Speed-Up Heuristics for the Traffic Flow Estimation with Gaussian Process Regression

Thomas Liebig

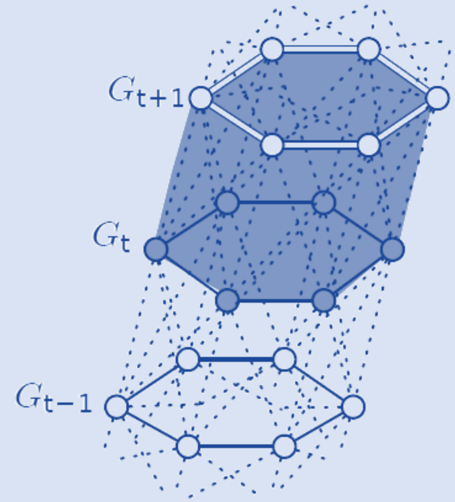
TU Dortmund

technische universität dortmund
Fakultät Informatik
Lehrstuhl für Künstliche Intelligenz

Predicttive Segment Cost Estimation

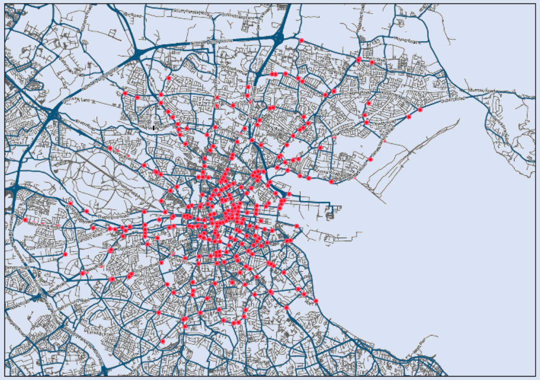
- ❑ Predict future sensor readings with Spatio-Temporal Random Fields

  [Piatkowski et al 13]



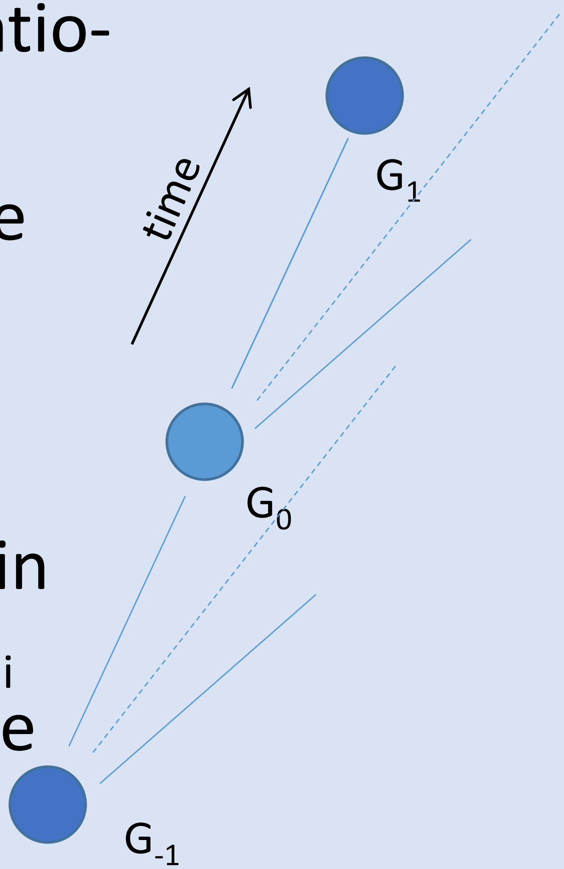
- ❑ Impute values for unobserved locations using Gaussian Process Regression

  [Liebig et al 12]



