ETCPS: An Effective and Scalable Traffic Condition Prediction System

Dong Wang

Institute for Interdisciplinary Information Sciences
Tsinghua University, China

February 21, 2017

`Joint work with` Wei Cao, Mengwen Xu, Jian Li
IIIS, Tsinghua Univ.

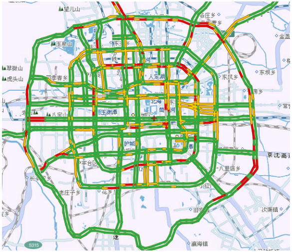
Outline

1. Our Goal
2. Challenges
3. Useful Observations
4. Traffic condition prediction system
5. Experiment Study
6. Conclusion

Our Goal

Predict the traffic condition of each road in the urban area

- after a few minutes or hours
- using the current and historical traffic conditions
  - extracted from taxi trajectories



Outline

1 Our Goal

2 Challenges

3 Useful Observations

4 Traffic condition prediction system

5 Experiment Study

6 Conclusion

Challenges

- **Data Explosion**
  - Large volume of GPS data.
- **Hard to extract patterns**
  - from traffic condition time series.
- **Varying Patterns**
  - Patterns vary significantly with time.
- **Hard to distinguish**
  - congestion and taxi picking passengers.

Outline

1 Our Goal

2 Challenges

3 Useful Observations

4 Traffic condition prediction system

5 Experiment Study

6 Conclusion

Useful Observations

Recall: **Hard to extract patterns from traffic conditions time series.**

- o_t^i: traffic condition of road r_i at time t.
  - the mean travel speed of taxi traveled on r_i at time t.
- $\mathrm{Org}^i = \{o_t^i | t = 1, \ldots, T\}$: traffic conditions time series
- Transforming Org^i can reveal very strong autocorrelations
  - **Observation 1**: Expectation-reality gap
  - **Observation 2**: First order difference of traffic condition series

Expectation-Reality Gap

- Expected traffic condition: $\mathrm{Avg^i} = \{a_t^i | t = 1, \ldots, T\}$
- Expectation-Reality **Gap**
  $\mathrm{Gap^i} = \mathrm{Org^i} - \mathrm{Avg^i} = \{g_t^i | g_t^i = o_t^i - a_t^i, t = 1, \ldots, T\}$
  - $g_t < 0$, traffic condition at t is more congested than usual.

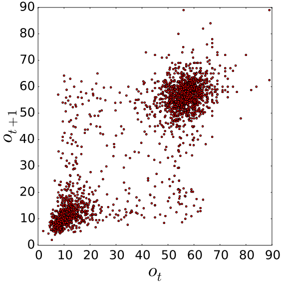


Figure: o_t and o_{t+1}

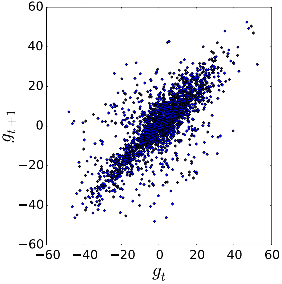
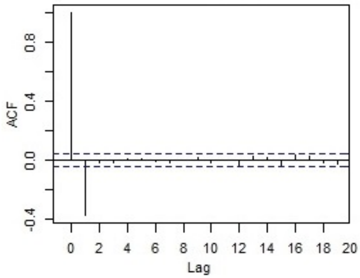


Figure: g_t and g_{t+1}

First order difference of traffic condition series

- $\mathrm{Diff}(\mathrm{Org}) = \{\delta_t^i | \delta_t^i = o_t^i - o_{t-1}^i, t = 2, \ldots, T\}$
- Autocorrelation[1] of $\mathrm{Diff}(\mathrm{Org})$ is shown with ACF[2].



[1] The autocorrelation of a random process describes the correlation between values of the process at different times with a time lag τ.

[2] ACF (Auto Correlation Function).

