Discovering Communities in Global Flight Route Graph

Fengyu Cai [1] **Liangwei Chen** [1] **Junze Li** [1] **Wanhao Zhou** [1]

Abstract

Global flight route information reveals the connection between different countries and regions, which naturally takes on a community structure. In this project, we utilize the the global flight route graph with network analysis and machine learning techniques to discover such communities, both in weakly supervised (clustering with assigned initial centers) and unsupervised (community detection) fashion. We visualize our results and give our analysis behind the phenomenon.

1. Introduction

Open flights possesses several databases containing global flight and airport information. Route database gives detailed information of 67663 routes between 3321 airports. Additionally, airport database possesses geological information, which potentially helps enrich the graph representation.

Construction of the global flight route graph helps unveil the connections between different countries and regions. As a common sense, the density of the graph is not necessarily uniform across the graph. In particular, communities emerge on different parts of the graph. In a such community, flights are more concentrated inside and are relatively sparse when reaching to other communities.

In this project, we aim to discover such communities from the graph. We start from the basic data processing and network exploration to give us some preliminary insight of the graph. In the exploitation part, we use (i) clustering algorithms to partition the graph into different sets, as well as (ii) community detection algorithms to automatically find desired communities. Visualization tools are combined with each step to verify the effectiveness of our approach intuitively. We finally conclude with case study and analysis.

[1]School of Computer and Communication Sciences, EPFL, Switzerland. Correspondence to: Fengyu Cai <fengyu.cai@epfl.ch>, Liangwei Chen <liangwei.chen@epfl.ch>, Junze Li <junze.li@epfl.ch>, Wanhao Zhou <wanhao.zhou@epfl.ch>.

2. Data Acquisition and Preprocessing

Data are publicly available at Open flights[1]. Route database contains all the information for nodes and edges of our graph. There are in total 67663 edges and 3321 nodes of the initial graph. We accompany the graph with detailed geological information of each airport, given in the airport database. Key attributes are listed below.

- Airport ID: Unique OpenFlights identifier.
- Latitude / Longitude: Decimal degrees.
- Tz database: Timezone in "tz"format, eg. "America/Los_Angeles".
- Airline: 2-letter (IATA) or 3-letter (ICAO) code.
- Source / Destination airport: 3-letter (IATA) or 4-letter (ICAO) code of the airport.

We follow the procedure below to clean our data. Since we need the geological information in the airport database, we remove flights which contain unrecorded airports. Flight records with unspecified source or destination airport will also be removed. Additionally, with careful inspection we noticed some airports contain wrong information, e.g., Los Alamitos AAF Airport.

We obtained in total 3188 airports with 66771 flight routes. We construct both unweighted adjacency matrix U and weighted adjacency matrix W to represent the final graph. To be precise, elements of U are zeros and ones, where ones indicate there is a flight route between two airports; entries of W are obtained using the Laplacian kernel calculated on geodesic distance whenerver there is a flight route between two airports:

$$W_{i,j} = \exp\Big(-\frac{\text{GeodesicDist}(i,j)}{\sigma}\Big) \cdot (1-\delta_{i,j}) \qquad (1)$$

$\delta_{i,j}$ denotes whether i and j are the same node.

We make the natural assumption that in this scenario, graph should be undirected: if there exists a flight from A to B, then B should also be able to access A. Our adjacency matrices are finally processed to be symmetric.

[1]https://openflights.org/data.html

Table 1. Key properties of the graph

ATTRIBUTE	VALUE
CLUSTERING COEFFICIENT	0.49
GIANT COMPONENT DIAMETER	12
# CONNECTED COMPONENTS	7
AVERAGE DEGREE	11.73

3. Graph Analysis and Visualization

We explore the graph to gain preliminary insight into the problem.

3.1. Degree Distribution

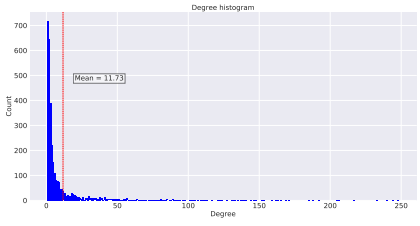

Figure 1. Degree distribution of the graph. The distribution typically follows the power law and thus the graph is the so-called scale-free network. Average degree is about 11.73.

The degree distribution of the graph is illustrated in Figure 1.

