Understanding urban mobility using crowdsourced GPS data

Dr. David McArthur, Dr. Yang Wang et al.
Urban Big Data Centre

• UK-wide data service for researchers who want to use big data to address a range of urban challenges, both local and global.
• Funded by the UK Economic and Social Research Council
• Objectives
  • **Develop novel solutions for using and sharing urban big data** including the infrastructure, tools and expertise to access such data;
  • **Provide high quality training and outreach activities** to equip researchers and decision-makers with the skills and knowledge to use big data to inform public policy debates and business innovations;
  • **Deliver cutting-edge research** to develop methods and tools to analyse urban big data as well as exemplar projects on substantive urban issues.
Plan for the talk

• I’ll be talking about one of my projects at the UBDC

• It’s about one way in which we might get different (better?) transport data
Some traditional data sources and limitations

• Surveys
  • Expensive
  • Small samples
  • Hard to complete
  • May lack detail

• Sensor networks
  • Requires (expensive) infrastructure
  • Lacks origin/destination info
  • Not multimodal
(Some) Big data
Some limitations

- Hard/expensive to access
- Pre-processing of data can be a black box
- May be tied to one mode
- Spatial coverage may be patchy
- Unknown biases
Smartphone data

• Smartphones offer the chance to collect rich data
  • Independent of operator
  • Information for all travel modes
  • Full door-to-door OD and route data
  • Potentially real-time
Catch! Project

• The Catch! (citizens at the city’s heart!) app is a journey planning app
• It passively collects GPS trajectories
• It utilises the phones’ sensors to infer travel mode
• Users get to contribute data to improve transport planning in their city
• Insight from the data can feed back into better journey suggestions
• Funded by Innovate UK
Catch! App

• Includes real time information on road and public transport performance
Catch! App

Travel

Favourites

no favourites yet

Nearby

Bus
Train
Tube

Great Titchfield Street / Oxford Circus (Stop OJ)
East to Archway, Bakers Arms, H...

398, 10, 25, 98, 55, 73

299m

Great Titchfield Street / Oxford Circus (Stop OP)
West to Hammersmith, Notting...

55, 10, 73, 98, 390, 25

390m

Wardour Street (Stop OM)

Journey planner

start

end

current location

select destination...

7 Carmelite Street, London, United Kingdom

London

The British Museum

COVENT GARDEN

Buckingham Palace
Morning rush hour in London

Analysing infrastructure at a number of levels and over long than normal periods can highlight city-wide as well as junction-specific traffic flow issues.
The Consortium

SMEs
- TravelAi, The Behaviourlist, Elgin Roadworks, Placr
- App development, data sources, citizen messaging, impact assessment

Research organisations
- University of Glasgow (UBDC), University of Leeds (CDRC), Transport Systems Catapult
- Data cleaning, anonymization, aggregation, analysis

Local Authorities and cities
- Coventry, Ipswich, Leeds, Newcastle, Oxfordshire
- Citizen access, sounding boards, pilots, data sources, advocates
Data from the app

- Person identifier
- Latitude, longitude
- Time
- Inferred mode
- Collected every 5 seconds (may change)
Processing

• Begin by removing nonsensical points e.g. points where the travel speed is unrealistic

• Assign the points onto the transport network (map matching)
Map matching approaches

• Geometric approach
  • node-to-node, node-to-link, curve-to-link

• Topological approach
  • geometric approach plus connectivity of the road network

• Advanced approaches
  • Weight based or probabilistic algorithms
# Four open source libraries

<table>
<thead>
<tr>
<th>Name</th>
<th>Category</th>
<th>Logic</th>
</tr>
</thead>
</table>
| Mapillary, map matching (mm) in Python    | Combined Geometric and Topological | - Selecting road links that are close to the sampling points,  
- connectivity of road network is of concern but only inside the points bounding box.                                                                                                               |
| Multiple Hypothesis Technique (MHT) based | Combined Geometric and Topological | - Multiple route candidates are kept in memory when matching each sample point at a time,  
- connectivity, distance to samples and turning directions are of concern all the time,  
- path is determined by consistent updating scores the candidates road segments.                                                                                                               |
| GraphHopper in Java                       | Advanced - probabilistic           | - Select candidate closest to GPS points,  
- probability is calculated based on their distance to samples,  
- sorting route by minimizing probabilities,  
- assign points to road edges.                                                                                                      |
| Barefoot in Java                          | Advanced - Hidden Markov Model     | - Probability lattice is calculated where  
  - emission probabilities are the chances of GPS points observed at the road segment are reduced when distance increases.  
  - transition probabilities are the chances of movement between road segments due to road connectivity at consecutive times.  
  - the best path is calculated by Viterbi algorithm from the probability lattice.                                             |
Aggregation of output

- Journey information can be aggregated at the link-level
- Can provide mode-specific counts of users/journeys
- Can provide a very fine-grain temporal scale

Average weekend speeds
Aggregation output

- Information can be aggregates at the level of junction
- How long do different road users have to wait at junctions?

