Powering Social Network Analysis with Graph Data Model

An Example of the Meetup.com Visual Analytics

Yiwei Zhang
Engineering
Columbia University
yz2698@columbia.edu

Mengge Li
Engineering
Columbia University
ml3695@columbia.edu

Rongyao Huang
QMSS
Columbia University
rh2648@columbia.edu

Rahul Gaur
Computer Science
Columbia University
rg2930@columbia.edu

Abstract— **The increasing popularity of various social network services, such as Facebook, Twitter and Meetup, provides a great opportunity to study what we care about and how we interact with others. Because of the complexity of relationship presented in social network data, however, research and applications are limited by the query efficiency of traditional relational data model and SQL-like language. This project intends to power social network analysis with a new type of data model, the graph data model. A special type of event-based online social service, Meetup, is examined. By storing data in a graph DB, Neo4j, we were able to efficiently extract information, conduct analysis and generate dashboard analytical visuals through real time user query.**

A video of the dashboard demonstration can be viewed on YouTube: http://youtube.com/watch?v=csp1YnoJtnM

Keywords-component; social network analysis; community detection; graph database; neo4j; meetup; visual analytics

I. INTRODUCTION

We live in the era of "big data" where rapid digitalization and advances in information extraction greatly deepen our understanding of this world. Google search engine services, Dropbox cloud storage, Spotify music recommendation, while we are enjoying the numerous data driven products, some people have to concern themselves with the choice and implementation of data models.

The long existed relational data model is proved efficient in handling a single representation of an aggregate entity. But when the relationships between entities get increasingly complicated, or when multiple views of the domain are desired, this data model becomes inadequate.

However, the very nature of the human society is a large network, characterized by rich connections between individuals. With the popularization of Facebook and Twitter, social network analysis comes into the spotlight. People have developed various theories and models for studying the relationships among interacting units. This calls for a better data model that can efficiently implement the analysis and even builds them into data products. And the answer is: graph data model.

Motivated by this idea, this project attempts to power social network analysis with graph database. We selects the Meetup.com as our targeted social network because it has several desirable attributes: 1) it is organized around interest groups which reflect trends in industry, academia and people's recreational life; 2) it is both online and offline, which adds value to the analysis; 3) it has an excellent API which eases the data collection. And we are trying to answer 2 questions: 1) what's the Meetup.com social world look like? 2) Can we achieve real time network visual analytics with the help of graph database?

II. RELATED WORKS

Social interaction has long been researched in sociology. Moreover, as various social media platforms gain popularity, the online network has also become increasingly popular in the computer science research field. Among all the social network theories, "small world" is probably the most fundamental and renowned. According to the theory, each of us maintains a set of acquaintances, and these acquaintances have their own acquaintances. This fast expansion implies that we reach anyone through a few hops. This theory comes from the famous Milgram's "small world" experiment [4], which shows that we can know anyone through on average 6 people. This theory has allowed people to explore the social network in a more mathematical way.

Community Detection

One important way of exploring the social network is detecting community structure, as it shed light on the organization of the complex system and on their function [2]. In a graph, nodes can often be grouped into sets such that each node is densely connected internally. Complex systems are usually organized in compartments, which have their

own role and/or function. In the network representation such compartments appear as sets of nodes with a high density of internal links, whereas links between compartments have a comparatively lower density. These subgraphs are called communities, or modules, and occur in a wide variety of network systems [3].

There are various *Community Detection* algorithms that are used on huge graphs. These algorithms can be used for clustering the nodes and node-pairs. One of most efficient algorithm is the algorithm by M. Girvan and M.E Newman [3]. It is a hierarchical divisive algorithm in which links are iteratively removed based on the value of their betweenness, which expresses the shortest paths between pairs of nodes that pass through the link. The method implemented by them uses the modularity maximization technique. The modularity of Girvan and Newman estimates the goodness of a partition based on the comparison between the graph at hand and a null model [3]. The idea of detecting communities by optimizing a modularity was proposed by Newman [3]. The idea behind detecting community structures to provide an insight into how network functions and topology affect each other. These insights can be used to improve the spectral clustering of graphs. We use this modularity maximization method for our project. We apply it on various groups in the Meetup dataset (which act as nodes) for an optimal clustering and community detection combined with various iterations (mostly 100). The method used in our project is called the Louvain Method. The original idea for the method is due to Etienne Lefebvre who first developed it during his Master thesis at UCL (Louvain-la-Neuve) in March 2007. The method was improved and tested with Vincent Blondel, Jean-Loup Guillaume and Renaud Lambiotte and is now known as the "Louvain method" because, even though the co-authors now hold positions in Paris, London and Louvain-la-Neuve, the method was devised when they all were at the University "catholique de Louvain" [11]. The method was initially used for unweighted graphs, but can easily be applied for weighted graphs by using the greedy optimization approach for different modularity.

