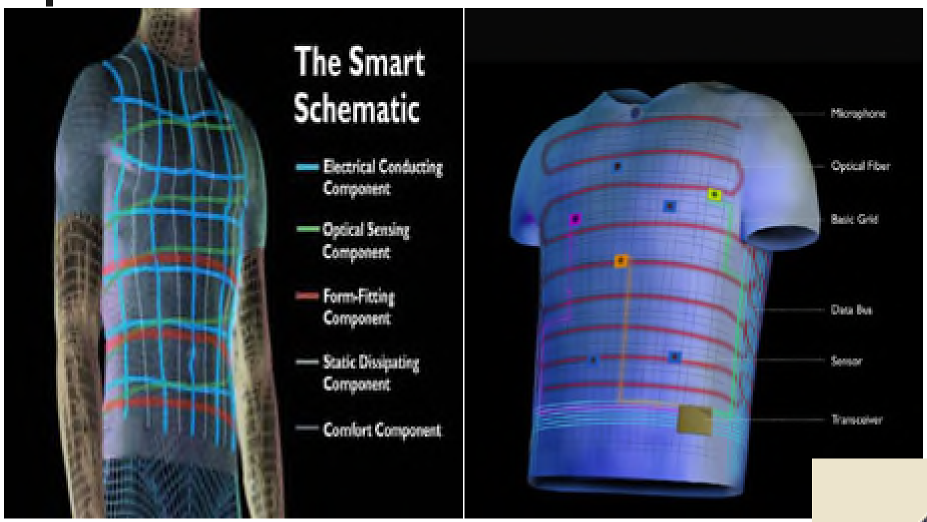
Designing a Crowd-sensing Service: Greener Transportation

Tarek Abdelzaher
University of Illinois at Urbana Champaign

The Rise of Crowd-Sensing

Force #1: More Sensors in Personal and Social Spaces

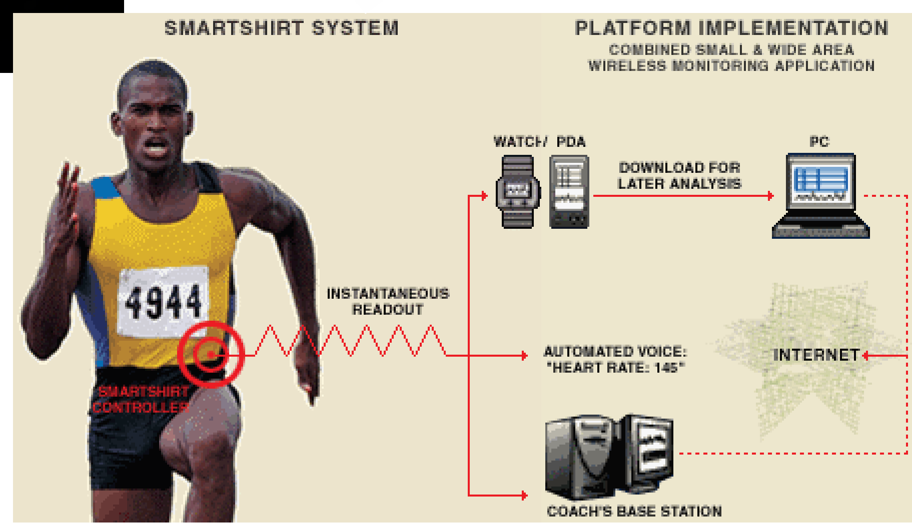
Early Examples (2005+)



http://www.sensatex.com

GPS

Nike -iPod

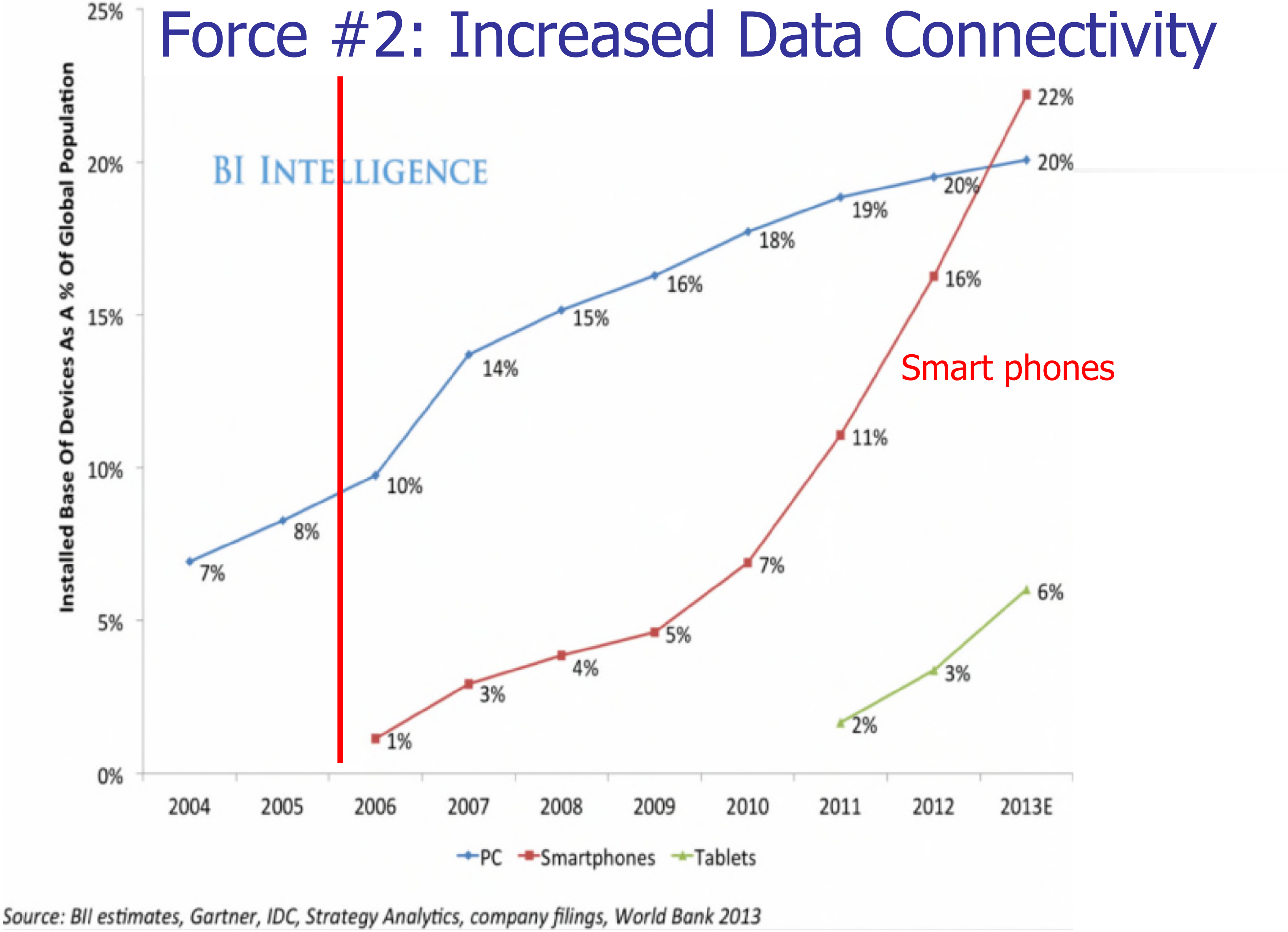


Spot

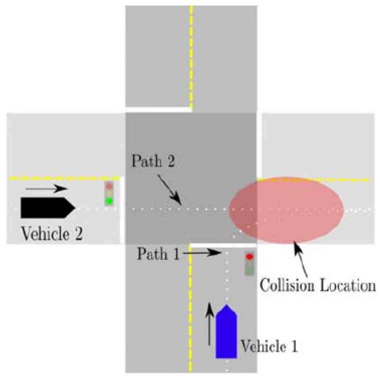
Wii

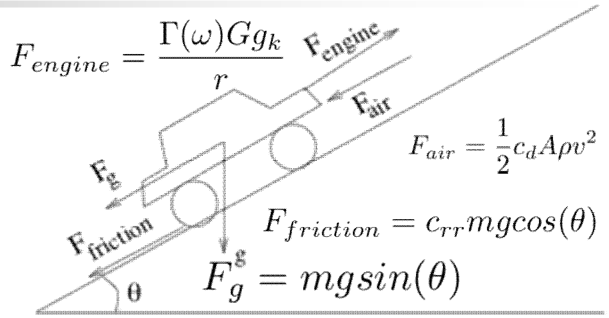
The Rise of Crowd-Sensing

Force #2: Increased Data Connectivity

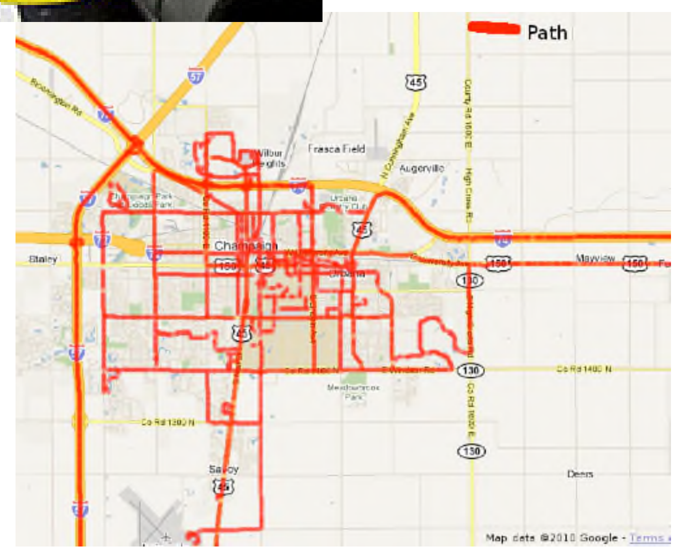


A Modern Crowdsensing Application: Sustainable Transportation





$$F_{car} = F_{engine} - F_{friction} - F_{air} - F_g$$



Transportation

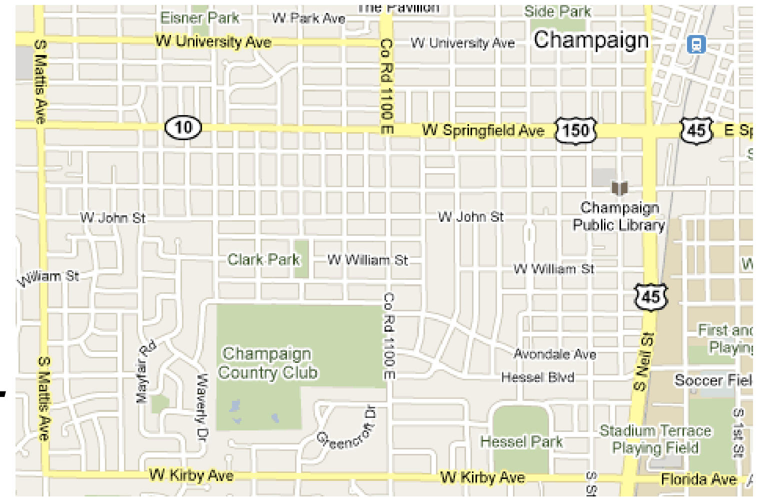
EPA Statistics (USA)

