A Network Tomography Approach for Traffic Monitoring in Smart Cities


Student
Ruoxi Zhang
rz9p4@mst.edu
Computer Science


Student
Sara Newman
sn976@mst.edu
Computer Science


Visiting Scholar
Marco Ortolani
marci.ortolani@unipa.it
Computer Science

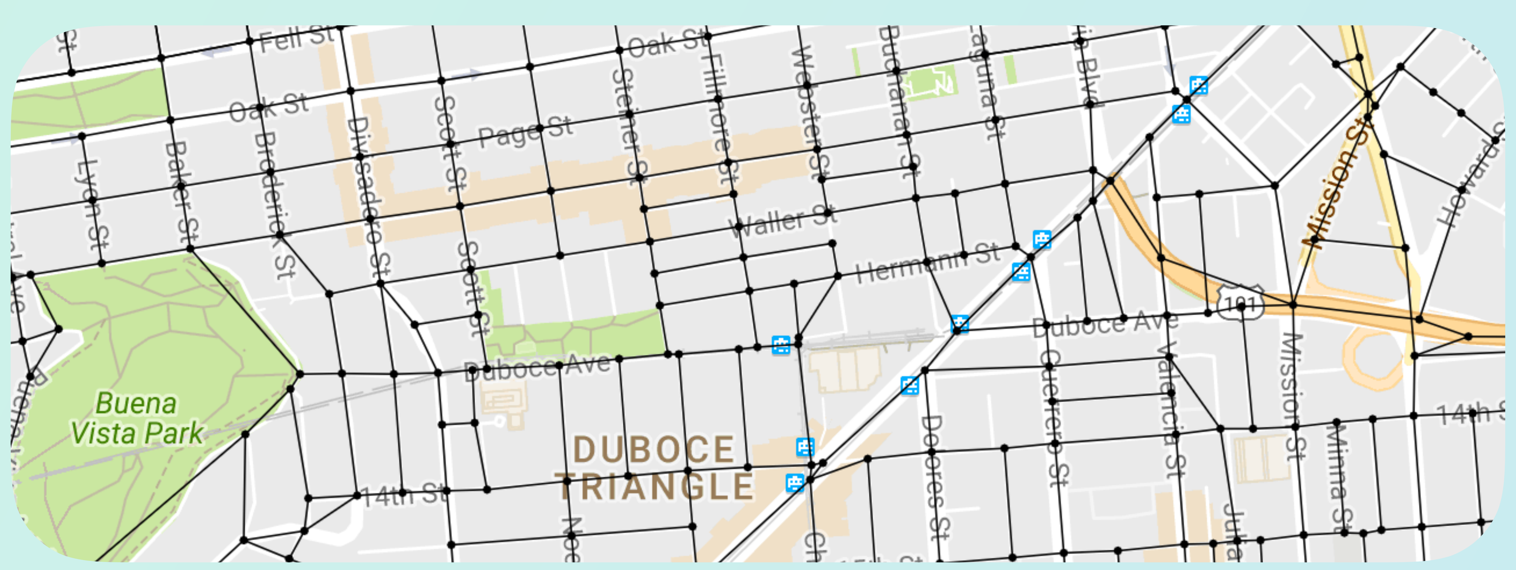

Faculty Advisor
Dr. Simone Silvestri
silvestri@cs.uky.edu
Computer Science

Abstract

Various urban planning and managing activities required by a Smart City are feasible because of traffic monitoring. As such, this project proposes a network tomography-based approach that can be applied to road networks to achieve a cost-efficient, flexible, and scalable monitor deployment. Due to the algebraic approach of network tomography, the selection of monitoring intersections can be solved through the use of matrices, with its rows representing paths between two intersections, and its columns representing links in the road network. Because the goal of the algorithm is to provide an inexpensive monitor set, this problem can be translated into a minimization problem over a matroid, which can be solved efficiently by a greedy algorithm. This approach is applied to both real road networks, based on downtown San Francisco, and synthetic, QuadTree-based road networks. The solution retrieved by the proposed approach is compared to the results of Clarkson's algorithm, a greedy 2-approximation algorithm designed to solve the weighted vertex cover problem.

