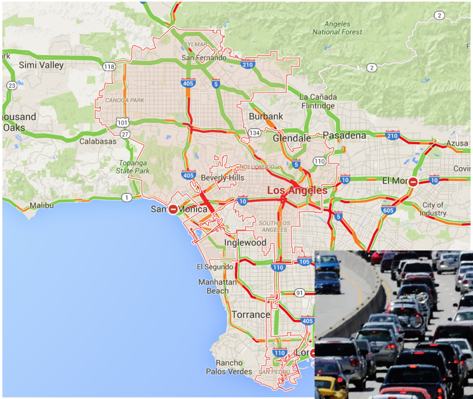
Adaptive Routing

Gaussian Process Dynamic Congestion Models

Tobias Bischoff, Brennan Young, Ivy Jiyun Xiao, Luciana Cendon

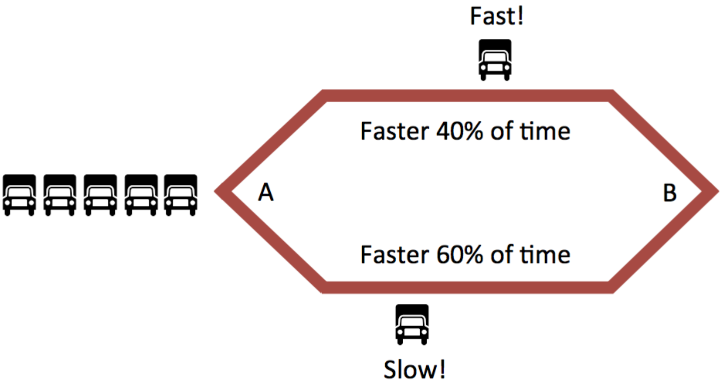
Introduction: The Problem



Introduction: The Problem

- Collectively route fleet of vehicles through a city
- **Goal:** Minimize total travel time
- In the absence of traffic sensors becomes an exploration-exploitation problem
- Must be done in real-time due to constantly changing traffic conditions

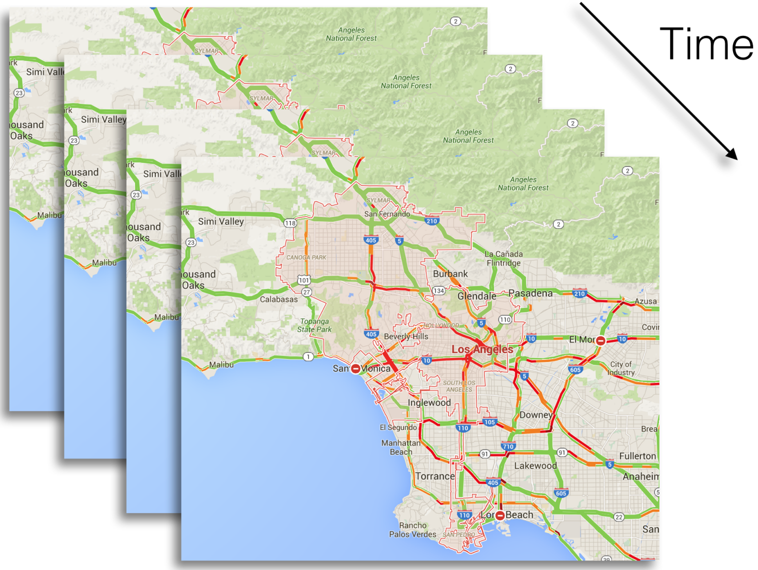
Introduction: The Problem



Introduction: Today's Approach

1. Build a probabilistic model of traffic conditions that can be updated quickly using new data
2. Use a UCB-type algorithm to route vehicle fleet in real-time using the canonical routes
3. Show that this heuristic algorithm is as good as omniscient algorithms

Model: Input Space



Model: Requirements

What kind of requirements exist for our traffic model?

- Reliably captures uncertainty in traffic conditions with time
- Makes posterior (after data collection) predictions and can be updated quickly with new data
- Can be evaluated quickly since we like real-time usability

Model: Gaussian Processes (GP)

Model the distribution of travel times as a GP with input points $x = (r, z)$, where r denotes a road segment and z a point in time.