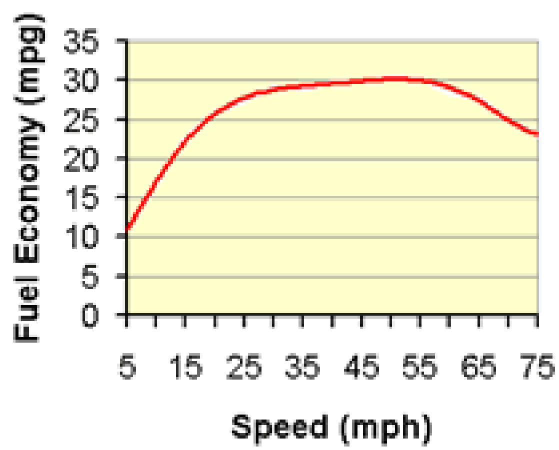
- 200 million light vehicles on the streets in the US
- Each is driven 12000 miles annually on average
- Average MPG is 20.3 miles/gallon
- **118 Billion Gallons of Fuel per year!**

- **Savings of 1% = One Billion Gallons**

GreenGPS: Fuel Efficient Vehicular Navigation

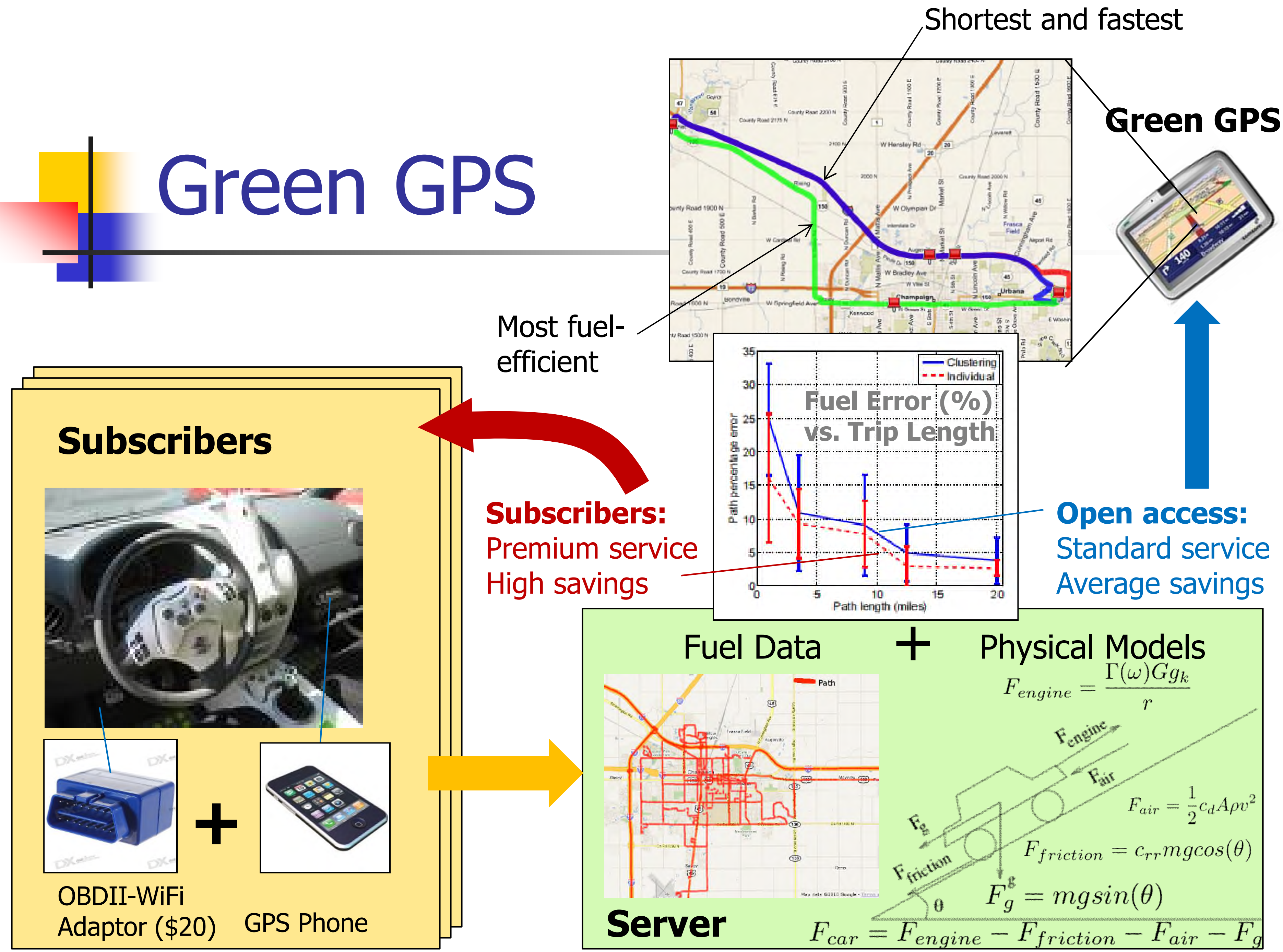
- Find the most fuel-efficient route (instead of a fastest or shortest)
- Fuel-efficient route is *different* from shortest or fastest route
  - Congestion → shortest may not be fuel efficient
  - MPG vs. speed is non-linear → fastest may not be fuel efficient





Source: US EPA

Green GPS

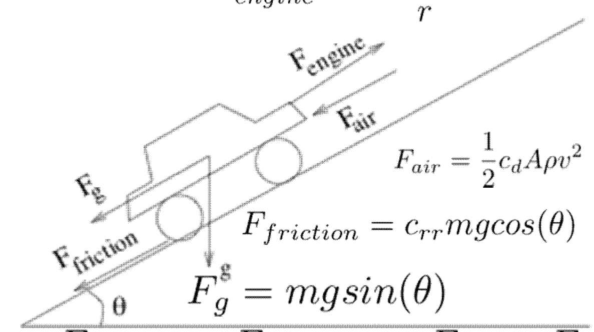

Shortest and fastest
Green GPS
Most fuel-efficient
Subscribers
Subscribers:
Premium service
High savings
Fuel Error (%) vs. Trip Length
Clustering
Individual
Path percentage error
Path length (miles)
Open access:
Standard service
Average savings
OBDII-WiFi Adaptor ($20)
+
GPS Phone
Fuel Data
+
Physical Models
Path
Server
$F_{engine} = \frac{\Gamma(\omega) G g_k}{r}$
$F_{air} = \frac{1}{2} c_d A \rho v^2$
$F_{friction} = c_{rr} m g cos(\theta)$
$F_g = m g sin(\theta)$
$F_{car} = F_{engine} - F_{friction} - F_{air} - F_g$

Fuel Consumption Model

- Simple model for fuel consumption derived from first principles
- The model is then is approximately recast in terms of easily measurable crowdsensed parameters (e.g., locations of stop signs, traffic lights, speed limits, and actual traffic conditions)

$$gpm = k_1 m \bar{v}^2 \frac{ST + \nu TL}{\Delta d} + k_2 m \frac{\bar{v}^2}{\Delta d} + k_3 m cos(\theta) + k_4 A \bar{v}^2 + k_5 m sin(\theta)$$

$$F_{engine} = \frac{\Gamma(\omega) G g_k}{r}$$



$$F_{car} = F_{engine} - F_{friction} - F_{air} - F_g$$

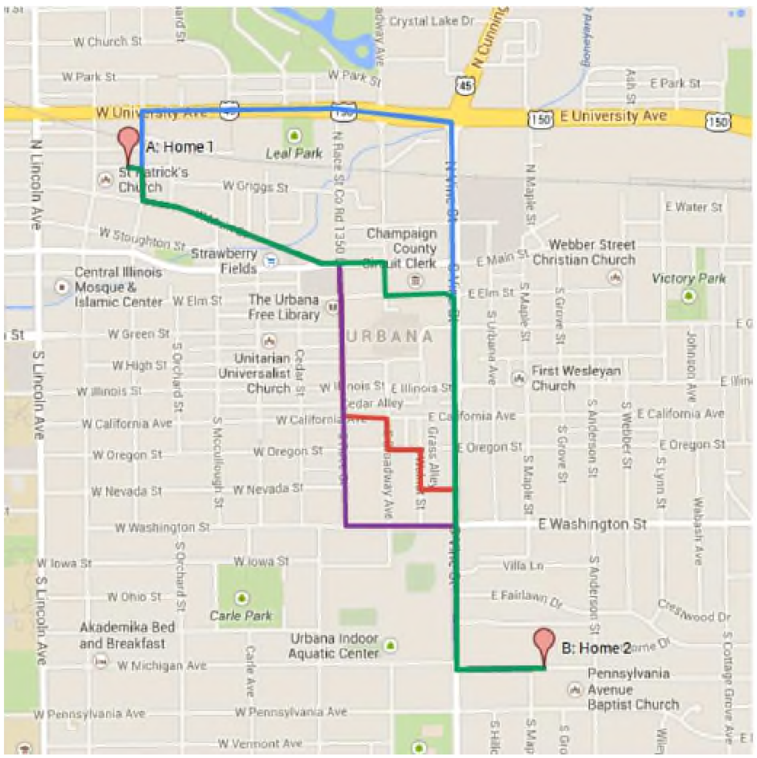
Fuel Consumption Examples

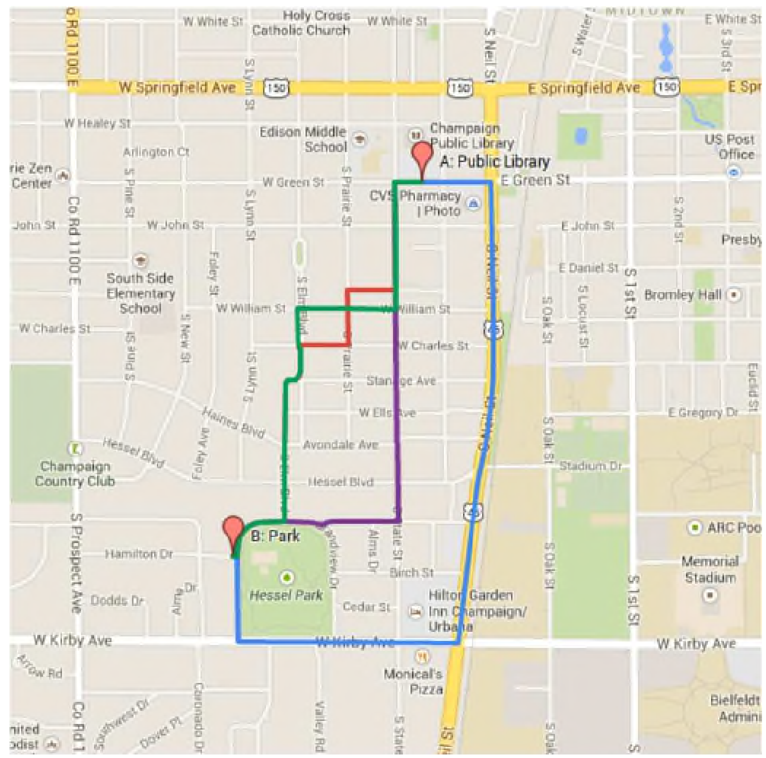
- Experiments on five cars, each does *four round-trips* between 2 landmarks in Urbana-Champaign on *fastest* and *shortest* routes, showing that neither wins consistently in being energy-optimal