Approach

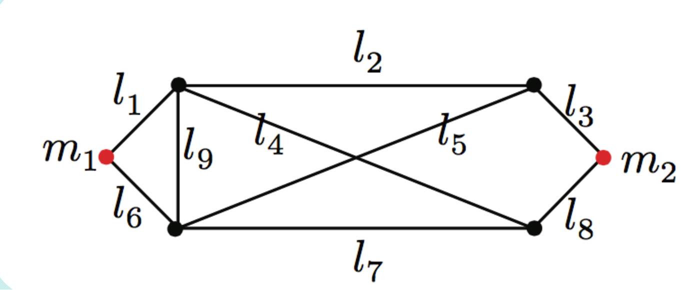
A map can be viewed as an undirected graph G = (V,E), where V is the set of nodes (intersections) and E is the set of links (streets). With a subset of nodes selected as monitors, measurements of paths between every pair of monitors are captured. A cost is introduced for placing a camera at each selected node.



The goal is to maximize link identifiability and minimize monitor deployment cost by monitor selection over a set of all possible monitor locations. The cost should be minimized while achieving the same inference error on links.

Network tomography provides a methodology for inferring internal network state through external, end-to-end measurements. It outputs solution with highest likelihood according to a predefined model.

This approach can be applied by introducing a routing matrix **R** and the end-to-end measurement vector **b**. By solving *Rx* = *b*, the unknown vector **x** will be calculated.



$$
\begin{array}{l}
\mathcal{P}_1: l_1\, l_2\, l_3\\
\mathcal{P}_2: l_6\, l_7\, l_8\\
\mathcal{P}_3: l_6\, l_5\, l_3\\
\mathcal{P}_4: l_1\, l_2\, l_5\, l_7\, l_8\\
\mathcal{P}_5: l_1\, l_4\, l_8\\
\mathcal{P}_6: l_6\, l_7\, l_4\, l_2\, l_3\\
\mathcal{P}_7: l_1\, l_9\, l_7\, l_8\\
\mathcal{P}_8: l_6\, l_9\, l_2\, l_3
\end{array}
\Rightarrow \mathbf{R} =
\begin{pmatrix}
1&1&1&0&0&0&0&0&0\\
0&0&0&0&0&1&1&1&0\\
0&0&1&0&1&1&0&0&0\\
1&1&0&0&1&0&1&1&0\\
1&0&0&1&0&0&0&1&0\\
0&1&1&1&0&1&1&0&0\\
1&0&0&0&0&0&1&1&1\\
0&1&1&0&0&1&0&0&1
\end{pmatrix}
$$

Traffic measurements often include noise, making the linear system unsolvable. Consequently, a parameter, Δ, is introduced, representing the maximum noise in the measurements. The linear system then becomes:

$$
\begin{array}{ll}
\text{minimize} & \Delta\\
\text{subject to} & R\mathbf{x} \le \mathbf{b} + \Delta,\ R\mathbf{x} \ge \mathbf{b} - \Delta\\
& \mathbf{x} \ge 0
\end{array}
$$

This problem is modeled using matroids, in which each line of the linear system is a vector in an |V|-dimensional space with an associated cost of monitor deployment. A basis of the matroid is a set of linearly independent vectors whose linear combination can represent any other elements in the vector space.

Every basis of the matroid provides the same inference error but different costs. The goal is to find the basis with minimum cost. The objective function, however, is submodular, which has the property that, as the size of the solution set increases, the marginal improvement gained by adding an additional element decreases.

Therefore, the problem is to minimize a submodular function over a matroid constraint. This can be solved by a greedy algorithm with an approximation factor of 1 - 1/e.

Algorithm 1 NT

```
1:  Calculate set, P, of all paths between each pair of nodes
2:  Calculate the costs, c_i for each p_i ∈ P
3:  B ← empty matrix
4:  M ← ∅
5:  while B ≠ basis do
6:      Calculate L such that ∀l ∈ L, l ∈ P,
7:                and is linearly independent from B
8:      Find p ∈ L such that p has minimum cost c
9:      B ← B ∪ p
10:     M ← M ∪ e, such that e the set of end nodes of p
11:     P ← P − p
12:     ∀p_i ∈ P, update the cost of c_i
    return B
```

Result