Capstone Project Course - Columbia University Data Science Institute

Final Report

Twitter Graph - Synergic Partners

Casey Huang, Claire Liu, Jordan Rosenblum, Steven Royce

0. Abstract

The goal of this project is to characterize and map, using network science and text mining techniques, the online Twitter conversation surrounding 'Data Science' and 'Big Data.' Specifically, we are interested in identifying Twitter influencers in a manner that would facilitate effective marketing campaigns by targeting individuals who can diffuse information in an efficient manner. To this end, the network was constructed by projecting the "retweet" and "mention" layers into one projected layer that captures both information diffusion processes. Community structures were identified in the projected network using K-Clique, Modularity, Random Walk, and the Mixed Membership Stochastic Blockmodel. We subsequently identified influencers within each community by utilizing various centrality metrics and analyzed user profiles to gain further insight into the demographics of the communities. Latent Dirichlet Allocation (LDA) was explored in our analysis to incorporate textual data into characterizing the communities. While each of the community detection algorithms we explored has its merits, for our sparse network of tweets, we found that modularity and random walk produced the most coherent communities based on user demographics and influencers. Finally, the network and user demographics of each community were represented in an interactive visualization to allow for further exploration.

1. Introduction and Project Summary

Twitter is a microblogging service with more than 300 million active users worldwide[1] and users share information via tweets, a message with 140-character limit. Social media, including Twitter, has played an important role in disseminating information[2]. The analysis

[1] http://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/
[2] Lerman and Ghosh 2010. "Information Contagion: An Empirical Study of the Spread of News on Digg and Twitter Social Networks"

of social networks can reveal individuals who are able to influence others, and detecting these influencers is important in many different fields including politics, online advertising and marketing campaigns[3]. Our work focuses on identifying these influencers for a specific topic.

Given limited resources, a challenge in a marketing campaign is identifying effective individuals to target on Twitter. We can exploit the graph properties of the Twitter network to identify these targets by using various centrality measures. However, with centrality measures alone, we run the risk of marketing to two individuals who are already well connected and potentially lose the overall effectiveness of the marketing campaign due to redundancy. Thus, this necessitates the need for understanding how information spreads among smaller groups of users, and subsequently looking at the influencers. We used community detection algorithms to partition the network in a meaningful way and looked at the various centrality measures to identify influencers. While our focus was on a marketing campaign, this approach can be extended to other fields as well.

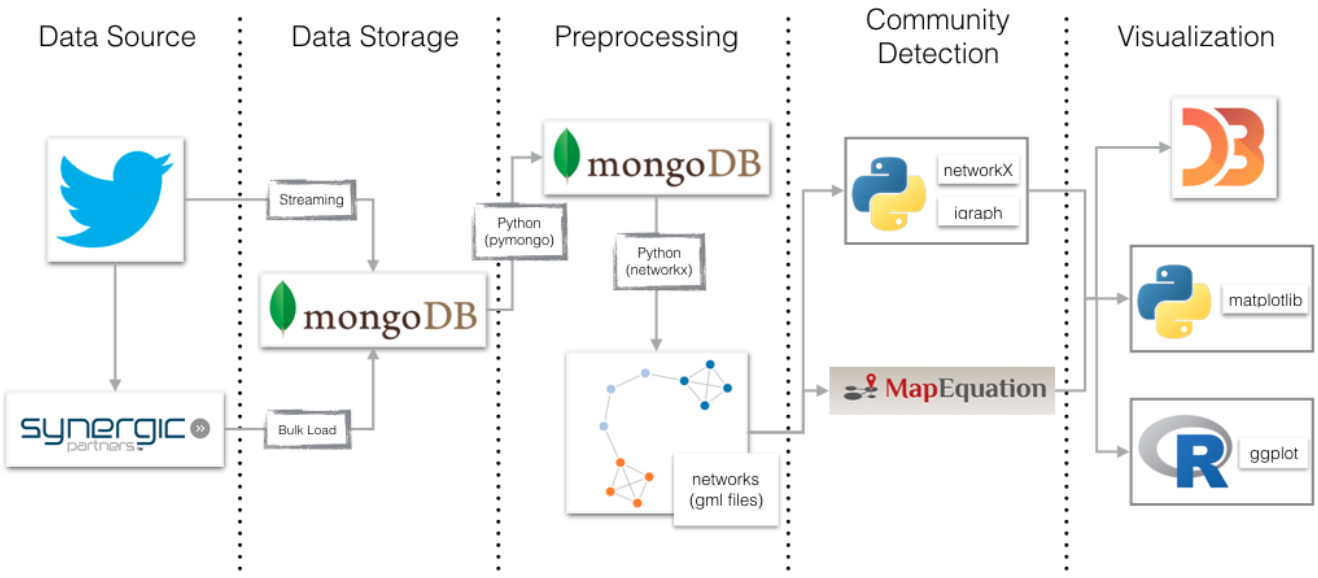


Figure 1: Project Flow

The high-level overview of the project is illustrated in **Figure 1**. The two data sources for the tweets surrounding 'Data Science' and 'Big Data' used in the project are Twitter (direct

[3] Zaman et al, 2010. "Predicting Information Spreading in Twitter"

streaming) and Synergic Partners (bulk load). The raw data were stored in MongoDB and were preprocessed in order to change the data structure from the tweet based structure to a user based structure (see **Appendix: 8.1**). Then, the preprocessed data were used to construct network files and were used as the input to the following community detection algorithms: K-Clique, Modularity, Random Walk, and the Mixed Membership Stochastic Blockmodel. The outputs were used to identify influencers and for visualizing the characteristics of the detected communities.

