Analysis of Campus Crowd Gathering and Movement Patterns Based on Density Clustering Algorithms (Postprint)

Authors: Guo Yubin, Wu Yuhang, Zhou Zhefan, LI Ximing

Date: 2019-01-03T00:00:00+00:00

Abstract

This study conducts mining and analysis on wireless network log data from a university to obtain the distribution of crowd gathering points and movement patterns on campus. First, a distributed statistical algorithm is employed to count the number of wireless network connections for each building on campus; then an R-tree index is constructed for the latitude and longitude coordinates of building centroids, and the R-tree leaf nodes are grouped to partition the campus into several sections; next, a density clustering algorithm is utilized to cluster the latitude and longitude coordinates of building centroids within each campus section to obtain a regional division of the campus; finally, the clustering results and statistical results are combined to identify crowd gathering areas and inter-regional movement patterns. The research findings can provide references for school bus route planning, shared bicycle deployment, and campus functional zone planning.

Full Text

Preamble

Vol. 37 No. 3
Application Research of Computers
ChinaXiv Partner Journal

Analysis of Crowd Aggregation and Movement on Campus Based on Density Clustering Algorithm

Guo Yubin, Wu Yuhang, Zhou Zhefan, Li Ximing†
(College of Mathematics & Information, South China Agricultural University, Guangzhou 510642, China)

Abstract: This paper analyzes wireless network log data from a university to study patterns of crowd aggregation and movement on campus. First, distributed statistical algorithms are used to count wireless network connections per building. Then, an R-tree index is constructed for the central latitude-longitude coordinates of campus buildings, and R-tree leaf nodes are grouped to partition the campus into several sections. Density clustering algorithms are applied to the building center coordinates within each section to obtain a detailed campus area division. Finally, crowd aggregation areas and inter-area movement patterns are derived from the clustering and statistical results. The findings provide references for school bus route planning, shared bicycle deployment, and campus functional zone planning.

Keywords: wireless network; log data; R-tree; density clustering; crowd aggregation and movement

0 Introduction

With the rapid development of mobile internet technology, most universities in China have completed or are implementing comprehensive wireless network coverage. Consequently, campus wireless network log data has grown exponentially. At the authors' university, where dormitories and office areas have full coverage, authentication data alone reached 252,515,722 records (over 250 million) in the first half of 2018. These logs primarily contain information including mobile device MAC addresses, connection/disconnection times, connected wireless access point (AP) names, and disconnection reasons. Analyzing this data reveals patterns in campus crowd distribution and movement, which can inform decisions such as shuttle bus scheduling, shared bicycle deployment, route design, and campus functional zone planning.

Analysis of crowd aggregation and movement patterns represents a hot research topic in the big data era. Reference [1] proposed a method for mining points of interest and travel planning from GPS trajectories, discovering ten major points of interest and travel patterns between them using individual location history data. Reference [2] analyzed six months of trajectory data from numerous mobile phone users to uncover spatiotemporal regularities in human mobility. Reference [3] examined massive GPS trajectory data from private vehicles to identify frequent travel modes and predict traffic hotspots and congestion likelihood. Beyond trajectory data, some studies utilize Wi-Fi network logs to analyze crowd aggregation or movement patterns. Reference [4] established a mobility model from Wi-Fi logs, deriving movement speeds between points using device connection/disconnection times to extract user mobility characteristics and accurately describe movement behavior.

Clustering is an effective analytical method for studying crowd aggregation and movement patterns. Reference [5] employed clustering techniques to perform two-stage clustering on user trajectory data from spatial and temporal dimen-

sions, thereby constructing user interest regions. Reference [6] used DBSCAN (density-based spatial clustering of applications with noise) to extract and analyze crowd aggregation hotspots, finding high coverage between hotspots in the same time period but significant differences between different time periods. In addition to clustering algorithms, researchers have also used neural networks [7] and statistical methods [8] to discover crowd movement patterns.

Some scholars have specifically studied crowd aggregation and movement patterns on campuses. Reference [9] applied DBSCAN to trajectory data from Wuhan University students to extract aggregation points and analyze activity time distributions. While this work provides a reference for student behavior research using GPS data, its limitation lies in the small and unrepresentative dataset from volunteers. Reference [10] analyzed Wi-Fi network logs from Concordia University in Montreal to identify network user activity types within buildings, such as attending classes, office work, and network usage in public areas, and proposed a search algorithm to associate the same activity types across multiple days. This study focused on a small movement range within buildings without considering inter-building movement.