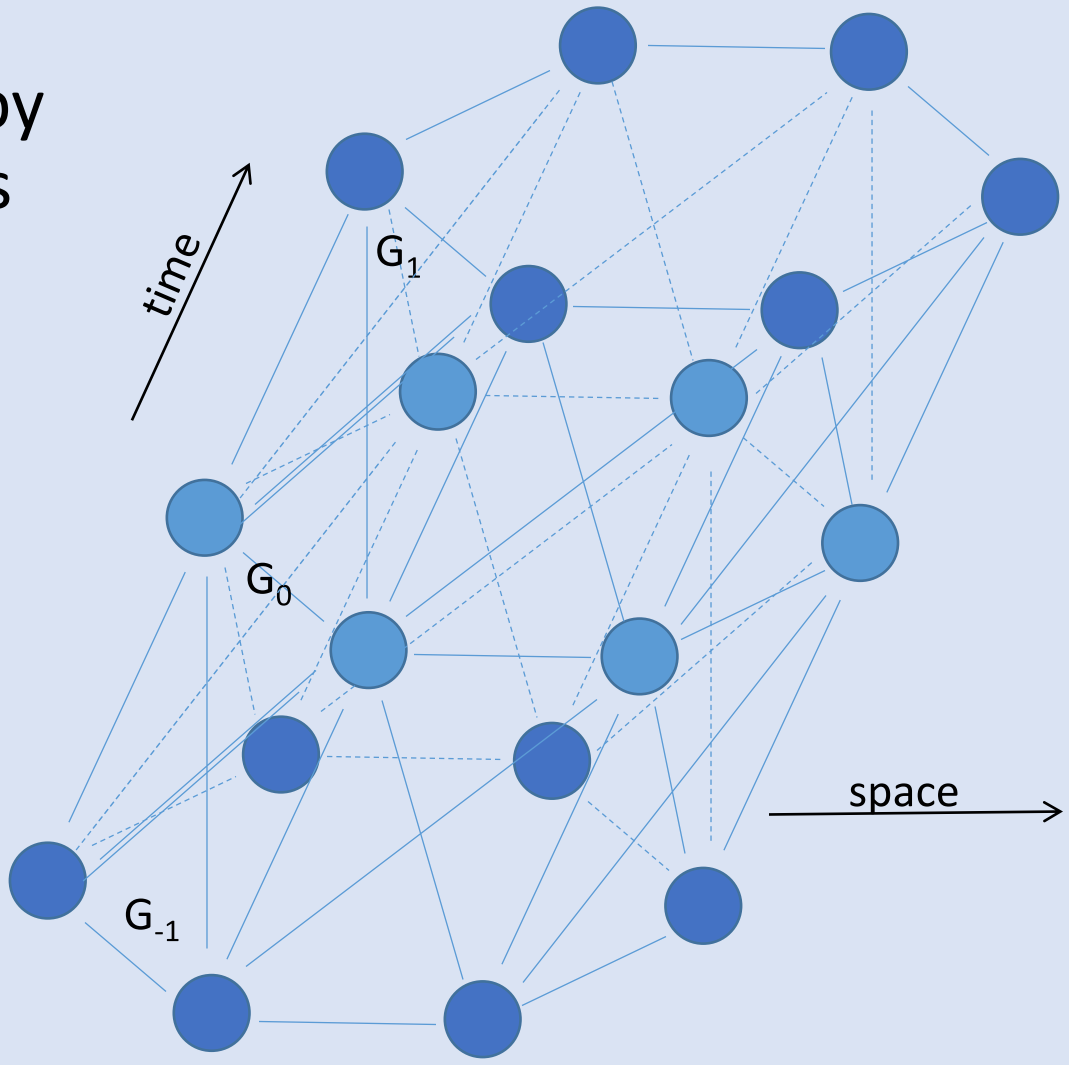
Spatio-Temporal Random Field (STRF)

- Discrete model of spatio-temporal values
- Spatial graph G_0 of the sensors generates measurements
- Joint measurements create a temporal chain G_1—G_2—G_3 … every G_i replicates the structure of G_0



Spatio-Temporal Random Field (STRF)

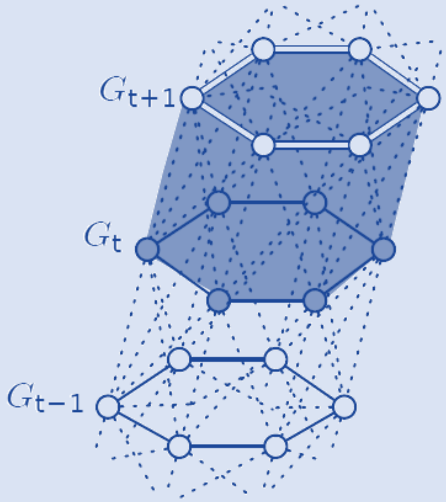
- Parts are connected by spatio-temporal edges (adjacent vertices in t-1, t, t+1)
- Induces generative probabilistic graphical model
- Allows prediction of maximum-a-posterior (MAP) estimates



Predicttive Segment Cost Estimation

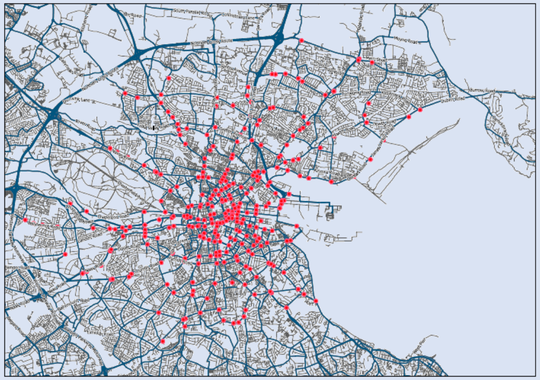
☑ Predict future sensor readings with Spatio-Temporal Random Fields