LinLog Energy Model

LinLog is an energy model which uses the cut ratio as measure of the coupling of two disjoint sets of nodes. Using this measure, its proved that in minimum energy drawings of the LinLog model, clusters are clearly separated from the remaining graph and the and the distance of each cluster to the remaining graph is interpretable with respect to properties of the graph (more precisely, the distance is approximately inversely proportional to the coupling). This method has been successfully applied to networks with tens of millions of nodes and hundreds of edges. Informally, a cluster is a set of nodes with many internal edges (high cohesion) and few edges to nodes outside the set (low coupling) [2].

In our project, we chose to implement *LinLogLayout* method. It optimizes LinLog and related energy models to compute layouts of the graphs combined with Girvan and Newman's Modularity to compute clustering [2][3]. The end result is clusters of various group Id's based on their cohesion.

Event/Group – Based Social Network

Different from the classical social network structure as presented in the case of Facebook and Twitter, the group/event-based characteristic of Meetup has made it an interesting and special type of network to study. The Meetup online network is organized around group: users can choose to join multiple groups of their interest. While the offline network is organized around events that are hosted by groups. In summary, the online and the offline networks share user base, but can be quite different in structure. Sander and Seminar [1] attended 40 social events in Meetup and concluded that participants in Meetup social events have social structures instead of just strangers meeting strangers. Liu, and his colleges [8] also conducted a comprehensive research on the Meetup community structure using web-crawled data from Oct 2011 to Jan 2012: clear definitions are given and properties are analyzed.
Based on the previous research and results, our study used an updated dataset that's collected in November 2014.
Notice there are no time series in our dataset, but instead of having just the group and tag id, we now have their names in character strings. Armed with graph database, this allows us to zoom in on individual topics and groups, which generates micro analytical results that are of real world implications.

III. SYSTEM OVERVIEW

In this section, we first present the theoretical framework of constructing and analyzing the Meetup.com social network, and then describe the system architecture of our real time analytical platform.

3.1 Theoretical Framework

Like any other social networks, social network of Meetup capture social interactions among the users. However, what is different is that Meetup incorporates two kinds of interactions: online social interactions and offline social interactions. In this part, we will give definitions of the structure.

3.1.1 Network Definitions

A System of Two Networks: Meetup users interact both online and offline.

$$G = < N, E^{on}, E^{off} >$$

Online Social Interactions: Users are in the same group, andi nteract with each other through the internet.

$$G^{on} =< N, E^{on} >$$

Offline Social Interactions: Users attend events that have a real world location.

$$G^{off} =< N, E^{off} >$$

The Network of Groups: group network looks at the Meetup community at a more aggregated level, but allows us to better interpret the community structure. Group interactions are based on shared members.

$$G = < U, E >$$

Notice that all the above graphs are undirected, however their edges are weighted to represent the strength of the interactions. So here comes the definition of weights.

Online Social Network Graph: Users n_i and n_j are connected in the online social network G^{on} if they are members of the same social group. Let g_r denote a group with $|g_r|$ members, then $(n_i, n_j) \in N^{on}$ if and only if $\exists g_r$ such that $n_i \in g_r$ and $n_i \in g_r$. We consider users of a smaller group more closely connected than those of a larger group. Therefore, we can define the edge weights:

$$w_{i,j} = \sum_{g_r} \frac{1}{|g_k|}$$

Offline Social Network Graph: Similarly, the offline social network graph G^{off} is constructed in the same way, but it based on the shared events between the users. Users n_i and n_j are connected in the online social network G^{off} if they are members of the same social group. The number of the members of the events are defined as $|e_r|$. Thus, the definition of the weights comes as:

$$w_{i,j} = \sum_{e_r} \frac{1}{|e_k|}$$

Social Network Graph of Groups: Just in the same method, the social network of groups. However, the size parameter is set as the shared number of users between the groups. Define it as s_r, so the formula of weights is:

$$w_{i,j} = \sum_{s_r} \frac{1}{|s_k|}$$

By now the three social network graphs are fully constructed.

