GESONGEN: An Interface for Generating and Visualizing Geosocial Networks

Julius Hoffmann[1 [0009-0007-2053-7904]], Panagiotis Bouros[1 [0000-0002-8846-4330]], and Theodoros Chondrogiannis[2 [0000-0002-9623-9133]]

[1] Johannes Gutenberg University Mainz, Germany
jhoffm09@students.uni-mainz.de, bouros@uni-mainz.de
[2] Norwegian University of Science and Technology, Trondheim, Norway
theodoros.chondrogiannis@ntnu.no

Abstract. The ubiquity of mobile location-aware devices and the proliferation of social networks gave rise to *geosocial networks*, where users not only form social connections but also perform geo-referenced actions. Examples include traditional social networks with geo-annotated posts, e.g., Twitter, and networks that directly offer geosocial services, e.g., Yelp. However, despite the strong interest by both the industry and academia in geosocial networks, a limited number of datasets are in fact publicly available. To fill this gap, we present GESONGEN, an interactive interface for generating geosocial networks. We built upon a recently proposed generation process which combines independently generated graphs and geospatial data or re-uses existing datasets. GESONGEN can visualize geosocial networks, with different view options, and to modify networks.

Keywords: Geosocial networks · generator · visualization.

1 Introduction

The proliferation of location-based services and social networks gave rise to *geosocial networks*, which model both the social interactions of users and their geo-referenced actions, e.g., check-ins. Examples are typical social networks extended with geospatial information such as X (formerly, Twitter) and Facebook, and networks directly offering geosocial services such as Yelp and Foursquare.

Despite the interest from the research community and the industry on query processing [3,7,17], indexing [20,21], recommender systems [15,19] and on tasks such as influence maximization [5,16] and community search [6,9], a limited number of geosocial networks are in fact publicly available. For instance, Yelp[3] offers an official dump for academic purposes, and SNAP's page[4] offers a Brightkite and a Gowalla dump. Another option for acquiring geosocial network datasets is to use official APIs e.g., from X[5] and Foursquare[6], similar to [15,19][7]. However,

[3] https://www.yelp.com/dataset/
[4] http://snap.stanford.edu/data/index.html#locnet
[5] https://developer.twitter.com/en/products/twitter-api
[6] https://location.foursquare.com/developer/reference/places-api-overview
[7] https://archive.org/details/201309_foursquare_dataset_umn

these APIs restrict the number of requests per day; for unlimited downloads, fees are charged. A common practice to deal with the limited availability of real datasets is to generate synthetic ones. Synthetic geosocial networks can be used for benchmarking the efficiency and the robustness of geosocial queries, for hypothesis testing, "what-if" scenarios, and simulations.

Existing generators. Network and spatial data generation have individually received significant attention in the past. For the first, the goal is to generate synthetic networks whose properties match the ones in real networks. Real-world social networks in particular, typically exhibit a vertex-degree distribution that follows a power law, and a small diameter ("small-world" phenomenon, or "six degrees of separation"). Under this, the majority of the proposed models [1,8,23,24] use some form of preferential attachment to progressively construct a synthetic network, adopting a "rich get richer" approach. For spatial data, Beckmann and Seeger [4] presented a generator that was later extended to build Spider [12,22].

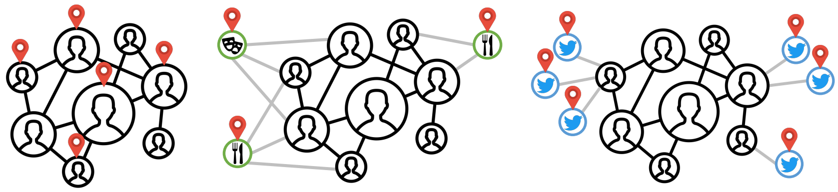
In contrast, generating geosocial networks has received limited attention. One approach is to leverage geo-simulation by creating a digital twin of a real urban environment and then use co-location between agents to form social connections [11,13,14]. Alizadeh et al. [2] adopted a different approach, modifying the generation process of the Erdős-Rényi [8], Barabási-Albert [1] and Watts-Strogatz [23] models so that geographic information dictates how network vertices are connected. Gallagher et al. [10] extended this approach. The vertices are no longer randomly positioned in space; instead the distribution of real geospatial data is used. Lastly, Sarsour et al. [18] proposed to fully decouple the generation of the two data components. The authors presented three types of synthetic networks for different real-world geosocial networks (see Section 2) and a generation process that combines the output of a graph and a spatial data generator.

