

Code Description

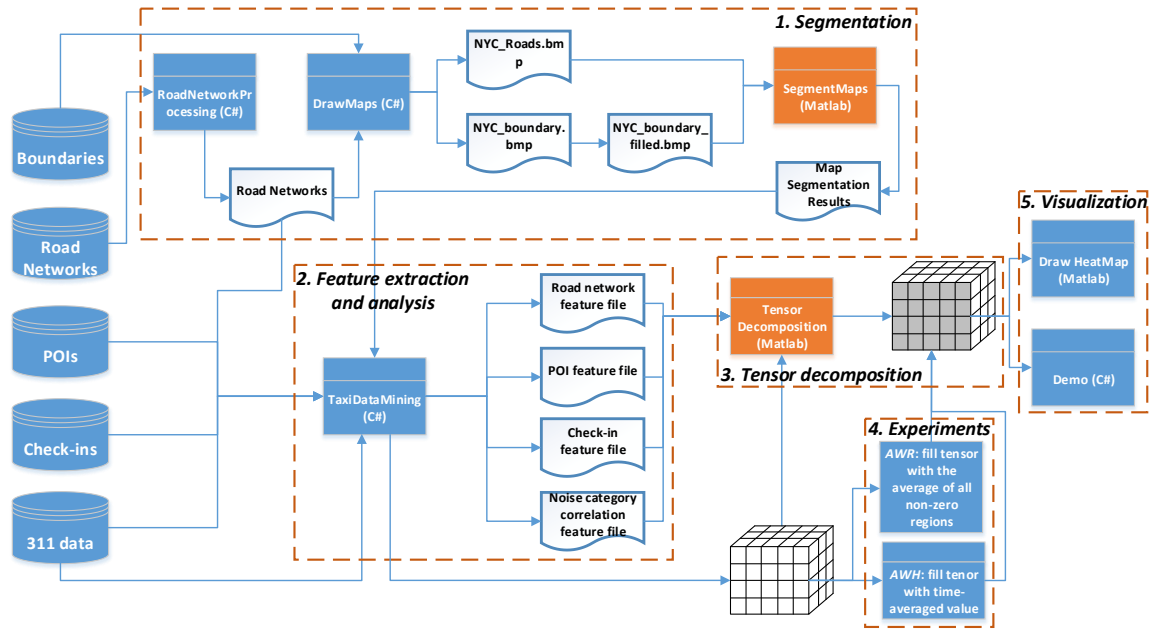
The document introduces a part of the codes that implement the research reported in [1]. The codes are organized into two parts, enclosed in two file folders and detailed respectively in later sections:

Please cite the following two papers when using the dataset.

[1] Yu Zheng, Tong Liu, Yilun Wang, Yanchi Liu, Yanmin Zhu, Eric Chang. [Diagnosing New York City's Noises with Ubiquitous Data](#). In Proceedings of UbiComp 2014.

[2] Wang, Y., Zheng, Y., Liu, T. [A noise map of New York City](#). In Proc. of UbiComp 2014.

The following figure shows the overview of all codes. The released codes are colored by orange.



1. Segmentation Maps:

The codes in this file folder are used to segment NYC map into disjointed regions according to its major roads (level ≤ 5). Each region is a basic spatial unit to diagnose noise according to 311 data of NYC in our following work. Major roads and the boundary of NYC are extracted from the original datasets, and drawn in bitmap files *NYC_Roads.bmp* and *NYC_boundary.bmp*, respectively. We colored the region in NYC black, and stored in *NYC_boundary_filled.bmp*.

1. The code *mapseg.m* is used to segment NYC according to *NYC_boundary.bmp*, *NYC_boundary_filled.bmp* and *NYC_boundary.bmp*. After that, each pixel is labeled with a number, representing the **region id** it belongs to. The final segmentation result, containing the region ids of all pixels, is stored in file *NYC_862.txt*.

2. Tensor decomposition

The codes in this folder are used to realize our method for inferring noise, which is named *context-aware tensor decomposition* method. We modeled noise with a 3-D tensor according to noise complaints' regions, time spans and categories. As 311 data is very sparse, a large number of elements in the tensor are vacant. We filled the tensor by exploiting tensor decomposition, and combined the features extracted from extra datasets. Noise tensor and feature matrices have been extracted from original datasets.

1. File folder *TensorData* encloses the text files of features extracted from original datasets.

2. ***TransformTxt2Mat.m*** reads the text files and transforms them by normalization. The results are saved as .mat files enclosed in ***TensorMat*** file folder.
3. ***TensorForExperiment.m*** runs tensor decomposition (TD) with four different settings: (1) TD without features, (2) TD plus spatial feature extracted POIs and road networks (TD+B), (3) TD+B plus check-in feature (TD+B+D), and (4) TD+B+D plus the feature of noise category correlations (TD+B+C+D). To evaluate the performance, we randomly choose 70% data as train data and other 30% data as test data. The error metrics of the results of TD are RMSE and MAE.
4. ***TensorForDemo.m*** obtains the result of tensor decomposition with all three features without choosing 70% data as train data. The result is stored in .mat files, in folder ***TensorResult***.

Contact:

Dr. Yu Zheng, yuzheng@microsoft.com

Lead Researcher at Microsoft Research

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