Car	Route	Better Route	Difference
Honda Accord 2001	Home1 to Mall	Shortest	31.4%
	Home1 to Gym	Shortest	19.7%
Ford Taurus 2001	Home2 to Restaurant	Shortest	26%
Toyota Celica 2001	Home2 to Work	Fastest	10.1%
Nissan Sentra 2009	Home3 to Clinic	Fastest	8.4%
Honda Civic 2002	Home4 to Work	Fastest	18.7%

Finding Fuel-efficient Routes

- Example: Compare the GreenGPS route (green) to the fastest (blue), shortest (red), and Garmin EcoRoute (purple)





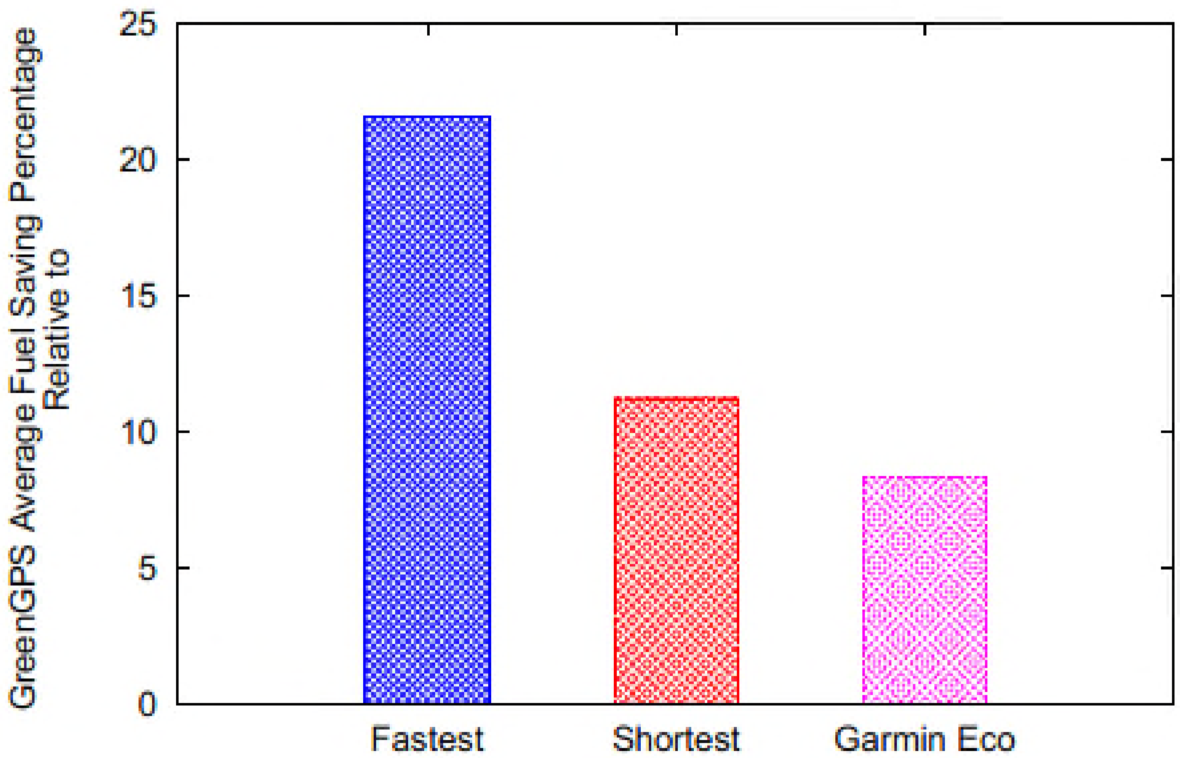
A Human Subjects Study

- 2000+ miles driven to evaluate GreenGPS

Car Make	Car Model	Car Year	Car Class	City MPG	Hwy MPG	Miles Driven	Individual Error %	General Error %	Cluster-based Error %
Toyota	Camry	2004	Mid-Size	24	33	80	1.55	8.44	1.72
Chevrolet	Impala	2002	Large	21	32	69	3.02	17.16	2.48
Ford	Ranger	2008	Van	15	19	29	0.89	25.26	5.26
Toyota	Corolla	2000	Compact	31	38	259	6.06	10.68	6.01
Buick	LeSabre	2002	Large	20	29	54	3.38	7.46	2.45
Ford	E-250	2011	Van	13	17	99	3.59	7.93	3.59
Toyota	Corolla	2010	Compact	26	35	53	4.31	18.47	9.32
Toyota	Celica	2001	Sub-Compact	28	34	497	4.94	11.69	4.94
Nissan	Altima	2006	Compact	24	31	95	3.83	7.04	3.83
Subaru	Impreza	2010	Sub-Compact	19	24	26	0.09	3.82	4.74
Toyota	Corolla	2004	Compact	32	40	141	3.67	13.59	3.67
Mazda	Mazda6	2003	Mid-Size	23	29	62	3.94	18.5	3.94
Audi	A4	2005	Compact	22	31	88	6.86	14.58	6.86
Toyota	Camry	2012	Mid-Size	25	35	90	4.96	7.59	4.96
Subaru	Impreza	2010	Sub-Compact	19	24	69	9.22	15.47	8.23
Hyundai	Santa-Fe	2001	Sport-Utility	21	28	87	3.3	17.92	3.3
Ford	Taurus	2002	Mid-Size	20	28	65	4.01	5.51	5.06
Mitsubishi	Eclipse	2002	Sub-Compact	23	30	184	5.32	15.91	5.32
Nissan	Altima	2010	Mid-Size	23	32	103	2.44	9.59	2.44
Mitsubishi	Galant	2002	Mid-Size	21	28	112	4.45	12.19	8.11
Toyota	Celica	2000	Compact	28	34	882	6.24	8.74	6.06
Toyota	Camry	2004	Mid-Size	24	33	57	0.73	13.76	2.21
Average Error Percentage (magnitude):							4.91	11.33	5.07

End Result: Fuel Savings

- The bottomline: percentage of fuel is saved over fastest, shortest, and GarminEco routes:



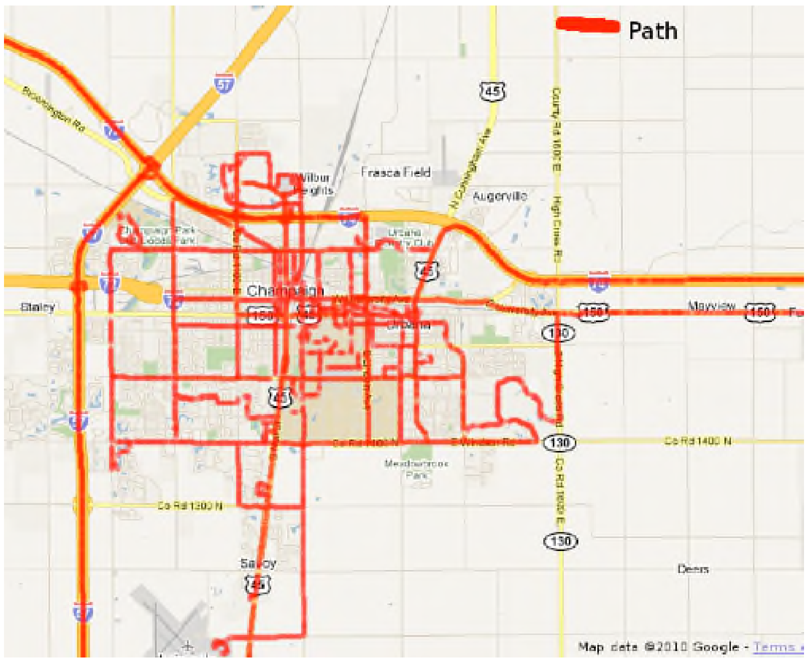
End Result: Fuel Savings

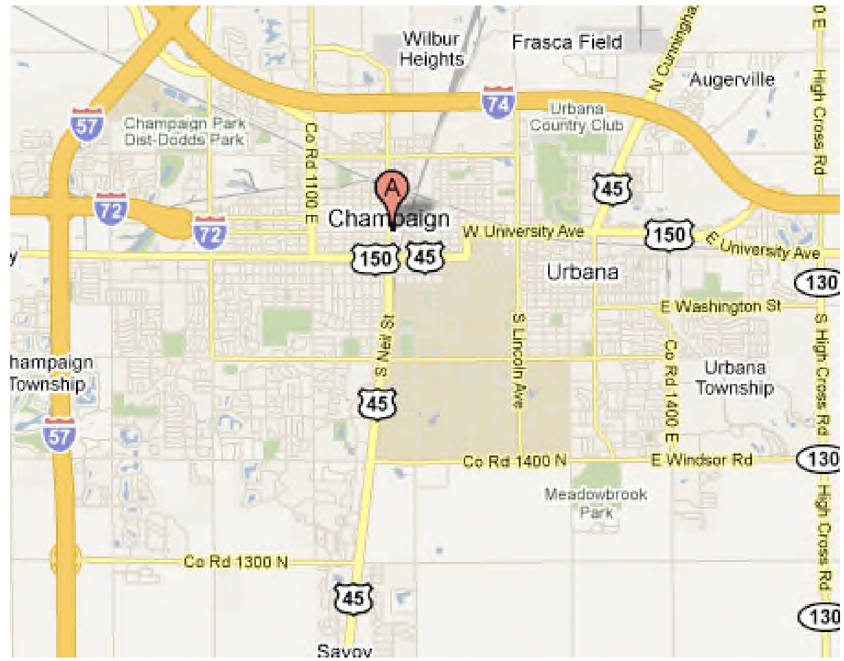
- The bottomline: percentage of fuel is saved over fastest, shortest, and GarminEco routes:

Crowdsensing challenge #1

Extrapolation from Sparse Data
(Conditions of Sparse Deployment)

Extrapolation from Sparse Data





Fuel consumption of
A few cars driven on a
few roads

Predict fuel consumption of
any car on any road

Generalization and Modeling

- Regression modeling:
  - Problem: one size does not fit all. Who says that Fords and Toyotas have the same regression model?
- Regression model per car?
  - Problem: Cannot use data collected by some cars to predict fuel consumption of others.
- Challenge: Must jointly determine both (i) regression models and (ii) their scope of applicability, to cover the whole data space within an acceptable modeling error.

