Integration Non-Traditional ITS Data Sources for Rural Areas

National Rural Intelligent Transportation Systems Conference
Huntington, WV
Tag-Based Data

- MTC (the regional MPO in the Bay Area) has a wonderful data set that is exactly like Bluetooth – toll tag readers!
- They’ve been doing this for years
- PeMS is the archive data system for Caltrans
- Framework already in place in PeMS to hold tag-based data
- Top screen is showing locations of tag readers
- This is a screen that shows the ability to select any two points for analysis
Tag-Based Data

- We store the statistics of the travel time between all pairs of readers every 5 minutes.
- For privacy reasons, we're required to toss the raw data within 24 hours (we do it faster than that).
- Aggregation allows us to plot various quantities over time.
- Here we're showing:
  - Travel time
  - June 9-16th
  - Minimum and Median
  - This route goes over Bay Bridge (HOV lanes)
Tag-Based Fusion

- Example of Fusion of tags and loops
- Matching tags across all possible readers can give you a crude estimate of OD
  - The toll tags are only a sampling of the vehicles (penetration rate)
  - The tag readers only read a fraction of the tags (detection rate)
- But you know the true volume passing by every station from the loops
- It’s possible to fuse loop detector data and tag data to produce accurate OD estimates per shift
- Once Bluetooth deployments are dense enough we’ll be able to do the same thing with that data (cheaper)
Bluetooth Data and Motivation

- Bluetooth detectors (and tag-based data in general) are becoming an important source of data for agencies
  - Easy to deploy
  - Allow for the relatively inexpensive collection of travel time
- But they don’t collect volume data
  - Only sampling the vehicles that have Bluetooth devices (that it can read)
- The total flow on a freeway is an important input to performance measures
  - Many spatial measures require it
  - VMT, VHT, Delay, LOS, etc
- Talk Agenda:
  - Project to fold Bluetooth data into an existing ADMS
  - Investigations into computing relevant performance measures
  - Can we use spare Bluetooth measurement devices in conjunction with dense ITS devices to compute performance?
  - What else can we do by fusing data?
Strengths and Limitation of Two Data Sources

• Two data sources
  – Link data: Bluetooth travel times
  – Point data: Loop detector volume

• Two technologies have certain drawbacks when used separately

• By fusing the two technologies, one can obtain accurate performance measures in a cost effective manner

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Volume</th>
<th>Travel time</th>
<th>Intra-link measurements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bluetooth and other link data</td>
<td>Inadequate Depends on penetration and detection rates (5-10%)</td>
<td>Adequate Ground truth, yet sampled</td>
<td>Inadequate Only measuring end-to-end travel time</td>
</tr>
<tr>
<td>Loop detector and other spot sensors</td>
<td>Adequate Accurate and complete</td>
<td>Inadequate Spot measured speed, Interpolated, temporal mean (~10% error if densely instrumented)</td>
<td>Semi-Adequate Only if there is dense spacing</td>
</tr>
</tbody>
</table>
Outline for Fusing Data

• **Travel Times**: estimation of Bluetooth travel times
  – Many techniques for filtering Bluetooth data
  – We use k-nearest neighbor running median smoothing
  – Can estimate different quantiles (say 20th and 80th percentiles) to estimate average travel times for fast travelers and slow travelers separately
    • Useful when the trips consist of fast HOV traffic and slow non-HOV traffic

• **Volumes**: Obtain loop detector traffic volumes
  – Spot speeds are available in some cases, but we ignore them for simplicity for now

• **Performance Measures**: If we have…
  1. Volume-only: VMT
  2. Travel time only: Travel time reliability (buffer time, planning time etc.)
  3. **Volume + travel time**: VHT, Delay
Note About Delay

• In a dense loop-based system, “negative” delay isn’t counted
• But a Bluetooth-based system naturally incorporates this
• Hence we shouldn't expect that the delays are going to match at all
Fusion Scenario: Dense

- Suppose Bluetooth readers are located at the start, A, and end, B, of a route
- Point sensors over route A-B could be deployed with varying degree of density
  - Sparse: a single sensor over the whole route
  - Dense: one every quarter mile
- We propose different fusion methods for different degrees of loop density
- Computation methods:
  - “All-loop method”:
    - Use link-wise speed + volume from point sensors
    - This is typical PeMS method
    - We’ll call this the ground truth for this study
  - “Dense-fusion”:
    - Use blue-tooth speed (intra-route links are assumed to have same speed)
    - VHT = VMT (computed from loop data) / average speed (computed from blue-tooth data)
    - Delay = VHT - VMT / 70 mph (assuming 70mph is freeflow speed)
Fusion Scenario: Sparse