Compared with trajectory data, campus wireless network logs offer larger volumes and greater stability, better reflecting crowd aggregation and movement. This paper processes and analyzes wireless network logs from a university. First, a distributed statistical algorithm based on the MapReduce computing model counts wireless network connections for each building' s central latitude-longitude coordinates. The statistical result represents the total connection count for all wireless APs in each building. Then, an R-tree index is constructed for all building center coordinates, and R-tree leaf nodes are grouped to roughly partition the campus into different sections. Density clustering algorithms are then applied to the center coordinates within each section to obtain a detailed campus area division. Finally, crowd hotspot distribution areas and inter-area movement patterns are derived by combining clustering results with statistical results. This paper uses R-tree indexing to address the dynamic changes in center coordinates, enabling long-term, continuous processing of campus network log data without being affected by network expansion or AP instability, thus increasing experimental flexibility. Additionally, implementing core algorithms using the MapReduce computing model improves big data processing efficiency. [Figure 1: see original paper] shows the overall algorithm flowchart.

1.2 DBSCAN Algorithm

DBSCAN is a density clustering algorithm that discovers clusters of arbitrary shapes using high-density connectivity [12]. The basic idea is that for each object in a cluster, its neighborhood within a given radius must contain at least a minimum number of objects. Let the sample set be a positive integer, with

the following definitions:

a) The -neighborhood of a point contains all samples in the dataset whose distance to the point is not greater than .

b) A core object is a point whose -neighborhood contains at least minPts samples.

c) A border point is not a core point itself but lies within the -neighborhood of another core point.

d) A noise point is any point that is neither a core point nor a border point.

e) Directly density-reachable: point q is directly density-reachable from point p if q lies within p' s -neighborhood and p is a core object.

f) Density-reachable: point p is density-reachable from point q if there exists a sequence of samples where each point is directly density-reachable from the previous one.

g) Density-connected: points p and q are density-connected if there exists a core object sample from which both p and q are density-reachable.

Based on these definitions, the DBSCAN algorithm proceeds as follows:

a) Select any sample P from dataset D and mark P as read.

b) If P is a core object, find all density-reachable points in P's -neighborhood.

c) If P is a border point with no density-reachable objects, temporarily label P as noise.

d) Repeat steps a-d until all objects in D are marked as read.

e) Find the maximum density-connected object set for all directly density-reachable points in the -neighborhood of core objects. Repeat until all core object neighborhoods have been processed.

DBSCAN offers strong advantages including noise resistance and the ability to discover clusters of arbitrary shapes [13]. Its drawbacks include the need to identify all density core objects and the requirement to manually determine parameters and minPts, which introduces human error. Parameters and minPts significantly affect clustering quality: if is too large, most points cluster into one group; if is too small, clusters split. If minPts is too large, points in the same cluster may be marked as non-core objects; if minPts is too small, too many core objects are discovered. To address these limitations, scholars have proposed many improved algorithms, such as M-FDBSCAN [14] and HDBSCAN [15].

1.1 R-Tree

An R-tree is a hierarchical, height-balanced multi-level data structure that naturally extends B-trees to multi-dimensional data space [11]. Each R-tree node corresponds to a minimum bounding rectangle (MBR) that encloses the minimal spatial range of all child nodes. [Figure 2: see original paper] shows an R-tree example, where (a) illustrates data item distribution with solid-line boxes representing MBRs of spatial objects (which can be two-dimensional coordinates such as latitude-longitude, representing the maximum range of several 2D shapes), and dashed boxes representing index spaces of intermediate node entries. [FIGURE:2(b) shows the corresponding R-tree structure.

Assuming m (2 m M/2) is the minimum number of index items (data items) a node must contain, an R-tree must satisfy the following properties:

a) All intermediate nodes except the root contain between m (m = M/2) and M index records.

b) The root has at least two leaf nodes unless it is itself a leaf node.

c) Each leaf node contains between m (m = M/2) and M data items.

d) All nodes require the same storage space.

e) All leaf nodes are at the same level.

2.1 Source Data Description

The experimental data comes from wireless network equipment logs at a university, comprising 39,478,898 records. describes the wireless network log structure.

