Data Preprocessing

March 14, 2019

1 Spatio-Temporal Data Mining

Dataset and Data preprocessing
Dr. Mitra Baratchi, Leiden University
Hossein A. Rahmani, University of Zanjan

1.1 Dataset Properties

The **original_data.csv** file includes the check-ins information by users. Each line of the files follows the following format (*tab* separated format):

user_ID POI_ID coordinate(atitude and longitude) checkin_time(hour:min) date_id
p.s., check-ins made on the same date have the same date_id and there are 151589 check-ins.

2 Data Preprocessing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format.

2.1 What does Data Preprocessing mean?

Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing.

In this file, we will read the data and make some data preprocessing. First, we read the check-ins data file and calculate the check-in of each user and location pair with a indicator function (1 if a POI is checked by user u, otherwise 0).

The CSV (Comma Separated Values) format is the most common import and export format for spreadsheets and databases. It is one of the interesting format for data. We import CSV in python to work with CSV files.

```
In [5]: import csv
```

Here, we create a **user_checkins** set to check the check-in of each user and location pair and **location_geo** dictionary to make each location geographical information.

```
In [9]: location_geo = dict()
        user_checkins = {}
        with open('dataset/original_data.csv', 'r') as original_data:
            # read the check-in CSV file
            check_ins = csv.reader(original_data, delimiter='\t')
            for check_in in check_ins:
                # add a location as key and the latitude and longitude as value
                location_geo[check_in[1]] = (check_in[2], check_in[3])
                # add user checkin
                if (check_in[0], check_in[1]) in user_checkins:
                    user_checkins[(check_in[0], check_in[1])] += 0
                else:
                    user_checkins[(check_in[0], check_in[1])] = 1
```

In the next step, we will save the preprocessed data into two files. Here, we make a file with three columns: **User_id**, **Location_id**, and **Checkin_frequency** for users' checkins and the second file as *geo data* to store the location information (i.e. **Location_id**,**latitude**,**longitude**).

```
In [7]: with open('preprocessed_data.csv', 'w', newline='') as preprocessed_data:
            checkins_writer = csv.writer(preprocessed_data, delimiter='\t')
            for checkin_info in user_checkins:
                checkin = [checkin_info[0], checkin_info[1], user_checkins[checkin_info]];
                checkins_writer.writerow(checkin)
```

```
In [8]: with open('geo_data.csv', 'w', newline='') as geo_data:
            geo_writer = csv.writer(geo_data, delimiter='\t')
            for lid, geo in location_geo.items():
                geo = [lid, geo[0], geo[1]];
                geo_writer.writerow(geo)
```

Now, if you have done everything correctly, you can see the **preprocessed_data.csv** and **geo_data.csv** beside of this file.

2.2 References

This data is used in the experiments of the following paper (originally, it is for the Foursquare location-based social network):
Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, Nadia Magnenat-Thalmann: ***Time-aware point-of-interest recommendation****, SIGIR, 2013*