- Use historical data to define a GP over travel times, which can be easily updated using collected data fro routed vehicles.
- Basically an infinite-dimensional Gaussian distribution
- Given a set of points $x_1, \ldots, x_N$ (here street segments and time), we have that $p(f(x_1), \ldots, f(x_N))$ is jointly Gaussian with mean $\mu(x)$ and covariance $\Sigma_{ij} = k(x_i, x_j) + \sigma_y^2 \delta_{ij}$
- Can estimate $\mu(x)$ and $\Sigma(x)$ from noisy observations
- Can make posterior predictions by conditioning on seen data

Foundations: Gaussian Processes Inference

- Get some data $Y = [Y_1, \ldots, Y_T]$ at $X = [(r_1, z_1), \ldots, (r_T, z_T)]$
- Calculate an estimate of the mean $\hat{\mu}$ and covariance $\hat{\Sigma}$
- Realize that the joint distribution of unseen and seen data y_* is

$$(\hat{y}, y_*) \sim \mathcal{N}\left((\hat{\mu}, \mu_*), \begin{pmatrix} \hat{\Sigma} & \Sigma_* \\ \Sigma_*^T & \Sigma_{**} \end{pmatrix}\right) \quad (1)$$

- Conditioning on data leads to posterior predictive PDF
- This approach allows for fast posterior prediction and updating with new data (**major advantage of GP**)

Foundations: Gaussian Processes Inference

The estimation equations are given by

$$\mu_* = \mu(X_*) + \Sigma_*^T \Sigma^{-1}(\hat{y} - \hat{\mu}) \tag{2a}$$

$$\Sigma_* = \Sigma_{**} - \Sigma_*^T \hat{\Sigma}^{-1} \Sigma_* \tag{2b}$$

What now?

- Need a framework for estimating $\hat{\mu}$ and $\hat{\Sigma}$ quickly
- Main issue is smoothing procedure for time coordinate

Gaussian Process Dynamic Congestion Models

Recall:

- $r \in \mathcal{R}$: Road segments
- $z \in \mathcal{Z}$: Contexts (time)
- $f(r, z)$: Travel speed of road r at time z

A Gaussian process prior is fully specified by:

- $\mu(r, z) = \mathbb{E}[f(r, z)]$
- $k((r, z), (r', z')) = \mathbb{E}[f(r, z) - \mu(r, z))(f(r', z') - \mu(r', z'))]$

Gaussian Process Dynamic Congestion Models

How should we compute μ and k?
Estimate them from historical data.

Estimating μ

Key assumption: Temporal regularity. $\mu(r, z)$ for future times will be like $\mu(r, z)$ in the past.

Estimating μ

Example historical data set S:

Road	Time	Speed
1	2014-03-25 18:35:32	18
1	2014-03-25 18:43:13	15
1	2014-03-25 18:43:13	13
1	2014-03-25 18:43:13	12
1	2014-03-26 04:00:32	88
1	2014-03-26 04:01:19	96
1	2014-03-26 04:03:01	75
1	2014-03-26 17:10:12	15
2	2015-05-16 06:14:19	12
2	2015-05-17 00:01:54	80
	$\vdots$	

Estimating μ: A Naïve Approach

$$\mu(r, z) = \text{mean}\{y_{r,z'} | (r, z', y_{r,z'}) \in S\} \text{ (for arbitrary } z' \in \mathcal{Z})$$

In words: $\mu(r, z)$ is estimated as the mean of the speeds observed for road r.

Estimating μ: A Naïve Approach

Road	Time	Speed
1	**2014-03-25 18:35:32**	**18**
1	**2014-03-25 18:43:13**	**15**
1	**2014-03-25 18:43:13**	**13**
1	**2014-03-25 18:43:13**	**12**
1	**2014-03-26 04:00:32**	**88**
1	**2014-03-26 04:01:19**	**96**
1	**2014-03-26 04:03:01**	**75**
1	**2014-03-26 17:10:12**	**15**
2	2015-05-16 06:14:19	12
2	2015-05-17 00:01:54	80
	⋮	

$\mu(1, z) = 41.5$

Estimating μ: A Naïve Approach

E. Del Mar Blvd. at 17:41

E. Del Mar Blvd. at 23:02

Estimating μ: Slightly better

Define $\tau(z)$ to be the time of day associated with z, and then:
$\mu(r,z) = \text{mean}\{y_{r,z'} | ((r, z', y_{r,z'}) \in S) \wedge (\tau(z') = \tau(z))\}$

In words: $\mu(r,z)$ is estimated as the mean of the speeds observed for road r with time of day matching z.