- "Sparse-fusion"
  - Assumptions:
    - "X" loop detectors on the route (where X is small)
    - Smooth volume over the route: Spot volume(s) is extrapolated to the whole route
    - Constant travel time over the route: for each intra-route link, travel time is assumed to be the same over the whole route
  - Hourly VHT = hourly loop volume * hourly single-vehicle travel time (veh-hr)
    - Hourly single-vehicle VHT is the average of the 12 5-min travel times in the given hour
  - Hourly Delay = hourly loop volume * hourly single-vehicle delay (veh-hr)
    - Hourly single-vehicle delay is the hourly single-vehicle VHT – free flow travel time (minimum of hourly single-vehicle travel time)

- Problem:
  - Flows aren’t constant along a route

- Possibly use Bluetooth data to estimate volume?
  - Bluetooth detection rates are (approximately) known or can be estimated from nearby loop data
  - In this case, one could use Bluetooth detections (not matches) to estimate the volume at each reader and use them as additional volume data along the route
# Fusion Scenarios: Summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Loop deployment</th>
<th>Data source</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>All-loop</td>
<td>Dense loop</td>
<td>Loop volume, loop speed</td>
<td>Serve as the ground truth in the study</td>
</tr>
<tr>
<td>Dense-fusion</td>
<td>Dense loop</td>
<td>Loop volume, bluetooth speed</td>
<td></td>
</tr>
<tr>
<td>Sparse-fusion with X loops</td>
<td>Sparse loop</td>
<td>Loop volume, bluetooth speed</td>
<td>As the number of loop stations grow, this approaches dense-fusion</td>
</tr>
</tbody>
</table>
Application: I-99NB Site

- 99NB south of Sacramento:
  - Two bluetooth readers
    - Between Elk Grove and Laguna, abs pm 287.73
    - MLK and 99, abs pm 296.02
    - Distance: 8.29 mile
    - Route “222 to 221” is 99NB

- PeMS Route
  - PeMS computes travel times over loops as well
  - PeMS Route is a similar route but slightly longer
  - Distance: 12.4 mile from PM 286.3 to 298.7
  - Densely instrumented: 24 stations, or ~2 stations per mile
  - Retrieved route travel time and hourly summary (VMT, delay, etc)
Characteristics of Site

- Traffic flows north in the morning period
- Passes through many suburbs: volume builds on the way to Sacramento
- Congestion is peak in the morning
Application: I-99NB Site

- Example from one day: Tue Jun 23, 2009
- Bluetooth reader information:
  - 16,542 records
  - 11,163 vehicles detected
    - 2,888 detected twice
- PeMS route travel time (top):
  - Lane 2 travel time over the route is shown
  - 5-min values of travel time
  - Ranges between 10-18 min
- Bluetooth travel time (bottom):
  - Running k-nearest neighbor algorithm
  - Showing median (and quantiles) of distribution
  - We'll use median as the representative value
  - Ranges between 7-16 min (shorter)
- Side note:
  - Individual travel times vary much for the similar departure time
  - During stable hours, say Midnight-5AM, the middle 50th percentile travel times range between 7-8 min, more than 10% variation!
Application: Hourly Profile Fusion

- Comparison of the following methods:
  - All-loop (PeMS value), Black
  - Dense-fusion, Red
  - Sparse-fusion with 1 station (19 different estimates), Gray
  - Sparse-fusion with 2 stations (use the two stations at the beginning and at the end of the route), Blue

- Results for VHT and delay
- We can observe that route average from the full loop data is about the average of different fusion data from partial, single loop data
Application: Total Daily Delay Fusion

- Look at distributions
- Top plots show daily VHT and delay from various methods
  - Sparse-fusion with 1-station: the histogram represents the distribution of 19 different values
  - All-loop (black): ground truth
  - Dense-loop (red)
  - Sparse-loop with 2 stations (blue)
- For VHT, the absolute relative error from the two fusion methods are about 10%
- For delay, the absolute relative error from the two fusion methods are greater, ranging between 15-24%
- This isn’t bad!

<table>
<thead>
<tr>
<th>Method</th>
<th>VHT Relative error</th>
<th>Delay relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dense-fusion</td>
<td>-9%</td>
<td>-24%</td>
</tr>
<tr>
<td>Sparse-fusion with 2 stations</td>
<td>11%</td>
<td>15%</td>
</tr>
</tbody>
</table>
Summary

- Attempting to get performance measures from fusing tag-based data and loop-based data
- We proposed fusion methods to calculate hourly / daily route VHT and delay from bluetooth travel time and loop data, for various loop deployment densities
  - Dense-fusion, and sparse-fusion with x stations
- The method was applied to a 10 mile corridor the results were compared with full-loop data over the route
  - Single-station Sparse-fusion accuracy depends a lot on which station data is used.
  - For the study route, daily VMT varies greatly from station to station, though the entire route was only ~10 mile; consequently, the fusion VHT and delay estimates when a single count station is used have relative error of about 50%
  - Two-station sparse-fusion accuracy was better (~11% error rate for VHT)
  - Dense-fusion accuracy was similar (-9% error rate for VHT)
- There are other applications out there for fusing the data!