We observe that the distribution follows the power law approximately, demonstrating the network is scale-free. In scale-free networks, hubs are nodes that have more connections than others. Communities are likely to form around these hubs and detecting these communities help us to analyze the structure of the graph.

3.2. Basic Properties of the Graph

Key properties of the graph are listed in Table 1. Notably, we will proceed with the largest connected component for future analysis. The largest connected component contains 99.19% of total nodes. An abstract illustration of the whole graph is given in Figure 2.

For other tiny components, we find that airports of these components belong to regions: New Caledonia, Greenland, Namibia, etc.

3.3. Centrality

We analyze the top influential airports by calculating centrality metrics, given in Table 2.

Betweenness centrality of a node v is given by the expression:

$$\mathrm{BC}(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{2}$$

where σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of those paths that pass through v.

PageRank centrality is based on the structure of incoming nodes; Eigenvector centrality is related to the largest eigenvector of the adjacency matrix.

We observe that airports with high centrality rankings are usually located in Europe or United States,

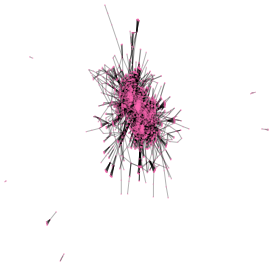

Figure 2. Visualization of the graph with all components. The largest connected component occupies the majority of the network and is used for later analysis.

Table 2. Rank of airports based on different centrality metrics

RANK	BETWEENNESS	PAGERANK	EIGENVECTOR
1	CDG, FR	ATL, US	AMS, NL
2	LAX, US	ORD, US	FRA, DE
3	ANC, US	ISL, TR	CDG, FR
4	DXB, AE	DFW, US	MUC, DE
5	FRA, DE	DEN, US	FCO, IT

3.4. Other Insights

We also gain insights regarding the aircraft preferences for different journeys. For intercontinental or long journeys, some popular aircrafts include: Airbus A380, Boeing 777-200LR, etc. For short journeys, popular aircrafts are: Avia BH-2, TG-DHP, etc.

3.5. Visualization

We visualize the graph against the globe before proceeding to exploit the network structure of the graph. We use the timezone information to categorize the airports into different

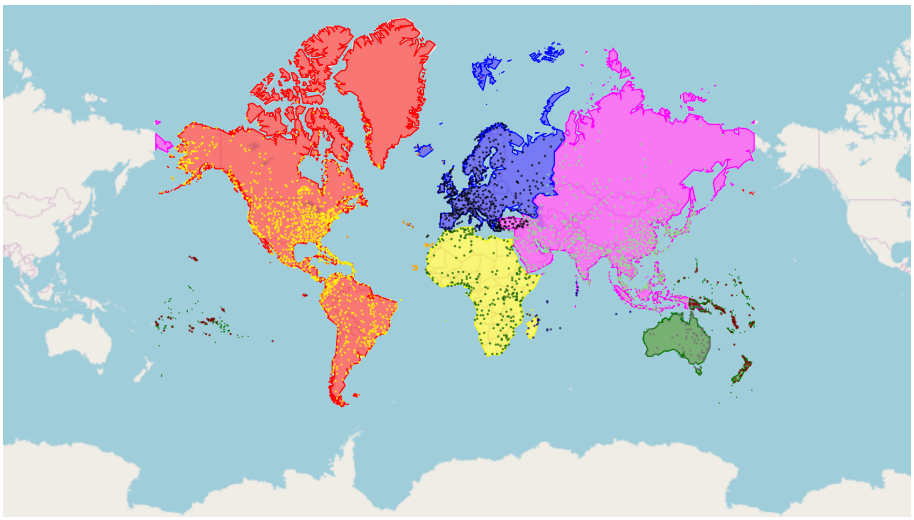

Figure 3. Visualization of global airports with borders set by continent. We use different colors for airports belonging to different continents. This will help to some extent with K-means clustering as the ground truth for each cluster.

groups. This coincides with the continental border as in Figure 3.

4. Learning on Graph

4.1. Feature Extraction

So far, we obtained the data matrix of dimension 3188×3188. For clustering methods, compressing the feature space to a smaller dimension can usually have better visualization results. Also, non-linear dimension reduction methods such as Laplacian eigenmap works typically well on graph data.