Estimating μ: Slightly better

Example: $\tau(\text{2016-07-12 13:14:18}) = \text{13:14:18}$
$\mu(1, \text{2016-07-12 13:14:18})$ "$=$" $\mu(1, \text{13:14:18})$

Remark: This choice of τ assumes that the temporal regularity "period" is one day. We gain empirical data for which to make our estimate at the cost of being unable to consider particular types of days (e.g. weekends, holidays).

Estimating μ: Slightly better

Road	Time	Speed	
1	**2014-03-25 18:35:32**	**18**	$\}\mu(1, 18{:}35{:}32) = 18$
1	**2014-03-25 18:43:13**	**15**	
1	**2014-03-25 18:43:13**	**13**	$\}\mu(1, 18{:}43{:}13) = 13.33$
1	**2014-03-25 18:43:13**	**12**	
1	**2014-03-26 04:00:32**	**88**	$\}\mu(1, 04{:}00{:}32) = 88$
1	**2014-03-26 04:01:19**	**96**	$\}\mu(1, 04{:}01{:}19) = 96$
1	**2014-03-26 04:03:01**	**75**	$\}\mu(1, 04{:}03{:}01) = 75$
1	**2014-03-26 17:10:12**	**15**	$\}\mu(1, 17{:}10{:}12) = 15$
2	2015-05-16 06:14:19	12	
2	2015-05-17 00:01:54	80	
	⋮		

Estimating μ: Even better

Use τ to partition time into intervals, e.g. 18:30-18:40.
Example: $\tau(\text{2016-07-12 13:14:18}) = \text{13:10-13:20}$
$\mu(1, \text{2016-07-12 13:14:18})$ "$=$" $\mu(1, \text{13:10-13:20})$
Side remark: We made a similar partitioning for the roads earlier, implicitly when defining $\mathcal{R}$.

Estimating μ: Even better

Road	Time	Speed	
1	**2014-03-25 18:35:32**	**18**	$\}\mu(1, 18{:}30\text{-}18{:}40) = 18$
1	**2014-03-25 18:43:13**	**15**	$\mu(1, 18{:}40\text{-}18{:}50) = 13.33$
1	**2014-03-25 18:43:13**	**13**	
1	**2014-03-25 18:43:13**	**12**	
1	**2014-03-26 04:00:32**	**88**	$\mu(1, 04{:}00\text{-}04{:}10) = 86.33$
1	**2014-03-26 04:01:19**	**96**	
1	**2014-03-26 04:03:01**	**75**	
1	**2014-03-26 17:10:12**	**15**	$\}\mu(1, 17{:}10\text{-}17{:}20) = 15$
2	2015-05-16 06:14:19	12	
2	2015-05-17 00:01:54	80	

$\vdots$

Temporal Smoothing

We can use temporal smoothing to further-improve estimates μ.
$Y_S^{(t)}(r) = \{y|(r,z,y) \in S \wedge \tau(z) = \tau(t)\}$
$H_S^{(t)}(y|r) = \beta H_S^{(t-\gamma)}(y|r) + (1-\beta)(1 - \mathrm{CDF}(y|Y_S^{(t)}))$
$\mu(r,z) = \mathbb{E}_{y\sim H}[y]$

- $Y_S^{(t)}(r)$: Empirical distribution of travel times
- β, γ, smoothing parameters

Estimating k

Main assumption: k can be decomposed by road and context. All roads have the same temporal covariance.

$$k((r,z),(r',z')) = k_1(r,r')k_2(z,z')$$

Estimating k_1

$$k_1(r, r') = \text{mean}\{(y_{r,z} - \mu(r, z))(y_{r',z} - \mu(r', z)) : \\ (r, z, y_{r,z}), (r', z, y_{r',z}) \in S\}$$

Note: The observations $(r, z, y_{r,z})$ and $(r', z, y_{r',z})$ should share exactly the same context z. We can employ temporal smoothing here, just as with μ.