This reports highlights the following analysis: (i) understanding the topology and structure of the network, (ii) determining who are the influencers of the network, and (iii) examining the underlying community structure of the network.

2. Data Analysis: Tweets to Network

2.1 Data Problems and Solution

The original project objective was to characterize the conversation surrounding Columbia University on Twitter. A major issue we had to deal with tweets about Columbia University was the quality of the data. For example, tweets containing the keyword 'Columbia' could be speaking about British Columbia, the District of Columbia, Columbia Sportswear, etc. Two approaches were considered to clean the data: (i) Filtering out tweets with irrelevant keywords (e.g. South Carolina), or (ii) Removing users who cannot reach the central nodes (e.g. '@Columbia, @DSI_Columbia). Both approaches were unsuccessful given that users incorrectly mention the '@Columbia' handle (in an attempt to refer to a 'different' Columbia), thereby drawing an edge between two unrelated topics (e.g. Columbia, SC and Columbia University).

To give a general sense for the topics being discussed in our datasets, the most popular hashtags are shown in **Appendix 8.2: Table 1** (only counting a hashtag once for a particular user). Despite the cleaning steps described above for the 'Columbia' network, most of the hashtags are still unrelated to Columbia University. For example, in the table, #scflood ('South Carolina Flood' in reference to Columbia, SC) and #cdnpoli ('Canadian Politics' in reference to British Columbia) both relate to a different Columbia. On the other hand, the hashtags for the 'Data Science' and 'Big Data' topic make sense.

Given the rationale discussed above, we decided to focus on the 'Data Science' and 'Big Data' dataset for our analysis. Our data has **394,545** total tweets from **169,017** distinct users ranging from Oct. 6th - Nov. 8th, 2015.

2.2 Network Construction

The mention and retweet networks were built using the mention and retweet fields contained within the streamed data (i.e. any user in a tweet prefaced with '@' or any tweet prefaced with 'RT,' respectively). For mentions, an edge was drawn from the person mentioning to the person mentioned and, for retweets, an edge was drawn from the original user to the user retweeting (i.e. in the direction of influence). We constructed the projected graph simply by taking the union of the mention and retweet graphs. Moreover, we removed nodes with 0 degree or with edges only to itself. We did not include the follower network given Twitter API and computational limitations. Moreover, the follower layer doesn't indicate active engagement. Specifically, the projected network was built as follows:

- Nodes: It is the union of nodes in the mention and retweet networks.
- Edges: If an edge is present between two nodes in at least one of the networks, then an edge is added in the combined network. The weight of the edge is calculated by summing the corresponding edge weights in the mention and retweet networks (i.e. mentions and retweets are treated equally).

General graph properties of the overall network are displayed in **Table 1**. Overall, it is an extremely sparse network.

Table 1: General Graph Properties

Nodes	Edges	Density	Diameter	Avg. Degree	Avg. Path Length	Clustering Coefficient
125,894	169,866	1.07e-05	25	2.70	6.99	0.0030

3. Community Detection

Community detection algorithms allow us to discover the organizational properties of a network. A general network is said to have some sort of community structure if its nodes can be grouped into (potentially overlapping) subsets of nodes such that each subset forms a more densely connected community as compared to the overall network. In order to shed light on the inter-community connectedness within the 'Data Science' and 'Big Data' network, we explored 4 different community detection algorithms. They all have their advantages and disadvantages as we will discuss in the subsequent sections. A high-level summary of these methods is shown in **Table 2**.

Table 2: Comparison of Community Detection Algorithms

Algorithm	Package	Directed?	Weighted?	Overlapping?
K-Clique	networkX (python)	Undirected	Unweighted	Overlapping
Modularity	iGraph (python)	Directed	Weighted	Non-overlapping
Random Walk	Map Equation	Directed	Weighted	Both
MMSB	lda (R)	Undirected	Weighted	Overlapping

3.1 K-Clique

The k-clique percolation method[4] builds up communities by first looking at all cliques of size k (i.e. fully connected sets of k nodes) within the network. A community is then formed by combining all adjacent k-cliques. Two k-cliques are defined as adjacent if they share k-1 nodes. We utilized the k-clique community detection function in networkX. One major downside to using k-cliques, however, is that the input must be an undirected, unweighted graph for the algorithm to make sense. Hence, we lost the directionality and importance of the retweets and mentions in our network. This method does allow for overlapping communities, however. **Appendix 8.2: Figure 1** is an example of how the algorithm works.

