International Journal of Science and Engineering Applications
Volume 5 Issue 6, 2016, ISSN-2319-7560 (Online)

Designing of Trajectory Similarity Calculation System Based on Latitude and Longitude

Le Wang
Shandong University of Science and Technology, Qingdao 266590, China

Lei Zhao
Shandong University of Science and Technology, Qingdao 266590, China

Cong Liu*
Shandong University of Science and Technology, Qingdao 266590, China

Abstract: This paper explains the algorithm of similarity calculation: K-means [1] clustering algorithm and Apriori algorithm, achieves the similarity calculation between two or more trajectories by using these two algorithms. Developing this system in the b / s structure model based on similarity calculation. This paper mainly describes system general design and system implementation. Among them, system general design includes the main business design of the system, system function module and database design; system implementation includes the setting of software and hardware and system function.This paper ends with a conclusion and prospect, pointing out problems and inadequacies, discussing the questions of the system functions, algorithm optimization and the like.

Keywords: similarity calculation; K-means clustering algorithm; Apriori algorithm; trajectory; b / s

1. INTRODUCTION

With the rapid development of mobile communication equipment facilities and positioning service, resulting in lots number of trajectories which based on the latitude and longitude, the vast number of data has attracted the attention of people. How to manage and use the information effectively has become a hot research currently. The trajectory model is a popular and promising approach which is used to find the similarity of the trajectory, that is also used to recommend interest points.

This paper implements the trajectory similarity calculation using the K-means clustering algorithm and Apriori algorithm, and displays trajectories visually by baidu gis map. Designing the trajectory similarity calculation system based on latitude and longitude in the b/s structure mode.

2. RELATED WORK

Ying et al. [2] propose a method to compare user similarity semantically on the level of frequent patterns. They use PrefixSpan to mine frequent patterns and develop a similarity measure, called maximal semantic trajectory pattern similarity (*MTP similarity*). Maximal trajectory patterns, or maximal patterns, are those patterns that are not contained in any other frequent patterns. In the MTP similarity measure, the comparison between users is based on the comparison between maximal patterns. *Chen et al.* [3] improve the MTP similarity measure by remedying a defect which is that when comparing two identical users using the similarity measure the similarity value is not necessarily one, and extend it to take temporal information into account. *Ruipeng LU* [4] proposes a new similarity measure called the *CPS-based similarity measure*, by directly comparing two users' frequent pattern sets instead of being based on the comparison of patterns. This measure eliminates the above-mentioned defects of the (improved) MTP measure.

3. ALGORITHM OF SIMILARITY CALCULATION

3.1 K-means Clustering Algorithm

The basic philosophy of K-means algorithm is taking *k* as a parameter, and dividing *n* object into *c* cluster, making the similarity in the cluster high while between clusters low. It uses the mean value of objects in each cluster to calculate the center of the cluster to make it as similarity standard. When the algorithm starts, it selects *c* point from all point as initial central point of each cluster, then iterates the residual point one by one, and entrusts its recent cluster with it, then revises the center of gravity of the cluster. Repeat like this, until the criterion function restraining.

The distance between each point is indicated by the distance between two points, which usually uses Euclid distance, and its formula is as equation (1):

$$D(i,j)=\sqrt{\left|x_{i1}-x_{j1}\right|^2+\left|x_{i2}-x_{j2}\right|^2+.....+\left|x_{ip}-x_{jp}\right|^2} \quad (1)$$

P is the number of the attributes of each point.

The value of the cluster center is the mean value of the cluster, its formula is as equation (2):

$$m_i=\frac{1}{n}\sum_{i=1}^{n_i}x_i \quad (2)$$

In the formula, *n* is the number of mid-value of the cluster.

Its computational process is as follows:

The input: the number of the clusters (k) and the number of its data object (n).

The output: *k* cluster, which make the criterion function the smallest.

(a) *K* data object are selected stochastically, and these *k* object initial represents the centre of each cluster.

(b) According to the mean value of the data object of the cluster, each data object is assigned to the nearest cluster.

(c) Renew the mean value of each cluster.

(d) The second and third one is executed repeatedly, until the criterion function is restraining.