Contributions. We present GEoSOcial Network GENerator (GESONGEN)[8], an interactive interface for geosocial networks with a threefold mission. First, our system enables users to generate synthetic geosocial networks by building on the generation process in [18]. Second, users can visualize generated geosocial networks; GESONGEN offers three view options, two that prioritize the social or the spatial component, and a third, combined view option. Finally, users can also modify a network by adding, removing, or editing vertices and edges.

2 Synthetic Geosocial Networks

A *geosocial* network is a labeled graph $G = (V, E)$ where every edge $(u, v) \in E \subseteq V \times V$ represents a relationship between the entities modelled by vertices u and v. The set $V_s \subseteq V$ contains *spatial vertices*, i.e., those are associated with a geometry $v.geom$ in the two or three-dimensional space , e.g., a point or a polygon. We revisit the three types of synthetic geosocial networks introduced by Sarsour et al. [18] which are fully supported by GENSOGEN. Fig. 1(a) illustrates the first type, denoted by G_s. All vertices represent the same type of entity, e.g.,

[8] https://gesongentool.github.io and https://github.com/pbour/geosocialgenerator



(a) Type G_s (b) Type G_c (c) Type G_p

Fig. 1. Types of synthetic geosocial networks: non-spatial vertices and edges in between them are drawn in black, spatial vertices are marked by a red pin. Without loss of generality, we draw only undirected edges.

users of the network, and the edges model relationships between such entities, e.g., *FRIEND_OF*. Some vertices are also spatial (marked by a red pin). An academic geosocial network created on co-authorship is an example of G_s type, where geospatial information models the affiliation of a researcher.

Fig. 1(b) depicts the second type of geosocial networks. G_c models two disjoint sets of entities as vertices. Specifically, the network comprises a set of non-spatial vertices, i.e., social vertices, and the edges connecting these vertices that model social relationships. Spatial vertices can be connected to one or more social vertices, but never to each other. The social vertices and the edge between them are in black, spatial vertices in green, also marked by a red pin, and the edges connecting social and spatial vertices in gray. Foursquare is an example of the G_c type with gray edges capturing *CHECKED_IN* actions by the users.

The third type of network G_p, is shown in Fig. 1(c). Similarly to G_c, G_p also models two disjoint sets of entities using social and spatial vertices. But, in contrast to G_c, each spatial vertex is always connected to exactly one social vertex. The social vertices and edges of the network are again in black, and the spatial vertices, in blue with a red pin. Social networks such as X or Facebook are examples of G_p networks, where the spatial vertices represent geo-annotated posts (namely, tweets in X). Naturally, as users make multiple posts, a non-spatial vertex can be connected to multiple spatial ones.

3 The GESONGEN System

GESONGEN adopts a typical client-server architecture, illustrated in Fig. 2. The frontend, a Web-based application developed in React, is responsible for collecting the parameters required to generate a synthetic geosocial network, and for visualizing and modifying networks. Visualization is powered by the D3.js Javascript Library and its d3-force-3d extension.[9] The backend, written in Python, uses Flask to provide a lightweight API that handles the requests between the two system components,[10] and is responsible for generating the

[9] https://d3js.org and https://github.com/vasturiano/d3-force-3d
[10] https://react.dev and https://flask.palletsprojects.com/

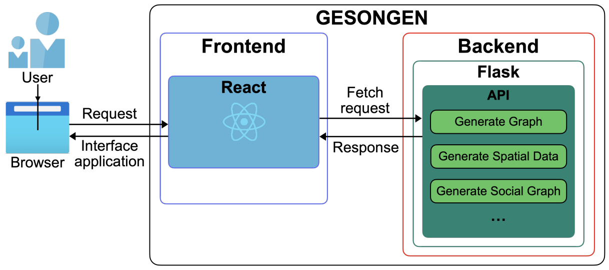


Fig. 2. GESONGEN architecture

networks according to the process in [18]. We use graph generators provided by NetworkX and the Spider spatial generator.[11]

The two system components communicate by exchanging `.gr` and `.co` files. Specifically, the backend stores the graph of the generated network inside a `.gr` file, and its spatial vertices inside a `.co` file. These files are then sent to the frontend using the Flask-based API for visualization. If the user decides to re-use the graph of an existing social network or an existing collection of geospatial objects, the frontend also sends a `.gr` or a `.co` file to the backend, respectively. Figures 3 and 4 detail the two file types. the first lines specify important metadata, i.e., the number of vertices and edges and the number of spatial vertices along with the type of their geospatial information (currently points or boxes). The rest of the lines in the `.gr` file list the edges with an optional label for each, while in the `.co` file the geometries of the spatial vertices.

