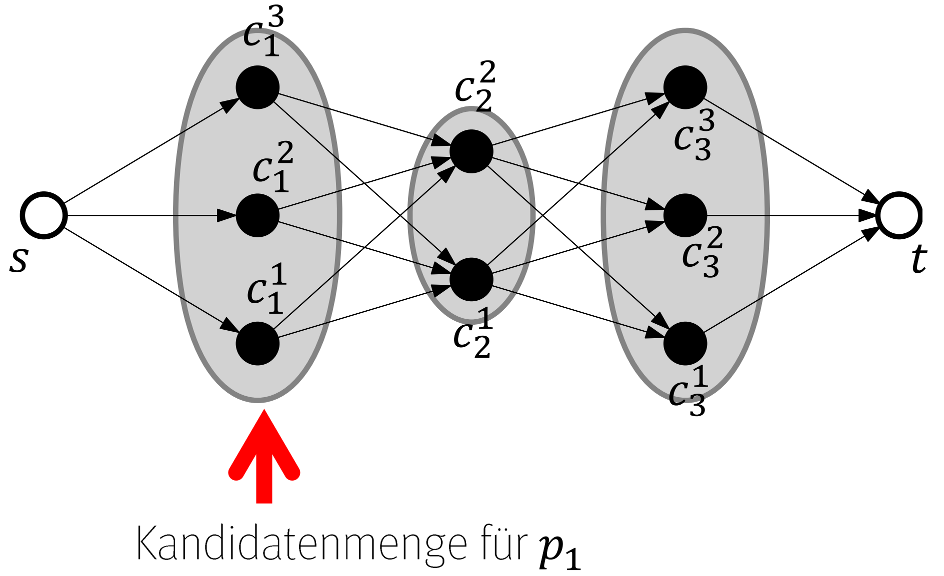
Ziel:

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

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

maximal ist.

Modellierung als Graph-Problem:



Map Matching

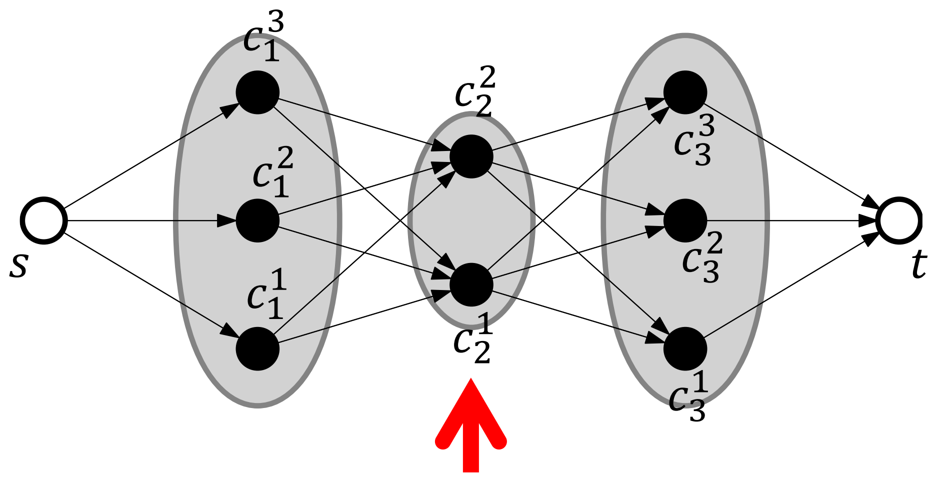
Ziel:

Wähle für jeden Punkt $\boldsymbol{p_i}$ einen Kandidaten-punkt $\boldsymbol{c_i}$ 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 Graph-Problem:



