Traffic Estimation and Optimization

Charles Wang, Bobby Lee, David Howard
EE 127A Optimization Model
University of California, Berkeley
charleswang007@gmail.com, bobbylee25@gmail.com, davidsonofjohn@berkeley.edu

***Abstract*— This project leverages probe vehicle data collected from the Mobile Millennium project to perform traffic estimation while analyzing spatial traffic patterns. We use what we have learned throughout the class EE 127A about optimization and experiment with real data and important applications.**

Keywords-Optimization, Least-Squares, Linear Programming

I. INTRODUCTION

Our main goal is to understand and develop an optimal road network with efficient movement of traffic and minimal traffic congestion problems. If time allows, we will study traffic congestion, which has a significant impact on economic activity throughout the world. By creating an accurate and reliable traffic monitoring and control system, we will have better understanding of interactions between vehicles, drivers, and infrastructure and hence improve our traffic networks.

Our final project will leverage probe vehicle data collected from the Mobile Millennium project to perform traffic estimation while analyzing spatial traffic patterns. Mobile Millennium is a joint effort of CCIT (California center for innovative transportation), Caltrans, Nokia, and UC Berkeley's Department of Civil and Environmental Engineering in the area of traffic monitoring. The goal of Mobile Millennium was to test traffic data collection from GPS-equipped cell phones driving on a stretch of a highway located in The San Francisco Bay Area. One hundred vehicles carrying the GPS-enabled Nokia N95 drove along a 10-mile stretch of I-880 from 9:30am to 6:30pm.

In addition to this data, we will construct a network model of the measured traffic flows. This will allow us to refine our approximations by observing such traffic flow invariants as preservation of vehicles. As we approach our TRAFFIC project, we will be using what we have learned throughout EE 127A about least-square estimation, convex optimization and linear programming, and experimenting with real data and applications downloaded from Mobile Century.

II. PROBLEM SOLVING

Probe vehicles reported their location periodically (on average once per minute) along with an identifier. From these location measurements, we use an algorithm to map the locations on the road network, filter the noise of the GPS and reconstruct the path of the vehicle between successive locations. We will estimate traffic conditions on a network with *m* links (road segments). The information on the path of a probe vehicle between successive measurements is stored with the following format: $a_i \in R^m$ describes the path of the vehicle, $y_i \in R$ represents the travel time of the vehicle (around 60 seconds) and $t_i \in R$ represents the time at which the vehicle sent its location. The j^{th} entry in the vector a_i represents the proportion of link *j* that was traveled by the vehicle: 0 if the vehicle did not drive on link *j*, 1 if the vehicle fully traversed link *j*, a value in (0, 1) if the vehicle traveled on a fraction of link *j*. We denote by *A* the matrix whose i^{th} row is ${a_i}^T$. The *n* rows of the matrix correspond to different path travel times of vehicles on the network.

A. Average Travel Time Estimation

The goal of this project is to use least square estimations and appropriate regularization to provide traffic estimates and study spatial traffic patterns in San Francisco. Let $x \in R^n$ represent the average travel time on each link of the network. It is the variable that we try to estimate. For a vehicle, with a path represented by a_i and the travel time y_i, we have $y_i = {a_i}^T x + v_i$ where v_i represents some noise in the measurements.

Followings are some of our observations:

- The ordinary least-squares (OLS) problem is a particularly simple optimization problem that involves the minimization of the Euclidean norm of a ‘‘residual error" vector that is affine in the decision variables. Given n measurements of path travel times, we would like to estimate the average travel time on each link of the network using least-square estimation:

$$\underset{x \in \mathbb{R}^m}{\text{minimize}} \quad ||Ax - y||_2^2 \qquad [1]$$

where $A \in \mathbb{R}^{m \times n}$, $y \in \mathbb{R}^m$ are given. The combination of *A*, *y*: (*A*,*y*) is often referred to as the problem data. The solution of the above optimization problem is given by

$$x^* = (A^T A)^{-1} A^T y \qquad [2]$$

if the matrix A is full column rank.

- We impose a constraint $x > 0$ to have meaningful results, since every entry in x represents a travel time on a link of the network.
- In general, the matrix *A* may not be full column rank, therefore the closed-form solution cannot be applied. We remedy equation [1] by applying the so-called regularized least squares. We solve the following problem instead:

$$\underset{x\in\mathbb{R}^m}{\text{minimize}} \quad ||Ax-y||_2^2+\lambda||x-\hat{x}||_2^2 \qquad [3]$$

For reasonable values of lambda, this formulation generates an approximation to the actual travel times which is near the observed mean travel times, but is adjusted by the measurements from the current time-step. This problem can be reformulated as a least-squares problem as follows:

$$\min_{x\in R^m} \| \begin{pmatrix} A \\ \sqrt{\lambda} I_m \end{pmatrix} x - \begin{pmatrix} y \\ \sqrt{\lambda}\hat{x} \end{pmatrix} \|_2 \qquad [4]$$

Observe that the two norms are independent, given x.