Estimating k_2

We make the additional assumption that $k_2(z, z')$ depends only on the difference $|z - z'|$.

$$k_2(z, z') = \text{mean}\{(y_{r,z_1} - \mu(r, z_1))(y_{r,z_2} - \mu(r, z_2)) : \\ (y_{r,z_1}, y_{r,z_2} \in S) \wedge (|z_1 - z_2| = |z - z'|)\}$$

Summary of Assumptions

- μ displays temporal regularity with respect to z (in the context of this paper, the regularity period is assumed to be 1 day).
- Time can be partitioned into intervals of non-zero length (in the context of this paper, into 10-minute intervals). Roads, similarly, are partitioned into road segments.
- The covariance kernel can be decomposed; $k((r, z), (r', z')) = k_1(r, r')k_2(z, z')$
- $k_2(z, z')$ depends only on $|z - z'|$

Summary of Equations

- $\mu(r,z) = \text{mean}\{y_{r,z'} | ((r,z',y_{r,z'}) \in S) \wedge (\tau(z') = \tau(z))\}$
- $k((r,z),(r',z')) = k_1(r,r')k_2(z,z')$
- $k_1(r,r') = \text{mean}\{(y_{r,z} - \mu(r,z))(y_{r',z} - \mu(r',z)) :$
  $(r,z,y_{r,z}),(r',z,y_{r',z}) \in S\}$
- $k_2(z,z') = \text{mean}\{(y_{r,z_1} - \mu(r,z_1))(y_{r,z_2} - \mu(r,z_2)) :$
  $(y_{r,z_1}, y_{r,z_2} \in S) \wedge (|z_1 - z_2| = |z - z'|)\}$

Algorithm

Assumption: The road congestion conditions behave according to GPDCM.

We'll use this later when we get to the PCR algorithm.

Let's start with the simplest algorithm.

Algorithm - Overall Structure

N trucks (N source locations and N destinations), 1 road map.

At each time step t:

The algorithm

1. Receive observations $Y_t = \{y_{t,1}, ..., y_{t,N}\}$, $y_{t,n}$ is observed speed of the roads, i.e., truck's travel speed
2. Receive state of vehicles $L_t = \{l_{t,1}, ..., l_{t,N}\}$, $l_{t,n} = (r, \tau)$, r is a road, τ is the expected amount of time needed to travel r
3. **Planner**: Output (Routing) decisions $\Psi_t = \{(r_1, \tau_1), ..., (r_N, \tau_N)\}$
4. Gives decisions Ψ_t to the trucks
5. $t \leftarrow t + 1$

Trucks execute them.

Algorithm - Planner

Planner(L_t, Y_t) at each time step t:
Select route that:

- minimize expected travel time for each truck.
- maximize information gain of the road congestion situation

Exploitation vs. Exploration

Algorithm - Planner - Exploration

How to quantify exploration?

Use Information Gain:

$$IG(y_\psi, f) \equiv H(f) - H(f|y_\psi)$$

,where H is information entropy, f is our congestion model, $y_\psi = \{y_{r,\tau}\}, (r, \tau) \in \Psi$ are the observations made by road segments traveled. Both f and y_Ψ are r.v.

The reduction in entropy of our congestion model f given observations y_Ψ.

Algorithm - Planner - Exploration

- Simple greedy selection is effective in maximizing $IG(y_\Psi, f)$
- For each vehicle $n = 1, 2, ..., N$
  Find the observation that maximizes Incremental Information Gain

$$\delta IG(\hat{y}|y_\Psi, f) = IG(y_\Psi \cup \{\hat{y}\}; f) - IG(y_\Psi; f)$$

Algorithm - Planner - Exploration

Why recompute information gain each iteration?

Information gain of routes change after the algorithm has committed to a route.

Algorithm - Planner - Exploration

Account for long-term values in the exploration term

$$\delta IG(y_{\vec{r}}|y_\Psi, f) = IG(y_\Psi \cup \{y_{\vec{r}}\}; f) - IG(y_\Psi; f)$$

, where $\vec{r}$ is an entire route.