GESONGEN is designed with extendability in mind. We can replace or extend the generators in the backend, by incorporating other graph or spatial data libraries. As long as the generated network is stored inside a `.gr` and a `.co` file, the system can operate as described above. We only need small changes in the frontend to modify or include new user parameters for the generation process.

3.1 Frontend

We first elaborate on the frontend. Fig. 5 exemplifies GESONGEN's main screen comprising two frames. The left frame is further split into two subframes. The top subframe lists the necessary input for generating a new geosocial network:

(1) the type of the synthetic network, i.e., one of G_s, G_c, G_p;
(2) the nature of the social and the spatial component, i.e., generated or uploaded by the user via a `.gr` and `.co` file - in the first case, the user provides the values for the generation parameters;
(3) the parameters for the combiner to be used (see Section 3.2).

The bottom left subframe and the entire right frame visualize a network.

Visualization options. GESONGEN offers three visualization options, called *views*. The *social* view (as the word suggests) prioritizes the social component

[11] https://networkx.org and https://spider.cs.ucr.edu

```
{number of vertices} {number of edges}
{source vertex id} {target vertex id} {label (opt)}
...
```

```
5 6
1 2 FRIEND_OF
2 3 FRIEND_OF
1 3 FRIEND_OF
1 4 CHECKED_IN
2 4 CHECKED_IN
2 5 CHECKED_IN
```

Fig. 3. Format (top) and an example (bottom) of a `.gr` file

```
{number of spatial vertices} {type: point or box}
{vertex id} {geometry}
...
```

```
2 point
4 0.9366755118414897 0.4555963756560951
5 0.30657038904474687 0.6100456763118138
```

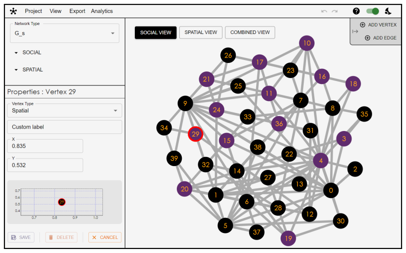
Fig. 4. Format (top) and an example (bottom) of a `.co` file

of the network. The right frame visualizes the network graph; the user is allowed to select the colors for the vertices (both spatial and non-spatial), and the edges. The left frame (bottom subframe) is used to partially display geospatial information of the network, i.e., for the spatial vertices selected by the user. Fig. 5(a) exemplifies the social view. The second visualization option, called the *spatial* view, prioritizes the spatial component of the network. The right frame now plots the geometries of all spatial vertices, while the left subframe is used to partially display the social this time information. Specifically, we draw for the selected spatial vertex its directly connected vertices of the network. Fig. 5(b) provides an example of the spatial view. Finally, the third option - the *combined* view, visualizes both components of the geosocial network at the same time, as the name suggests. The combine view defines two layers; the top layer displays the social vertices of the network and their edges, while the bottom layer plots the spatial vertices. Fig. 5(c) exemplifies the combined view.

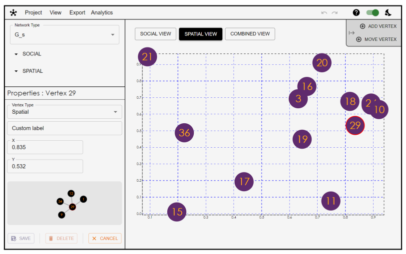
Statistics. GESONGEN provides evidence on how realistic generated geosocial networks are. We consider statistics previously used in [11] for the same purpose, i.e., the vertex degree and its distribution histogram, the diameter of the network and the number of contained triangles. Fig. 6 exemplifies the statistics screen.