We perform the Laplacian eigenmap (Belkin & Niyogi, 2003) on our obtained unweighted and weighted adjacency matrices A. Laplacian matrix L is defined as $L = D - A$, where D is the diagonal matrix representing the degree of each node. In the normalized cut problem, we are interested in solving the vector u satisfying $Lu = \lambda Du$. We perform the eigenvector decomposition on the normalized Laplacian matrix $L_n = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$. The associated eigenvectors are v_k, s.t. $L_n v_k = \lambda_k v_k$. The final solution is $u_k = D^{-\frac{1}{2}} v_k$, with eigenvalue λ_k.

We exclude the eigenvector of the smallest eigenvalue, and the remaining eigenvectors are the candidates of the new feature space.

4.2. Clustering

We perform: (i) Spectral clustering on both unweighted and weighted adjacency matrices; (ii) K-means clustering on Laplacian eigenmap embeddings obtained from unweighted and weighted adjacency matrices with *initialized* clustering centers.

All visualization results are displayed in this section as well as in the appendix.

Using K-means on Laplacian eigenmap embeddings of weighted adjacency matrix yields meaningful results, as illustrated in Figure 4. After extensive experiments, we choose the final dimension as 2, i.e., we are using two eigenvectors of Laplacian eigenmaps two represent our data. This yields good results and can be visualized in 2-D plot.

Notably, in K-means clustering, the final results are heavily susceptible to the parameters of the algorithm. This is where we regard our method as weakly supervised method. If clusters are initialized randomly, the converged result might differ greatly between different initial clusters. Here we utilize the information from the attribute: *Tz database*. We assume a priori that there are in total 9 clusters, belonging to each continent as defined in the *Tz database*. We thus run our algorithm to cluster 9 groups. Next, we introduce the centrality ranking computed in Section 3.3. I.e., we assign the initial centers to be the topmost ranked airport in each continent. Table 3 summarizes all the center airports we used.

4.3. Community Detection

Though we obtained reasonable results from the previous section via K-means running on Laplacian eigenmap embedding, there still requires to some extent ground truth to have more stable performance. We resort to a more autonomous way of finding such communities on our graph.

We investigate two major algorithms from two different

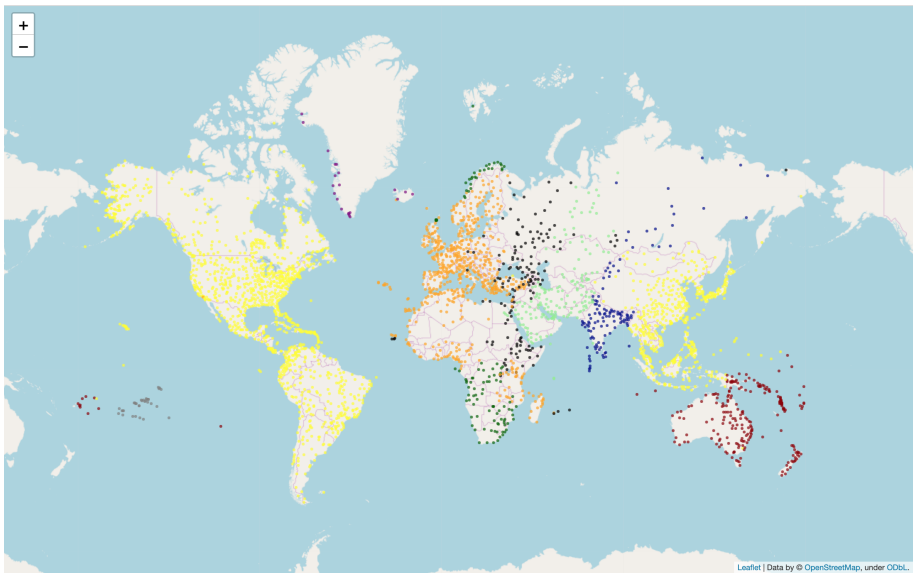

Figure 4. K-means on the Laplacian eigenmap of weighted adjacency matrix.

Table 3. City of center airport for each continent

CONTINENT	CITY
AFRICA	JOHANNESBURG
AMERICA	CHICAGO
ARCTIC	SVALBARD
ASIA	DUBAI
ATLANTIC	GRAN CANARIA
AUSTRALIA	SYDNEY
EUROPE	PARIS
INDIAN	MALE
PACIFIC	HONOLULU

families of community detection.