[Piatkowski et al 13]



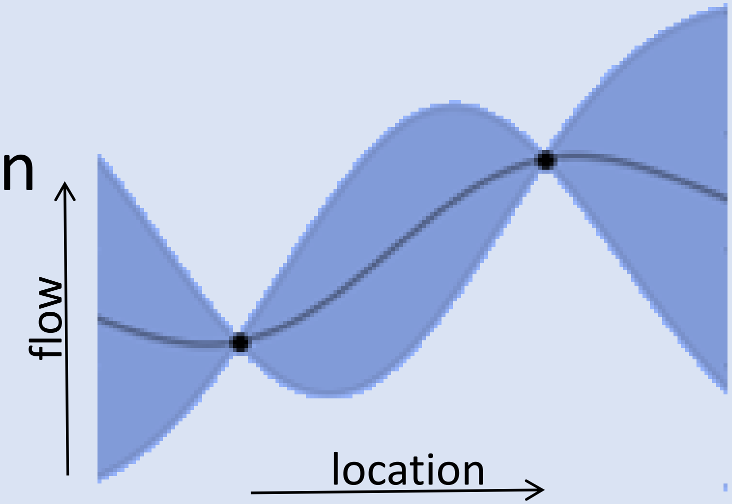
❑ Impute values for unobserved locations using Gaussian Process Regression

[Liebig et al 12]



Gaussian Process Regression

- Distribution between observed sensors depends on
  - distance,
  - path distance,
  - centrality of the sensors,
  - or others
- Shape is defined by kernel function
- Presumption: Multivariate Gaussian distribution



Gaussian Process

$$y_i = f_i + \epsilon_i, \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

$$P(\mathrm{f}|\mathbf{X}) = \mathcal{N}(0, K)$$

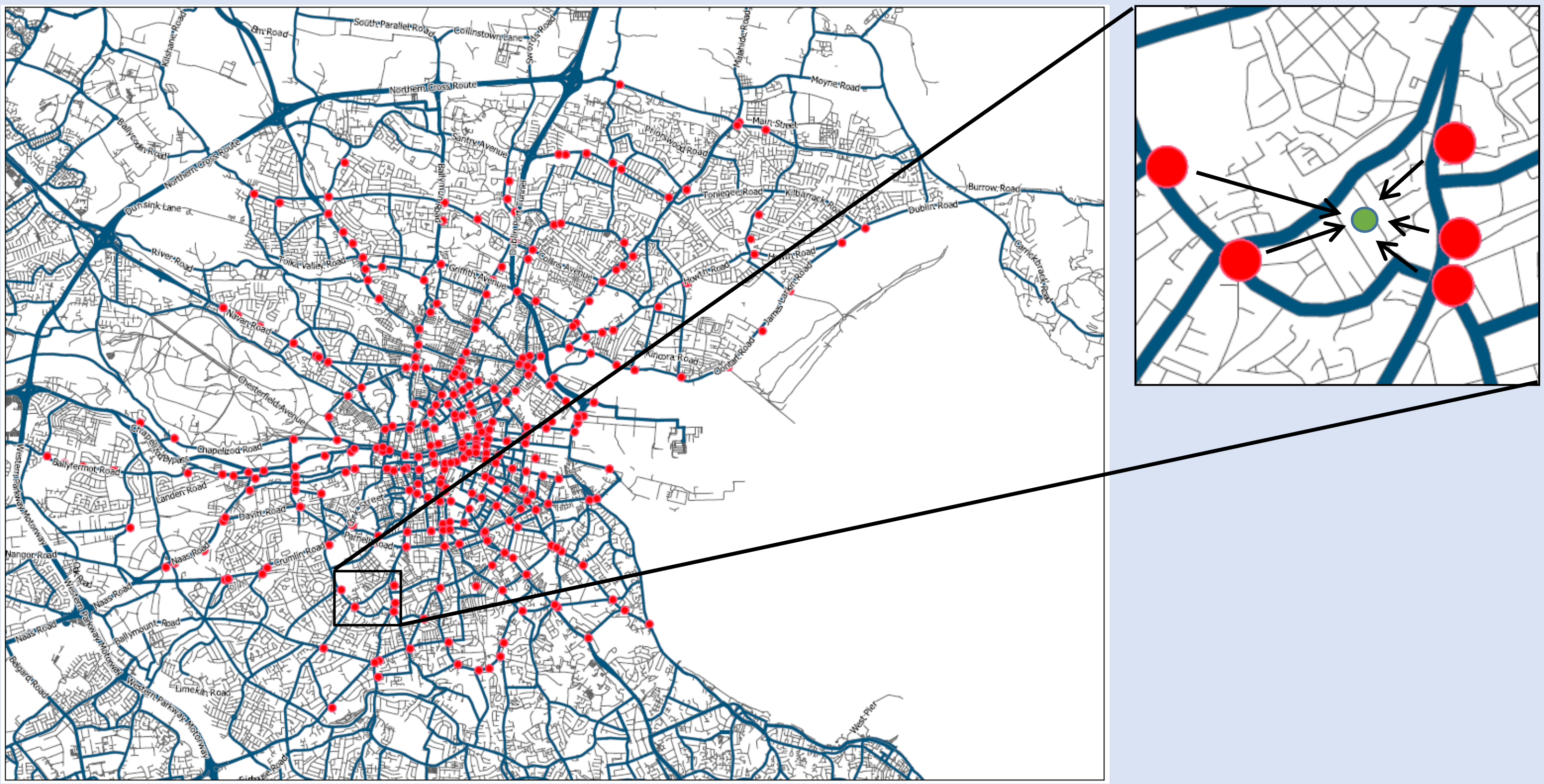
$$\begin{bmatrix} \mathrm{y} \\ \mathrm{f}_u \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \hat{K}_{-u,-u} + \sigma^2 I & \hat{K}_{-u,u} \\ \hat{K}_{u,-u} & \hat{K}_{u,u} \end{bmatrix}\right)$$