Idea #1: Data Clustering (Using Data Cubes)

- Data cubes are clustering technique that group all crowd-sensed data according to several *alternative* dimensions (clustering policies) such as by car make, model, or year.
- A regression model is then derived for resulting clusters
- Different clustering policies are evaluated in terms of their fuel prediction error to determine the best policy
- When a navigation request from a new vehicle arrives:
  - The best clustering policy is used to add the vehicle to existing clusters
  - The regression model for this cluster is used to predict the vehicle's fuel consumption

The Regression Cube Model

- Data cells correspond to:
  - Output attributes $Y_c = \{y_i\}$
  - Each associated with k input attributes $x_{i1}, \ldots , x_{ik}$, $X_c=\{x_{ij}\}$

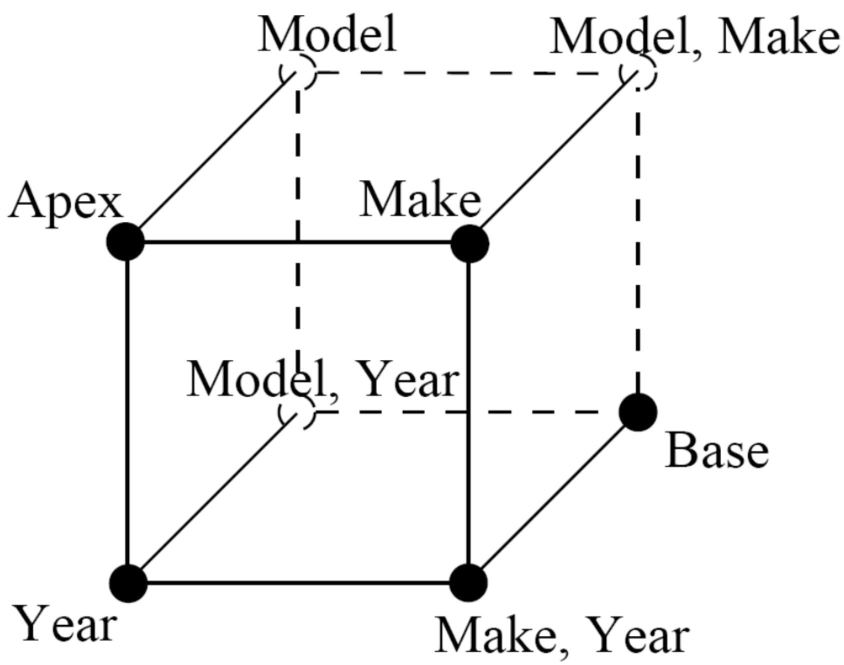
- Data cells store the following measures:
  - Regression model coefficients:

$$\hat{Y}_c = X_c \hat{\eta}_c$$

  - Regression modeling error:

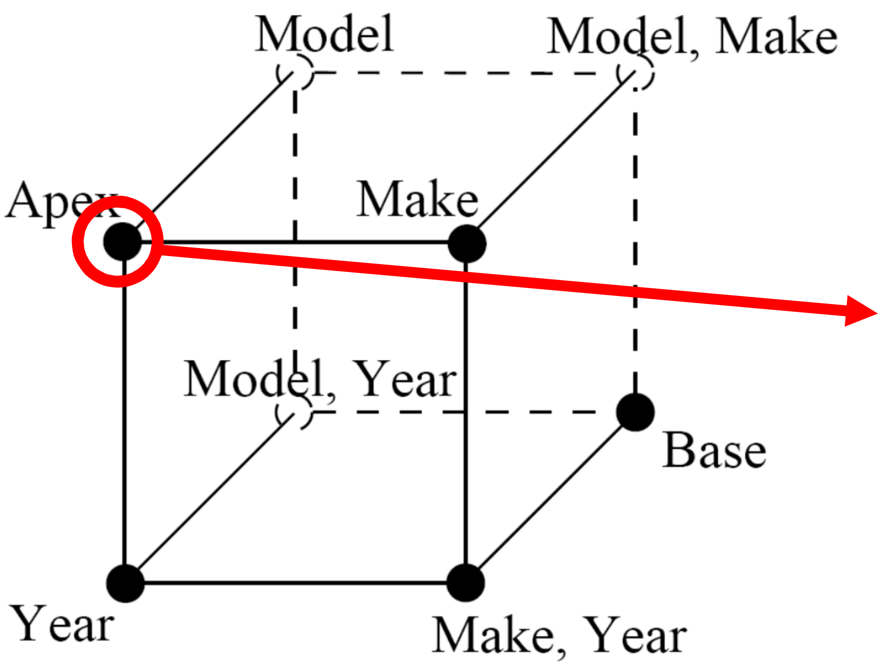
$$Err_c = (Y_c - X_c \hat{\eta}_c)^T (Y_c - X_c \hat{\eta}_c)$$

Example of Regression Cubes



- Goal: predict fuel consumption
  - Group by make, model, or year

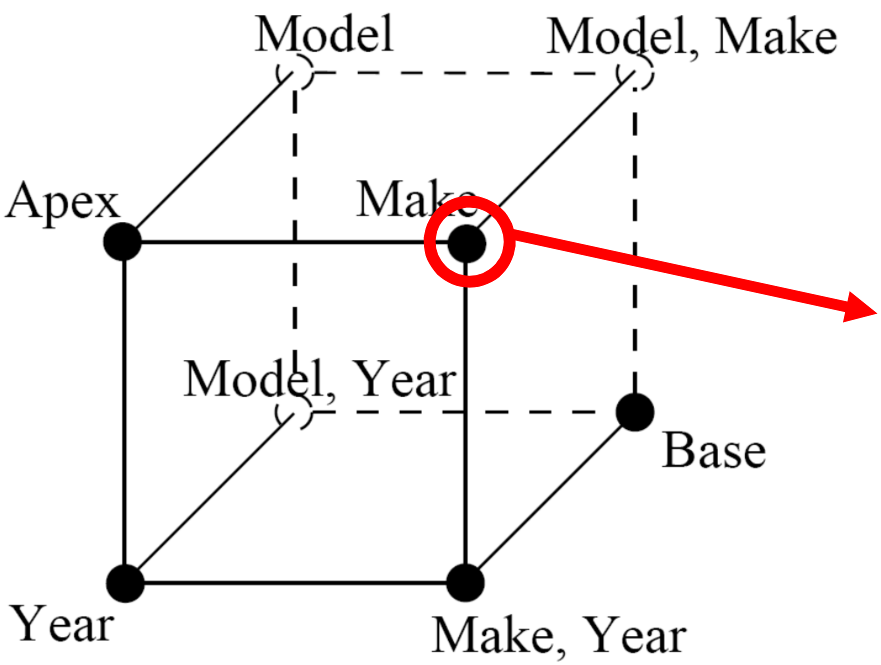
Example of Regression Cubes



Data Cells:

(*,*,*) – X, Y

Example of Regression Cubes



Data Cells:

(Toyota,*,*) – X_{c1} Y_{c1}
(Ford,*,*) – X_{c2} Y_{c2}
(Honda,*,*) – X_{c2} Y_{c3}

Data Cell Measures

- Main challenge: compute data cell measures recursively and without reprocessing raw data
- Measures can be classified as:
  - Distributive – $f(x_1, x_2, x_3) = f(f(x_1, x_2), x_3)$ - Efficient
    - Examples: sum, count
  - Algebraic/Compressible – An algebraic combination of distributive functions - Efficient
    - Example: average = sum/count
  - Holistic – Reprocess raw data - Inefficient
    - Example: median

The Challenge in Regression Cubes

- Main problem: Model parameters and estimation error are not distributive

$$\hat{Y}_c = X_c \hat{\eta}_c$$

$$Err_c = (Y_c - X_c \hat{\eta}_c)^T (Y_c - X_c \hat{\eta}_c)$$

An Efficient Representation

- Compressed representation of a cell c:
  - $\rho_c = Y_c^T Y_c$: scalar value
  - $\Theta_c = X_c^T X_c$: vector of size k
  - $\nu_c = X_c^T Y_c$: k by k matrix
  - n_c : number of samples

$$\rho_c = \sum_{i=1}^{m} \rho_i \qquad \nu_c = \sum_{i=1}^{m} \nu_i \qquad \Theta_c = \sum_{i=1}^{m} \Theta_i \qquad n_c = \sum_{i=1}^{m} n_{c_i}$$

- These matrices are distributive measures

An Efficient Data Cube for Fuel Consumption Regression Models

- Model coefficients:

$$\hat{\eta}_c = (X_c^T X_c)^{-1} X_c^T Y_c = \Theta_c^{-1} \nu_c$$

- Error:

$$\begin{aligned} Err_c &= (Y_c - X_c\hat{\eta}_c)^T (Y_c - X_c\hat{\eta}_c) = \\ &Y_c^T Y_c - (X_c\hat{\eta}_c)^T Y_c - Y_c^T X_c\hat{\eta}_c + (X_c\hat{\eta}_c)^T X_c\hat{\eta}_c = \\ &\rho_c - \hat{\eta}_c^T \nu_c - \nu_c^T \hat{\eta}_c + \hat{\eta}_c^T \Theta_c \hat{\eta}_c \end{aligned}$$

- Model coefficients and regression error are compressible measures

Idea #2: Model Reduction

- Independently find *the set of model parameters, L*, for each cell, such that:
  - The cell is reliable
  - Corresponding error is minimized
  - Challenge: Exponential number of *L*s

Attributes
Velocity (*v*)
Mass (*m*)
Frontal area (*A*)
Stop signs (*S*)

