Efficient Trajectory Retrieval

Sole Team Member: Hasan Pourmahmoodaghababa
uID: u1255635

1 The problem

A spatial trajectory is a sequence of waypoints $T = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where x_i, y_i show the latitude and longitude of a moving object. Mostly there is a time stamp attached to each waypoint too. The problem I am going to address in this project is trajectory similarity problem from information retrieval perspective. Thus the task is: given a query trajectory we should return the most similar trajectory from data to the query. This is somehow a document-term task. I will consider two types of queries:

- First type is a pair $q = (T, k)$, where T is a trajectory and k is an integer. The search machine should return k most similar trajectories to T in an ordered way.
- The second kind is a pair $q = (T, r)$, where T is a trajectory and r is a positive real number showing the range we are willing to get an ordered list of trajectories that lie in distance at most r from T in data.

2 My approach

I will use the approach studied in the research paper [1], which is an information retrieval perspective; in fact, building a vector space model. In the following I have listed the paradigm I have in my mind:

1. First I will calculate the pairwise distance of all trajectories in data employing a famous distance in order to get a ground truth ranked list for each trajectory in terms of proximity.
2. I will use all data points as a query for evaluation purpose.
3. Assuming the dataset lives in a rectangular bounded region R, we will decompose R into a large number of grids, say N. Then we will map every trajectory in our training data to an N-dimensional vector both in binary and frequency-type way.
4. As trajectories pass through a small number of grids, our representation of a trajectory in the latent space would be sparse (think of terms in a document versus corpus) and so it seems reasonable to reduce dimension in some way.
5. I will use a technique suggested in [1] to do a dimensionality reduction and will preserve the operator as a matrix in order to apply for query trajectories too. I will also try to apply other dimensionality reduction techniques like PCA or LDA.
6. Euclidean distance or cosine distance will be used as a similarity measure to create ranked lists as well as for comparing the mapped query with our index.
7. Depending on time, I would try a landmark-based featurization method for trajectories to map trajectories to a low dimensional Euclidean space and do the mentioned tasks for this representation as well and compare the performance with above vectorization method. I think this will work a bit worse but more efficiently as we will not need to calculate a huge dimensional vector for a trajectory and then do a dimensionality reduction. Indeed, the landmark-based vectorization will efficiently map each trajectory to a low dimensional vector, say 20-dimensional, and will do the same for a query in linear time in terms of the number of waypoints of the trajectory.

3 Datasets

I will use 3 trajectory data sets for evaluation: Car-Bus dataset and Character Trajectories dataset from UCI Machine Learning Repository, and Geolife Trajectory data set (a fairly big data) released by Microsoft.

4 Contribution

My contributions are:

- Using different dimensionality reduction techniques,
- Apply a kernel-based similarity measures in addition to cosine and dot product similarity measures,
- Using different datasets as a benchmark,
- Utlizing binary vectorization of trajectories,
- Using term-frequency type of trajectory vectorization,
- Applying dynamic time warping as a ground truth distance
- Using landmark-based technique (if time allows).

5 Evaluation

Evaluation would be through effectiveness and efficiency. Therefore, the model and performance would be judged by computing accuracy, recall, average precision and normalized discounted cumulative gain, for example. My plan is to do these evaluation at least on 2 datasets.

6 Implimentation

I will implement the whole project in Python.

References

[1] Apostolos N. Papadopoulos. Trajectory retrieval with latent semantic analysis. In *SAC'08 March 16-20*, pages 1089–1094.