Apply regularized laplacian kernel

$$K = \left[\beta(L + I/\alpha^2)\right]^{-1}$$

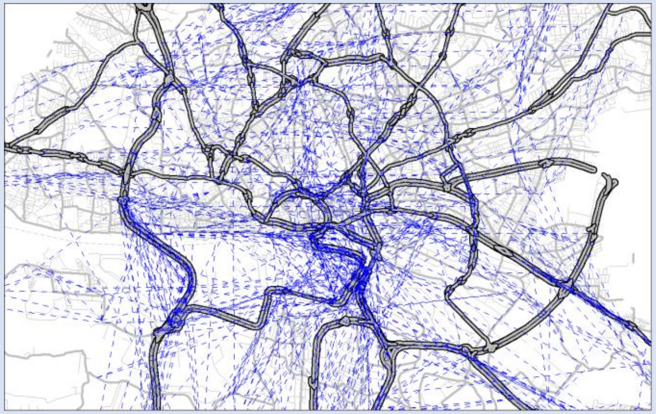
Tested also multivariate log-normality of the values using Mardia test

Speed Up Heuristic for Traffic Model



Chosing the Right Neighbours

- But, there exist far dependencies among the locations [Liebig 08]. e.g. Structure of a Bayesian Network on Hamburg:

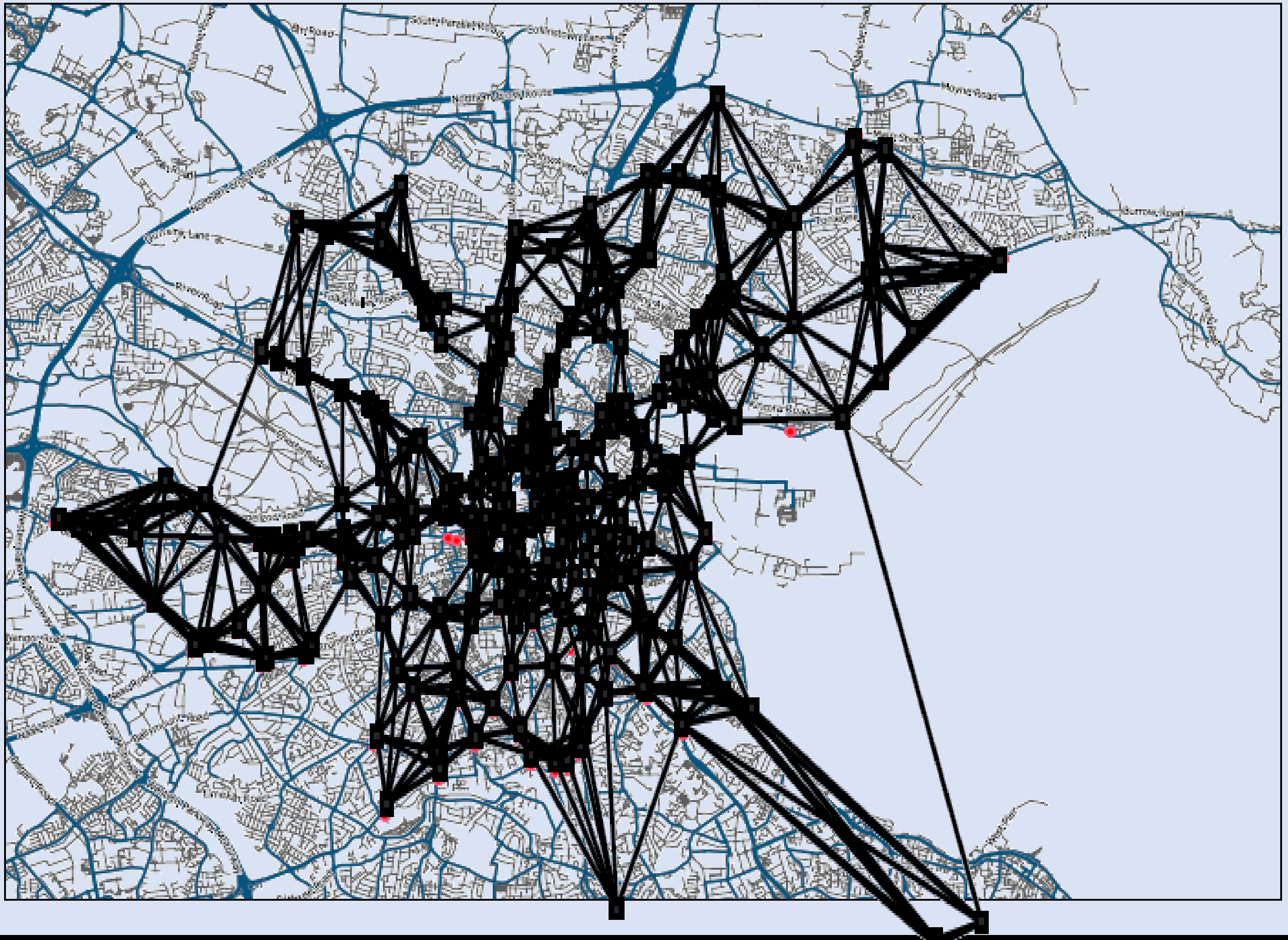


- many possible neighbourhoods to chose from
  - **spatial,**
  - shortest path,
  - attributes (street name, lanes, …) [May et al 08]
  - correlations in real trajectories