For our graph, we decided to choose k=3 given the sparsity of edges in our network. Using k=3, we found that 90% of the nodes in our graph do not belong to any community. This is

[4] https://en.wikipedia.org/wiki/Clique_percolation_method

simply because they do not belong to a clique of size 3. Hence, 10% of nodes are placed within at least 1 community, 0.9% are placed within more than 1 community, but only 0.2% are placed within more than 2 communities. Although there are few nodes placed in many communities, the ones that are make intuitive sense. For example, @BigDataTweetBot is a member of 55 different communities. This particular user is a bot that retweets everything containing the hashtag '#bigdata.' The graph properties of the 5 largest communities are outlined in **Appendix 8.2: Table 2**. The first community is fairly large, containing 5% of the total nodes, whereas the remaining communities are quite small, containing <1% of the total nodes.

The K-clique method for our network captured one big community and numerous smaller and closer-knit communities. The characteristics of a group of users in the Netherlands (2nd largest community) detected by K-clique is shown in **Appendix 8.2: Table 3**.

We considered other values of k as well. For example, setting k=4 would only place 1.7% of the total nodes within any community at all and the communities which formed were quite small. Generally, similar communities would form as compared to k=3, only split into smaller subsets. For example, the 1,119 nodes in the largest community using k=4 all fall within the largest community when using k=3, containing 6,089 nodes.

3.2 Modularity

The Louvain method[5] belongs to a class of community detection algorithms that seek to optimize modularity. Modularity is a clustering metric that measures the goodness of a community assignment based on the comparison of the internal (within-community) edge density and the randomized null model[6]. Exact maximization of modularity is NP-hard[7]. The Louvain method is a heuristic algorithm that greedily optimizes the modularity by assigning each node to its own community and iterating over the following two steps:

1) Local optimization to form 'small communities' by measuring the change in modularity (removal of a node from its community and placing it into one of its neighboring communities)
2) Aggregations of each 'small community' into one node

[5] https://perso.uclouvain.be/vincent.blondel/research/louvain.html
[6] Leskovec et al., 2010: "Empirical Comparison of Algorithms for Network Community Detection"
[7] Brandes et al., 2006: "Maximizing Modularity is Hard", arXiv:physics/0608255

The algorithm terminates when a maximum of modularity is attained. The Louvain method is one of the most popular community detection algorithm for large networks.

For our graph, 22% of the nodes fall within the 5 largest communities. The graph properties of these communities are outlined in **Appendix 8.2: Table 4**. The user characteristics of an interesting Spanish speaking community are shown in **Appendix 8.2: Table 5**.

3.3 Random Walk and Variations

A random walker traverses a graph with the probability of picking an edge proportional to the edge weights, and a small probability to jump to any other random node (as in PageRank). Map Equation[8] is an algorithm based on encoding the random walker's movements using coding theory. It tries to denote the path by the shortest representation.

Instead of using the Huffman Coding algorithm directly, it introduces two-level encoding, which differs from simple encoding in the following way:

1) Simple encoding: give a unique name to every node in the network
2) Two-levels: parent clusters receive unique names, but the names of nodes within clusters are reused

By this two-level encoding, we are able to both obtain the shortest representation and extract important graph structures, i.e. communities.

It can be shown that finding community structure in networks and minimizing the description length of a random walker's movements form a dual problem. The intuition of this duality is straightforward: encoding can be compressed if the network has regions/communities in which the random walker tends to stay for a long time.

We tried three variations of this method:

1) Two-level overlapping community structure
2) Two-level non-overlapping community structure

[8] Maps of information flow reveal community structure in complex networks (Martin Rosvall and Carl T. Bergstrom)

3) Multi-level non-overlapping community structure (achieved by recursively applying two-level algorithm) where communities are in tree structure and a supermodule could have several submodules.

The graph properties of the 5 largest communities are outlined in **Appendix 8.2: Tables 6 - 8**. We use the top level community assignment in the multilevel tree to be compute the statistics for multilevel model. Since it allows a sub-community structure, the largest community by multilevel accounts for 19% of the total nodes. For both two-level models, the community sizes are smaller. The user characteristics of an interesting community from France talking about big data technology for random walk multilevel are shown in **Appendix 8.2: Table 9**.

3.4 Mixed Membership Stochastic Blockmodel

The Mixed Membership Stochastic Blockmodel (MMSB)[9] is a probabilistic overlapping community detection algorithm. It captures the fact that a particular node might be explained by its membership in several overlapping groups, a property that is essential when analyzing real world networks. It models the probability that two nodes might be connected to each other as high if they share a similar membership vector. Using it, we get the community distribution for each user.

One of the advantages of probabilistic community detection is that it enables us to analyze to what extent a given user is shared among different communities. This measure is called bridgeness[10]. Intuitively, a user that belongs to only one of the communities has zero bridgeness, while one that belongs to all of the communities to exactly the same extent has a bridgeness of 1. The bridgeness is defined as the distance of its membership vector from the reference vector in Euclidean vector norm, inverted and normalized to the interval [0,1]. The users with the top ten bridgeness scores between communities are displayed in **Appendix 8.2: Table 10**.