Algorithm- Planner - Exploration

Two Approximations:
1. For each route $\vec{r} \in \Psi$, approximate y_r using $y_{\vec{r}} = \{y_{(r,\tau)}\}$,where r is a road, τ is expected travel time of the road.
2. f is a Gaussian process. Use GPDCM.

Algorithm- Planner - Exploration

Thus, the incremental variance reduction of running any route $\vec{r}$ given intermediate solution Ψ and recent observations $O \equiv (L, Y)$ can be written as

$$\delta IG(y_{\vec{r}}|y_{\Psi}, f) = k(\vec{r}, \vec{r}|O, \Psi) = \sum_{(r,\tau)\in E[\vec{r}]} k((r,\tau),(r,\tau)|O,\Psi,E[\vec{r}])$$

(Equation 10)

Algorithm - Planner - Balancing Exploration vs. Exploitation

Utility Function

$$U(\Psi) = \sum_{n=1}^{N} c_n(\Psi \cap P_n) + \alpha IG(y_\psi, f)$$

α is the tradeoff between exploration and exploitation
P_n are all feasible routes (any route that could end at the vehicle's destination)
c_n is the exploitation term that measures the difference in expected travel time between the selected route for vehicle n and the best route $\vec{r_n}$

Algorithm - Planner - Balancing Exploration vs. Exploitation

Maximize this Utility Function

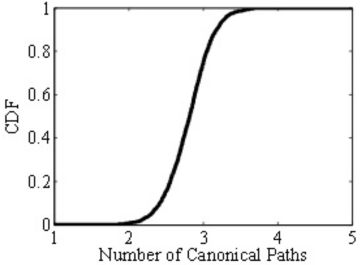
Incremental utility gain!

For each $n = 1, 2, ..., N$, maximize

$$\delta U_n(\vec{r}|\Psi) \equiv c_n(\vec{r}) + \alpha k(\vec{r}, \vec{r}|O, \Psi), \forall \vec{r} \in \Psi \cap P_n$$

Algorithm - Planner - Planning with Canonical Routes (PCR)

Sample K traffic scenarios from GPDCM $\rightarrow$ get travel speed of all routes at all times about each route (Complete Information) $\rightarrow$ compute optimal routes $\leftarrow$ **Canonical Paths**



On average 3.2 optimal(canonical) routes per vehicle.

Algorithm - Planner - Planning with Canonical Routes (PCR)

Recall maximizing Incremental utility
For each $n = 1, 2, ..., N$, maximize

$$\delta U_n(\vec{r}|\Psi) \equiv c_n(\vec{r}) + \alpha k(\vec{r}, \vec{r}|O, \Psi), \forall \vec{r} \in \Psi \cap P_n$$

$P_n =$ is now set of canonical routes $\approx 3.2!$

Experiments and Results

- Dataset:
  - **City 1:** One year (from January 2008 to December 2008) of GPS data from 15,000 taxis in Shenzhen, China.
  - **City 2:** One year (from January 2006 to December 2006) of GPS data from 5,600 taxis in Shanghai, China.
- Validation:
  - Gaussian Process Dynamic Congestion Model (GPDCM)
  - GP-UCB and PCR algorithm
- Performance Evaluation
- Preliminary Field Study

GPDCM Validation - Kolmogorov–Smirnov test

- K-S test: used in statistics to quantify the distance between two distributions

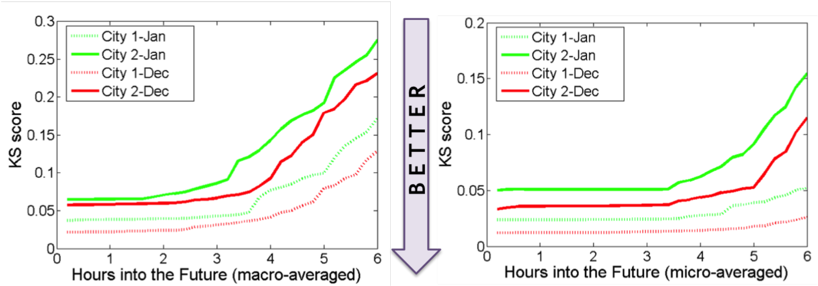
$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I_{X_i \le x}$$

$$I_{X_i \le x} \begin{cases} 1 \text{ if } X_i \le x \\ 0 \text{ otherwise} \end{cases}$$