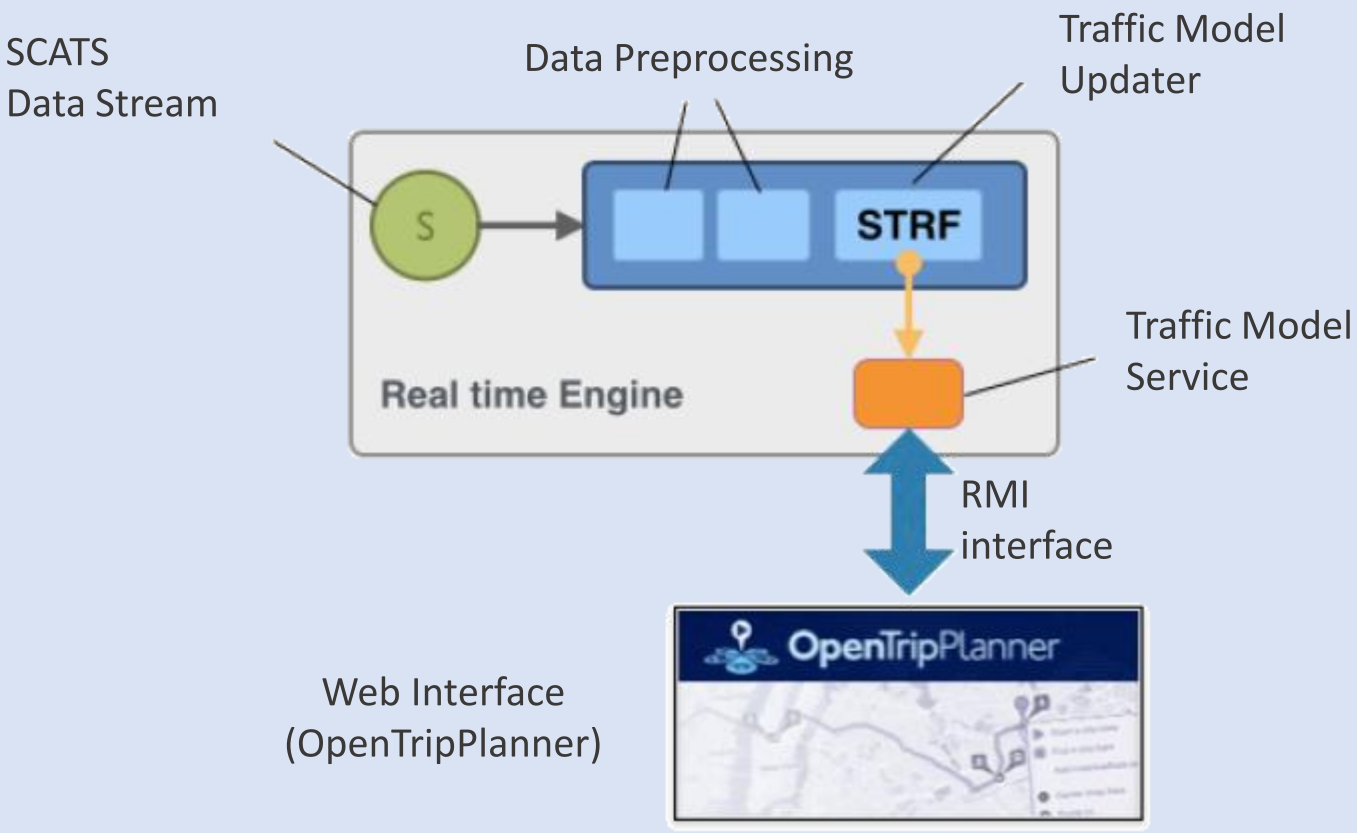
Turning the Models into Predictors

- Aggregation
  - Nodes: 7-Nearest neighbour graph among SCATS sensors
  - Traffic: 1 of 6 intervals
  - Time: 1 of 48 time slices for a 24 hour day
- Training instances: January till March‘12; Test: April
- Test for prerequisites: multivariate normal distribution
- We estimate the traffic flux at 5000 locations among the city for easy tractibility

STRF - Aggregation

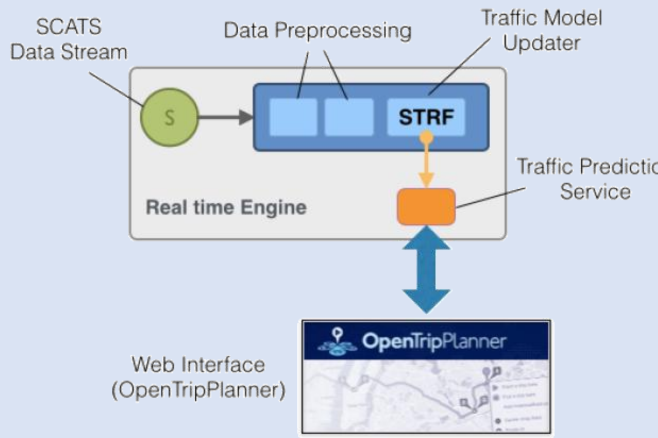


Application: Situation Aware Trip Planner



Interface to OpenTripPlanner (OTP)