Average vehicle waiting time
Semantic Trajectory

- Another strand of our work deals with the semantic annotation of the GPS trajectories
Models or Frameworks for Semantic Trajectory Development
Paradigm of Trajectory Data Mining (Y. Zheng)
Models or Frameworks for Semantic Trajectory Development

Weka-STPM – an open source toolkit
Models or Frameworks for Semantic Trajectory Development

Semantic Trajectory Platform Architecture in (Z. Yan et al)

Trajectory computing platform

Trajectory annotation platform
An Extra Data Anonymization Layer to Semantic Trajectory Framework 1

• Current frameworks have no facilities for data anonymization and data sharing;

• Two main sources of contextual information: road network and geographical regions or points through Map Matching and stop/move Detection and Annotation;

• Our contribution:
Adding an extra data anonymization layer to the framework to better:
(a) Protect individual users’ privacy
(b) Develop a workflow including methods and algorithms towards such a goal using a raster/grid based generalization structure.
An Extra Data Anonymization Layer
Stop Detection

Stay point detection is

- the set of geographical locations that an individual stays at for a certain amount of time (Li et al., 2008)

Knowing the stay points allows us to

- infer activities that are conducted at different locations (Liao et al. 2007, Ye et al. 2009),
- Segment the trajectory with separate travel purposes (Zheng 2015)
- Find points where modes are switched e.g. walking to train (Zheng et al. 2008, , Patterson et al. 2003, Liao et al. 2007, Gonzalez et al. 2008).
Stop Detection Methods

Threshold based Approaches
taking GPS embedded or calculated parameters, such as speed, dwelling time, clustering density and ‘power-off’ gap durations (Ashbrook and Starner 2003, Schuessler and Axhausen 2008, Srinivasan et al. 2008). Some problems:
• threshold settings are arbitrary and require additional information about the raw GPS data
• speed values are unreliable due to limitations of GPS

Density based Approaches
• spatial clustering algorithms (e.g. DBSCAN ) assumes there are a larger number of points clustered around significant locations (Schoier and Borruso, 2011).
• The algorithm scans for a minimum number of tracking points ($MinPts$) around a randomly selected unvisited points within a pre-specified search radius ($eps$) then further aggregates clusters if they are densely connected (Ester et al. 1996).
• Approaches such as Hinneburg and Keim (1998), Ankerst et al (1999) and recent work proposed by Campello et al (2013) try to simplify the parameters
• Other proposed improvements include introducing temporal and other dimensions (Birant and Kut 2007, Hwang 2013). ST-DBSCAN (Birant and Kut 2007), temporal DBSCAN (Hwang et al. 2013), interpolate missing GPS points (Hwang et al. 2017).
• DBSCAN is less sensitive to noise and can detect stops with arbitrary shapes. It doesn’t work well with large temporal gaps, loss of GPS signal or movement inside a house
Stop Detection in Semantic Trajectory Mining (Y. Zheng et al.)

- Density based method: distance between each points to all other points until the final distance and the duration of the set of points exceeds the thresholds. The algorithm loop and add points into the candidate stop until the clusters is no longer expansible.

- A supervise model: features including a) minimum bounding ratio (MBR), average and centre distance to road segments, duration and speed for last stop, b) term frequency invers document frequency (tf-idf) for Point of Interests, c) repetitive historical visits, to filter out the clusters caused by slow speed.

- The method is designed for taxi stop location detection, therefore, it is transport network constrained.

```
Algorithm 1: ParkingCandidateDetection
Input: A road network G, a trajectory Tr, distance threshold δ, time threshold τ
Output: A set of parking candidates P = {P}
1. i ← 0, M ← |Tr|, P ← Ø, δ ← δ;
2. while i < (M − 1) do
   3. j ← i + 1; flag ← false;
   4. while j < M do
      5. dist ← Distance(p_i, p_j);
      6. if dist < δ then j ← j + 1; flag ← true;
      7. else break;
   8. if p_j+1, t − t_i > τ and flag = true then
      9. for each point p ∈ Tr[i, j] and p ∈ P do
         10. P.Add(p); /* build a candidate */
      if j = j + 1 then
         11. P.Add(Ø); P ← Ø;
         12. /* add the minimum bounding box of P into P */
      i ← i + 1;
13. return P
```
Stop Detection in Semantic Trajectory Mining
Weka-STPM

• An Intersection based Stop and Move Trajectories (IB-SMoT) approach where GPS trajectories are spatially intersected with pre-defined geographical file to look for durations that spend inside each stop shape to determine a stop.

• A clustering-based Stop and Moves of Trajectories (CB_SMoT) algorithm based on DBSCAN but clusters speed values of the trajectory. By apply the algorithm, slower speed part of trajectories are clustered. If stay duration is greater than a threshold, stay point is detected.