Outline

1 Our Goal

2 Challenges

3 Useful Observations

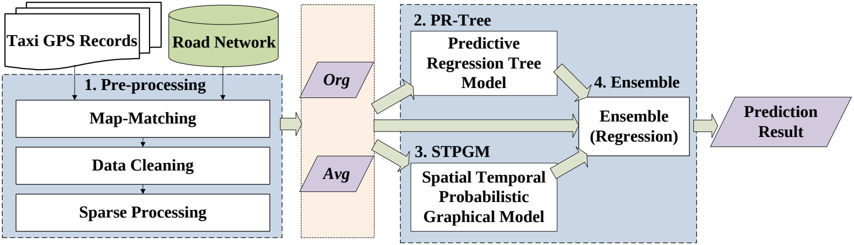
4 Traffic condition prediction system

5 Experiment Study

6 Conclusion

System Overview

- Predictive Regression Tree (PR-Tree)
- Spatial Temporal Probabilistic Graphical Model (STPGM)



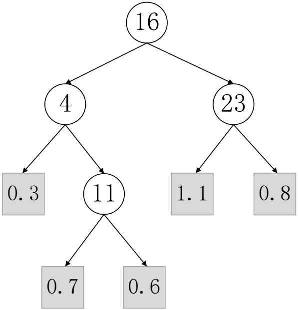
Predictive Regression Tree (PR-Tree)

Intuitions:

- regression tree based model
- g_{t+1}/g_t is piecewise linear based on the observation.
- Approximate g_{t+1} by $\hat{g}_{t+1} = g_t \cdot R(g_t)$.
- Use a decision tree to learn the function $R(.)$
- By estimating g_{t+1}, we thus obtain $\hat{o}_{t+1} = a_{t+1} + \hat{g}_{t+1}$.

Structure (PR-Tree)

- Each inner node is associated with a splitting value.
- Each leaf node has an output value θ.
- Inner nodes split the input space into several subspaces.
- Leaf nodes indicate the corresponding value of the subspaces.



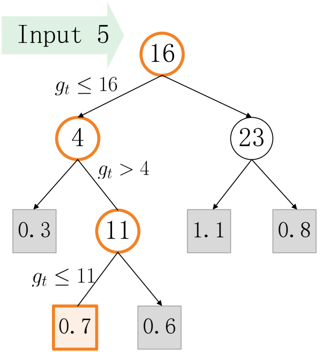
- Given input g_t, binary search the value of corresponding subspace and take it as the estimation of $R(g_t)$.

Example (PR-Tree)

Given a PR-Tree of road r_i, the current traffic condition $o_t = 45$, assume the expected traffic condition on t and $t+1$ are $a_t = 40$, $a_{t+1} = 43$.
Objective: predict o_{t+1}.

Solution:

- $g_t = o_t - a_t = 5$
- put g_t into the PR-Tree, we get $R(g_t) = 0.7$.
- approximate g_{t+1} by $\hat{g}_{t+1} = g_t \cdot R(g_t) = 3.5$
- estimate o_{t+1} by $\hat{o}_{t+1} = a_{t+1} + \hat{g}_{t+1} = 46.5$.

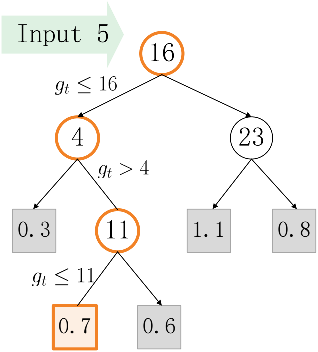


Example (PR-Tree)

Given a PR-Tree of road r_i, the current traffic condition $o_t = 45$, assume the expected traffic condition on t and $t+1$ are $a_t = 40$, $a_{t+1} = 43$.
Objective: predict o_{t+1}.

Solution:

- $g_t = o_t - a_t = 5$
- put g_t into the PR-Tree, we get $R(g_t) = 0.7$.
- approximate g_{t+1} by $\hat{g}_{t+1} = g_t \cdot R(g_t) = 3.5$
- estimate o_{t+1} by $\hat{o}_{t+1} = a_{t+1} + \hat{g}_{t+1} = 46.5$.