	Error	Reliable
L = {v}	0.031	yes
L = {m}	0.152	yes
L = {A}	0.043	yes
L = {S}	0.056	yes

Computing data Cell Confidence

- Measure of confidence:
  - Probability at which the actual coefficients are far from the estimate

$$Pr[||\hat{\eta_c} - \eta_c|| > \delta]$$

$$Pr[||\hat{\eta_c} - \eta_c|| > \delta] \leq \frac{k\sigma^2}{\delta^2 \lambda_{min}(X_c^T X_c)}$$

$$\hat{\sigma}^2 = \frac{Err_c}{n_c}$$

- Reliable Cell:

$$\frac{k\hat{\sigma}^2}{\delta^2 \lambda_{min}(\Theta_c)} < 0.05$$

Idea #2: Model Reduction

- Independently find *the set of model parameters, L,* for each cell, such that:
  - The cell is reliable
  - Corresponding error is minimized
  - Challenge: Exponential number of *L*s

	Error	Reliable
L = {v}	0.031	yes
L = {m}	0.152	yes
L = {A}	0.043	yes
L = {S}	0.056	yes

Attributes
Velocity (*v*)
Mass (*m*)
Frontal area (*A*)
Stop signs (*S*)

Idea #2: Model Reduction

- Independently find *the set of model parameters, L,* for each cell, such that:
  - The cell is reliable
  - Corresponding error is minimized
  - Challenge: Exponential number of *L*s

Attributes
Velocity (*v*)
Mass (*m*)
Frontal area (*A*)
Stop signs (*S*)

L = {v}
L = {m}
L = {A}
L = {S}

→

	Error	Reliable
L = {v, m}	0.021	no
L = {v, A}	0.030	yes
L = {v, S}	0.028	yes

Idea #2: Model Reduction

- Independently find *the set of model parameters, L*, for each cell, such that:
  - The cell is reliable
  - Corresponding error is minimized
  - Challenge: Exponential number of *L*s

Attributes
Velocity (*v*)
Mass (*m*)
Frontal area (*A*)
Stop signs (*S*)

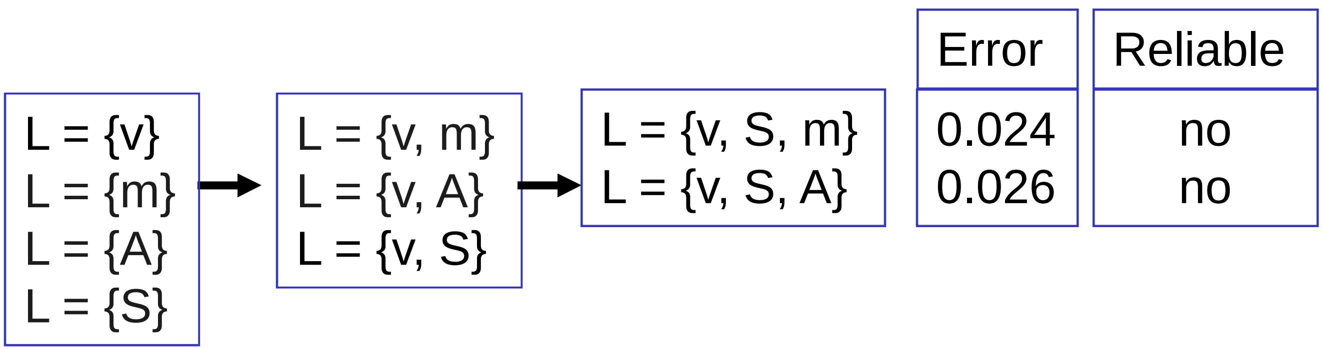
L = {v}
L = {m}
L = {A}
L = {S}

→

	Error	Reliable
L = {v, m}	0.021	no
L = {v, A}	0.030	yes
L = {v, S}	0.028	yes

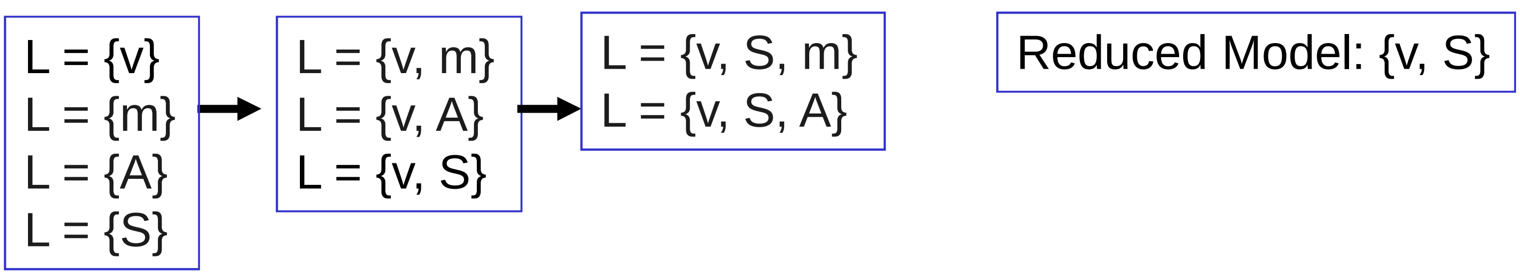
Idea #2: Model Reduction

- Independently find *the set of model parameters, L,* for each cell, such that:
  - The cell is reliable
  - Corresponding error is minimized
  - Challenge: Exponential number of *L*s



Idea #2: Model Reduction

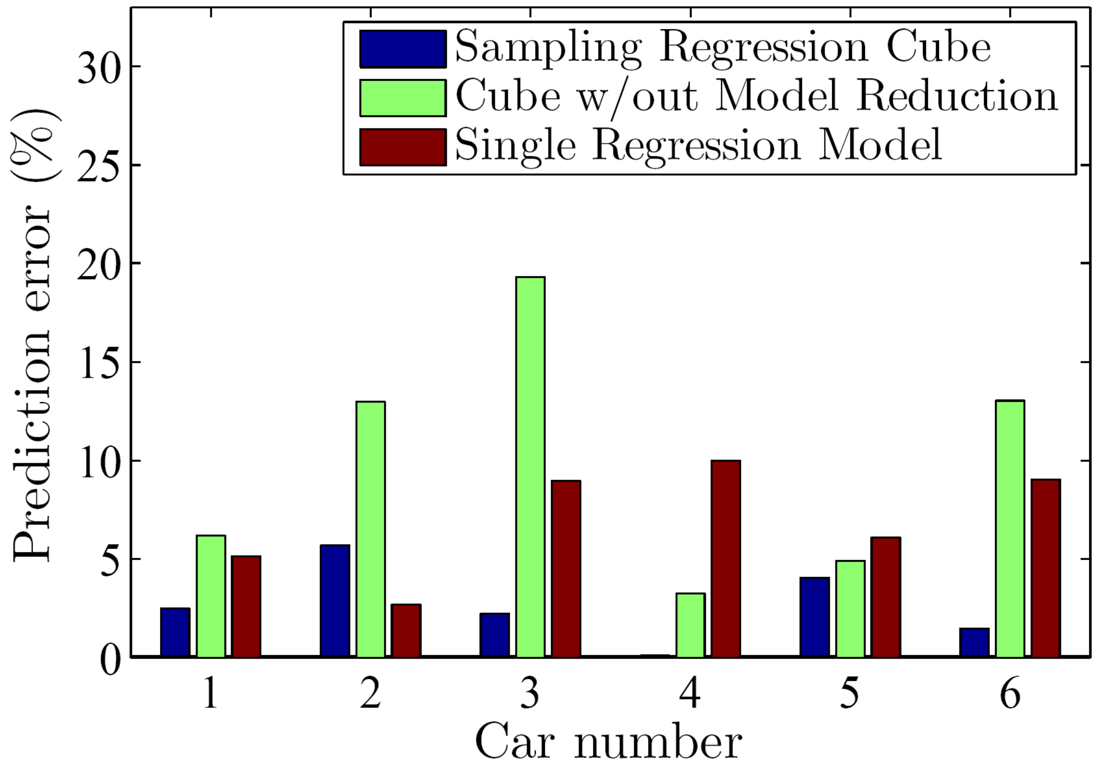
- Independently find *the set of model parameters, L,* for each cell, such that:
  - The cell is reliable
  - Corresponding error is minimized
  - Challenge: Exponential number of *L*s



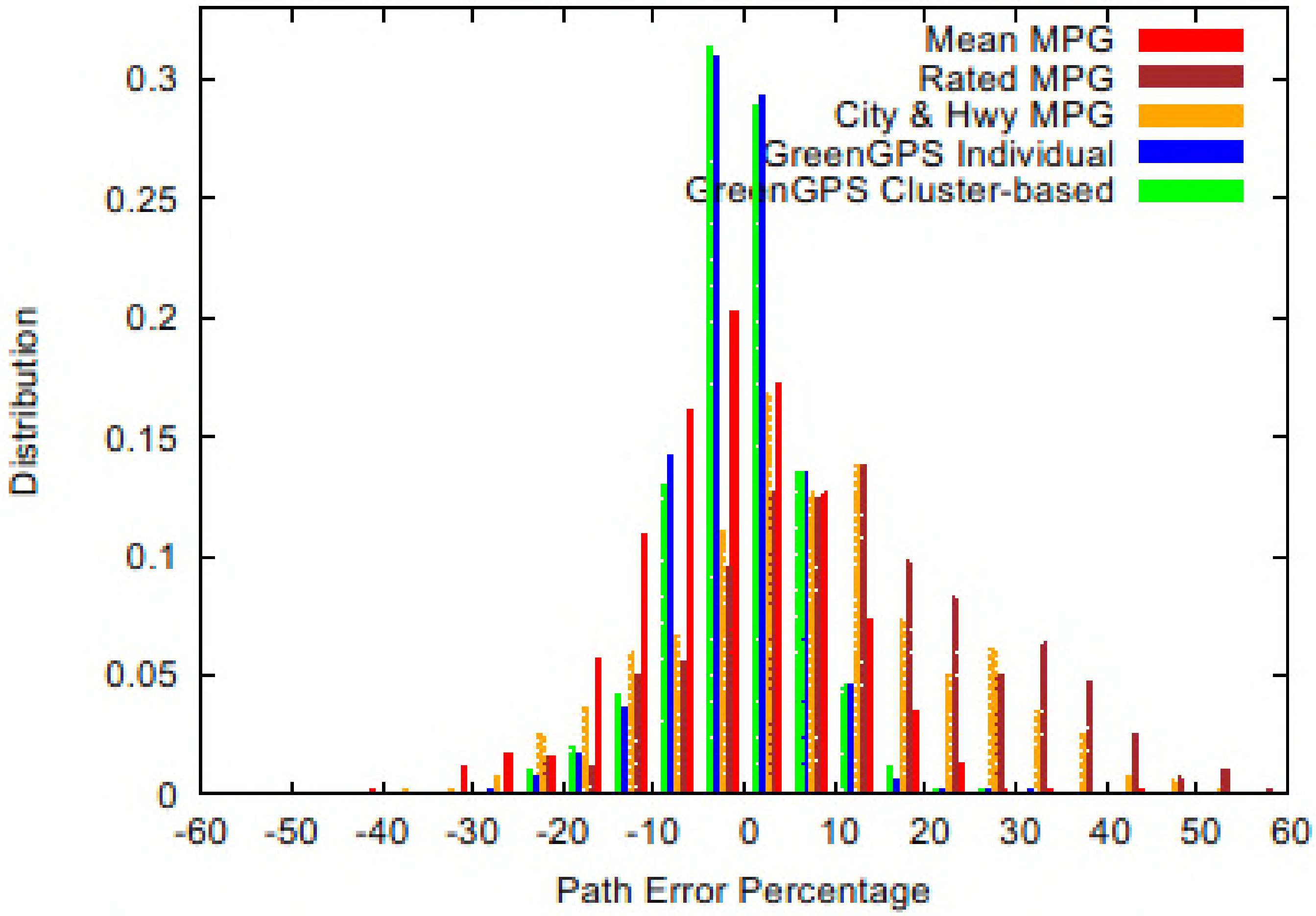
Accuracy Results

- The sampling regression cube improves prediction accuracy significantly

- A regression cube without model reduction is even worse than a single model!