[9] P. K. Gopalan and D. M. Blei. Efficient discovery of overlapping communities in massive networks. *Proceedings of the National Academy of Sciences*, 110(36):14534–14539, 2013.
[10] Tamás Nepus and Andrea Petróczi. Fuzzy communities and the concept of bridgeness in complex networks. *PHYSICAL REVIEW* E 77, 016107 2008

4. Results and Discussion: Comparing Methods

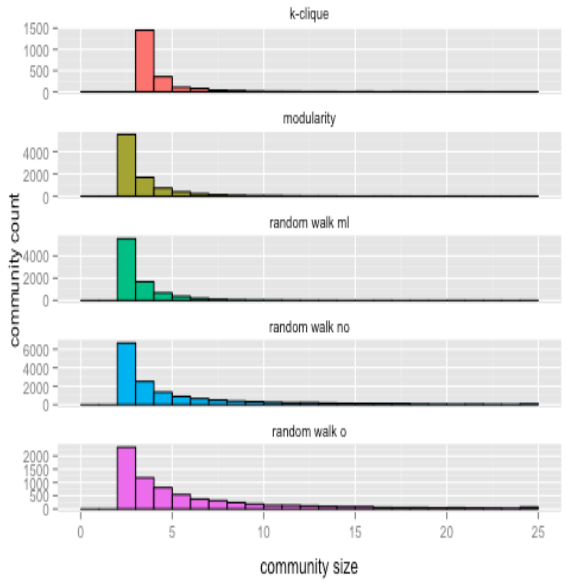
We employed a number of different measures to understand how communities are formed by the different algorithms. We looked at (i) size distributions of the communities, (ii) percent of outgoing edges from communities, (iii) overlap of communities between different methods, (iv) number of communities each node belongs to for the overlapping methods, (v) intra-community structure, and (vi) influencers within the largest communities.

4.1 Community Size Distribution

The community size distributions are shown in **Figure 2**. The shapes of the distributions for modularity and all three random walk implementations are similar to one another: most of the communities detected by these algorithms fall between 2 and 3 users. For K-clique, most communities fall between 3 and 4 users which makes intuitive sense based on the input of k=3.

We also explored how well these algorithms divide users to different communities. In order to measure this, we looked at how many of the top communities would be needed to account for 75% and 90% of the total users in the network (shown in the table in **Figure 2**). The result shows that communities formed by modularity and random walk multilevel are top-heavy: less than 2% of communities are needed to account for 75% of the users in the network.

Community size distribution



Community size overview

Algorithm	Number of Communities	75% of users	90% of users
K-clique	2,135	696 (32%)	1,556 (72%)
modularity	9,133	87 (1%)	3,085 (33%)
random walk ml	9,124	196 (2%)	3,077 (33%)
random walk no	15,872	4,327 (27%)	9,581 (60%)
random walk o	7,574	1,797 (24%)	4,159 (55%)

*total users (for each algorithm) is defined as the total number of distinct users across all communities

Figure 2: Community size distribution and overview

4.2 Inter-Community Connectedness

We also explored the inter-community connectedness for each of the 5 methods (**Figure 3**). This was accomplished by counting the number of edges leaving a given community as a percentage of the total number of edges in that community[11]. By definition of community detection, we want this percentage to be lower. For modularity and all of the random walk methods, this is the case. For K-clique, however, there are more communities with a high percentage of outgoing nodes. Intuitively, cliques are hard to form so nodes will talk to other people outside of their cliques as well. Since our graph is fairly sparse, many 3-cliques will have outgoing edges to nodes which are not in an adjacent k-clique. Hence, K-clique is not the correct community detection method to use for our network.

[11] Total outgoing edges / (Total outgoing edges + Total edges within community)

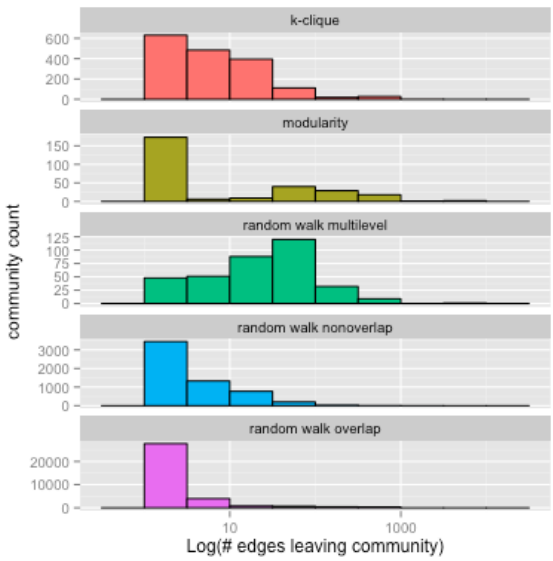
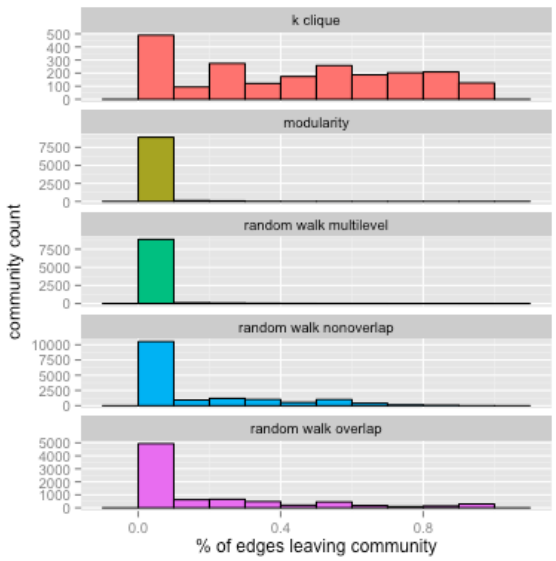
Total # of edges leaving community	Percentage of edges leaving community
	