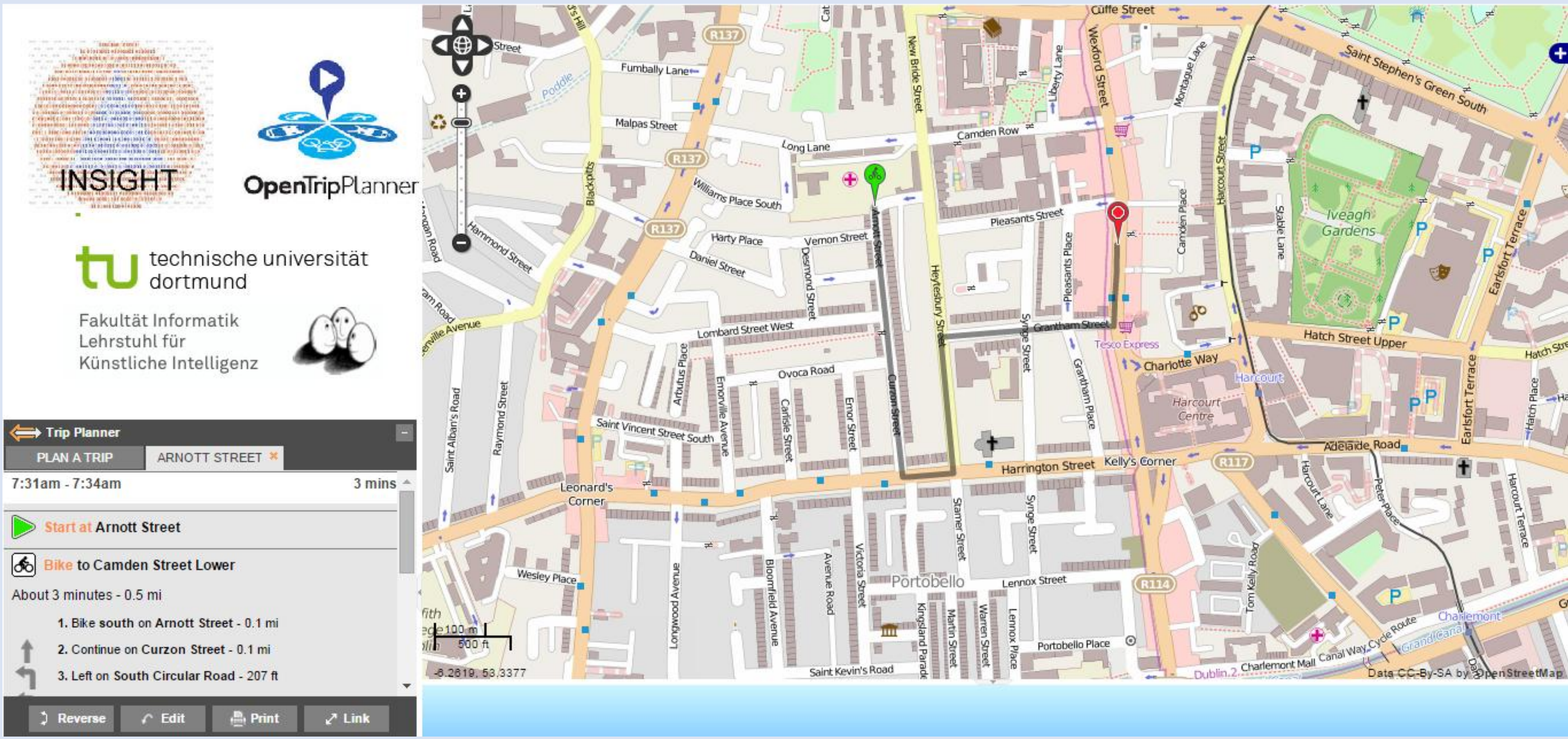
- ❑ OTP supports multimodal trip planning, traffic network derived by OpenStreetMap (vehicular routing not advisable)
- ❑ OTP calculates trip with A*
- ❑ OTP queries for cost at (location, time)
- ❑ Traffic Prediction Service (STRF, Gaussian PR) estimates the cost



Interface:

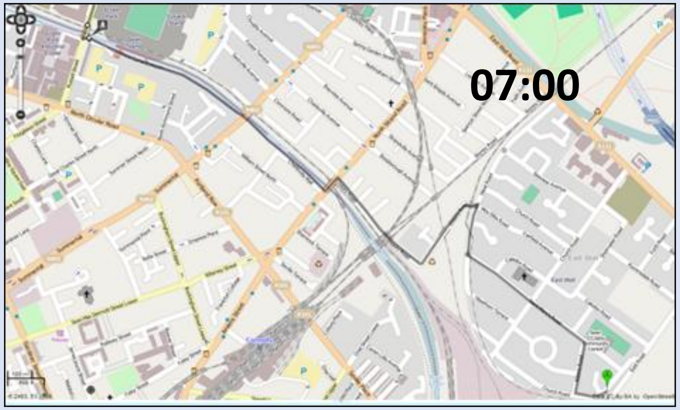
