Path Travel Time Estimation

The document describes the codes and datasets we have used in paper [1].

**Download the data and codes** [here](http://research.microsoft.com/apps/pubs/?id=217493).

The datasets consist of four parts:

1. The processed real-time traffic flow (over a short period) and the historical traffic patterns (over a long period) on each road segment in Beijing.
2. Geographical features of each road segment extracted from road network data and points of interests.
3. One-day trajectory data generated by over 30,000 taxicabs in Beijing.
4. Query paths generated by two drivers to test the performance of our method.

The codes are comprised of two parts: Context Aware Tensor Decomposition (CATD) Algorithm and Optimal Concatenation (OC) Algorithm.

The datasets and codes cannot only partially replicate the results we present in paper [1] but also enable other applications, such as gas-effective routing, fine-grained air pollution alert, and business location choosing. For more details, please refer to paper [1].

**Reference:**

Please cite the following papers when using the dataset.

[1] Yilun Wang, Yu Zheng, Yexiang Xue. [Travel Time Estimation of a Path using Sparse Trajectories](http://research.microsoft.com/apps/pubs/?id=217493). In the Proceeding of the 20th SIGKDD conference on Knowledge Discovery and Data Mining (**KDD 2014**).

[2] Yu Zheng, Licia Capra, Ouri Wolfson, Hai Yang. [Urban Computing: concepts, methodologies, and applications](http://research.microsoft.com/apps/pubs/?id=211950). ACM Transaction on Intelligent Systems and Technology. 5(3), 2014

[3] Yu Zheng. [Trajectory Data Mining: An Overview](http://research.microsoft.com/apps/pubs/?id=241453). ACM Transaction on Intelligent Systems and Technology. 6(3), 2015

Please cite paper [1] and [2] when using this dataset.

# Code Description

## Context Aware Tensor decomposition (CATD)

The first part of code is to supplement the missing values in the tensor Ar, which denotes the travel time of each user on each road segment at different time intervals, as illustrated in Figure 1. The code is written in Matlab, stored in “*Code/CATD*” folder, executing a context-aware tensor decomposition (CATD) with an element-wise solver. Detail explanation regarding the code is shown in the Matlab file named “*catd.m*”.



Figure 1. Context-aware tensor decomposition

## Optimal Concatenation (OC)

This part of codes is to find the most optimal concatenation of sub-trajectories to best estimate the travel time of path. The code is comprised of three major components: dynamic programming, frequent trajectory pattern mining, and suffix-tree-based index. The C# project of OC is in “*Code/OC*”folder, with a hierarchical structure of classes shown in Figure 2:



Figure 2. Architecture of the codes for optimal concatenation

Class description:

1. main.cs: this class is the main function for Optimal Concatenation including dynamic programming, trajectories loading, and invert list creating, etc.
2. tensorDec.cs: this class is the interface between CATD and OC, it reads the core tensor and factor matrices from CATD and computes the recovered results in the original tensor.
3. traj.cs: this class stands for the definition of trajectories, which contains the road segments sequence, travel time, length, invert list, etc.
4. trie.cs: this class is the suffix tree (trie) for frequent trajectory pattern mining and suffix-tree-based index. It contains functions of various operations and indexing in suffix tree including trajectory indexing, travel time indexing, support of trajectories, suffix tree construction, suffix tree output, etc.
5. trieNode.cs: this class stands for the nodes in suffix tree. Each node contains road segments ID, support, child, index for trajectories traversed on this node, travel time, etc.

# Data Description

## Tensor Data

Tensor data, stored in “*Data/TensorData*”folder, contains three parts, demonstrated as matrices X, Y and tensor A in the following figure, respectively. The dataset is used in CATD. For the detailed descriptions of the figure please refer to paper [1].

1. *temporal X.txt* is the temporal feature matrix X based on Tensor A, representing the correlation between different time slots in terms of the coarse-grained traffic conditions.
2. *geospatial Y.txt* is the geospatial feature matrix Y extracted from Beijing’s Road Network in 2012 and Beijing’s POI file in Quarter 3, 2012, capturing the similarity between different road segments in geographic spaces. The columns, separated by blank, are defined as follows:

Road Segment ID, Length of a road segment, Number of Lanes, Speed constraint, Direction, Level, Tortuosity, Number of connections, Schools, Companies & Offices, Banks & ATMs, Malls & Shopping, Restaurants, Gas stations & Vehicle services, Scenic spot, Hotels & Residences, Transportations, Entertainments & Living Services, sum of POIs.

1. *geospatial feature.txt* contains each road segment’s original geospatial features, such as length, speed limit and number of lanes. The schema is the same as “geospatial Y.txt”.
2. *road connection.txt* describes the structure of Beijing network. The columns, separated by blank, are defined as follows: Road segment ID, Start Node ID, End Node ID. The road network can be reconstructed based on the three entries. That is, if two road segments share a common node ID, then they are connected. The direction information of a road segment can be found in *geospatial feature.txt.* 0 means one-way, start node id to end node id. 1 means bi-directional.
3. *tensor.txt* is the tensor A in CATD consisting of real-time tensor Ar and historical tensor Ah, denotes the traffic conditions based on the taxi trajectories data over a long period of time (form Sep. 1st, 2013 to Oct 31st, 2013). Time slot 1-4 is tensor Ah and 5-8 is tensor Ar, where time slots 8 is the current time slot (8:30-9:00 on Oct. 25, 2013). Time slot 1-4 means 7:00-9:00 and so do time slot 5-8. Each line in this file stands for a driver on one road segment in one time slot format as “road segments ID, time slots, user ID, travel time”.

## Trajectory data

“*Sampled Trajectory data.txt*”, stored in “*Data/TrajectoryData*”folder, contains trajectories generated by over 30,000 taxicabs on Oct. 25, 2013, in Beijing. The time of day from 0 am to 11 pm is divided into 48 time slots, each of which lasts for 30 minutes. This is only one day of the trajectory dataset that were used to mine the frequent trajectory patterns (and further to estimate the travel time of a path in OC).

The schema of each trajectory is in a format of:

“*road segments ID, user ID, time slots, travel time*”.

Two trajectories are separated by a blank line.

## Test Data

Query paths, stored in “*Data/TestData*”folder, is the test data used as ground truth to verify our approach. This dataset contains two files.

1. *query path.txt* is the query paths generated from taxi trajectory data. (Please note to remove this data from trajectory data)
2. *in the field.txt* is in the field query paths recorded by two drivers carrying a GPS logger in Beijing.

Query paths in these two files are in a format of:

*First line: travel time*

*Second line: road segments sequence*

**Contact:**

Yu Zheng, [yuzheng@microsoft.com](mailto:yuzheng@microsoft.com)

Lead Researcher at Microsoft Research

<http://research.microsoft.com/en-us/people/yuzheng/>