Route Prediction from Trip Observations
Jon Froehlich (UW) and John Krumm (MSR)
What if we could **predict** a driver’s route?

- road grade
- road curvature
- traffic conditions
HEV Charge/Discharge Control System Based on Navigation Information

Convergence Transportation Electronics Association 2004
Nissan Motor Company

road grade

traffic conditions
**Predestination:**

Inferring Destinations from Partial Trajectories

Trip starts, uniform destination probability

4 squares south, half of region eliminated

More squares in trip, ¾ of region eliminated

Ubiquitous Computing 2006
John Krumm and Eric Horvitz
Our **Goal:**

predict a vehicle’s entire route as it is driven
Data Collection

Microsoft Multiperson Location Survey

- GPS data collection initiative
- Started in 2005
- 252 subjects
  - Volunteer to drive with GPS recorder
  - Avg. 15.1 days of data per person
- 2.2 million GPS location points

Garmin Geko 201

Seattle
Greater Seattle
Washington
A subset of raw GPS data
Motivation

Data Collection

Trips

Route Detection

Route Prediction

Closest Match

Future Work

Threshold
We need to transform this raw GPS data into trips...
From GPS Data to Trips

• A trip describes a driver’s path through time and space using time stamped GPS data

• Three stage transformation process:
  1. Trip Segmentation: segment the trips into multipoint trip objects
  2. Trip cleansing: clean the trips by removing invalid data points
  3. Trip filtering: filter the trips to eliminate false trip objects
A subset of segmented, cleaned, and filtered trip data
# Overview of Trip Data

14,468 trips / 240 subjects

<table>
<thead>
<tr>
<th>Description</th>
<th>Avg.</th>
<th>Med.</th>
</tr>
</thead>
<tbody>
<tr>
<td>trip distance (miles)</td>
<td>7.7</td>
<td>4.2</td>
</tr>
<tr>
<td>trip time (min)</td>
<td>16.3</td>
<td>11.5</td>
</tr>
<tr>
<td>num trips / day</td>
<td>4</td>
<td>3.9</td>
</tr>
<tr>
<td>num trips / subject</td>
<td>60.3</td>
<td>50</td>
</tr>
<tr>
<td>num days of data / subject</td>
<td>15.1</td>
<td>13</td>
</tr>
</tbody>
</table>

**High Level Trip Stats**

**Greater Seattle Area**
Trips to Routes

- A **trip** describes a **driver’s path** through **time** and **space** using time stamped GPS data.
- A **route** is simply an **abstraction of a trip** (or trips) **without the temporal component**.
  - That is, a route is a collection of latitude, longitude pairs that define a directed path.
- A **regular route** is a path that a driver drives often.
For all points in Trip A, we find the closest trip segment in Trip B.

The average $d_{AB}$ represents the trip similarity score $Score_{AB}$ from Trip A to Trip B.
We repeat the algorithm to calculate the similarity score $\text{Score}_{BA}$ from Trip B to Trip A.
The final trip similarity score between Trip A and Trip B is:
\[
\text{Score}_{AB} + \text{Score}_{BA} \over 2
\]

We use this trip similarity score to automatically detect routes. Trips that are very similar are along the same route.
Route Detection

• We create routes from trip data by comparing every trip in a subject’s dataset.

• The result of each trip by trip comparison is the previously described trip similarity score.

• These scores are stored in a trip similarity matrix.

• We repeatedly combine trips with the lowest scores (most similar) into routes.
Our Clustering Technique

• **Dendrogram Clustering**: a hierarchical clustering technique
  – Recursively clusters data points until a pre-specified threshold is reached

• In our case:
  – We repeatedly combine trips into clusters until the lowest score in the similarity matrix is > 0.05 miles
  – The size of the trip cluster represents how frequently that route was traveled
Example: detect the three routes
Motivation
Data Collection
Trips
Route Detection
Route Prediction
Closest Match
Threshold
Future Work

Dendrogram Cluster

No scores below our cutoff threshold of 5

Final Route Matrix

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>5</td>
<td>8.5</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>0</td>
<td>3.5</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>C</td>
<td>8.5</td>
<td>3.5</td>
<td>0</td>
<td>5.5</td>
<td>7.5</td>
</tr>
<tr>
<td>D</td>
<td>9</td>
<td>4</td>
<td>1.5</td>
<td>0</td>
<td>5.5</td>
</tr>
<tr>
<td>E</td>
<td>14</td>
<td>9</td>
<td>5.5</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>F</td>
<td>16</td>
<td>11</td>
<td>7.5</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