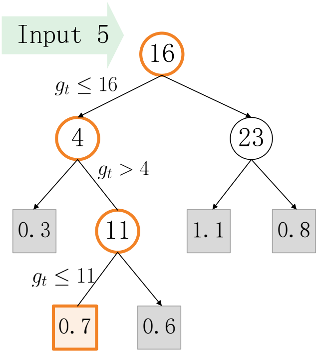


Example (PR-Tree)

Given a PR-Tree of road r_i, the current traffic condition $o_t = 45$, assume the expected traffic condition on t and $t+1$ are $a_t = 40$, $a_{t+1} = 43$.
Objective: predict o_{t+1}.

Solution:

- $g_t = o_t - a_t = 5$
- put g_t into the PR-Tree, we get $R(g_t) = 0.7$.
- approximate g_{t+1} by $\hat{g}_{t+1} = g_t \cdot R(g_t) = 3.5$
- estimate o_{t+1} by $\hat{o}_{t+1} = a_{t+1} + \hat{g}_{t+1} = 46.5$.

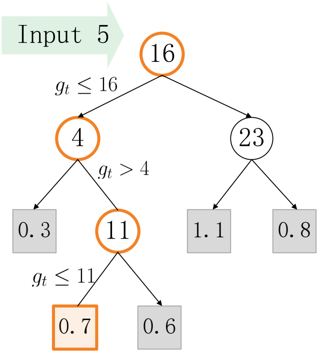


Example (PR-Tree)

Given a PR-Tree of road r_i, the current traffic condition $o_t = 45$, assume the expected traffic condition on t and $t+1$ are $a_t = 40$, $a_{t+1} = 43$.
Objective: predict o_{t+1}.

Solution:

- $g_t = o_t - a_t = 5$
- put g_t into the PR-Tree, we get $R(g_t) = 0.7$.
- approximate g_{t+1} by $\hat{g}_{t+1} = g_t \cdot R(g_t) = 3.5$
- estimate o_{t+1} by $\hat{o}_{t+1} = a_{t+1} + \hat{g}_{t+1} = 46.5$.



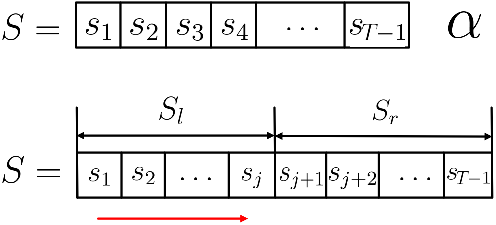
Training Method (PR-Tree)

- Input: Series $\mathrm{Gap} = \{g_1, \ldots, g_T\}$.
- Objective: Find $R^* = \mathrm{argmin}_R \sum_{t \in [1,T)} (g_{t+1} - R(g_t) \cdot g_t)^2$.
- Algorithm:
  - Step 1.1: Define $s_j = (g_j, g_{j+1})$ and sort all s_j with $s_j^{(1)}$.
  - Step 1.2: Define $f(S) = \min_\alpha \sum_{j \in [1,T)} (s_j^{(2)} - \alpha \cdot s_j^{(1)})^2$

$$S = \boxed{s_1 \mid s_2 \mid s_3 \mid s_4 \mid \quad \cdots \quad \mid s_{T-1}} \quad \alpha$$

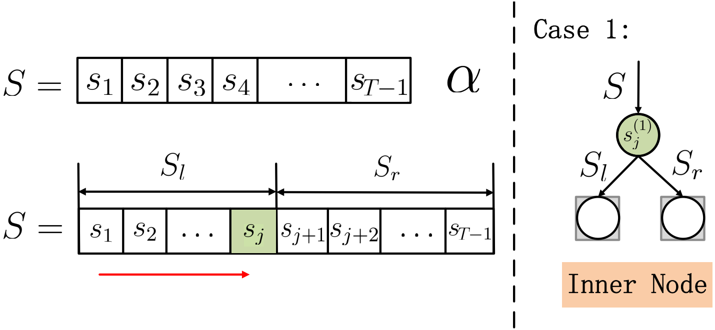
Training Method (PR-Tree)