- ❑ Java RMI interface to a persistent service of the streams framework [Bockermann,Blom 12]

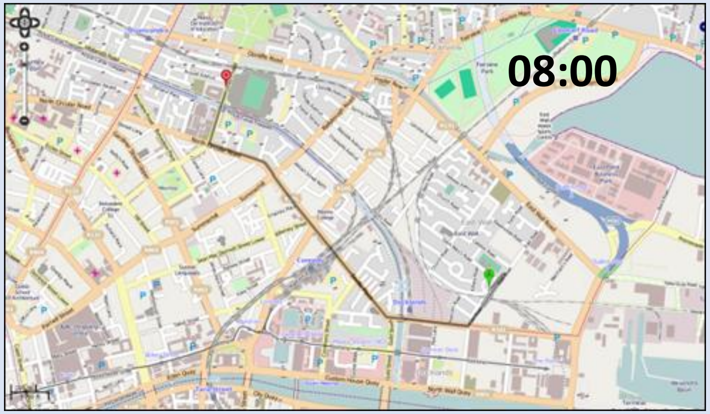
User Interface



Example

- ❑Applied to 8th April at different times





Summary

- Spatio-temporal traffic cost predictions
  - Spatio-Temporal Random Fields
  - Gaussian Process Regression
- Prediction as service in the streams framework
- OpenTripPlanner exploits traffic estimates for trip calculation