In , the SEQ field is a unique log identifier, DATE indicates the log generation date, TIME specifies the exact generation time down to microseconds, and the MESSAGE field contains log content including mobile device MAC addresses, connected wireless AP names, and device connection status information (online, offline, or roaming). Additionally, all wireless AP information is summarized in .

includes AP_NAME (wireless AP name), AP_MAC (unique MAC address identifier), CEN_GPS (building center point latitude-longitude coordinates), and BUILDING (building name). In this table, since wireless APs within the same building are relatively close and numerous, the building' s center point coordinates represent all APs' actual coordinates, simplifying building representation while marking building locations.

2.2 Data Preprocessing

Data preprocessing primarily involves selecting logs from the study period, removing erroneous data based on error tags in the MESSAGE field, and extracting and saving each device' s location change information. Erroneous data includes authentication errors and wireless AP equipment failures. Location change information is exemplified in , which shows records for mobile device 0000.00d1.ab46 on March 31, 2017, indicating connections at the Foreign Languages College at 07:56:44, Life Sciences College at 08:18:40, and Forestry College at 14:34:04. Since only the first record is retained when MAC, DATE, and CEN_GPS fields are consecutively identical, effectively stores daily CEN_GPS changes, corresponding to user location changes throughout the day.

The specific preprocessing steps are:

a) Save wireless network log data and wireless AP information tables to HDFS.

b) Use regular expressions to extract mobile device MAC addresses and connected building names from the MESSAGE field, denoted as MAC and BUILDING respectively. Match building center coordinates from the building name and AP table, denoted as CEN_GPS. Add DATE and TIME fields and save all information to a temporary table.

c) Join the wireless AP table with the temporary table on CEN_GPS, then sort by MAC, DATE, and TIME. Retain only the first record where MAC, DATE, and CEN_GPS are consecutively identical to obtain the preprocessed result.

3 Experiments and Results Analysis

Experiments were conducted on a Hadoop cluster consisting of one master node and three slave nodes, each configured with an i7-8700K CPU and 8GB RAM, running CentOS 7.0 and Hadoop 2.7.

The experiments consist of two processes: person-time statistics and density clustering. In the statistics phase, since preprocessed data maps all AP coordinates within a building to the building' s center coordinates, calculating connection counts for each center coordinate yields the total connections for all APs in that building. In the clustering phase, an R-tree index is first built for all center coordinates. Based on the school' s functional zone division, R-tree leaf nodes are grouped into seven groups (A-G), with duplicate nodes assigned to the nearest group to avoid redundant results. DBSCAN clustering is then applied to each group, using parameters = 863.92 and minPts = 3 based on

reference [16]. The study area is divided into 10 clusters, with group and cluster distributions mapped to match actual conditions, as shown in [Figure 4: see original paper].

presents clustering results and connection count matching across different time periods. Groups A, B, F, G, and H each yield one cluster (A1, B1, F1, G1, H1), while groups C, D, and E each produce two clusters (C1, C2; D1, D2; E1, E2). Statistics include monthly totals, daily averages, and daily averages for morning (6:00-12:00), noon (12:00-14:00), afternoon (14:00-18:00), evening (18:00-24:00), and midnight (00:00-6:00) periods.

Daily average data shows clusters C1, E1, G1, D1, and B1 have higher person counts, indicating dense crowd distribution. Morning aggregation areas are C1, E1, G1, and D1; noon aggregation areas are C1, E1, G1, and D1; afternoon dense areas are E1, C1, and G1; evening aggregation areas are C1, G1, E1, D1, and B1; midnight aggregation areas are D1, B1, and G1.

Inter-area movement analysis reveals the largest flow between E1 and C1. Other significant flows occur between A1-A2, C1-D1, and B1-G1. Regions with large internal movement disparities include D1 (difference of 132) and C1 (difference of 106).

and [Figure 5: see original paper] show statistics for each area' s movement frequency. Areas E1, A1, B1, C1, D1, and G1 exhibit frequent movement, with E1 being the most active (997 inbound, 1047 outbound). C1 ranks second (926 inbound, 1032 outbound), followed by D1, G1, and A1.

Morning and afternoon movement patterns are detailed in and . Morning movement is most frequent in C1 (926 inbound, 657 outbound), with internal movement calculated as 2,294 (65.58% of total morning movement). E1 follows (507 inbound, 707 outbound, 2,127 internal, 63.66%). Afternoon patterns show C1 and E1 remain the most active, with internal movements of 1,387 (60.01%) and 1,638 (62.54%) respectively.