- Input: Series $\mathrm{Gap} = \{g_1, \ldots, g_T\}$.
- Objective: Find $R^* = \mathrm{argmin}_R \sum_{t\in[1,T)} (g_{t+1} - R(g_t) \cdot g_t)^2$.
- Algorithm:
  - Step 2: Find $S_l \in \mathrm{Prefix}(S)$ s.t. $f(S_l) + f(S_r)$ is minimized. $(S_r = S - S_l)$.



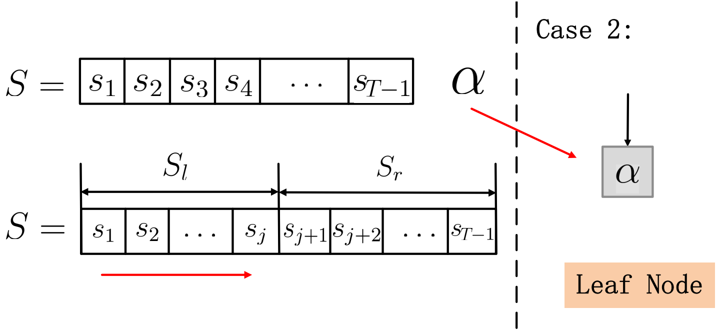
Training Method (PR-Tree)

- Input: Series $\text{Gap} = \{g_1, \ldots, g_T\}$.
- Objective: Find $R^* = \text{argmin}_R \sum_{t \in [1,T)} (g_{t+1} - R(g_t) \cdot g_t)^2$.
- Algorithm:
  - Step 3: If $f(S_l) + f(S_r) < f(S) - \gamma$, split the current node recursively. Let the splitting value be $s_j^{(1)}$.



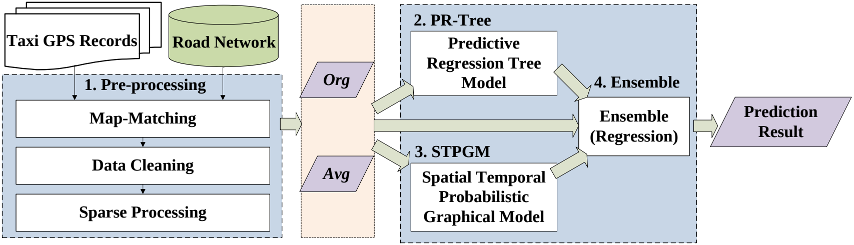
Training Method (PR-Tree)

- Input: Series $\mathrm{Gap} = \{g_1, \ldots, g_T\}$.
- Objective: Find $R^* = \mathrm{argmin}_R \sum_{t\in[1,T)} (g_{t+1} - R(g_t) \cdot g_t)^2$.
- Algorithm:
  - Step 4: Otherwise, let the current node be a leaf node and set the output value as $\mathrm{argmin}_\alpha \sum_{j\in[1,T)} (s_j^{(2)} - \alpha \cdot s_j^{(1)})^2$.



System Overview

- Predictive Regression Tree (PR-Tree)
- Spatial Temporal Probabilistic Graphical Model (STPGM)



Spatial Temporal Probabilistic Graph (STPGM)

Intuitions:

- PR-Tree does not consider the correlation between the road segments;
- Some roads are easily affected by its neighbors.