Microsoft Research

University of Washington

Computer Science and Engineering
Route Prediction

• We attempt to predict a driver’s **entire route** based on **previous** trip history

• Our algorithms are based on the observation that drivers are highly regular
  
  – A **repeat trip** is a trip that occurs more than once along a route
  
  – **39.3%** of the trips in our dataset are **repeat trips**
  
  – For **67 / 240 subjects**, the repeat trip rate was **greater than 50%**

  • That is, **one out of every two** trips for these drivers is along an established route
After approximately 1 month of observation, the number of repeat trips reaches 50%.
The top ten most frequently traveled routes account for 50% of a driver’s trips.

The most frequently traveled route accounts for 12% of a driver’s trips.

This line represents the hypothetical case where no repeat trips occurred in our dataset.

average cumulative distribution of trips in routes
Basic Premise

• As a trip progresses, we find which previously driven route, if any, the driver is on

Trip A

Route 1

Route 2

Closest Match: Route 2
Testing Setup

• Tested two route prediction algorithms on:
  – 14,468 trips
  – 240 subjects

• Leave one out approach
  1. One test trip is left out of a subject’s dataset
  2. Remaining trips clustered into routes
  3. Test trip is then virtually driven in 5% increments
  4. Route prediction algorithms applied
  5. Repeat steps 1 – 4 on every trip from each subject
1. Closest Match Algorithm

- **Input:**
  - The current trip
  - The route database

- **Output:** an ordered list of the routes most similar to the current trip
  - The *closest matching* route (index zero of ordered list) is taken as the predicted route
After 50% of trip has been driven, the correct route is, on average, within the top 2 matches.

After 1 mile, correct route within top 8 matches.

After 5 miles, correct route within top 5 matches.
At halfway, the correct prediction is within the top 10 matches over 90% of the time.

Halfway into a trip, we can correctly predict 17% of the routes.
2. Threshold Match Algorithm

• **Input:**
  - The current trip (and travel distance)
  - Distances to 1\textsuperscript{st} and 2\textsuperscript{nd} closest routes ($d_1$ & $d_2$)
  - The route database

• **Output:** the predicted route and a confidence measure
  - Confidence measure represents how often the route prediction has been correct in the past with the same parameters
When $d_1 \leq 0.05$ miles and grows large $d_2 > 1$ mile, our accuracy is greater than 85%.

... and as $d_1$ grows small, our accuracy increases as expected.

As $d_2$ grows large.
When $d_1$ is small and $d_2$ is large the accuracy trend becomes more pronounced as the trip progresses.
A high density of trips where both $d_1$ and $d_2$ are small.
Future Work

• We only incorporated one feature into our route prediction: geographic distance

• Other features to explore:
  – Partial route matching
  – General route popularity
  – Common destinations amongst area population
  – Optimal path behavior
  – Driver familiarity with area
  – Identifying the driver
  – Identifying passengers in the car
  – Temporal aspects such as start time and route recencies
Summary

• We provided a methodology for automatically extracting routes from raw GPS data without knowledge of the underlying road structure

• We presented a detailed discussion and analysis of repeat trip behavior from a real world dataset of 14,468 trips from 252 drivers

• We developed and evaluated two algorithms that used a driver’s trip history to make route predictions of their current trip
Thank You!

- **Contact Information:**
  - Jon Froehlich: jfroehli@cs.washington.edu
  - John Krumm: john.krumm@microsoft.com

- **Acknowledgements:**
  - Eric Horvitz, Kayur Patel, Scott Saponas, and Mike Toomim
Trip Segmentation

1. Sort each subject’s raw GPS data chronologically
2. Find gaps between two consecutive recorded points \((P_1, P_2)\) of three minutes or more
3. If a gap is found, \(P_1\) becomes end point of last trip and \(P_2\) the beginning point of the current trip
Trip Cleansing

Invalid Starting Point

Invalid Starting Point Removed

Remaining Valid Trip
Trip Cleansing

Invalid GPS Point
(Green Trip Segment=397.8 mph)

Invalid GPS Point Removed
Trip Filtering
Trip Filtering