Error Distribution in Fuel Prediction



Crowdsensing Challenge #2

Privacy

Possible Solution: Privacy via Anonymity

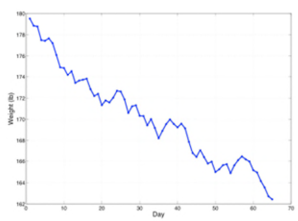
- Share data (e.g., GPS trajectory), but not user ID
- Problems?

Alternative Idea: Data Perturbation

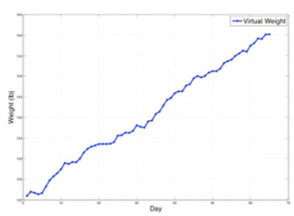
- Develop perturbation that preserves privacy of individuals
  - Cannot infer individuals’ data without large error
  - Reconstruction of community distribution can be achieved within proven accuracy bounds
  - Perturbation can be applied by non-expert users

Intuitive Approach

Real user

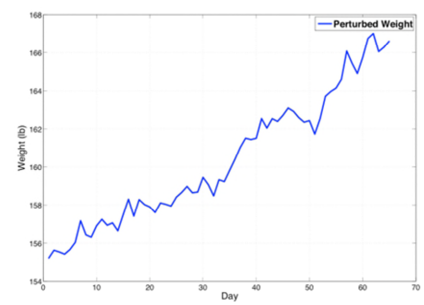


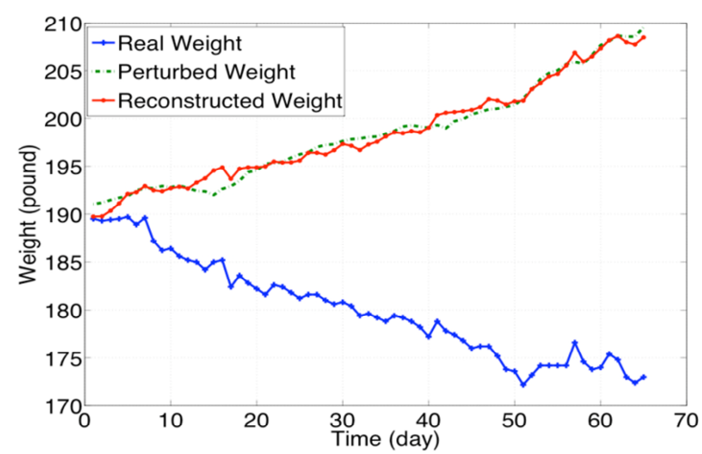
Virtual user



Add virtual user curve to real curve

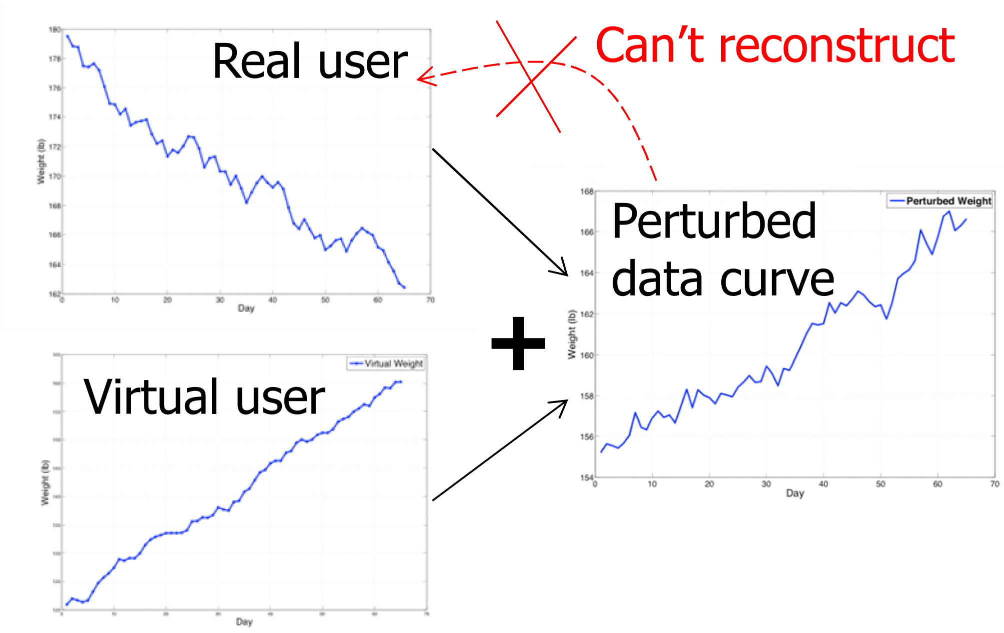
Perturbed data curve





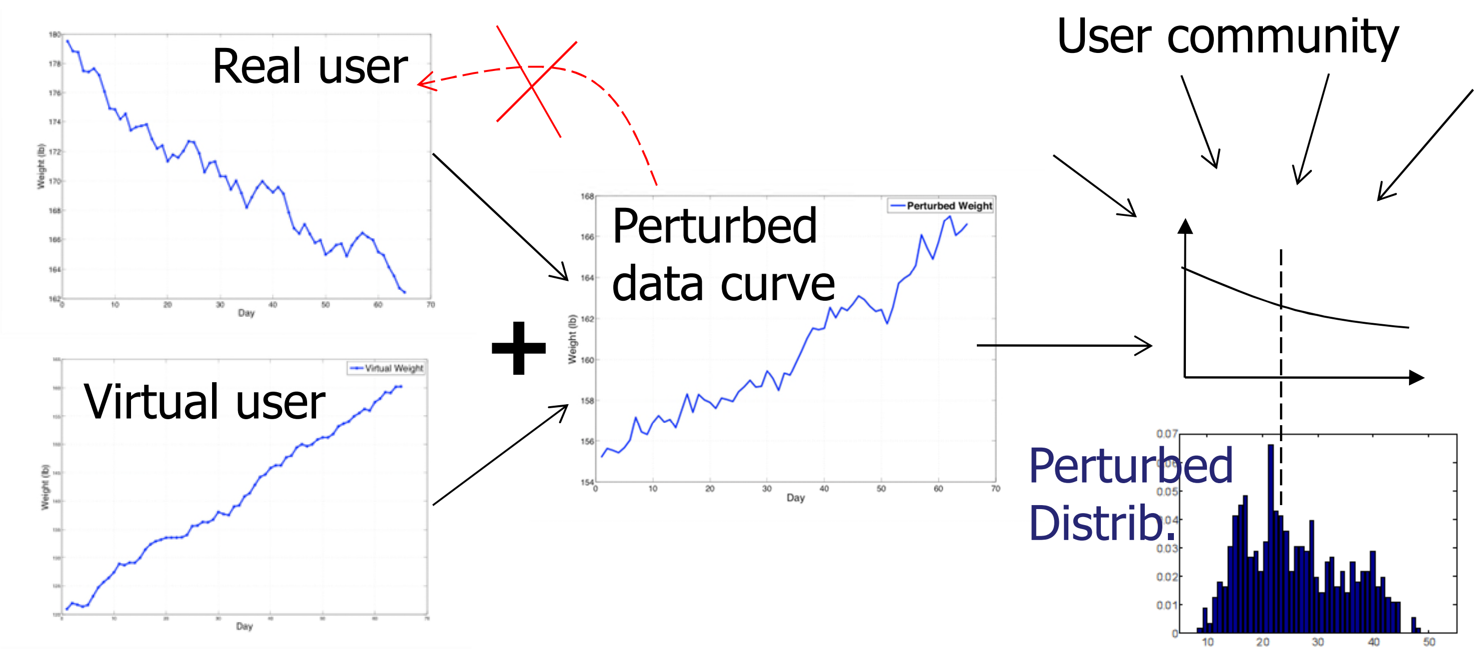
Intuitive Approach

- Client adds noise time-series with co-variance that largely mimics covariance of actual data (overlap in frequency domain)