Kandidatenmenge für $\boldsymbol{p_2}$

Map Matching

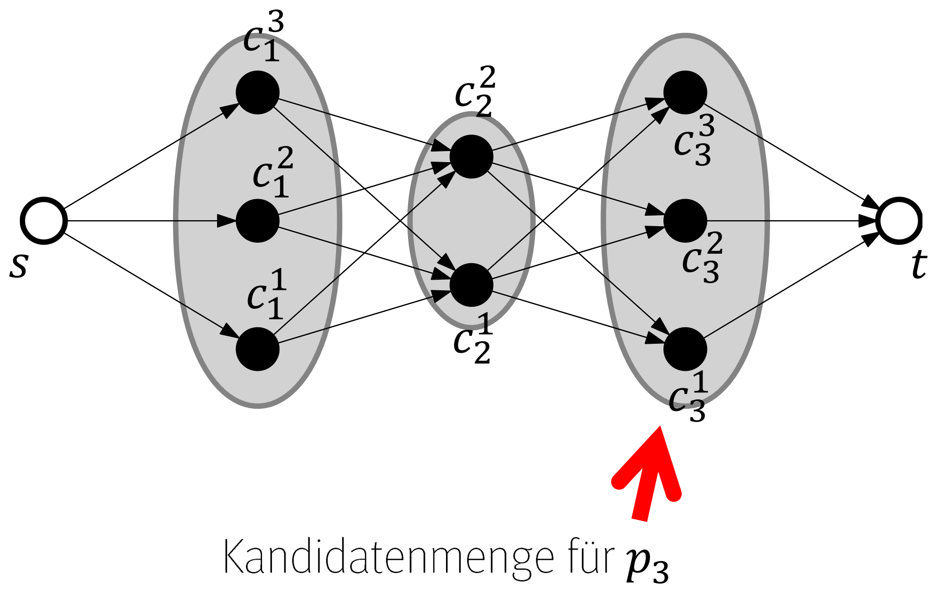
Ziel:

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.

Modellierung als Graph-Problem:



Map Matching

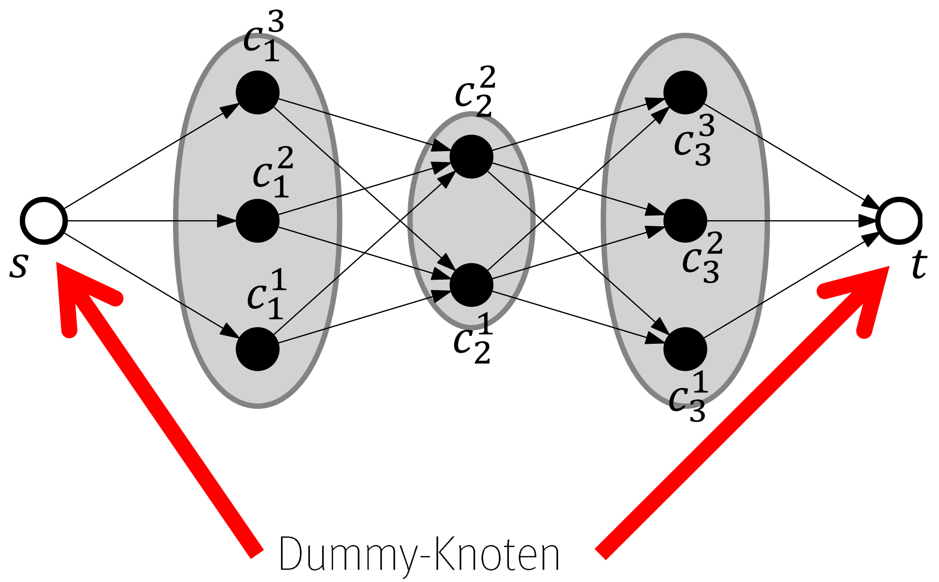
Ziel:

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

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

maximal ist.