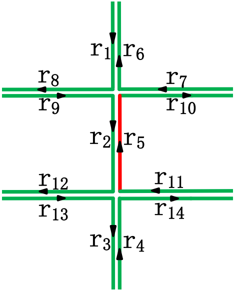


Figure: Road condition in t

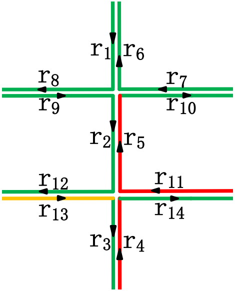
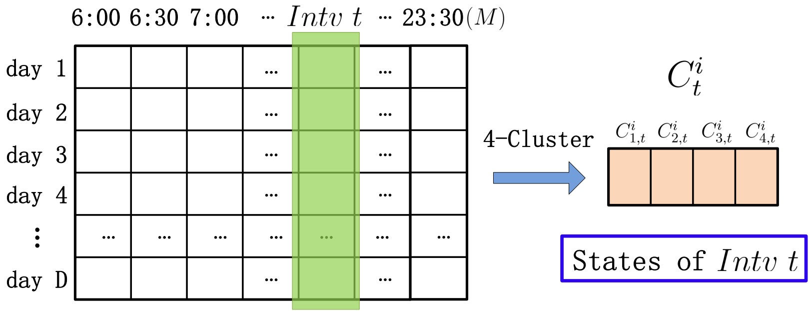


Figure: Road condition in $t+1$

States of STPGM

Discretize the traffic conditions into different states.



Example:

- *Intv* $t_1 = $ 6:00-6:30, $C_{t_1}^i = \{44, 48, 52, 58\}$(km/h)
- *Intv* $t_2 = $ 8:30-9:00, $C_{t_2}^i = \{15, 25, 32, 38\}$(km/h) **More Congested**

Parameter Learning of STPGM

- Consider a road r_i, let $\{r_i\} \cup Neib(r_i) = \{r_{i_1}, \ldots, r_{i_n}\}$ and the corresponding states at time t are $\{c^{i}_{x_i,t}, c^{i_1}_{x_{i_1},t}, c^{i_2}_{x_{i_2},t}, \ldots, c^{i_n}_{x_{i_n},t}\}$.
- Our goal:

$$\begin{aligned} & P(s^i_{t+1} = c^i_{x_i,t+1} | s^{i_1}_t = c^{i_1}_{x_{i_1},t}, s^{i_2}_t = c^{i_2}_{x_{i_2},t}, \ldots, s^{i_n}_t = c^{i_n}_{x_{i_n},t}) \\ \propto \quad & P(s^i_{t+1} = c^{x_i}_{i,t+1}, s^{i_1}_t = c^{i_1}_{x_{i_1},t}, , \ldots, s^{i_n}_t = c^{i_n}_{x_{i_n},t}) \end{aligned} \tag{1}$$

- The state space in Equation 1 explodes exponentially.

Parameter Learning of STPGM

- Approximate Equation 1 by

$$P(s^i_{t+1} = c^{x_i}_{i,t+1}) \prod_{j=1}^{n} P(s^t_{i_j} = c^{x_{i_j}}_{i_j,t} | s^i_{t+1} = c^i_{x_i,t+1})$$

- Only need to calculate
  - $P(s^i_{t+1} = c^{x_i}_{i,t+1})$
  - $P(s^t_{i_j} = c^{x_{i_j}}_{i_j,t} | s^i_{t+1} = c^i_{x_i,t+1})$

Prediction of STPGM

Given the traffic conditions of the road network at time t:

- Obtain the states of r_i and $Neib(r_i)$ at time t
- Use Equation 1 to infer the probability of each state for r_i at time t
- Select the state with the largest probability as the predicted state and the corresponding cluster center as the predicted traffic condition.

Ensemble

- The performances of PR-Tree and STPGM differ in different roads.
  - For roads which are rarely affected by their neighbors **PR-Tree wins**
  - For the roads which are highly affected by its neighbors, especially the roads that only few GPS records are observed **STPGM wins**

- Our prediction system ETCPS combines these two models, and the experiments show that it achieves a higher accuracy for the prediction.

Alternate of the model input

- Recall:
  - Gap as the input of PR-Tree.
  - Org as the input of STPGM.
- The input of our models can be any time series:
  - Gap, Org, $\mathrm{Diff(Org)}$, $\mathrm{Diff(Kal)}$ $\ldots$

Note: Kal is Org filtered with Kalman filtering.