4.3.1. GREEDY MODULARITY MAXIMIZATION

The algorithm (Chen et al., 2014) is based on a criterion called modularity to evaluate the communities detected. Suppose we have a family of such detected communities C, and c_i is a specific community in C. Formally, the modularity of a partitioned network is defined as:

$$Q = \frac{1}{2|E|} \sum_{ij} \left[A_{ij} - \frac{d_i d_j}{2|E|} \right] \delta_{c_i,c_j} \tag{3}$$

$$= \sum_{c_i \in C} \left[\frac{\left|E_{c_i}^{in}\right|}{|E|} - \left(\frac{2\left|E_{c_i}^{in}\right| + \left|E_{c_i}^{out}\right|}{2|E|} \right)^2 \right] \tag{4}$$

where $\left|E_{c_i}^{in}\right|$ is the number of edges between nodes within community c_i, $\left|E_{c_i}^{out}\right|$ is the number of edges from the nodes in community c_i to the nodes outside c_i, and $|E|$ is the total number of edges in the network, where d_i is the degree of node i, A_{ij} is an element of the adjacency matrix, δ_{c_i,c_j} denotes whether c_i and c_j are the same community.

Modularity favors a community that has more edges within the same community than expected number of edges while preserving the degrees of the vertices. Maximizing the modularity will increase the gap.

The greedy algorithm starts with each node representing an independent community and merges communities with the highest $\Delta Q_{c_i,c_j} = 2\left(\frac{|E_{c_i,c_j}|}{2|E|} - \frac{|E_{c_i}||E_{c_j}|}{4|E|^2}\right)$. The algorithm stops when all vertices are in the same community and outputs the partition with the largest value of modularity.

4.3.2. LABEL PROPAGATION

Label propagation (Raghavan et al., 2007) similarly starts with each node representing a different label. For a given node v_i with neighbours $v_{i1}, v_{i2}, \ldots, v_{ik}$. Labels of each node v_i is denoted as $L_{v_i}(t)$ at time t. Node v determine its label at time t based on the its neighbours' labels of the previous timestamp. I.e.,

$$L_{v_i}(t) = f\left(L_{v_{i1}}(t), \ldots, L_{v_{im}}(t), \ldots, L_{v_{ik}}(t-1)\right) \tag{5}$$

f here is some function that returns the label occurring with the highest frequency among neighbors. If every node has a label that the maximum number of their neighbors have, then stop the algorithm. At the end of the propagation process, nodes having the same labels are grouped together as one community.

4.3.3. COMPARISON

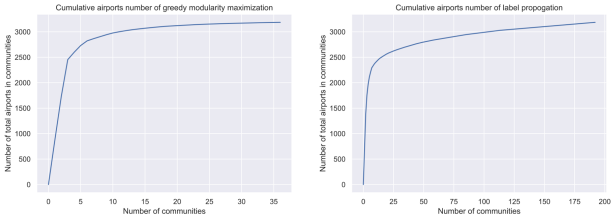

Figure 5. Cumulative number of airports in communities.

Figure 5 illustrates all the cumulative number of airports in top populated communities. Top 5 communities discovered by greedy modularity maximization accounts for over 90% of the airports, where as we need around 25 communities in label propagation to account for that portion. This means the communities found by greedy modularity maximization are more accurate since we are expecting in total around 10 communities.

4.4. Case Study

Detailed illustration of communities found by the above two algorithms will be given explicitly in the appendix. Here we zoom in to evaluate the quality of our automatically discovered communities. We will stick to analyze the communities in greedy modularity maximization.

Ryan Airlines. Ryanair is an Irish budget airline. In 2016, Ryanair was the largest European budget airline. From Figure 6, we observe that all its flights operate within one community denoted as purple circles. Connections of the network are dense and more of point-to-point connection. We aim to find communities where the connections inside itself are dense and the route map of Ryan Airlines demonstrate this property.

Turkey Airlines. It interconnects many communities (denoted in different colors) as it is more international and has its center near the Istanbul airport. We expect this behavior since its area of service goes far beyond that of Ryanair.

5. Conclusion

In this report, we explore the global flight routing network to figure out the potential community among the airports.

Firstly, we build up the network, analyze its properties. Since the whole graph is disconnected, we choose the largest connected component which accounts for 99.19% of the total nodes.