3.1.2 Network Properties

The Meetup social network possesses important properties. Among them, degree, clustering coefficient and power law distribution are interesting to look at.

Degree

In graph theory, the degree (or valency) of a vertex of a graph is the number of edges incident to the vertex. The degree of a vertex v is denoted $\deg(v)$ or $\deg v$. In the social network, degree is all # of the people the user is connected to. And in our online and offline social network graph, the edges connected to the users have the weights, the definition of the weights is delivered in previous section.

Clustering Coefficient

A clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together. Evident suggests that in most real-world networks, and in particular social networks, nodes tend to create tightly knit groups characterized by a relatively high density of ties; this likelihood tends to be greater than the average probability of a tie randomly established between two nodes.

The clustering coefficient is based on triplets of nodes. A triplet consists of three nodes that are connected by either two (open triplet) or three (closed triplet) undirected ties. A triangle consists of three closed triplets, one centered on each of the nodes. The global clustering coefficient is the number of closed triplets (or 3 x triangles) over the total number of triplets (both open and closed).

The clustering coefficient is formally defined as:

$$C = \frac{3 \times number\ of\ triangles}{number\ of\ connected\ triplets\ of\ vertices} = \frac{number\ of closed\ triplets}{number\ of\ connected\ triplets\ of\ vertices}$$

Power Law Distribution

In statistics, a power law is a functional relationship between two quantities, where one quantity varies as a power of another. For instance, the number of cities having a certain population size is found to vary as a power of the size of the population.

Given a relation

$$f(x) = ax^k$$

Scaling the argument x by a constant factor c causes only a proportionate scaling of the function itself. That is,

$$f(cx) = a(cx)^k = c^k f(x) \propto f(x)$$

3.1.3 Community Detection

We utilized the Lovain Algorithm for community detection in our project. It is a simple yet efficient algorithm used for identifying communities in a large network. It uses a greedy optimization strategy. It aims to optimize the modularity partition of the network. The optimization is performed in two steps:
1) The method looks for small communities by optimizing modularity locally.
2) It then aggregates the nodes belonging to the same community and builds a new network whose nodes are the communities.
These steps are then repeated iteratively until a maximum modularity is attained and a hierarchy of communities is produced.

3.2 Real Time Analytical Platform

The Meetup Visual Analytics Platform is an end-to-end data product that allows users to generate visuals of their interested Meetup topics. It is built upon the GitHub projects of the Meetup-Analytics-Dashboard by Kenny Bastani and the Neo4j Swagger by Mat Tyndal. As shown in Figure 3.1, the system is constructed with 4 major components: the initial Meetup API data pull, the Neo4j graph data storage, the web app analytical queries, and the dashboard data visualization.

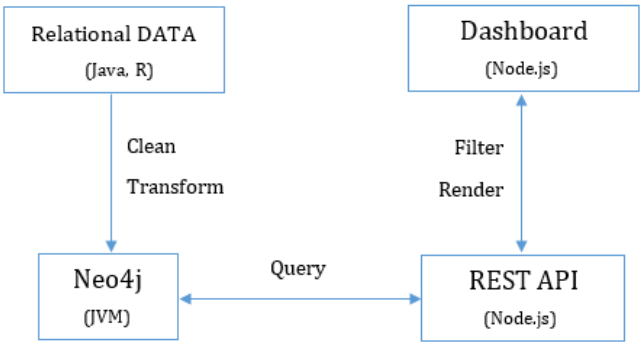

Figure 3.1. System Architecture

The following sections will elaborate on each component in this system.

3.2.1 Raw Data Poll

A Java program was written to pull data from the Meetup.com server through API and stored them in 7 csv files representing pair relationships. Table 3.1 lists the relationships captured by these files.

File Name	***Stored Relationship***
groupIdName	group id → group name
tagIdName	tag id → tag name
eventIdName	event id → event name
userGroup	user id → group id (one user can join multiple groups)
eventGroup	group id → event id (one group can host multiple events)
eventLocation	Event id → latitude, longitude (an upcoming event is marked by its lat lon location)

Table 3.1 Relational Data Storage

Basic data investigation in R reviewed that the whole dataset contains 17311 groups, 1598876 events, 1075859 users and 20780 tags. Duplicates were found in groupIdName.csv and were removed. Also, we looked at number of groups per tag and found it a long tail distribution. Almost 50% of the tag names were exclusive to one group and were therefore less likely to be searched. In order to increase the query efficiency of our platform, we decided to retain only the tags that were shared by at least 5 groups. This reduced the number of tags to 4848, but preserved 92.45% of the group-tag relationships.