Figure 3: Histograms of outgoing edges from each community (communities with zero outgoing edges are not included in the figure)

4.3 Overlapping Communities of Different Methods

In order to get a qualitative sense for how the different community detection algorithms compare with each other, we analyzed the overlap of some of the largest communities. Firstly, we found that the largest communities for each of K-clique, modularity, and random walk multilevel have quite a bit of overlap between the 3 methods. Moreover, the top 2 communities for modularity are both subsets of the largest community for random walk multilevel (i.e. modularity splits the largest community in random walk multilevel into a few smaller subsets). Lastly, we found that a community largely consisting of users from France forms in both random walk multilevel and in modularity. These results are presented in **Figure 4.**

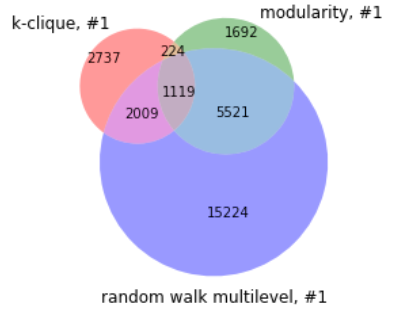
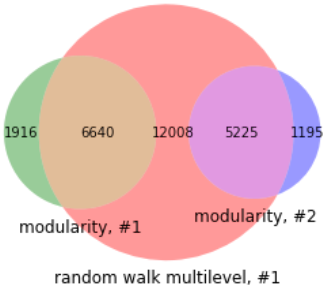
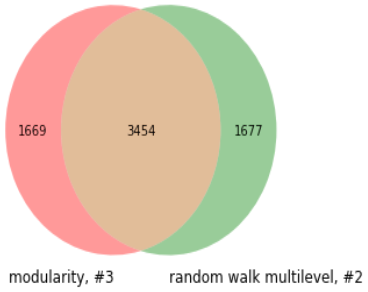
Largest community overlap	Largest random walk (19%) vs. top 2 modularity	France community forms in random walk and modularity
		

Figure 4: Overlapping communities of different methods

4.4 Number of Communities each node belongs to

For the overlapping methods, we analyzed how many nodes were members of multiple communities. **Table 3** illustrates these findings. Interestingly, K-clique results in only 10% of nodes belonging to a community at all and even fewer belonging to multiple communities. Random walk overlap places all nodes within a community, however, many of these communities have only 1 member.

Table 3: Overlapping community membership

Overlapping Method	> 0 communities	> 1 community	> 2 communities
K-clique	12,653 (10.1%)	11,53 (0.9%)	281 (0.2%)
Random Walk Overlap	125,894 (100.0%)*	11,338 (9.0%)	4,220 (3.4%)

The most interesting part of these results is looking at users which belong to the most number of communities. We can use this to determine who serves as a bridge between communities. In both methods, '@BigDataTweetBot,' a user we mentioned in detail earlier in the paper, belongs to the most number of communities (55 for K-clique and 3,122 for random walking overlap). This is encouraging since this bot also scores highest in bridgeness score within the Mixed Membership Blockmodel discussed earlier. Other users who are present in the most number of communities are less consistent between methods but there still is quite a bit of overlap. They include LinkedIn, Forbes, and YouTube.

4.5 Intra-community Structure

The clustering coefficient of a graph is a measure of how closely the nodes are connected together. It ranges from 0 to 1, where 0 means the graph has a star structure, and 1 means it's a clique.

Among the community detection methods used, k-clique formed communities that are generally denser than those of other methods. This is expected based on how K-clique forms communities. Modularity gives communities that are more loosely knitted, but not as much as Random Walk's result. The 3rd largest community in Multilevel Random Walk has a clustering coefficient of 0 (a star), where all 3,664 users in that community were retweeting the same tweet about the founder of DataSift stepping down as CEO. Interestingly, the communities detected by two-level Random Walk in general are more like stars. The top 3 communities for nonoverlapping and the top 5 for overlapping are exact star structure graphs or very close to it. For those communities, the single user that got retweeted by everyone else is basically the influencer of the community.

4.6 Influencers

In order to measure the importance of a user within a given community, we considered several centrality metrics[12]:

- Degree - "How many connections does this person have?"
- Betweenness - "How likely is this person to be in the shortest path between two other people in the network?"
- Closeness - "How fast can this person reach anyone in the network?"
- Eigenvector - "How well is this person connected to other well-connected people?"

For each community detection algorithm with the exception of K-clique, which we decided does not make sense for our network, we computed the top most influential user for each of these centrality metrics. The results can be found in **Appendix 8.3: Tables 11 - 14**. In general, the most influential users make sense and there is usually an overlap between the various centrality metrics.