Secondly, given a priori knowledge (the number of clusters), we run spectral clustering to classify the airports. However,

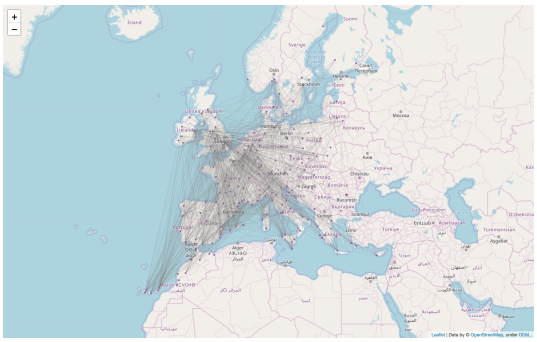

Figure 6. Route map of Ryan Airlines

Figure 7. Route map of Turkey Airlines

with random initialization, the clustering result is variant and unstable. Therefore, later in the process of clustering (K-means), we select some outstanding candidates as initialized centers and feed them into the model. Candidates are chosen by centrality ranking in the largest connected component of the graph. We average three different centrality metrics and choose the topmost airport for each continent. Accordingly, the result has been much more similar to the real continental distribution.

To get rid of the need of supervision, finally, we perform the community detection using two unsupervised algorithms, Greedy Modularity Maximization and Label Propagation. Different from clustering methods used before where we need to assign the number of clusters, community detection methods will return a list of communities detected. Focusing on the main part of the graph, we only take top 6 communities containing over 90% of nodes in consideration. Furthermore, we analyze the quality of the communities we discover using two representative airlines, regional airline (Ryanair) and international airline (Turkey Airlines). Typically, flight routes of Ryan Airlines spread within a single community and are dense, taking on the characteristic of point-to-point connection. Whereas for the international airline, it covers multiple communities all over the globe and connections between communities are sparse.

References

Belkin, M. and Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

Chen, M., Kuzmin, K., and Szymanski, B. K. Community detection via maximization of modularity and its variants. *IEEE Transactions on Computational Social Systems*, 1 (1):46–65, 2014.

Raghavan, U. N., Albert, R., and Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007.

Appendix

In this section, we give our remaining visualization results. Figure 8 gives all the flight routes in the database with airports labeled by different colors denoting the continent.

In Section 4.2, we perform spectral clustering on both unweighted and weighted matrices. Figure 9 and Figure 10 give results respectively.

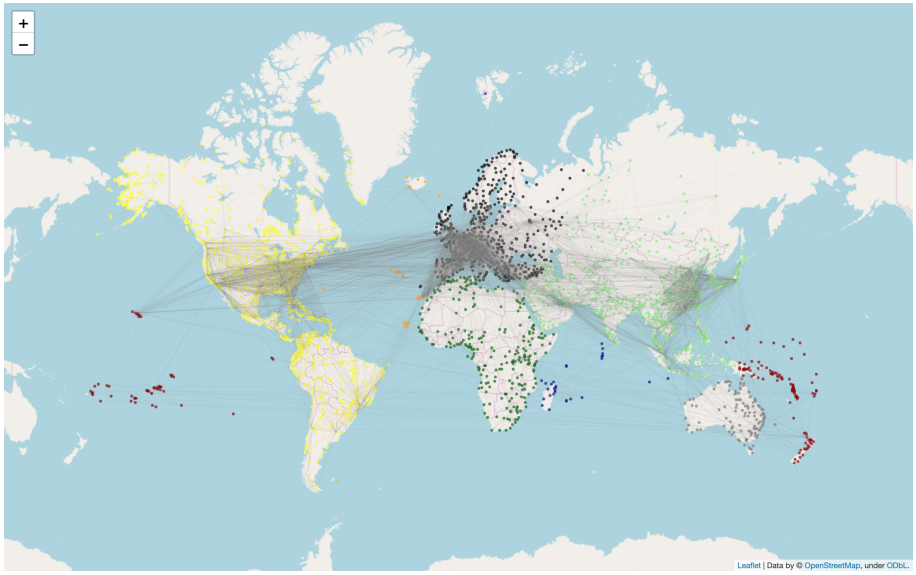

Figure 8. Visualization of all flight routes with grouped airports.