3.2.2 Neo4j Network Construction

The next step is to transform the data structure from relational to graph. Among the existing graph databases, we chose Neo4j because of its detailed documentation, supportive user community, and various applications that can be of reference.

The Neo4j graph system is composed of three essential elements: node, property, and edge. The Meetup.com network has 4 type of nodes: group, user, event, and tag. For our desired analytics, only group, event and tag will be imported as nodes. Figure 3.2 demonstrates the constructed nodes and their relationships.

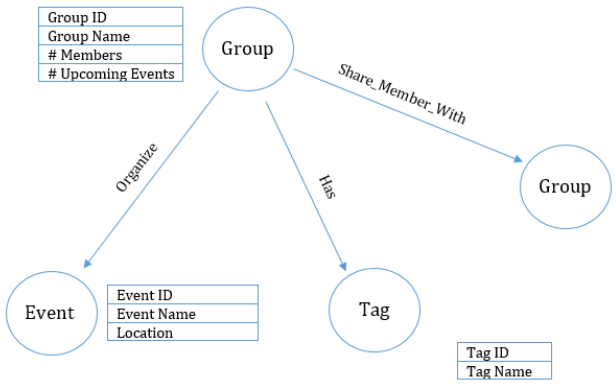

Figure 3.2 Neo4j Network Structure

Figure 3.3 is a Neo4j web interface screenshot that demonstrates part of the group-tag relationship in our database.

Figure 3.3 Screenshots of Neo4j Web Interface

The whole graph system has 1,611,035 nodes, 4,875,568 properties, and 3,599,309 edges.

3.2.3 REST API Construction

In order to connect the Neo4j graph database with the front-end web dashboard, 4 REST API are constructed. The REST API is a fork of Neo4j Swagger. Swagger is specification and complete framework implementation for describing, producing, consuming, and visualizing RESTful web services.

For each API, we specify dashboard user input as parameters, write cypher queries to extract desired node-edge information, and then map the queries to node.js actions that can then feed the returns to web dashboard.

Figure 3.4 is a screenshot of the Neo4j Swagger REST API we constructed.

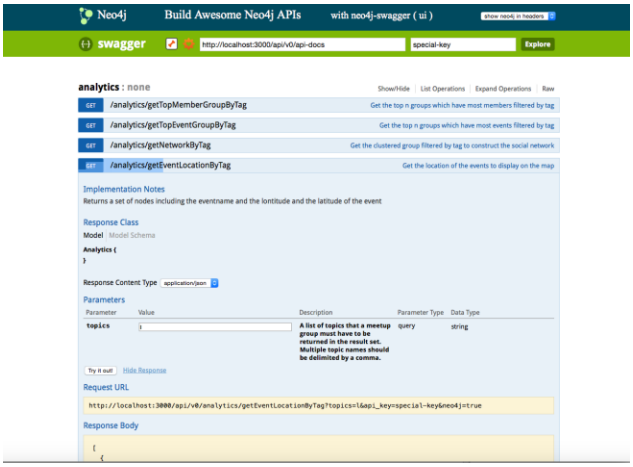

Figure 3.4. Swagger API Demo

3.2.4 Dashboard Construction

With the neo4j database and REST API in hand, we are now ready to build the Meetup Visual Analytics dashboard. This web application uses JavaScript to communicate with the REST API and generate several interactive graphs containing information about groups and events that are related to users' desired topics.

The visual construction of this dashboard uses bootstrap for page styles, highcharts.js for bar graphs, google map API for the event map visual, and d3.js for the group network representation.

IV. TOOLS AND ALGORITHMS

In this section, we will briefly describe all the tools that we used for conducting social network analysis and building up the platform. We will also present a detailed discussion of the Lovain Algorithm and the LinLogLayout Algorithm for community detection and visualization.

Tools

This project requires a comprehensive set of tools to be used in each phase. Table 4.1 gives a full list.