→ situation aware Trip Planner

Next steps

- ❑ Event detection by comparison of prediction with current measurement → SCATS ISA
- ❑ Enhancing through bus data
- ❑ Enhancing through crowd-sourcing component

Speed-Up Heuristics for the Traffic Flow Estimation with Gaussian Process Regression

technische universität dortmund
Fakultät Informatik
Lehrstuhl für Künstliche Intelligenz

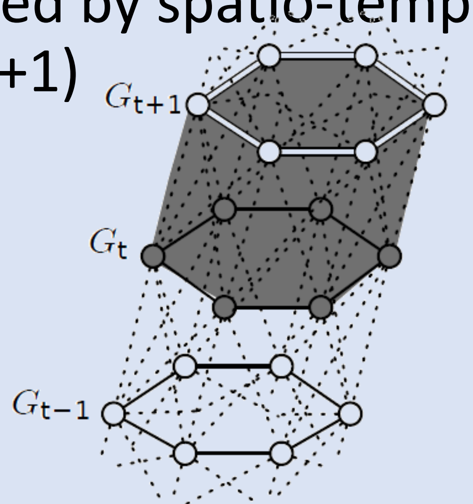
Thomas Liebig

TU Dortmund

Spatio-Temporal Random Fields

[Piatkowski et al 13]

- Spatial graph G_0 of the sensors generates measurements
- Joint measurements create a temporal chain G_1—G_2—G_3 … every G_i replicates the structure of G_0
- Parts are connected by spatio-temporal edges (adjacent vertices in t-1, t, t+1)



- Induces generative probabilistic graphical model
- Allows prediction of maximum-a-posterior (MAP) estimates

Spatio-Temporal Random Fields (STRF)

[Piatkowski et al 13]

- Full joint probability mass function:

$$p_{\boldsymbol{\theta}}(\boldsymbol{X}=\boldsymbol{x}) = \frac{1}{\Psi(\boldsymbol{\theta})} \prod_{v \in V} \psi_v(\boldsymbol{x}) \prod_{(v,w)\in E} \psi_{(v,w)}(\boldsymbol{x})$$

- With the potential

Weight for every state of the sensor and time i in [1..T]

$$\psi_v(\boldsymbol{x}) = \psi_{s(t)}(\boldsymbol{x}) = \exp\left\langle \sum_{i=1}^{t} \frac{1}{t-i+1} \boxed{\boldsymbol{Z}_{s,i}}, \boxed{\phi_{s(t)}(\boldsymbol{x})} \right\rangle$$

Indicator vector 1 at state s(t)

- Regularized MLE for parameter estimation
- MAP estimates

$$\hat{\boldsymbol{x}} = \arg \max_{\boldsymbol{x}_{V\setminus U} \in \mathcal{X}} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{V\setminus U} \mid \boldsymbol{x}_U)$$

Gaussian Process Regression

[Liebig et al 12]

- traffic flux **y** generated by stochastic process on Indexset *X* of the edges *E*

$$\mathbf{y} = f(\mathbf{x}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2)$$

$$f(\mathbf{x}) \sim \mathcal{GP}(0, K)$$

$$k(x_i, x_j) = cov(y_i, y_j)$$

- Vector of traffic flux **y** is multivariate normal distributed (we test this by Mardia test)

$$\begin{bmatrix} \mathbf{y}_s \\ \mathbf{y}_u \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K_{s,s} + \sigma_n^2 I & K_{s,u} \\ K_{u,s} & K_{u,u} \end{bmatrix}\right)$$

Gaussian Process Regression

[Liebig et al 12]

- conditional probability distribution of $\mathbf{y}_u$ for given evidence $\mathbf{y}_s$ at measurement locations x_s

$$P(\mathbf{y}_u | x_u, x_s, \mathbf{y}_s) \sim \mathcal{N}(\mu, \sigma^2)$$

$$\mu = K_{u,s}(K_{s,s} + \sigma_n^2 I)^{-1} \ \mathbf{y}_s$$

$$\sigma^2 = K_{u,u} - K_{u,s}(K_{s,s} + \sigma_n^2 I)^{-1} \ K_{s,u}$$

- We apply Regularized Laplacian Kernel, works also well in [Liebig et. al. 12]

$$K = \left[\beta(L + I/\alpha^2)\right]^{-1}$$

- Hyperparameters are found by grid search = 0.5