Outline

1. Our Goal
2. Challenges
3. Useful Observations
4. Traffic condition prediction system
5. Experiment Study
6. Conclusion

Experiment Setting

- **Data set**:
  - Taxi Trajectories: Generated by 12,000 taxicabs in Beijing, from Nov. 1 to Dec. 31, 2012.
  - Road networks: 10,812 roads (**Standard**) and 101,672 roads (**Sparse**).

- **Measurement**:
  - $\mathrm{MAE} = \frac{1}{|E|}\sum_{i=1}^{|E|}\sum_{t=1}^{T}|p_t^i - o_t^i|$ **Mean Absolute Error**
  - $\mathrm{MRE} = \frac{1}{|E|}\sum_{i=1}^{|E|}\sum_{t=1}^{T}|p_t^i - o_t^i|/o_t^i$ **Mean Relative Error**
  - $\mathrm{MSE} = \frac{1}{|E|}\sum_{i=1}^{|E|}\sum_{t=1}^{T}(p_t^i - o_t^i)^2$ **Mean Squared Error**

Performance Evaluation

- **Performance of different models**

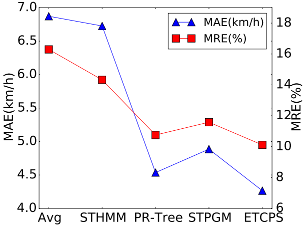


Figure: All Models (STD)

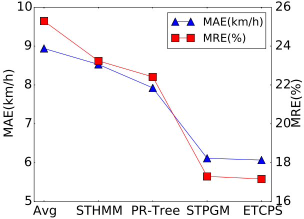


Figure: All Models (Sparse)

STHMM (Spatio-Temporal Hidden Markov Model) is proposed by Yang, B. et al in VLDB 13, which is similar to STPGM, using Mixture Gaussian to model traffic condition states, and using coupled HMM to model the interactions among roads.

Performance Evaluation

- **Verifying the observed patterns**: using $\mathrm{Diff(Org)}$ and Gap as input, the performances improve significantly.

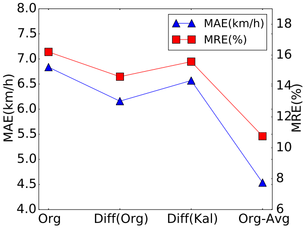


Figure: PR-Tree(STD)

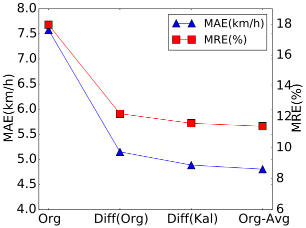


Figure: STPGM(STD)

- $\mathrm{Diff(Kal)}$ represents the first order difference of Kalman Filtered traffic condition time series. $\mathrm{Org\text{-}Avg}$ represents the Gap.

Performance Evaluation

- **Training time cost**

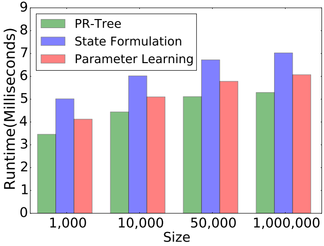


Figure: Training time cost

Size	PR-Tree	State Formulation	Parameter Learning
10^3	0.04	1.72	0.22
10^4	0.46	17.46	2.09
$5 * 10^4$	2.14	88.1	10.09
10^5	3.26	176.6	19.61

Figure: Training time cost (Minutes)

Outline

1. Our Goal
2. Challenges
3. Useful Observations
4. Traffic condition prediction system
5. Experiment Study
6. Conclusion

Conclusion

- A very fundamental but challenging task
  - Data Explosion
  - Hard to extract patterns
  - Varying Patterns
  - Hard to distinguish
- We present two useful observed correlations in the traffic condition data, which are the bases of our design.
- Our method
  - Predictive Regression Tree Model (PR-Tree)
  - Spatial Temporal Probabilistic Graph Model (STPGM)
- Our system provides high-quality predictions and can easily scale to very large datasets.

Thank you!