Keyword-based Community Search Over Spatial-Social Networks

Halie Eckert (hec171@kent.edu), Gavin Hulvey (ghulvey@kent.edu), Sydney Zuelzke (szuelzke@kent.edu)

Advisor: Dr. Xiang Lian (xlian@kent.edu)

Department of Computer Science, Kent State University
Kent, Ohio 44242

Abstract

A spatial-social network is a combination of spatial road network and social network, where users in the social network check in at some points of interest (POIs) on the road network. Recently, spatial-social networks have gained much attention from both academia and industry, due to its wide spectrum of real-world applications such as the study of user behaviors in social networks, online advertising and marketing, and so on. Therefore, it has become increasingly important to design and build a system for efficiently and effectively managing, querying, and visualizing spatial-social networks.

In this project, we tackle a useful problem, keyword-based community search over spatial-social networks (KCS-SSN), which retrieves a community of users who are constrained by query keywords (w.r.t. user interests/preferences and POI keywords) and have high social and spatial cohesiveness (e.g., constraints of k, d-truss and average road-network distances). We propose efficient and effective algorithms to enable fast retrieval of community answers, and design a user-friendly graphical user interface (GUI) to interact with users.

Introduction

- Combine spatial road data and social network data with users sharing common keywords and points of interest (POIs)
- Keyword search problem with a graphical user interface application
- Built with Python
- Allow users to input query constraints
- Several algorithms that take into account: physical closeness, degree of interest, relationships with POI keywords, relations between other users with keywords, and
- Large datasets

Contributions

- Further research on effective programs for community query searching
- Provide a user-friendly interface to plot, visualize, and analyze spatial real networks
- Focus on keyword-based community searching

Problem Definition

- Large gaps in data when only considering user relationships
- Keyword-based Community Search Over Spatial-Social Networks: a method to retrieve a community of users within a constraint of query keywords and cohesiveness
- Considers physical closeness, degree of similarity, relationship of POIs, keywords, and multiple user login locations
- Each user can have more than one login location
- GUI must allow user interactivity (Figure 1)
- Community Search Problem: Must take use of road networks and social networks to search for communities (Figure 2)

Spatial-Social Network Model

Dataset Sizes

- Users
- Relations
- Keywords

Figure 1: A Summary Query View

Figure 2: A Social and Road Network Overview

Future Work

- Performance could better be better optimized
- Use a SQL database instead of CSV files for datasets
- Add more options for querying communities
- Modify the interface to be more user-friendly and attractive
- Allow for more interaction
- Allow for customizing datasets to see relationships better

Acknowledgements

Thank Kent State University and the Choose Ohio First program for the funding and opportunities that provide to students such as ourselves to be able to explore research works.

Solutions

Distance Problem Solution

To accommodate for users having multiple check-in locations, we use a formula to calculate average distances between users.

Querying Road Networks

When querying road networks, we return users who:

A. Have visited at least 1 POI with 1 keyword in common
B. Have an average real path distance less than user-defined r

To find the direct paths, we implement the Dijkstra algorithm. After these constraints are satisfied, we return the community of users.

Querying Social Networks

When querying social networks, we return users who:

A. Satisfy the k, d-truss - User-defined values k and d are used to control community cohesiveness and diameter respectively
B. Have an degree of similarity score greater than user-defined η
C. Have at least 1 keyword in common

To find the direct paths, we implement the Dijkstra algorithm. After these constraints are satisfied, we return the community of users.

Results

Speed

- Combination of multi-thread data chunking and offline precomputation
- Startup speed was brought down to 7.21s. Initially start times were well over 30 seconds.
- Querying and graphing is also significantly faster, with relatively little change between commands.

Displaying Datasets

- Displaying datasets as a timely and meaningful manner was achieved by PyQGraph and Pyvis
- Pyvis was used for the summary of relationships for interactive nodes and edges
- PyQGraph was used to display the large datasets faster and with more simplicity
- sklearn for summary graph clustering

Querying Communities

- The California road network is used
- Users satisfying the user defined k, d, and η values are displayed, on the real network (Figure 1 right chart) as well as in an interactive network (Figure 1 left chart).
- Query execution time increases the more broad the user defined parameters are. It is recommended that queries start small with gradual increases.