[12] High-Performance Big-Data Analytics: Computing Systems and Approaches (pg. 380)

Based on the clustering coefficient of a particular community, different centrality metrics should be used. For example, betweenness and eigenvector centralities don't make sense for communities which are stars (i.e. clustering coefficient of 0) with directed edges radiating from the only center. This is because there is no path from the outer nodes to any other node. An example of this can be seen in community 1 of random walk non-overlapping (**Appendix 8.3: Table 13**).

5. Exploratory Work: LDA for Tweets

In order to characterize the conversation between communities, we applied LDA, which is a Bayesian probabilistic model aiming to analyze a massive collection of documents. The model is defined in a generative way: Each document is treated as grouped data. Topics, as a distribution over vocabulary, are shared across documents. The topic proportions vary from group to group. Each word in a document is generated from its topic proportion. Here, we treat each tweet as a document and used the python package Gensim to train a LDA model using 20 topics, 1 pass over the data, and 100 iterations. The example topics are listed in **Table 4.** We can clearly see topic 1 as "distributed computing systems," topic 2 as "hiring data scientist" and topic 3 as mainly foreign languages. We do observe less coherent topics (e.g. topic 4).

Table 4: Overlapping community membership

Topic 1	Topic 2	Topic 3	Topic 4
hadoop	jobs	gran	business
amp	hiring	empleo	intelligence
spark	san	sistema	bigdatablogs
jose	silicon	portafolioco	central
applications	jose	marcas	artificial
computing	amp	turismo	startups
wellness	services	sectur	banking
mit	machine	herramienta	seattle
generador	predictive	informaci	marketing
solutions	engineer	datos	human

Ideally, LDA can both characterize communities and potentially give a sense about how separable those communities are using different community detection algorithms. Despite some of the topics not being coherent enough, we tried to integrate tweets by communities to get the topic proportion for top communities. The topic distribution of the top five communities using nonoverlapping random walk can be found in **Figure 5**.

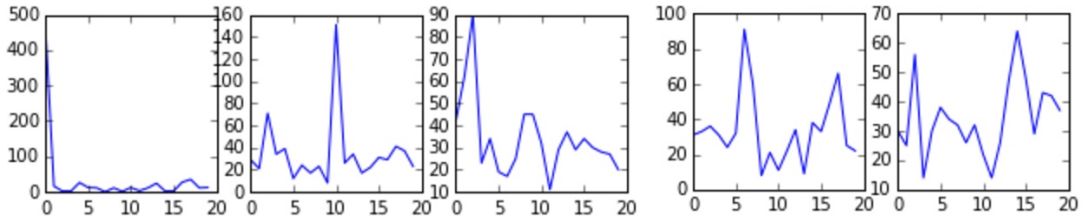


Figure 5: Topic distributions of top 5 communities (random walk nonoverlapping)

Though we didn't list all the top topics for each community here, we think it's promising given the varying topic proportions over communities. The reason for having ambiguous topics might be because of the limited performance of LDA on short length microblogs like tweets. The topic quality could potentially be improved by aggregating tweets by users over a longer time period of tweets (e.g. 1 year). Based on coherent topics, we could further:

- characterize communities using their conversations;
- potentially measure community quality by calculating similarities based on inferred topics.

6. Visualization

The visualization we made has two purposes:

- Give a straightforward view of the inter-community structure given a partition method
- Offer an interface to let people easily explore the user profiles and conversation in each community

The interactive visualizations we made based on these two goals are located at the following links and a sample screenshot can also be found in **Appendix: 8.4**:

- K-clique[13]
- Modularity[14]
- Multilevel Random Walk[15]

Since the sizes of communities decay rapidly, only the ~100 largest communities and corresponding edges among them are displayed. One limitation of this visualization is that the intra-community structures can't be shown at the same time.

7. Future Work

As discussed in the LDA section above, the amount of data we have is too small to construct meaningful topics. Ideally, we would want 1 year of tweets to work with. This would allow us to perform a comparison of meaningful topic distributions for top communities, providing another way for us to evaluate community quality for different community detection algorithms.

More analysis is also needed on other community detection algorithms. For example, the Mixed Membership Stochastic Blockmodel can be looked at in more depth to explore probabilistic membership vectors of users and to improve influencer detection by integrating the bridgeness score with a centrality metric.

To further combine the two overall approaches used (i.e. (i) text and (ii) network analysis), we found an implementation in SNAP that takes both edge structure and node attributes into account[16]. If we could aggregate the tweets on users, we would be able to integrate both approaches by extracting features from the user tweets to be the node attributes.

Lastly, streamlining the data collection and the processing workflow would be an essential next step. In addition, a natural extension to this work would be building a real-time data processing platform based on scalable algorithms. These will be necessary as the timeframe of the data increases.