3.2 Apriori Algorithm

Among the best known algorithms for association rule induction is the apriori algorithm. This algorithm works in two steps: In a first step the frequent item sets are determined. These are sets of items that have at least the given minimum support. In the second step association rules are generated from the frequent item sets found in the first step. Usually the first step is the more important, because it accounts for the greater part of the processing time.

In order to make it efficient, the apriori algorithm exploits the simple observation that no super set of an infrequent item set can be frequent .

The original pseudocode by Agrawal is offered in Algorithm1.

Algorithm 1. The Apriori algorithm.

(1) L_1 -{large 1-itemsets};

(2)**for** $(k = 2; L_{k-1} \neq \Phi; k++)$ **do begin**

(3) C_k = apriori-gen(L_k -1); *//New candidates*

(4) **forall** transactionst $t \in D$ **do begin**

(5) $C_t = subset(C_k, t)$; *//Candidates contained int t*

(6) **forall** candidates $c \in C_t$ **do**

(7) c.count++

(8) **end**

(9) $L_k = \{c \in C_k | c.count \geq \min \sup\}$;

(10) **end**

(11) $\text{Answer} = \cup_k L_k$;

In the algorithm *apriori-gen* is a function made up of two phases: union and pruning. In the union phase (see Algorithm 2), all k item sets candidates are generated.

Algorithm 2. Union phase of the Apriori.

insert into C_k

select $p.item_1, ..., p.item_{k-1}, q.item_{k-1}$

from $L_{k-1} p, L_{k-1} q$

where $p.item_1 = q.item_1, ..., p.item_{k-2} = q.item_{k-2}$,

$p.item_{k-1} < q.item_{k-1}$;

Now in the pruning phase (see Algorithm 3), which gives the name to the algorithm, all candidates generated in the union phase with some non-frequent (k-1) item set are removed.

Algorithm 3. Pruning phase of the Apriori.

forall item sets $c \in C_k$

forall (k-1)-subsets s of c **do**

If $s \notin L_{k-1}$ **then**

Delete c form C_k ;

4. OVERALL DESIGN

4.1 The Main Business Of the System

Firstly, the user logins the system. If the user does not have an account, the system will prompt the user to register in the system. After that the user can use the system properly. Secondly, the user needs to import trajectories into the system, and then selects two or more trajectories. Thirdly, the user can compare the similarity of trajectories to find useful rules and find similar trajectories. What' s more, the system will also recommend interest points to users according to two or more trajectories that users have already chosen. In the end, the user exits or logs off the system if the user has finished all operations.

The similarity calculation is the main business of the system that is ran automatically in the background. Above all, the data should be divided into k classes through k-means algorithm. Then, the frequent sequence is obtained according to apriori algorithm. Finally, the similarity calculation is achieved according to the concept of the longest common sub sequence .

4.2 System Function Module

This system can realize these functions: registration, login, exit, query and modify the user information, import the trajectory data, select the trajectory, view the trajectory through different ways (form ,text and baidu gis map), show the result of the similarity calculation and interest points and so on.

The structure of the main function module of the system is shown in *Fig.1*. The main function module of the system includes three parts: import data module, process data module and display data module. Import data module imports the trajectory to the database in order to use the data efficiently. Process data module clusters the trajectory, and calculates the similarity of the corresponding trajectory by using the similarity algorithm. Display data module displays the trajectory through baidu gis map or table. It's also shows the result of the similarity calculation to users in a friendly way.

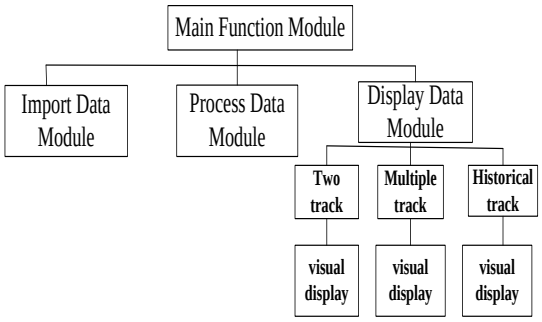

Figure 1. **Main Function Modules Of the System**

Import data module includes three parts: select data, read data and save data to database. The user will be alerted to select a file again, if the imported file format is wrong or the user does not choose a file.