B. *Traffic Pattern Study*

We have seen in class that a l_1 norm penalization has the property to provide results with a lot of entries equal to zero. We exploit this property of the l_1 norm to study spatial traffic patterns. For this we study the following optimization problem:

$$\underset{x\in\mathbb{R}^m}{\text{minimize}} \quad \tfrac{1}{2}||Ax-y||_2^2+\tfrac{\lambda}{2}||x-\hat{x}||_2^2+\mu||Kx||_1 \qquad [5]$$

The matrix $A \in R^{p\times m}$ represents information on the traffic patterns. For example, we can expect some neighboring links to have the same traffic conditions. In particular, we can encourage neighboring links to have the same speed. We realize that two links j_1 and j_2 with travel times x_{j1} and x_{j2} and length l_{j1} and l_{j2} have the same speed if and only if $\frac{l_{j1}}{x_{j1}} = \frac{l_{j2}}{x_{j2}}$ [6]

In this section, we will experiment our traffic pattern with different *K* matrices.

K matrix is produced as follows: For every intersection *u*, there are incoming links j_i and outgoing links j_o. For every (j_i,j_o) pair, there is a row at index *i* with value $1/l_i$ and at index *o* with value $-1/l_j$.

Followings are some of our observations:

- The matrix *k* would encourage on the solution *x* to the above optimization problem that vehicles travel with constant speed at the intersection.
- We experiment with our results on the data provided. We separate the data set into two different sets. The first set is called the training set and you use it to estimate the travel times *x*. Then use the second set (called validation set) and compute the estimation error using the value of *x* that you computed on the training set. We experimented with the data in several ways in an effort to translate the theoretical optimization model into a working model. We varied the optimization parameters λ and μ to minimize the objective function for a given data set, as well as training our model on various subsets of the data and studying the ability of the resulting model parameters to predict the behavior of the remaining observations. The results of these analyses are discussed in section IV.
- We also experimented with a similar K matrix to that described above. Instead of considering incoming/outgoing pairs, however, our second matrix considered n-1 pairwise differences between link speeds, effectively attempting to maintain the same speed for all links in an intersection using the l-1 norm trick.

III. IMPLEMENTATION

A. *Matlab CVX code to numerical implement the regularized least-squares optimization problem*

```
errors=[]
lamda=[]
values=[]
values=horzcat(linspace(0,1,11),linspace(1.5,40,10),logspac
e(2,4,10))
first=0
min_error=0
opt_x=[]
for k=values
cvx_begin
      variable x(815,1)
      minimize ((Ax-y)'(Ax-y)+k(x-x_hat)'(x-x_hat))
      subject to
      0x
cvx_end
error=norm(Ax-y,2)
if first==0
      min_error=error
      opt_x=x
      first=1
elseif min_errorerror
      min_error=error
      opt_x=x
end
errors=vertcat(errors,error)
end
plot(values,errors),ylabel('errors'),xlabel('lambda');
```

B. *Matlab code to generate K matrix*

```
K=[]
for i = 1:527
      row=incidence(i,:)
      for ej=find(row==1)
            for ek=find(row==-1)
                  vec=zeros(1,815);
                  vec(ej)=1/link_length(ej);
                  vec(ek)=-1/link_length(ek);
                  K=vertcat(K,vec);
            end
      end
end
```

C. *Matlab code to generate another K matrix (K_2)*

```
[m,n]=size(incidence)
K_2=[]
for i=1:m
      row=incidence(i,:)
      links=find(row)
      for j=1:length(links)-1
            vec=zeros(1,n);
```

```
            vec(links(j))=1/link_length(links(j))
            vec(links(j+1))=-1/link_length(links(j+1))
            K_2=vertcat(K_2,vec);
        end
end
```

D. Date set seperation (Notation: Training Set 1 = T_S 1; Validation Set = V_S)

K	T_S 1	T_S 2	T_S3	T_S 4	K2	
T_ S	1 - 3000	Random Split 50%	Random Split [0.9 0.7 0.5 0.3 0.1]	[400 1200 2000 2800 3600]	T_ S	1 - 3000
V_S	3001 -4419	Random Split 50%	Random Split [0.1 0.3 0.5 0.7 0.9]	[3600 2800 2000 1200 400]	V_S	3001 - 4419
Min error	910.7101	1.160			Min error	910.704
Opt λ	2.9167	1.5			Opt λ	2.9167
Opt μ	23.8846	1.5			Opt μ	23.8846

E. Geo-visualization result of using Google Earth

Please see VII. Attachment.

IV. Analysis

A. Average Travel Time Estimation

We use regularized least squares with cvx to numerically implement this optimization problem. To our surprise, the speed of lanes varies greatly, from 0 to almost infinity.