[13] http://casey-huang.neocities.org/twittergraph/kclique/forcedirected.html
[14] http://casey-huang.neocities.org/twittergraph/modularity/forcedirected.html
[15] http://casey-huang.neocities.org/twittergraph/rw_multilevel/forcedirected.html
[16] https://github.com/snap-stanford/snap/tree/master/examples/cesna

8. Appendix

8.1 Storing the tweets and users in MongoDB

The raw data was stored in MongoDB in a tweet oriented table. We preprocessed it and stored the graph in a user oriented view where each entry of a user has the following structure:

```
{
	userid:
	screen_name:
	name:
	description:
	location:
	statuses_count:
	followers_count:    /* number of 'followed' */
	friends_count:    /* number of 'following' */
	…
	mentions: {uid1: [tid1, tid2, …], uid2: [tid1, …], … }
	retweets: {uid1: [tid1, tid2, …], uid2: [tid1, …], … }
}
```

The fields "mentions" and "retweets" are dictionaries that store all the tweet ids in which this user mentions/retweets other users, grouped by the user id.

8.2 Supplementary Figures and Tables

Table 1: Top 5 unique hashtags by user[17]

'Columbia' (after filtering)	'Data Science' + 'Big Data'
columbia	bigdata
scflood	analytics
columbiagivingday	data
elxn42	datascience
cdnpoli	iot

[17] Hashtag only counted once for any given user

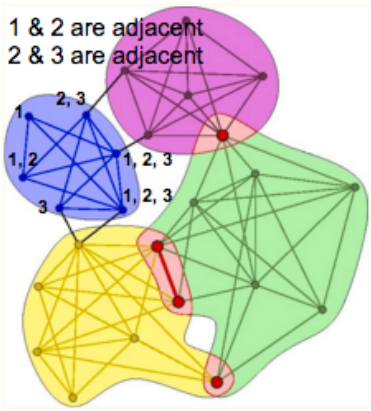


Figure 1: Illustration of K-Clique Communities with k=4

Table 2: General Community Properties from K-clique

Community	Nodes	Edges	Density	Diameter	Avg. Degree	Avg. Path Length	Clustering Coefficient
1	6,089 (5%)	29,146	0.00079	17	9.57	4.94	0.01800
2	43 (<1%)	149	0.08250	4	6.93	2.22	0.29363
3	26 (<1%)	72	0.11077	3	5.54	1.50	0.26655
4	24 (<1%)	84	0.15217	4	7.00	1.92	0.41423
5	23 (<1%)	52	0.10277	5	4.52	2.12	0.28261

Table 3: Characteristics of 2nd largest community (with 43 nodes) from K-clique

Location (user description)	count	Top words (user description)	count	Top hashtags (from tweets)	count
Amsterdam	5	van	9	#bigdataBK[18]	112
Noord-Brabant[19]	4	noord-brabant	8	#CloudComputing	41
Tilburg[20]	3	brabant	8	#Hadoop	27
Europe	2	provincie	6	#BigData	26
Eindhoven[21]	2	data	5	#NoSQL	21

[18] #bigdataBK relates mostly to the website: http://brabantkennis.nl/, which is a regional platform where data-driven knowledge about Noord-Brabant is collected.
[19] Noord-Brabant is a province in the south of the Netherlands.
[20] Tilburg is a city in Noord-Brabant.
[21] Eindhoven is the largest city of Noord-Brabant.

Table 4: General Community Properties from Modularity

Community	Nodes	Edges	Density	Diameter	Avg. Degree	Avg. Path Length	Clustering Coefficient
1	9,145 (7%)	13,132	0.00016	12	2.87	4.78	0.00306
2	6,681 (5%)	11,130	0.00025	13	3.33	4.41	0.00312
3	5,055 (4%)	6,545	0.00026	26	2.59	8.48	0.00716
4	4,151 (3%)	5,305	0.00031	22	2.56	6.60	0.01453
5	3,613 (3%)	4,365	0.00033	27	2.42	7.84	0.00854

Table 5: Characteristics of 5th largest community (with 3,613 nodes) from Modularity

Location (user description)	count	Top words (user description)	count	Top hashtags (from tweets)	count
Madrid	347	marketing	356	#BigData	1173
Barcelona	217	digital	209	#BDS15[22]	361
España	121	tecnolog	178	#coebigdata[23]	216
México	51	data	165	#Marketing	114
Spain	41	comunicaci	159	#OtraformadeverTV[24]	111

Table 6: General Community Properties from Random Walk Multilevel Nonoverlapping

Community	Nodes	Edges	Density	Diameter	Avg. Degree	Avg. Path Length	Clustering Coefficient
1	23,984 (19%)	40,433	0.00007	16	3.37	5.54	0.00512
2	3,702 (3%)	4,582	0.00033	23	2.48	6.88	0.00590
3	3,664 (3%)	3,664	0.00027	3	2.00	1.02	0.0
4	2,498 (2%)	3,049	0.00049	14	2.44	5.30	0.01225
5	1,721 (1%)	1,880	0.00064	10	2.18	3.34	0.00045

[22] #BDS15 mostly refers to the Big Data Spain 2015 conference (http://www.bigdataspain.org/)
[23] #coebigdata mostly refers to the organization Big Data CoE (Center of Excellence) based in Barcelona (http://www.bigdatabcn.com/)
[24] "Otra Forma De Ver TV" is the discussion topic brought by a Spanish company Sentisis (http://sentisis.com/) about marketing via social media versus tv.