3.2 Backend

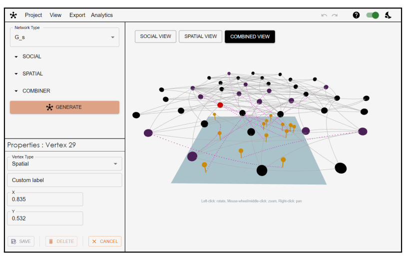
To generate a synthetic geosocial network, we combine a social graph with a collection of spatial objects. The two data components can be either generated or provided by the user. Therefore, we developed a *combiner* module for each network type in Section 2. Fig. 7 shows the generation process. For simplicity, assume that both components are generated. The backend receives the number of vertices in the social network, the edges type (i.e., directed or undirected) and



(a) Social view



(b) Spatial view



(c) Combined view

Fig. 5. Frontend, visualization options; spatial vertices in purple

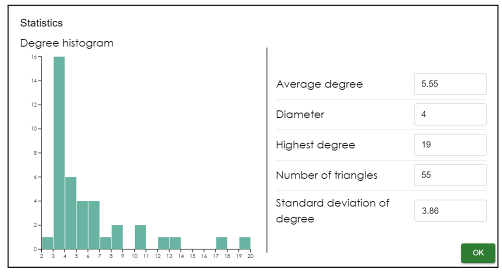


Fig. 6. Frontend, network statistics

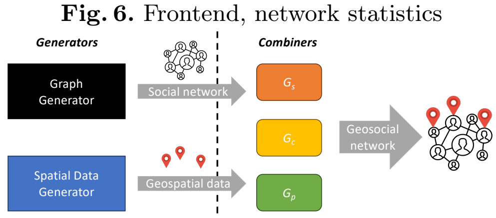


Fig. 7. Backend, generation process

their labels, and the number of geometries (i.e., the number of spatial vertices to be created). The remaining parameters depend on the model used by the graph generator, the distribution and the geometry type for the spatial generator.

We briefly discuss the available combiners, which receive a `.gr` and `.co` file, as inputs. The G_s combiner randomly selects a subset of the input social vertices to become spatial, assigning them a geometry from the `.co` file. These vertex-to-geometry assignments are stored in the output `.co` file; the output `.gr` file is identical to the input. Unlike G_s, the G_c and G_p combiners extend the input network graph. They create a new spatial vertex for each geometry found in the input `.co` file, and new edges to connect these spatial vertices to the social ones found in the input `.gr` file. The difference is in how these new edges are created; G_c connects each spatial vertex to multiple social ones, while G_p to exactly one. Hence, the combiners also receive the parameters of a normal distribution to determine how many connections every spatial vertex will have (G_c), or how many connections to spatial vertices, every non-spatial will have (G_p).

4 Demonstration Scenarios

We will demonstrate both the generation and the visualization features. For the first, we will walk the attendees through the process considering two scenarios:

Scenario 1: Generating networks from scratch. We will demonstrate how to generate a geosocial network using only the network and spatial generators in GESONGEN. We will use different parameters to generate all three types of geosocial networks supported by our system and show the effect these parameters have on the final network. The attendees will also be able to explore the generation process and to create synthetic geosocial networks themselves.

Scenario 2: Generating networks from existing datasets. We intend to also show how to generate a geosocial network from an existing network and/or a set of spatial objects; we will have various real-world social network and spatial datasets available. The attendees will have the opportunity to compare statistics of networks generated using real data and fully synthetic geosocial networks.

For the visualization, we will exemplify the three view options, i.e., social, spatial, and combined, explaining their advantages and limitations. Attendees will be able to interact with the interface, switching between different views, adding/removing vertices and edges from a generated network, and assigning labels. Moreover, they will be able to see the statistics for the generated networks and discuss how well they mimic real geosocial networks. Lastly, we will show the project management and export (to `.gr`/`.co` files) features of the system.

5 Conclusions

We developed the GESONGEN interactive interface to generate and visualize geosocial networks. The interface allows users to modify and extend generated networks by hand to meet their needs. GESONGEN is designed with extendability in mind. The generation process is decoupled from the network visualization which allows for incorporating additional graph and spatial generators. In the future, we plan to investigate this direction. Also, we will include new combiners to capture the correlation between socially interwoven and spatially close entities, and incorporate other generation approaches, e.g., from [10]. Another direction is to allow the visualization of larger graphs, by considering pre-rendering and interactive visualization techniques.