- Measures how well the prediction of future traffic conditions matches the real distribution observed in nature
- KS-Score:

$$Z_n = \sup_x |F_n(x) - F(x)|$$

GPDCM Validation - Kolmogorov–Smirnov test



Liu, Yue & Krishnan, TKDE 2015

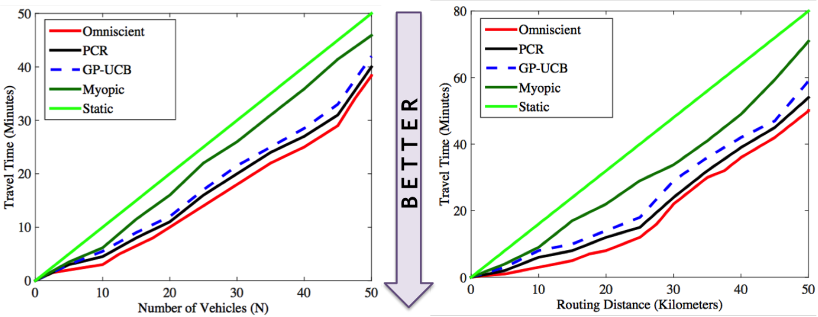
- Macro-averaged KS score: computes KS score for each day separately
- Micro-averaged KS score: aggregating all days into a single day
- GPDCM is an effective model for predicting the distribution of future traffic conditions given real-time observations

Routing Validation

GP-UCB and PCR algorithms were compared against conventional baselines and omniscient lower bound

- **Static Routing:** routing based on prior mean. Routes are not updated in response to real-time observations.
- **Myopic Routing:** routing based on posterior mean. Routes are updated in response to real-time observations but value of exploration is not considered.
- **Omniscient Routing:** routing according to perfect information of traffic conditions.

Routing Validation

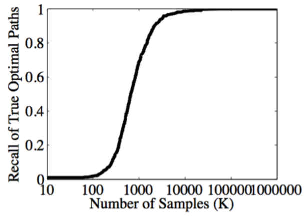


Liu, Yue & Krishnan, TKDE 2015

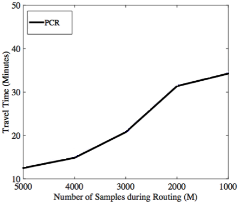
- PCR perform nearly as well as omniscient lower bound
- Both GP-UCB and PCR are effective at balancing the exploration/exploitation trade-off

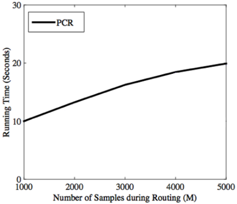
Performance Evaluation: Number of samples

- K should be large enough so all canonical paths are covered



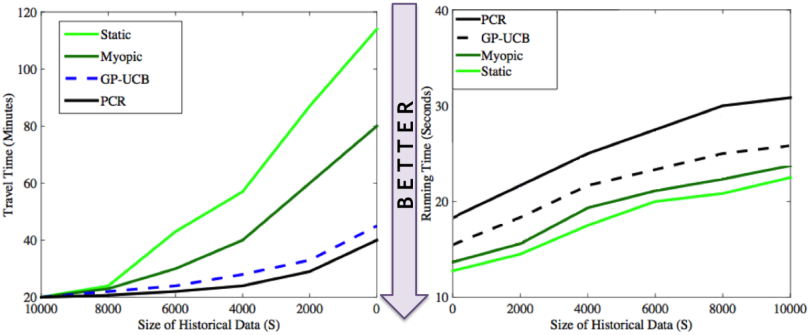
- PCR tolerates small degree of under sampling of traffic conditions