Table 7: General Community Properties from Random Walk 2-level Nonoverlapping

Community	Nodes	Edges	Density	Diameter	Avg. Degree	Avg. Path Length	Clustering Coefficient
1	3,573 (3%)	3,572	0.00028	1	2.00	1.0	0.0
2	1,627 (1%)	1,640	0.00062	3	2.02	1.91	0.00001
3	1,601 (1%)	1,600	0.00062	3	2.00	2.00	0.0
4	1,193 (1%)	1,614	0.00113	5	2.71	2.44	0.00154
5	995 (1%)	1,026	0.00104	5	2.06	2.37	0.00009

Table 8: General Community Properties from Random Walk 2-level Overlapping

Community	Nodes	Edges	Density	Diameter	Avg. Degree	Avg. Path Length	Clustering Coefficient
1	1,655 (1%)	1,671	0.00061	3	2.02	1.99	0.0
2	1,618(1%)	1,627	0.00062	2	2.01	1.91	0.0
3	1,415 (1%)	1,446	0.00072	4	2.04	2.19	0.00005
4	1,222 (1%)	1,240	0.00083	4	2.03	2.01	0.00004
5	685 (1%)	695	0.00148	2	2.03	1.89	0.00003

Table 9: Characteristics of 2nd largest community (with 3,702 nodes) from Random Walk Multilevel

Location (user description)	count	Top words (user description)	count	Top hashtags (from tweets)	count
Paris	550	digital	428	#BigData	1893
France	165	marketing	384	#data	372
Lyon	44	innovation	288	#BlogBD[25]	228
Nantes	35	communication	204	#in	216
Bordeaux	31	r	202	#FrenchTech	196

[25] #BlogBD mostly refers to the blog of a e-Business consulting company Business&Decision (http://blog.businessdecision.com/) whose headquarters is in Paris.

Table 10: Top influencers by bridgeness score

Top influencers	Bridgeness
BigDataTweetBot	0.9965
recuweb	0.9764
Ronald_vanLoon	0.9484
KirkDBorne	0.9399
analyticbridge	0.9058
innova_scape	0.8980
ClearGrip	0.8975
7wdata	0.8818
EvanSinar	0.6508
BernardMarr	0.6431

8.3 Influencers

Table 11: Influencers: Modularity

Community	Degree	Betweenness	Closeness	Eigenvector	Notes
1	orangebusiness	orangebusiness	orangebusiness	orangebusiness	[1]IT and communications services provider
2	infosecjerk[1]	JavierHAres	infosecjerk[1]	infosecjerk[1]	[1]Cyber security
3	FMarlidio[1]	Teligo[1]	The_Shelby_Neil	DrJessupS	[1]Spanish speakers
4	sitebymack[1]	sitebymack[1]	craigwilson	excontinent	[1]Web development and consulting
5	EmSixTeen	umbrant[1]	umbrant[1]	chris__nellis	[1]Software enginner at Cloudera

Table 12: Influencers: Random Walk Multilevel

Community	Degree	Betweenness	Closeness	Eigenvector	Notes
1	marcrojek[1]	Central_Colo[2]	marcrojek[1]	marcrojek[1]	[1]IoT enthusiast [2]Data centers
2	siemensindustry[1]	Remnant_Shadows[2]	giweqojyviku	RT_insurance	[1]Industrial tech [2]Programmer
3	o2mc[1]	snsharma[2]	o2mc[1]	cmphksar	[1]Big data ops [2]Applied stats
4	RonZimmernPHG	GautierLudovic[1]	GautierLudovic[1]	runitbymom	[1]Big data at Oracle France
5	jeremybowers[1]	Cassand00141441	Cassand00141441	Cassand00141441	[1]Interactive news at NYT

Table 13: Influencers: Random Walk Nonoverlapping

Community	Degree	Betweenness	Closeness	Eigenvector	Notes
1	MichelleOckers	ClementineWong	MichelleOckers	ClementineWong	Both from Sydney, AU
2	claudiusvr	claudiusvr	claudiusvr	claudiusvr	Big data and analytics technology advisor
3	Flavia_Flavv	Flavia_Flavv	Flavia_Flavv	MaddieWeirrrrrr	N/A
4	2MikeGarr[1]	MartaVilaRigat	2MikeGarr[1]	2MikeGarr[1]	[1]Digital media in natural sciences
5	readflowerchild	readflowerchild	readflowerchild	readflowerchild	Scientific healing

Table 14: Influencers: Random Walk Overlapping

Community	Degree	Betweenness	Closeness	Eigenvector	Notes
1	moha_doha[1]	moha_doha[1]	moha_doha[1]	multimedia_2016	[1]Author of novels and nonfiction
2	MediaMetricsGal	MediaMetricsGal	MediaMetricsGal	MediaMetricsGal	Analytics leader at Visa
3	multitaction	multitaction	multitaction	multitaction	Manufacturer of multitouch tech.
4	ofereliassaf	ofereliassaf	ofereliassaf	ofereliassaf	Software and Bitcoin enthusiast
5	NETSERPENTS	NETSERPENTS	NETSERPENTS	NETSERPENTS	Software dev. company

8.4 Visualization