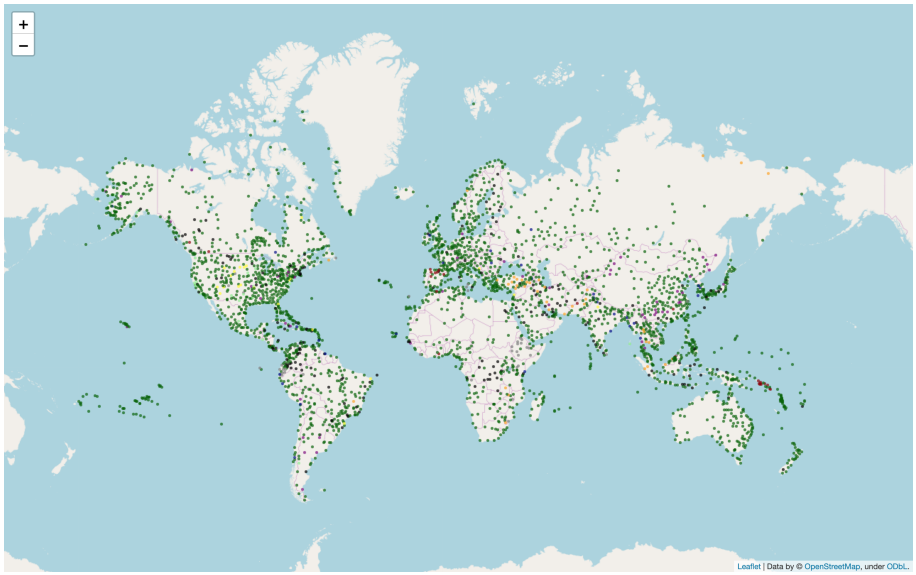

Figure 9. Spectral clustering on the unweighted adjacency matrix.

As mentioned in Section 4.2, we run K-means on the Laplacian eigenmap of unweighted adjacency matrix. Results obtained are shown in Figure 11.

Figure 12 and Figure 13 illustrate the communities discovered by greedy modularity maximization and label propagation respectively. We select the top 6 communities for visualization.

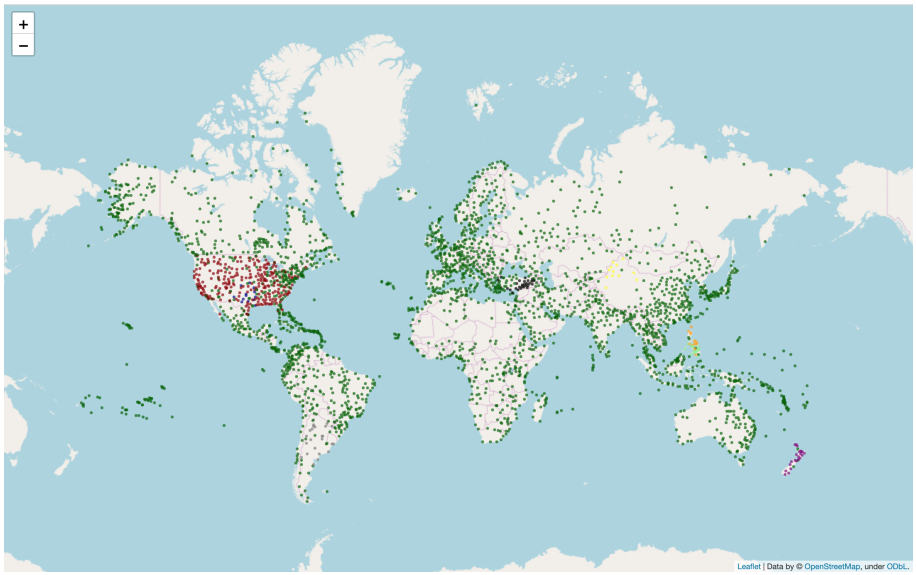

Figure 10. Spectral clustering on the weighted adjacency matrix.