B. Traffic Pattern Study

Our primary task in data analysis was to identify an appropriate training set from which to determine the parameters of our model. We focused on two schemes for choosing a validation set, which had surprising results given our other analyses.

The first and simpler of the schemes we studied was to simply choose some portion of the earliest observations – Our first attempt used the first 3000 of the approximately 4500 measurements as the training set, and the remainder as the validation set. This approach yielded reasonable results for the time necessary to construct the training set.

The second method we tried was randomly selecting a percentage of the observations for use as the validation set. While less easy to implement, this converged to an accurate model faster (using a smaller proportion of the observations) than the naïve block approach we used first.

A surprising behavior of both of these methods is that they did not appear to lose accuracy as the size of the validation set in proportion to the training set increased; i.e. neither method produced 'overtrained' models. As is clearly visible in figures 3 and 4 of the attachment, both schemes exhibit a linear relationship between size of the validation set and model accuracy, with optimal error being achieved with training sets comprised of 90% of the data. This behavior is surprising given the apparent independence of the observed data. Singular value decomposition of the matrix of observations showed that the first 700 singular values (of a total of 817) differed only by a factor of order unity, suggesting that the routes taken by the probes was highly uncorrelated.

Perhaps owing to their similarity, the different K matrices we experimented with seemed to have no distinct effects on the model produced, yielding very similar results given training sets and optimization parameters.

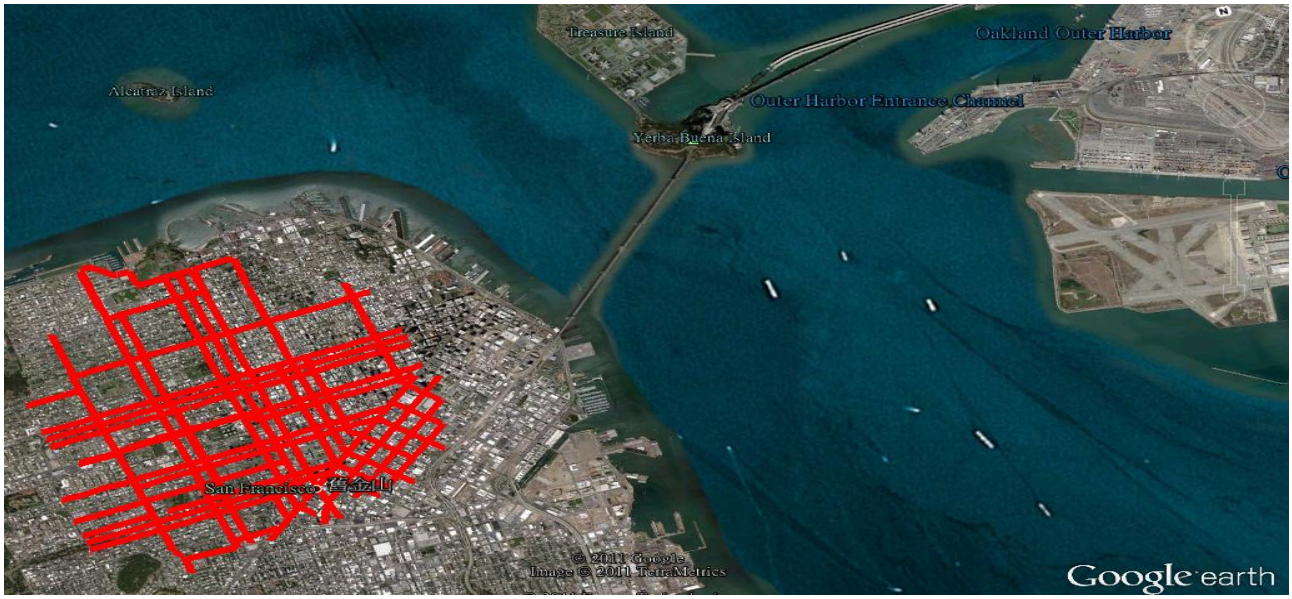
V. Conclusion

So far we have shown detail progress of our Traffic Project. Applying what we learned in EE 127A, we were able to estimate average travel time and study traffic patterns in San Francisco Bay Area.

VI. Acknowledgment

Special thanks to the guidance from Prof. Laurent El Ghaoui and GSI Aude Hofleitner

VII. ATTACHEMENT



Color Map	Average Estimation	Regularized Least-Square	K1 Training Set 1
			
K1 Training Set 2	K1 Training Set 3	K1 Training Set 4	K2 Training Set 1
			

Lane Speed	RGB	Color
0-1	000000	
1-9	FF0000	
10-99	00FF00	
100-999	0000FF	
1000-9999	00FFFF	
10000-99999	FF00FF	
>100000	FFFFFF	