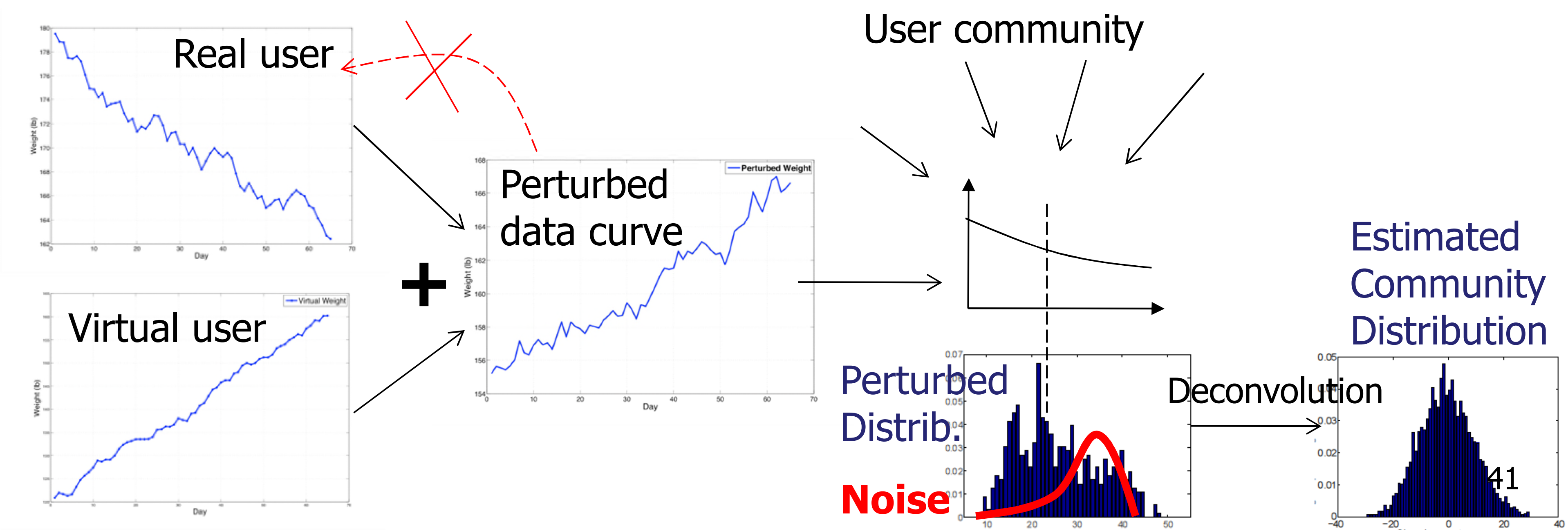
Intuitive Approach

- Client adds noise time-series with co-variance that largely mimics covariance of actual data (overlap in frequency domain)
- Users send their perturbed data to aggregation server

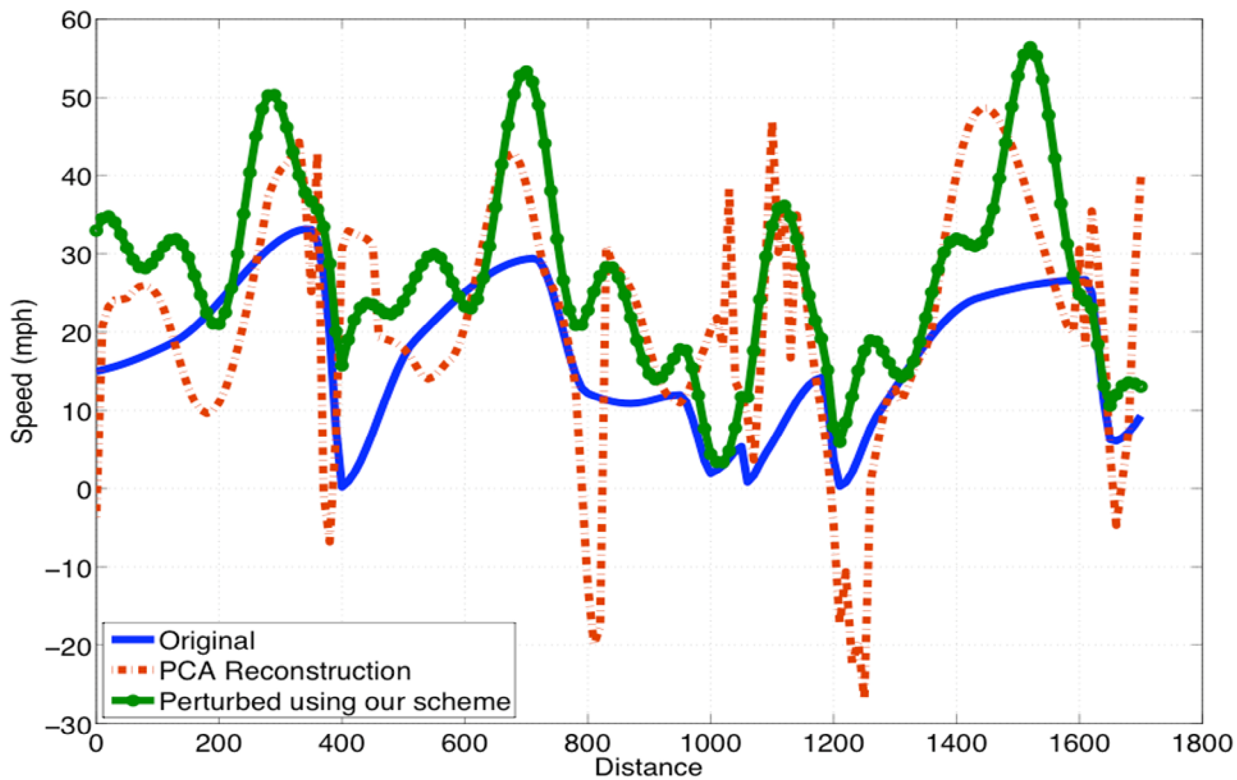


Intuitive Approach

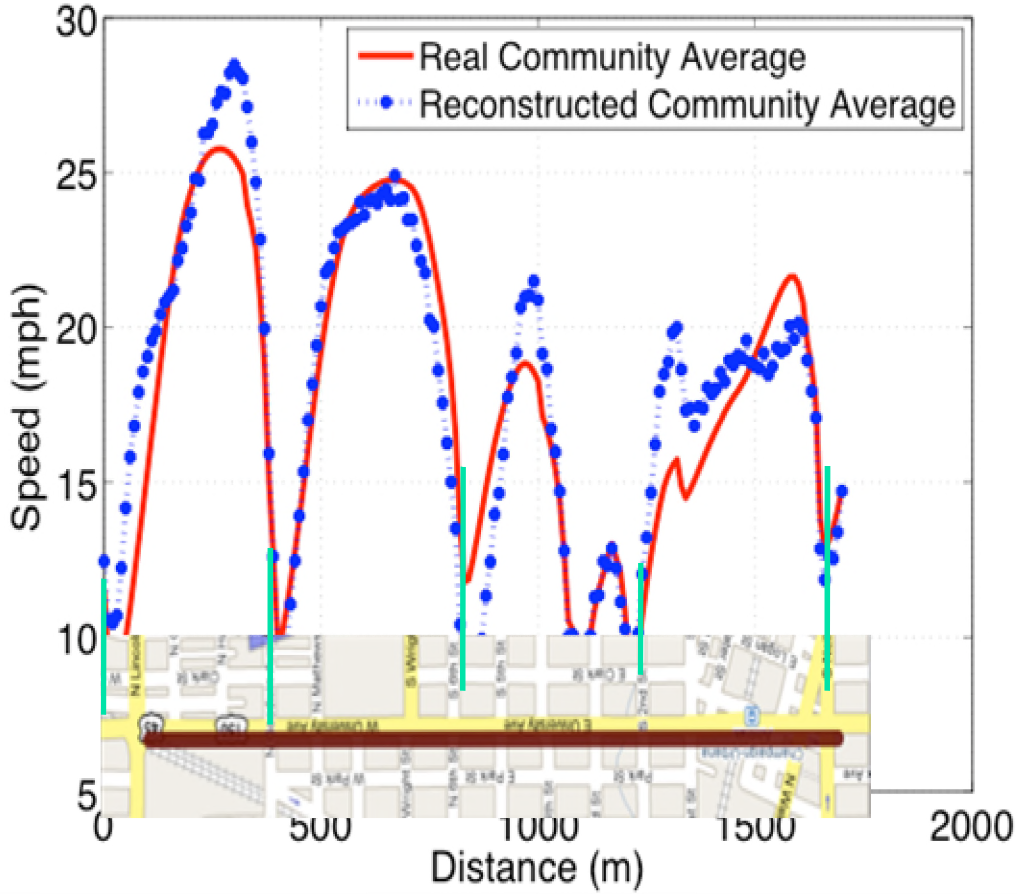
- Client adds noise time-series with co-variance that largely mimics covariance of actual data (overlap in frequency domain)
- Users send their perturbed data to aggregation server
- Given perturbed community distribution and noise, server uses de-convolution to reconstruct original data distribution at any point in time



