Experiment (3): prediction using a Markov Chain model

Introduction:

Based the previous experiment (extracting points of interest) we are going to conduct a number of experiments to predict next location and recommend new places.

The problem of predicting the next location of the user, based on history are commonly addressed by researches, we are working to develop the common techniques tackling the problem as a review in order to prepare a reference for comparison.

We used Markov chain model to predict the next state and we applied this to the POIs extracted from Geolife dataset, the results varied between users based on the number of POIs extracted from every user.

Problem definition:

Geolocation data is very important for bunch of applications, like GoogleNow and Foursquare, these applications are interested in the current location of the user, but predicting the next place the user would visit, will help the marketing campaigns, news feeds or traffic information to be more directive and in time, also for traffic congestion prevention, car sharing applications.

So the problem is how to find out the place where the user expected to visit given the current location and historical location data, in this experiment we used Markov chain model, as it is popular on predicting the next states given the current one, as we converted the GPS data into POIs(states) we can apply this model.

Methodology:

1- Data slicing:

After identifying the points of interest into a dataset, we divided it into training set 70% and test set 30%, the test set was extracting randomly (as cross validation) from the dataset, with the conservation of the transitions between states (POIs), we divided the dataset based on the daily sequence, as we considered states found in the same day as a single sequence of transitions between states (some other work divide the day into time separated periods).

For the extracted training set, we constructed a 2-column matrix of N rows, as the column one is a state, and column two is the state (place) where the user visited next, we did this modification to maintain the relations between the states without escaping any.

2- Markov model:

We used a Markov chain model, as it well recognized in solving stochastic processes, and sequential data, it's a statistical model consist of:

a. States: which are the points of interest.
b. Transitions probabilities: are the probability of moving from one point to another, as shown in Figure 1.

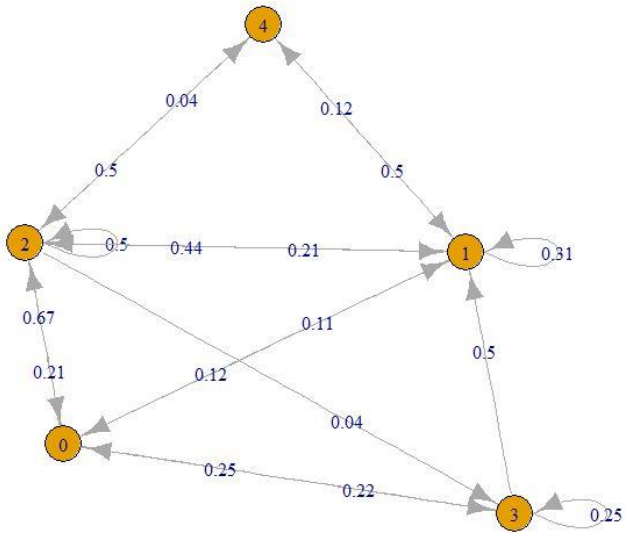

Figure 1 Sample of the Markov chain used on the dataset, the yellow nodes are the states (POIs) and the arrows are the transitions with probabilities upon.

The probability of being in state x depends on the previous states:

$$p(x) = p(x_l, x_{l-1}, x_{l-2}, \dots, x_1)$$

$$= p(x_l|x_{l-1}, x_{l-2}, \dots, x_1)\, p(x_{l-1}|x_{l-2}, \dots, x_1) \dots p(x_1)$$

The equation can be reduced to

$$p(x) = p(x_l|x_{l-1})\, p(x_{l-1}|x_{l-2}) \dots p(x_2|x_1)$$

Based on the first order Markov chain property, which states that the current state only depends on the previous state

$$p(x) = p(x_1) \prod_{i=2}^{L} p(x_i|x_{i-1})$$

3- Training the model:

Using the points of interest and the constructed sequences, we ought to construct the transition matrix of the Markov chain.

The state transition matrix was trained using the training sequences matrix by using maximum likelihood estimator (MLE), we used MLE to get the transitional probabilities between states, MLE maximizes the transitional probabilities which called the parameters (θ) of the Markov chain given the sequence matrix (D).

$$P(\theta|D)$$

First we calculate the probability of each state:

$$P(S_1) = \frac{S_1}{\sum_i S_i}$$

Then we calculate the probability of moving to state given another state for all states

$$P(S_2|S_1) = \frac{P(S_1, S_2)}{P(S_2)}$$

And probability of $P(S_1,S_2)$ can be calculated by the times S_2 followed S_1 in the whole sequence matrix.

4- prediction

After training the model, we can test its efficiency using the test set. We fetched a place (POI) from the test set with a consecutive place, then we set this location as an input for the Markov chain model. Using the transition matrix (P) and the vector of states (X), we applied the general rule of prediction:

$$x^n = x^{n-1}P$$

Results:

We tested the resulted place of the Markov model with the actual state found in the test set, then we count the matches and mismatches, to get the accuracy of the model

$$acc = \frac{n_T}{n_T + n_F}$$

Where:

n_T: Number of matches

n_F: Number of mismatches

User ID	Accuracy (%)	No. of matches	No. of mismatches
1	54.7	29	24
2	83.3	15	3
3	71.4	25	10
4	45.6	57	68
5	61.4	97	61
6	33.3	6	12
7	100	4	0
8	62.1	18	11
9	75	6	2
10	33.3	5	10

References:

[1] Gambs, Sébastien, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. "Show me how you move and i will tell you who you are."*Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*. ACM, 2010.