Project Phase	*Used Tools*
Web Crawling	Java
Data Cleaning	R: stringr, ggplot2, plyr
User Network Analysis	Python: networkx, Java, Gephi
Graph Data Storage	Neo4j with Cypher
REST API	Node.js, Cypher
Dashboard	Node.js, JavaScript, HTML, JQuery, highcharts.js, d3.js

Table 4.1. Tools

Algorithms

In the Lovain Algorithm, the modularity value is defined between -1 and 1 as:

$$Q = \frac{1}{2m}\Sigma_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j)$$

Here A_{ij} represents the edge weight between nodes i and j. k_i and k_j are the degrees of node i and j respectively. m is the total number of edges in the graph. c_i and c_j are the communities of the nodes.

There are two phases to calculate value Q. Each node is first assigned to its own community. Then for each community for each node i, the change in modularity is calculated for removing i from its own community and moving it into the community of each neighbor j to i. The second phase of the algorithm groups all of the nodes in the same community and builds a new network where nodes are the communities from the previous phase. Any links between nodes of the same community are now represented by self-loops on the new community node and links from multiple nodes in the same community to a node in a different community are represented by weighted edges between communities. Once the new networks is created, the second phase has ended and the first phase can be re-applied to the new network.

Besides the Lovain Algorithm, we used another algorithm called the LinLogLayout Algorithm to compute the graph layouts and graph clusterings. It reads a graph from a file, computes a layout and a clustering, writes the layout and the clustering to a file, and displays them in a dialog. LinLogLayout can be used to identify groups of densely connected nodes in graphs, like communities of friends or collaborators in social networks, related documents in hyperlink structures (e.g. web graphs), cohesive subsystems in software systems, etc. With a change of a parameter in the main method, it can also compute classical (i.e. readable) force-directed layouts [11].

Layouts and clusterings complement each other. On the one hand, only clusterings can faithfully represent inherently high-dimensional structures. (Layouts with more than 2 or 3 dimensions can be easily computed, but are difficult to understand for human viewers.) On the other hand, only layouts can show:

1. The relationship between clusters, e.g., whether their separation is clear or fuzzy, and which nodes form their interface [11].
2. The internal structure of clusters, e.g., whether a dense cluster is composed of even denser sub clusters; (layouts have no fixed resolution, and thus no "resolution limit".
3. The relationship between nodes and clusters, e.g., whether a node is central or peripheral to its cluster, or whether a node is closely related to several clusters [11].

V. DASHBOARD DESCRIPTION

The Meetup Visual Analytics Dashboard is divided into 5 major functional blocks. The general idea is that a user can type in his/her topic of interest, and if that matches any tag name in our Neo4j database, the dashboard will return its relation groups and events information in a way that's easily digestible. Below we will describe the functional blocks one by one.

User Query Window

User can type in any number of topics separated by comma and press enter to send request. Figure 5.1 is a screenshot of this block.

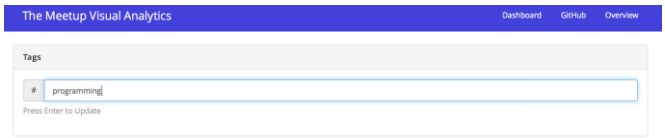

Figure 5.1 Dashboard Query Window

Most Popular Group

The top 5 groups that have the most members will be presented in a colorful bar char with the x-axis representing the group size and the group names listed in the right-hand size legend. Mouse-over individual bars reveal group details. Figure 5.2 is a screenshot of this block.

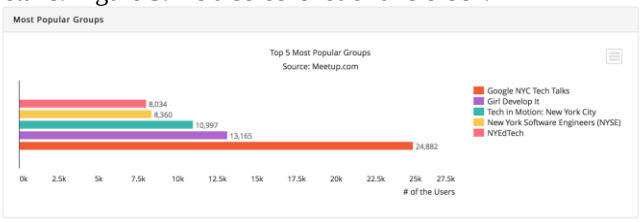

Figure 5.2. Most Popular Groups

Most Active Group

The top 5 groups that have the most upcoming events will be presented in a colorful bar char with the x-axis representing the number of events and the group names listed in the right-hand size legend. Mouse-over individual bars reveal group details. Figure 5.3 is a screenshot of this block.

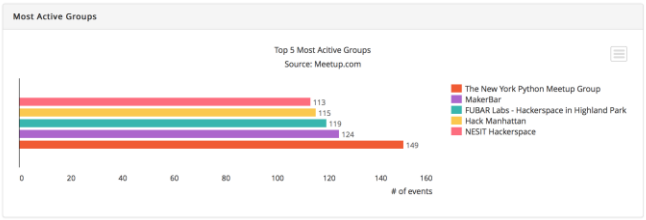

Figure 5.3. Most Active Group

Event Location Map

All the events related to the topics that a user type in will be marked as a blue dot on a map centered at New York. Multiple events hosted at the same location will make the dot darker. Figure 5.4 is a screenshot of this block.