Liu, Yue & Krishnan, TKDE 2015

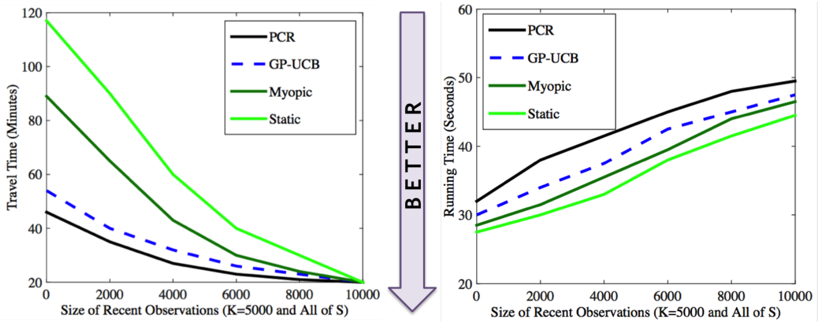
Performance Evaluation – Size Historical Data (S)



Liu, Yue & Krishnan, TKDE 2015

- GP-UCB and PCR relatively robust using smaller S
- GP-UCB 25% more efficient than PCR
- Approach is viable for real-time adaptive routing

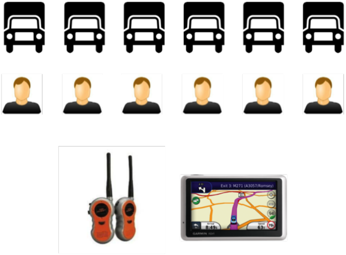
Performance Evaluation – Number Recent Obs. (Y)



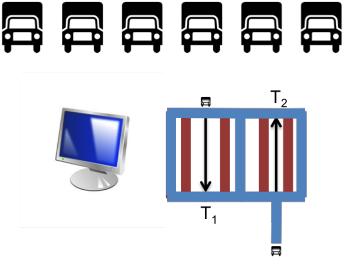
Liu, Yue & Krishnan, TKDE 2015

- Smaller Y leads to less reliable GPDCM posterior distributions (trade-off computational cost)
- Both GP-UCB and PCR tolerate significant reduction in Y

Preliminary Field Study



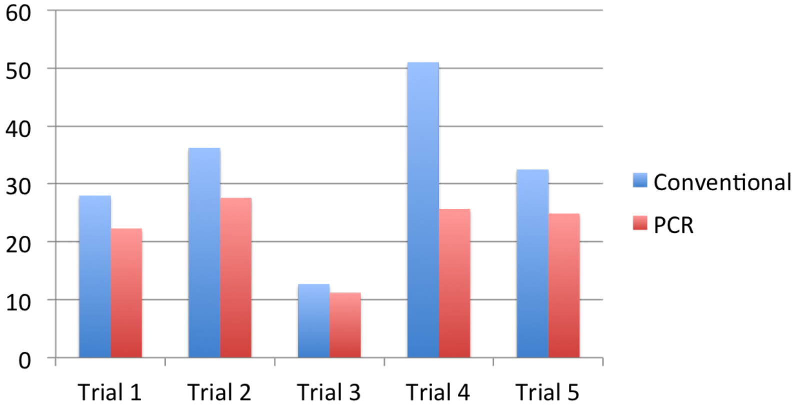
Conventional
"Myopic Routing"



Liu, Yue & Krishnan's approach
Planning using Canonical Routes

Liu, Yue & Krishnan, TKDE 2015

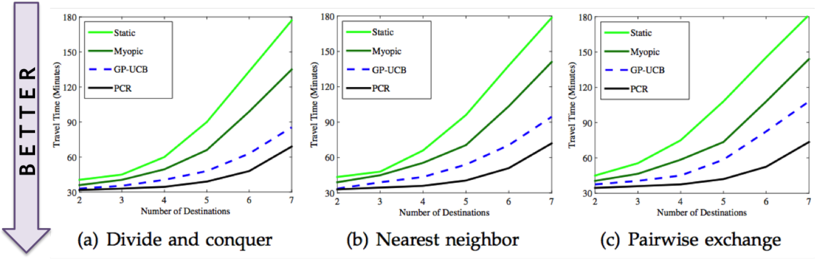
Preliminary Field Study



Liu, Yue & Krishnan, TKDE 2015

- 30% median reduction in travel time!

Routing to Multiple Destinations



Liu, Yue & Krishnan, TKDE 2015

- GP-UCB and PCR can be extended to multi-destination routing scenario (i.e Traveling Salesperson)
- GP-UCB and PCR perform better than static and myopic approach no matter the heuristic algorithm chosen

Any Questions?