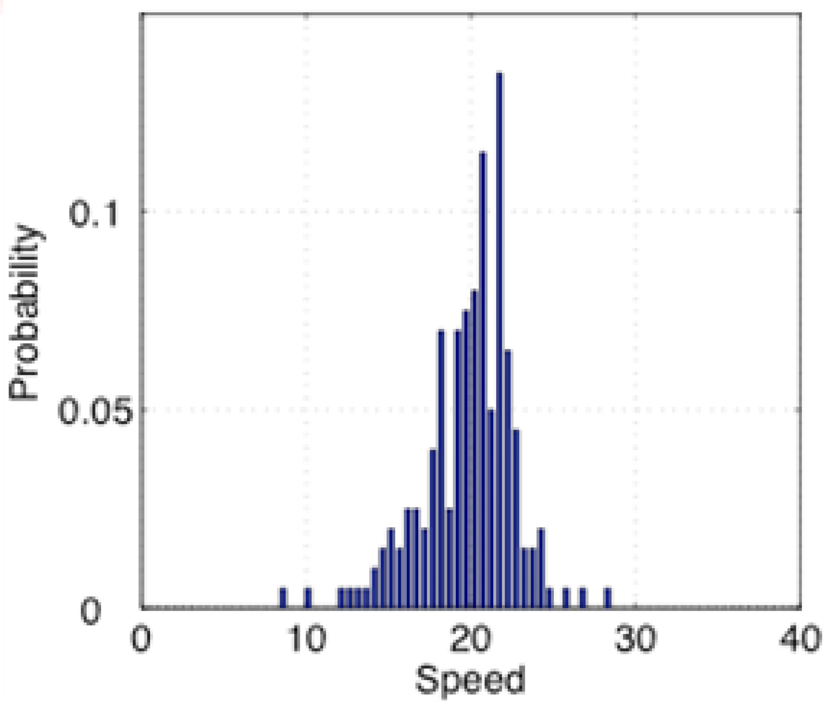
Perturbing Speed of Traffic



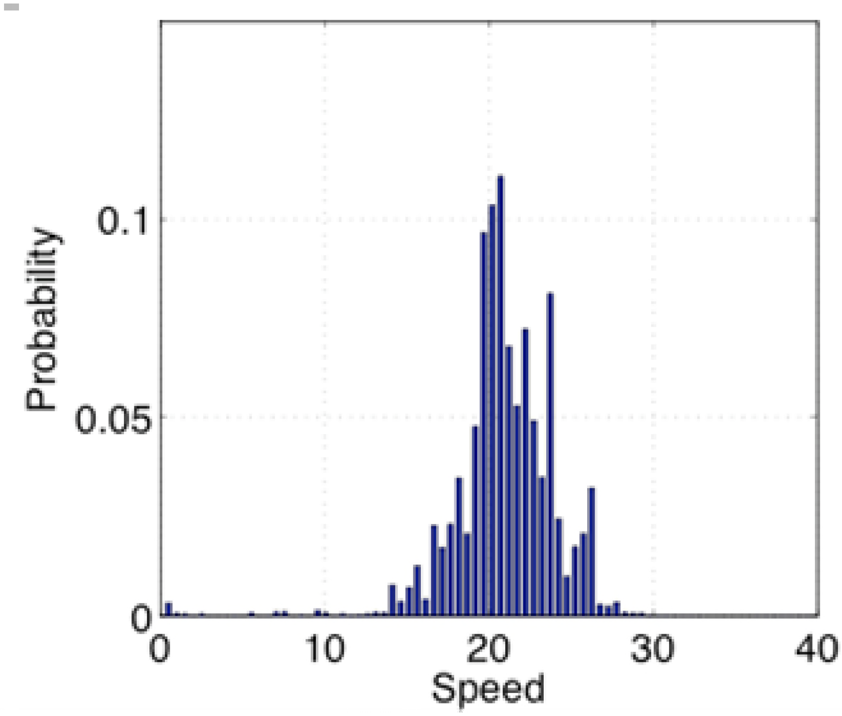
Reconstruction of Average Speed



Reconstruction of Community Speed Distribution



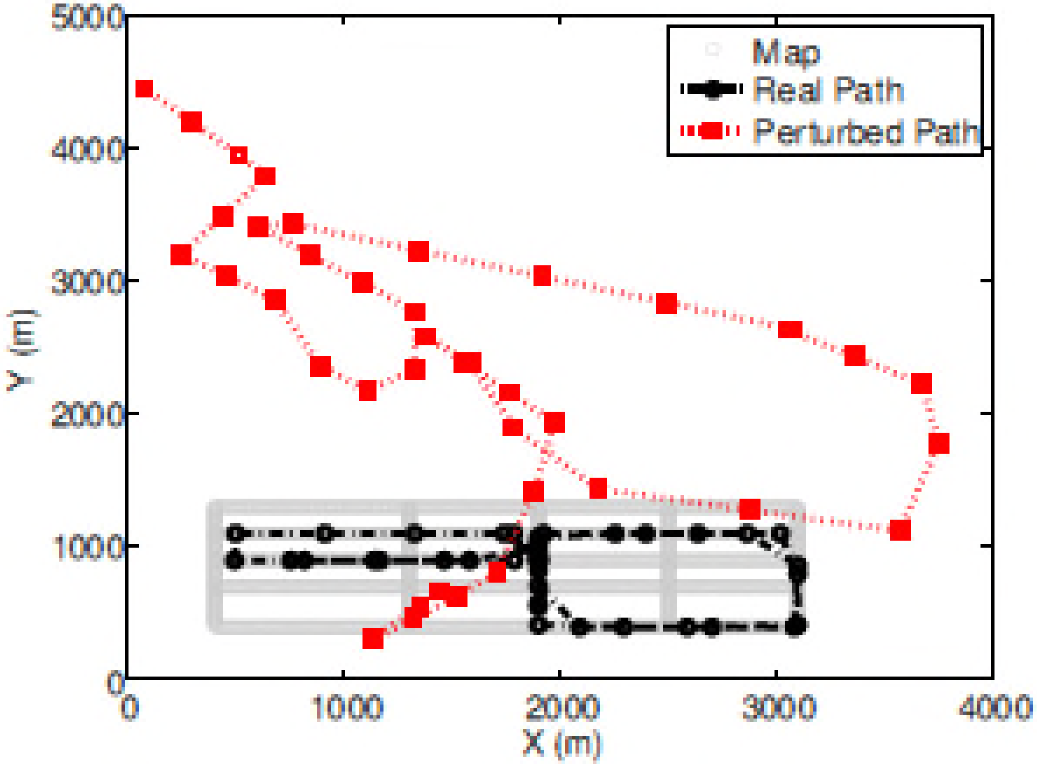
Real community distribution of speed



Reconstructed community distribution of speed

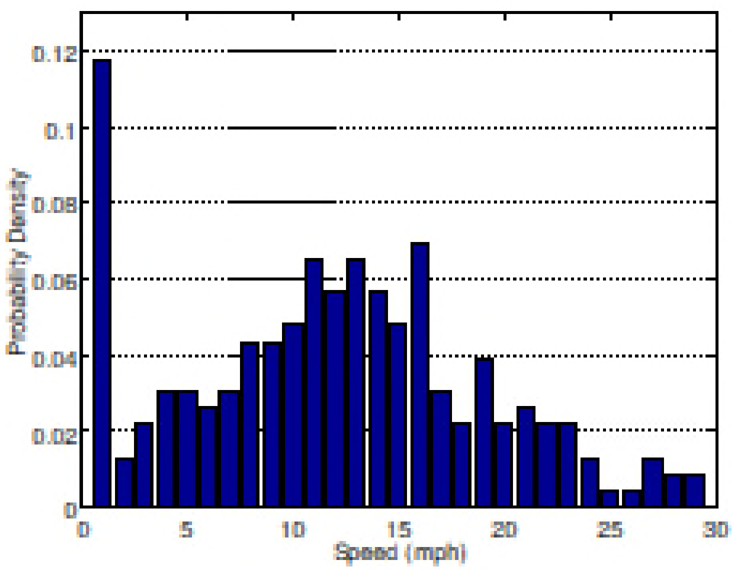
Perturbing Speed and Location

- Clients lie about both location and speed

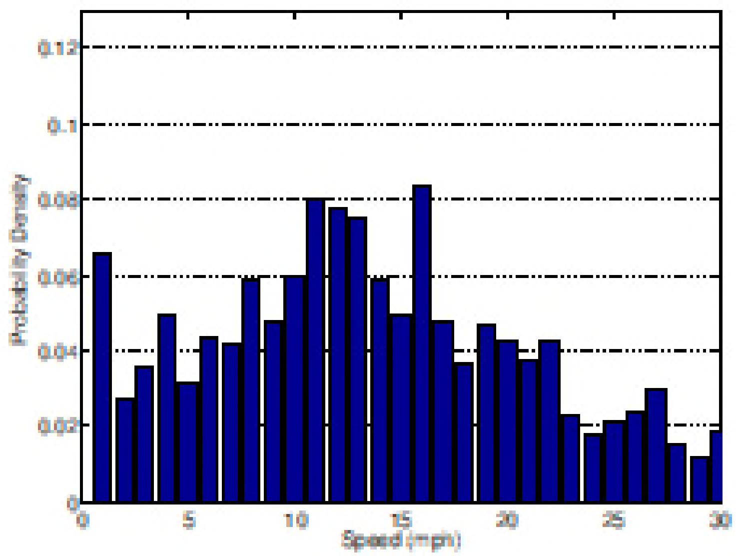


Reconstruction Accuracy

- Real versus reconstructed speed



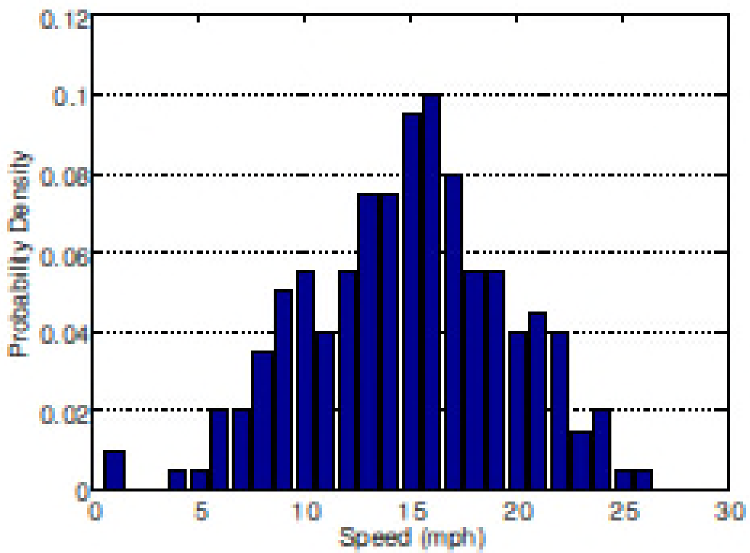
Real community distribution of speed



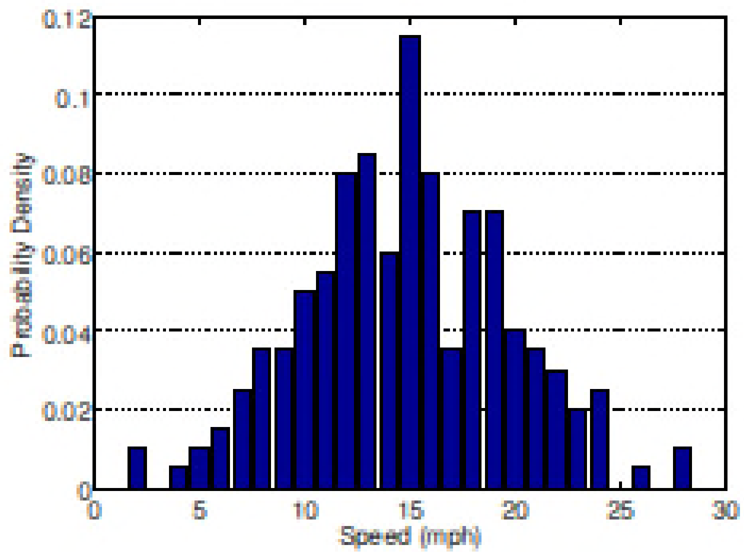
Reconstructed community distribution of speed

More on Reconstruction Accuracy

- Real versus reconstructed speed on Washington St., Champaign



Real community distribution of speed



Reconstructed community distribution of speed

How Many are Speeding?

- Real versus estimated percentage of speeding vehicles on different streets (from data of users who “lie” about both speed and location)

Street	Real % Speeding	Estimated % Speeding
University Ave	15.6%	17.8%
Neil Street	21.4%	23.7%
Washington Street	0.5%	0.15%
Elm Street	6.9%	8.6%

Conclusions

- A green navigation service was described that determines the most fuel-efficient routes for drivers using crowd-sourced information
- Several challenges were involved
  - Extrapolation from scarce data
  - Privacy
  - Handling unreliable sources
- Results
  - The service saves up to 20% of gas compared to typical navigation options.
  - Works well in conditions of sparse deployment
  - Leverages unreliable sources to construct accurate traffic maps
- Limitations:
  - Evaluated in a small town with little or no congestion
  - Benefits are potentially larger in bigger cities with more extreme traffic conditions: large-city evaluation left as future work