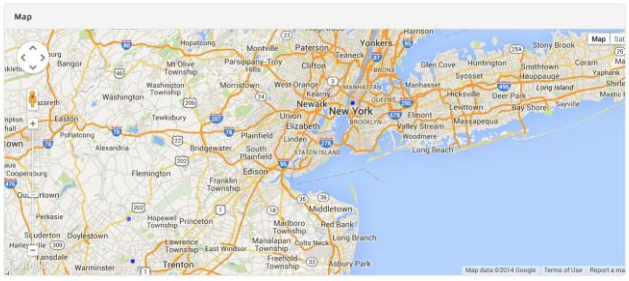

Figure 5.4. Events Location Map

Group Network

All the group related to the topics that a user type in will be presented as a network. Each node is a group, and the layout of the network is determined by the weight of edge between two groups. The bigger the weight, the closer it appears on the graph. Also, the color of each node represents which cluster it belongs. Figure 5.3 is a screenshot of this block.

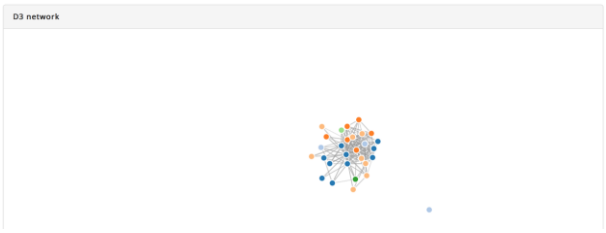

Figure 5.3. Group Network

The dashboard also has our project overview and the link to our GitHub account.

VI. EXPERIMENT RESULTS

In this section, we will discuss important properties of the Meetup network, results of community detection, and performance of the Neo4j database.

5.1 Meetup Network Properties

Average Degree & Clustering Coefficient

Table 5.1 presents the average degree and clustering coefficient of the Meetup online and offline network.

	Average Degree	*Clustering Coefficient*
Online network G^{on}	1660.1	0.443
Offline network G^{off}	157.3	0.246

Table 5.1. Meetup Network Properties

As can be seen, the average degree of G^{on} is much higher than G^{off}, which means that users have much interactions with others online while users are less likely to attend the offline events to know others physically.

Also we can notice that the clustering coefficient of G^{on} is much high than that of G^{off} too. By the definition of the clustering coefficient, we can know that the number of components of triangles in the G^{off} is less than that of G^{on}. We name the edge of the triangles in the social network the strong-tie, which mean the connection between the two users is strong, when the edge between two users is cut out, they can find another route through just one person. So when the edge is not a strong tie, it's a weak tie. If weak tie is cut out, the users is likely to lose the connection. As a result, there are more strong tie in the G^{on} and more weak tie in the G^{off}. The G^{on} is denser and strong connected which the G^{off} is more separated and venerable.

The result of the graph makes sense in the real world. People intend to have a lot of friends, and attend different groups, they may know someone very far away and share a lot of friends. However, users seldom attend offline events and people in the events are around the same location.

Power Law Distribution

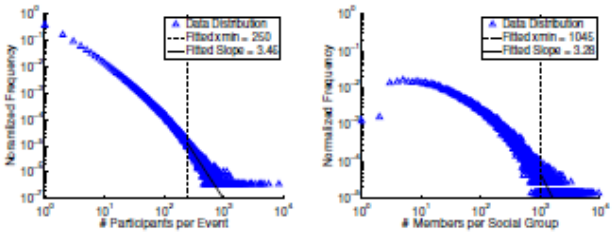

Figure 5.1 Power Law Distribution

As can be seen in the Figure 5.1, most of the events are small with just a few participants, bur large events with a large number of participants. Similarly, large groups do have significant presence. Both of these two graph is "big tail" style, which means the distribution fit the power law.

5.2 Meetup Community Detection

Using the Lovain Algorithm, we iterate through the entire dataset 100 times. The number of iterations is a parameter of the method minimizeEnergy of the classes MinimizerBarnesHut and MinimizerClassic (called in the main method). Increasing the number of iterations proved to be particularly helpful if the energy values printed by LinLogLayout still decreased notably during the last 5 percent of the iterations (e.g., from iteration 95 to iteration 100).