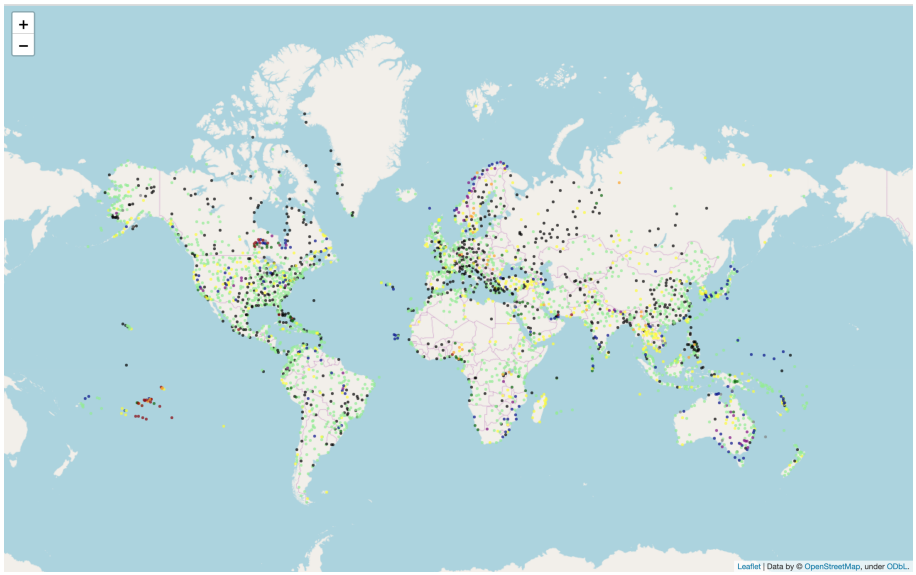

Figure 11. K-means on the Laplacian eigenmap of unweighted adjacency matrix.

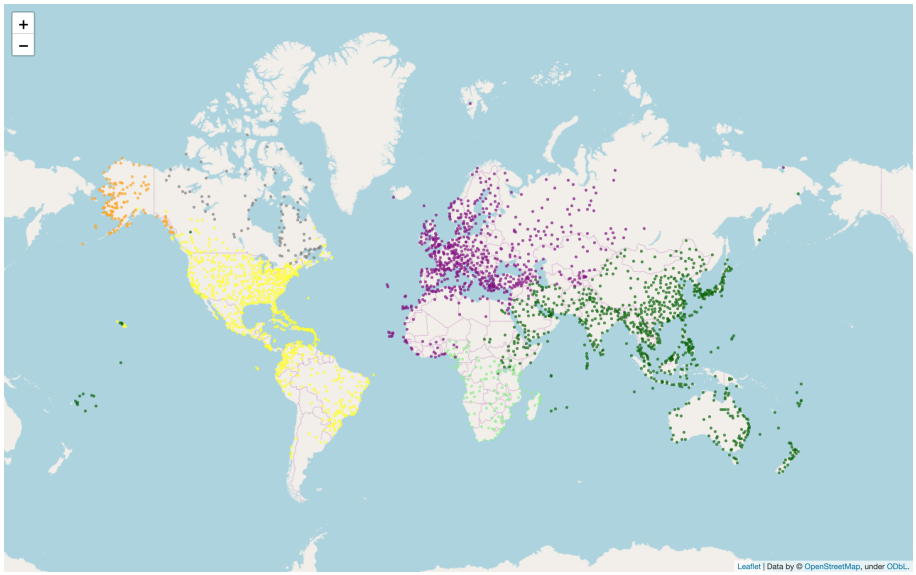

Figure 12. Communities discovered by greedy modularity maximization.

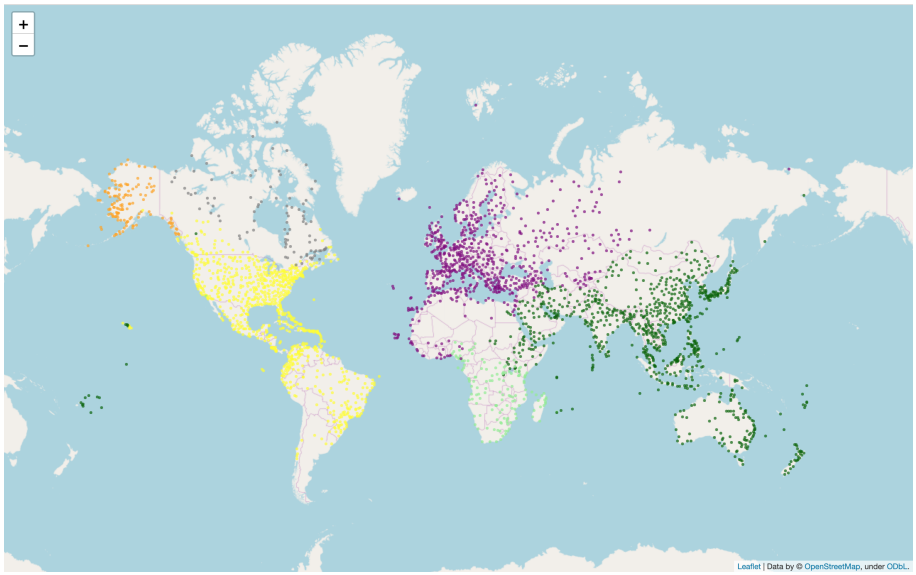

Figure 13. Communities discovered by label propagation.