The main function of process data module is to achieve cluster algorithm, and find the longest common sub sequence by using apriori algorithm.

Display data module is to show the similarity value and the trajectory information friendly.

4.3 Database Design

The system uses MySQL database, because MySQL database is multi thread, multi user operating SQL database server, and supports windows, Linux, UNIX, SUNOS etc. multiple operating system platforms, and is beneficial to transplant the system, could reduce pressure on the server. The database of this system is divided into four tables: User information table (user), trajectory file table (path), table associated with user table and trajectory file table(user_data), track detail information table (track_data). Among them, the user table stores user *ID*, user name, user password. Path table stores file *ID*, file name, file imported time. The user_data table stores the user *ID a*nd the file *ID* by associating the user table with the path table.Track_data table stores file *ID*, longitude, latitude, time, serial number, cluster mark, in which the path file *ID* and the user *ID* is auto-incremented .

5. SYSTEM IMPLEMENTATIONS

5.1 Software And Hardware Environment

This system runs on Windows 7. The following is the hardware and software environment of the system :

Hardware Environment:

CPU : Intel (R) Core (TM) -2450M CPU i5 @ 2.50 GHz

Memory: 4GB or above

Operating System: 64 bit

Broadband: 10.0M or above

Software Environment:

Compiler: MyEclipse 10.5

Java Running Environment: java 7.0

DataBase: MySQL 5.5

JSP Server: tomcat 6.0

5.2 System Function

The main function of the system is to obtain the longest common sub sequence through the frequent sub sequences that are based on the Apriori algorithm. The paper defines a method to calculate the similarity between two trajectories through the concept of the longest common sub sequence. The similarity between the two trajectories *P* and Q is defined as *sim(P,Q)*: $sim(P,Q) = \frac{2 \cdot lenLCS(P,Q)}{len(p) + len(Q)}$.

The *lenLCS (P,Q)* is the length of the longest common sub sequence of *P* and *Q*, *len(p)* is the length of the trajectory *p*. The *sim(P, Q)* is used to get the similarity of the trajectory. Finally, the system will present the trajectory map and the result of the similarity calculation to users in a friendly way.

6. SUMMARY

The structure of B/S (*browser/server*) was used in this system, because of its merits, which made the system developed easily, made the system more flexible, and made the maintenance and the management of system easier in the later stage.

The system has obtained the result of the user trajectory similarity calculation according to the K-means algorithm and Apriori algorithm. Meanwhile, for the result, this paper puts forward an recommendation mechanism to recommend interest points.

The design of the system can also be used to manage students to access to the Internet healthy and safe. As of way of learning and communicating for students, the internet and its management will play a vital role in students' physical and mental health as well as the improvement of their academic performance. Discovering the hiding and potential rules through computing the similarity of the Internet log for good students and bad students. Improving the poor performance of students in grades during the result of similarity calculation.

From an overall viewpoint of the system is successful, but also certain features of the complex operation, the user interface is also beautiful enough, the similarity calculation is not compared with other similarity calculations, these questions need to be improved in the future design and improvement. In the future work, we will try to find an efficient similarity calculation method through comparing with other various similarity algorithm. The results will be applied to calculate the similarity of trajectories and the similarity of the network log data more precisely.

7. REFERENCES

[1] Jiawei Hua, Michelline Kamber, Jian Pei. Data mining concepts and techniques: China Machine Press. 2012. 161-175. 288-306.

[2] Josh Jia-Ching Ying, Eric Hsueh-Chan Lu, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S_ Tseng. Mining user similarity from semantic trajectories. In Proc, International Workshop on Location Based Social Networks (GIS-LBSN), pages19-26. ACM, 2010.

[3] Xihui Chen, Jun Pang, and Ran Xue. Constructing and comparing user mobility profiles for location-based services. In Proc. 28th Annual ACM Symposium on Applied Computing (SAC), pages 261–266. ACM, 2013.

[4] Ruipeng LU. New User Similarity Measures Based on Mobility Profiles. ShanDong: ShanDong Univerity, 2014.