Because it's difficult to comprehend the clustering results of the whole dataset, here we chose a subset that's related to data and analytics. Indeed, Figure 5.2 allows us to visually comprehend the community structure. After sanity checking the group names, it turned out the green nodes are all related to analytics, and the red nodes to big data. This has proved the efficiency of the Lovain Algorithm to some extent.

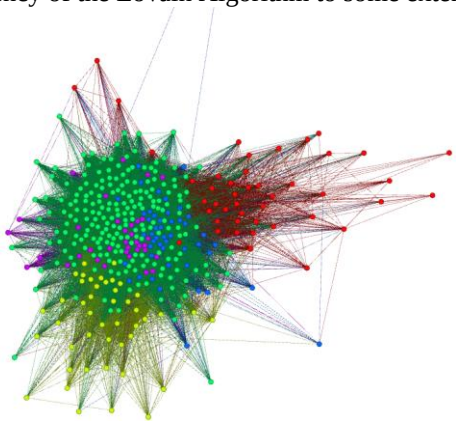

Figure 5.2 Community Structure of Data & Analytics Related Groups

5.3 Neo4j Performance

With a graph database that has over 1 million nodes and 3 million edges, our dashboard that runs at localhost is perfectly responding. From sending data request to visual rendering, it usually takes less than a second. This proves the query efficiency of Neo4j when the dataset possesses complicated relationship.

However, we do need to point that importing data into Neo4j is time consuming task. It took us almost two days to construct all the nodes and edges in Neo4j.

VII. CONCLUSION

With all the hard efforts, we are now able to answer the two questions proposed at the beginning of this report. 1) The Meetup network possesses vibrant interest-based communities both online and offline. Group tags usually serve as good indicators of the nature of a certain community. Although the online and offline network share the same user base, the former is much more closely knitted. 2) Using graph database to store network data, real time query and visual rendering is made possible on a single commodity computer. This will encourage us to conduct more graph DB based analysis and to build more related applications that benefit people.

This project is a collaborative efforts of the team Yiwei Zhang, Mengge Li, Rongyao Huang and Rahul Gaur. Yiwei contributed a great deal in data collection, the back end graph DB management and the REST API construction; the amazing dashboard visual representation is made possible largely because of Mengge, who cracked a comprehensive skill sets of HTML, JavaScript, Node.js and JQuery; Rongyao is responsible for adapting the overall system architecture, designing the visuals, implementing the d3 interactive network and writing up the report; Rahul's efforts focus on the network property analysis and community detection.

ACKNOWLEDGMENT

THE AUTHORS WOULD LIKE TO EXPRESS OUR GRATITUDE TO PROF. CHING-YUNG LIN FOR PROVIDING US GREAT SUPPORT AND ENTHUSIASTIC ENCOURAGEMENT. WE WOULD ALSO LIKE TO THANK BY KENNY BASTANI FOR DESIGN THE ORIGINAL SYSTEM ACHITECTURE AND MAKING IT OPEN SOURCE.

REFERENCES

[1] T. Sander, "E-associations: using technology to connect citizens: the case of meetup. com," 2005.

[2] A. Noack, "An energy model for visual graph clustering," in *Graph Drawing*, 2004, pp. 425–436.

[3] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[4] M. Mitzenmacher, "A brief history of generative models for power law and lognormal distributions," *Internet mathematics*, vol. 1, no. 2, pp. 226–251, 2004.

[5] S. Milgram, "The small world problem," *Psychology today*, vol. 2, no. 1, pp. 60–67, 1967.

[6] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, pp. 415–444, 2001.

[7] M. O. Lorenz, "Methods of measuring the concentration of wealth," *Publications of the American Statistical Association*, vol. 9, no. 70, pp. 209–219, 1905.

[8] X. Liu, Q. He, Y. Tian, W.-C. Lee, J. McPherson, and J. Han, "Event-based social networks: linking the online and offline social worlds," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 1032–1040.

[9] A. Lancichinetti and S. Fortunato, "Community detection algorithms: a comparative analysis," *Physical review E*, vol. 80, no. 5, p. 056117, 2009.

[10] M. Granovetter, "The strength of weak ties: A network theory revisited," *Sociological theory*, vol. 1, no. 1, pp. 201–233, 1983.

[11] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.