• A direction-based stops and Moves of trajectories: similar direction change controlled by a minimal direction change threshold, minimal amount of stop duration is used to verify if the direction change is noise or direction change reaches its end.
Stop Detection in Semantic Trajectory Mining
Yan (et al. 2013)

A combined velocity-based and density based methods.

• For the velocity-based method, the speed of successive GPS points are compared with the minimum value of observed speed on the by-passing road segment and the average travel speed of the user.

• further compare the time duration of the groupings of the stop episode with a minimal stop time threshold to eliminate congestion stops.

• Since speed value is not always reliable to indicate stop, the authors apply another supplementary density-based method to cover generic cases.

Velocity-based stop identification

**Algorithm 2: Velocity-based trajectory structure**

**Input:** a raw trajectory \( T_{raw} = [p_0, p_1, ... , p_n] \)

**Output:** a structured trajectory \( T_{str} = [\tau_0, \tau_1, ... , \tau_n] \) where \( \tau_i \) is a tagged trajectory episode (stop S or move M)

1. Begin
2. If initialization: calculate GPS instantly speed if needed \( \Omega \)
3. ArrayList: \( x, y, t \), gpsList = getGPSList \( \tau_{stop} \);
4. If no instant speed from GPS device then
5. Compute GPS instant speed \( s \), for all \( p = (x, y, t) \in \) gpsList;
6. For all \( p = (x, y, t) \in \) gpsList do
7. \( \Omega \) = getDynamic \( \Omega_{stop} = \Omega_{stop}(\rho, \theta_{stop}) \);
8. Compute GP point as a stop point \( S \) or a move point \( M \)
9. If instant speed \( s < \Omega_{stop} \) then
10. Tag current point \( p(x, y, t) \) as a stop point \( S \);
11. Else
12. Tag current point \( p(x, y, t) \) as a move point \( M \);
13. End
14. Compute episode(s): grouping consecutive same tags/"
15. For all the consecutive points with the same tag \( S \) do
16. \( \Omega_{stop} \) = getDynamic \( \Omega_{stop} = \Omega_{stop}(\rho, \theta_{stop}) \);
17. Compute stop episode
18. Get the time duration \( t_{stop} \) of these points;
19. If \( t_{stop} > t \) the minimal possible stop time then
20. \( \Omega_{stop} = \Omega_{stop}(\rho, \theta_{stop}) \), \( \Omega_{stop} \) = add the stop episode;
21. Else
22. Change the \( S \) tag to \( M \) for all those points; // as "congestion"
23. For all the consecutive points with the same tag \( M \) do
24. Compute move episode
25. Move = \( \text{computeMove}(\rho, \theta_{stop}, \text{duration}) \) // Create a move episode \( T_{raw} \), \( \text{move} \), \( \text{move} \) = add the move episode;
26. Return the structured trajectory \( T_{str} \);
A Raster Sampling based Method

A ‘top-down’ raster sampling method which directly queries a set of GPS records and samples those with significant differences

- Geographical attributed raster cells by nature impose spatial constraints while we try to sample temporal and other attributes inferred from the GPS records
- A data clustering method is performed at the final stage
- It does not sample the density of GPS records inside grid cells, but rather information such as total dwelling time

Advantages:

- requires only the setting of the raster cell size
- fast and accurate (compared to a travel diary)
Exploring GPS Indicators for Stop Detection

Data: a day-to-day episodes of one user’s one month, from 2016-07-12 to 2016-8-10 which are cleaned, and a travel dairy containing locations of stops in chronological order

Method:
- Top-down sampling method which depend on indicators including
  - (a) time difference between two consecutive GPS tuples,
  - (b) an rough estimation of single trip GPS dwelling time at a given cell,
  - (c) a dwelling time deducing the travel time observed before and after a given GPS record,
  - (d) an estimation by pulling actual dwelling time per visit.
- Natural Break (Jenks) with goodness of variance fit over 0.8, to cluster the cell values into groups then select stops
- Two baseline methods:
  - (e) using thresholds to select stops with higher GPS dwelling time
  - (f) detecting stops less ‘bounded’ with the road network through a map matching process, are chosen as baselines for comparison.
Summery: An extra raster/grid layer

Advantages:
- enable multi-level data sharing while protect privacy;
- facilitate stop detection and further stop/move segmentation;
- support fast and semantic enriched GPS queries without performing expensive spatial joins or intersections;
- ease further spatial/temporal activity pattern mining and place/route recommendations.

Limitations:
- extra processing time;
- stop/move segmentation is still a raster/vector combined method;
- top-down approach missing shorter stays;
- scalability needs to be investigated.
(Some) limitations of the project

• Will the app be used by enough people
• Will it drain people’s batteries?
• The data are biased; but how? Will this change over time?
Thank you for your attention.
Questions?