References

1. Albert, R., Barabási, A.L.: Emergence of scaling in random networks. Science **286**, 509–512 (1999)
2. Alizadeh, M., Cioffi-Revilla, C., Crooks, A.T.: Generating and analyzing spatial social networks. Comput. Math. Organ. Theory **23**(3), 362–390 (2017)
3. Armenatzoglou, N., Papadopoulos, S., Papadias, D.: A general framework for geo-social query processing. Proc. VLDB Endow. **6**(10), 913–924 (2013)

4. Beckmann, N., Seeger, B.: A benchmark for multidimensional index structures. Tech. rep., Philipps-Universität Marburg (2008), https://www.mathematik.uni-marburg.de/˜rstar/benchmark/distributions.pdf
5. Bouros, P., Sacharidis, D., Bikakis, N.: Regionally influential users in location-aware social networks. In: SIGSPATIAL. pp. 501–504 (2014)
6. Chen, L., Liu, C., Zhou, R., Li, J., Yang, X., Wang, B.: Maximum co-located community search in large scale social networks. Proc. VLDB Endow. **11**(10), 1233–1246 (2018)
7. Doytsher, Y., Galon, B., Kanza, Y.: Querying socio-spatial networks on the world-wide web. In: WWW. pp. 329–332 (2012)
8. Erdos, P., Renyi, A.: On the evolution of random graphs. Publ. Math. Inst. Hungary. Acad. Sci. **5**, 17–61 (1960)
9. Fang, Y., Cheng, R., Li, X., Luo, S., Hu, J.: Effective community search over large spatial graphs. Proc. VLDB Endow. **10**(6), 709–720 (2017)
10. Gallagher, K., Anderson, T., Crooks, A.T., Züfle, A.: Synthetic geosocial network generation. In: LocalRec. pp. 15–24 (2023)
11. Gallagher, K., Kotnana, S., Satishkumar, S., Siripurapu, K., Elarde, J., Anderson, T., Züfle, A., Kavak, H.: Human mobility-based synthetic social network generation. In: HANIMOB. pp. 23–26 (2022)
12. Katiyar, P., Vu, T., Eldawy, A., Migliorini, S., Belussi, A.: Spiderweb: A spatial data generator on the web. In: SIGSPATIAL. pp. 465–468 (2020)
13. Kim, J., Jin, H., Kavak, H., Rouly, O.C., Crooks, A.T., Pfoser, D., Wenk, C., Züfle, A.: Location-based social network data generation based on patterns of life. In: MDM. pp. 158–167 (2020)
14. Kim, J., Kavak, H., Manzoor, U., Crooks, A.T., Pfoser, D., Wenk, C., Züfle, A.: Simulating urban patterns of life: A geo-social data generation framework. In: SIGSPATIAL. pp. 576–579 (2019)
15. Levandoski, J.J., Sarwat, M., Eldawy, A., Mokbel, M.F.: LARS: A location-aware recommender system. In: ICDE. pp. 450–461 (2012)
16. Li, G., Chen, S., Feng, J., Tan, K., Li, W.: Efficient location-aware influence maximization. In: SIGMOD. pp. 87–98 (2014)
17. Mouratidis, K., Li, J., Tang, Y., Mamoulis, N.: Joint search by social and spatial proximity. TKDE **27**(3), 781–793 (2015)
18. Sarsour, A.A.R., Bouros, P., Chondrogiannis, T.: Towards generating realistic geosocial networks. In: LocalRec. pp. 25–28 (2023)
19. Sarwat, M., Levandoski, J.J., Eldawy, A., Mokbel, M.F.: Lars*: An efficient and scalable location-aware recommender system. TKDE **26**(6), 1384–1399 (2014)
20. Sun, Y., Sarwat, M.: A generic database indexing framework for large-scale geographic knowledge graphs. In: SIGSPATIAL. pp. 289–298 (2018)
21. Sun, Y., Sarwat, M.: Riso-tree: An efficient and scalable index for spatial entities in graph database management systems. TSAS **7**(3), 12:1–12:39 (2021)
22. Vu, T., Migliorini, S., Eldawy, A., Bulussi, A.: Spatial data generators. In: SpatialGems @ ACM SIGSPATIAL (2019)
23. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world'networks. Nature **393**(6684), 440–442 (1998)
24. Winick, J., Jamin, S.: Inet-3.0: Internet topology generator. Tech. rep., University of Michigan, Ann Arbor (2022), http://web.eecs.umich.edu/techreports/cse/02/CSE-TR-456-02.pdf