Modellierung als Graph-Problem:



Map Matching

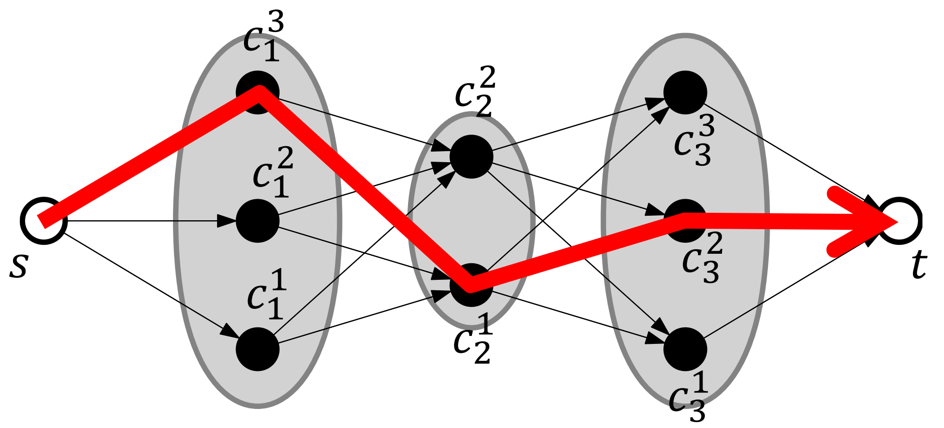
Ziel:

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.

Modellierung als Graph-Problem:



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

Map Matching

Ziel:

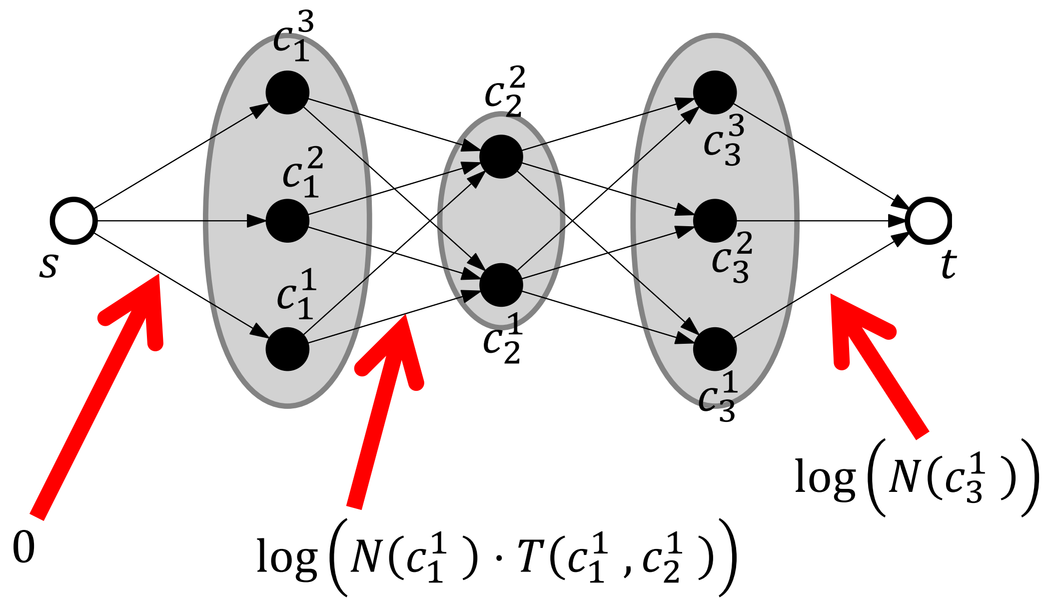
Wähle für jeden Punkt p_i einen Kandidaten-punkt c_i aus der Menge $\{c_i^1, \ldots, c_i^k\}$, so dass

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

maximal ist.