These findings indicate that C1 and B1 (dormitory areas), E1 (central area with library, administration, and teaching buildings), and D1 (mixed dormitory and teaching area) are primary crowd aggregation zones, matching actual conditions. Recommendations include adding entertainment facilities, shops, expanded fire exits, and enhanced security in aggregation areas. For movement patterns, frequent-movement areas like E1, A1, B1, C1, and D1 could benefit from increased traffic safety reminders, more shared bicycle deployments, and additional parking areas. Bus routes or bicycle lanes could be designed along connectivity patterns such as A1 A2 E1 D1 and G1 B1 E1 C1, with increased shuttle frequency for high-traffic periods.

4 Conclusion

This paper statistically analyzed and clustered wireless network log data from a university to obtain campus area divisions, crowd aggregation patterns, and movement regularities, providing references for both non-commercial and commercial activity planning. The approach involves: (1) removing irrelevant data and extracting device location changes; (2) counting connections per building center coordinate; (3) building an R-tree index for all center coordinates, grouping leaf nodes, and applying density clustering to each group; (4) matching cluster center coordinates with statistical results to obtain connection counts per cluster and calculating inter-cluster movement volumes.

The analysis provides valuable references for school bus routing, shared bicycle deployment, and campus functional zone planning. Using R-trees improves processing efficiency and allows dynamic addition or removal of wireless APs, enhancing experimental flexibility. Future work will track individual movement patterns and combine them with academic performance and internet usage data to analyze student campus behavior.

References

[1] Zheng Y, Zhang L, Xie X, et al. Mining interesting locations and travel sequences from GPS trajectories [C]// Proc of the 18th International Conference on World Wide Web. New York: ACM Press, 2009.

[2] González M C, Hidalgo C A, Barabási A. Understanding individual human mobility patterns [J]. Nature, 2008, 453(7196): 779-782.

[3] Giannotti F, Nanni M, Pedreschi D, et al. Unveiling the complexity of human mobility by querying and mining massive trajectory data [J]. VLDB Journal, 2011, 20(5): 695.

[4] Kim M, Kotz D, Kim S. Extracting a mobility model from real user traces [C]// Proc of IEEE International Conference on Computer Communications. Piscataway, NJ: IEEE Press, 2006: 1-13.

[5] Ji Yali, Gui Xiaolin, Dai Huijun, et al. Constructing user' s interest regions with two steps for trajectory privacy protection [J]. Chinese Journal of Computers, 2017, 40(12): 2734-2747.

[6] Zhang Wenxing. Analyze and Predict the hot spot regions [D]. Yinchuan: Ningxia University, 2017.

[7] Liao L, Patterson D J, Fox D, et al. Building personal maps from GPS data [J]. Annals of the New York Academy of Sciences, 2006, 1093(1): 249-265.

[8] Qi Jiaqian. Crowd behaviors analysis and abnormal trajectory detection based on surveillance data [D]. Beijing: Beijing Jiaotong University, 2018.

[9] Du Shenglan, Li Feng, Huang Changqing, et al. Trajectory-based activity pattern analysis of Wuhan University' s students [J]. Journal of Geomatics, 2017, 42(1): 91-95.

[10] Poucin G, Farooq B, Patterson Z. Activity patterns mining in Wi-Fi access point logs [J]. Computers Environment & Urban Systems, 2018, 67: 55-67.

[11] Guttman A. R-trees: a dynamic index structure for spatial searching [C]// Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM, 1984: 47-57.

[12] Zhang Wenyuan, Tan Guoxin, Zhu Xiangzhou. Application of stay points spatial clustering in hot scenic spots analysis [J]. Computer Engineering & Applications, 2018, 54(4): 263-270.

[13] Liu Shufen, Meng Dongxue, Wang Xiaoyan. DBSCAN algorithm based on grid cell [J]. Journal of Jilin University: Engineering and Technology Edition, 2014, 44(4): 1135-1139.

[14] Erdem A, Gündem T I. M-FDBSCAN: a multicore density-based uncertain data clustering algorithm [J]. Turkish Journal of Electrical Engineering & Computer Sciences, 2014, 22(1): 143-154.

[15] Melvin R L, Xiao J, Godwin R C, et al. Visualizing correlated motion with HDBSCAN clustering [J]. Protein Science, 2018, 27(1): 62-75.

[16] Lai Liping, Nie Ruihua, Wang Jiangping, et al. Improved DBSCAN algorithm based on MapReduce [J]. Computer Science, 2015, 42(S2): 396-399.

Note: Figure translations are in progress. See original paper for figures.