Modellierung als Graph-Problem

Definiere Kantenlängen:



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

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

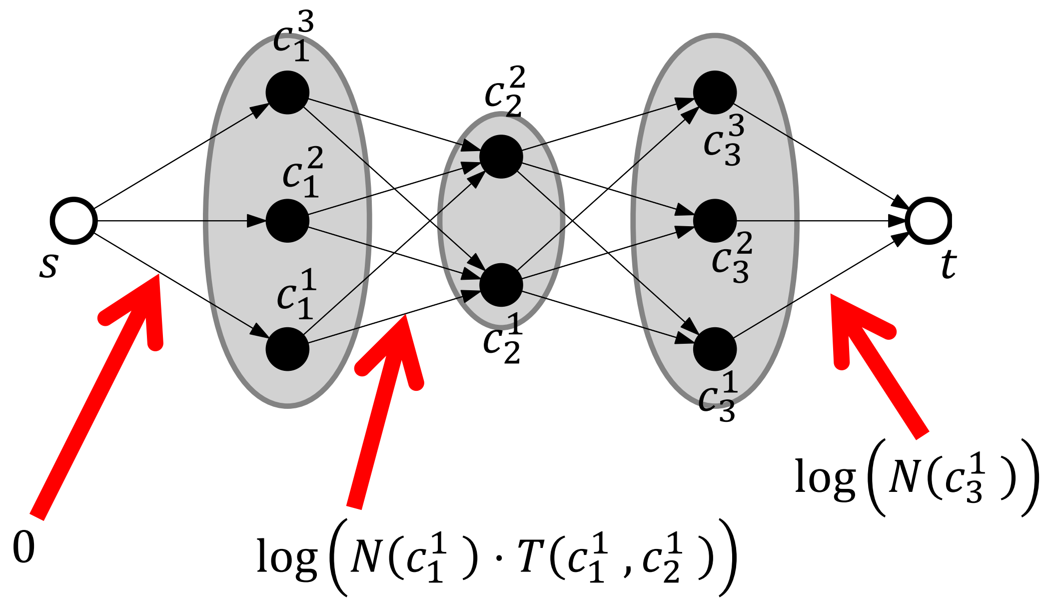
maximal ist.

Lösung:

Längster s-t-Pfad

Modellierung als Graph-Problem

Definiere Kantenlängen:



Map Matching

Ziel:

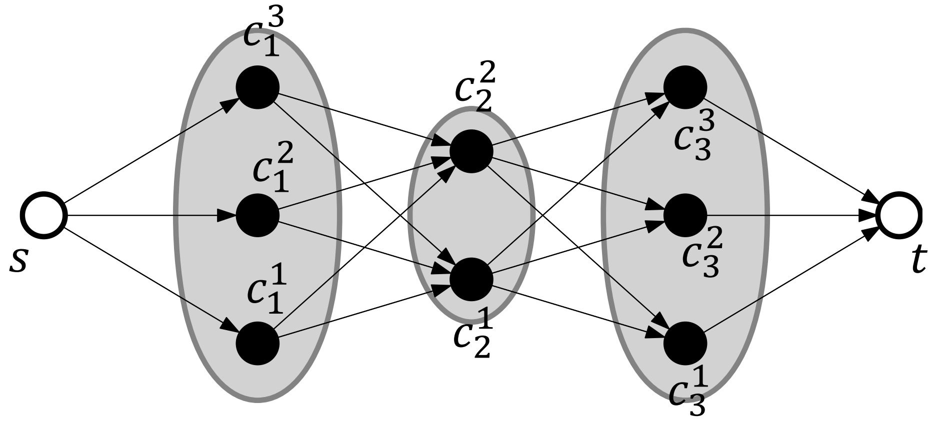
Wähle für jeden Punkt p_i einen Kandidaten-punkt c_i aus der Menge $\{c_i^1, \dots, c_i^k\}$, so dass

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

maximal ist.

Modellierung als Graph-Problem

Definiere Kantenlängen:



Lösung:

Längster s-t-Pfad

Für gerichtete azyklische Graphen mit m Kanten und n Ecken ist das Longest-Path-Problem in $O(mn)$ Zeit lösbar!