Social Computing
Final Project

Topical Topology: How Topics Determine Structure in Social Networks

Sean Chen
Sebastian Benthall

Fall 2011

Background

There is a lot of existing research about social network structure. But what about social network content? How does it relate to structure? Suppose I'm a blogger and I want to maximize my followers, should I cover a broad range of topics, or one topic deeply? If I want to advertise or send a political message, how can I identify and target communities of interest?

In this research, we use topic modeling to characterize the content of social network (Twitter) data, and look for relationships between topics and structure.

Plan

- We are narrowing our study to Twitter data.
- Use Latent Dirichlet Allocation (LDA) topic modeling techniques to identify topics in a Twitter data set.
- Using the topic model, infer the topic distribution of Twitter actors
- Mine for relationships, starting with topic entropy (see below)

Topic Model and Topic Entropy

A document (like a tweet, or collection of tweets) is a collection of words. A topic is a probability distribution over words. A document can be thought of as a being generated from latent topics. Looking at documents in this way finds latent semantics despite ambiguities in words

For this study, we used the implementation of Latent Dirichlet Allocation (LDA) in the MALLET software package. MALLET performs both topic modelling and topic inference functions.

For the first iteration of our study, we defined a new measure on a collection of documents, **topic entropy**. Defined for a given topic model, the topic entropy of a collection of documents is the Shannon entropy of the distribution of topics inferred on those documents. For this measure, we sum and normalize the topic distributions of individual documents to get the distribution for the entire collection.

This lets us investigate the topic entropy of a Twitter user, who is associated with a collection of documents (tweets). We investigated the relationship between topic entropy and number of followers.

Process

1. Target a Twitter account.
2. Do a snowball crawl.
    a. Get the metadata of that account, including number of followers.
    b. Get the connecting accounts (friends or followers of the target).
    c. Crawl all or part of the connecting accounts.
    d. Repeat the loop for *h* hops.
3. For each account, fetch the latest 200 tweets.
4. LDA topic model
    a. For each tweets, remove noise text and stop words
    b. feed the data to LDA library
5. LDA topic inference
    a. For each tweet, generate a percentage value for each topic.
6. Analyze results
    a. For each user, calculate the value for each topic.
    b. Normalize the value and calculate entropy.
    c. generate a matrix of user topic entropy vs number of followers.

Source code for our study is available on-line at:
https://github.com/sbenthall/topical-topology

Problems Faced

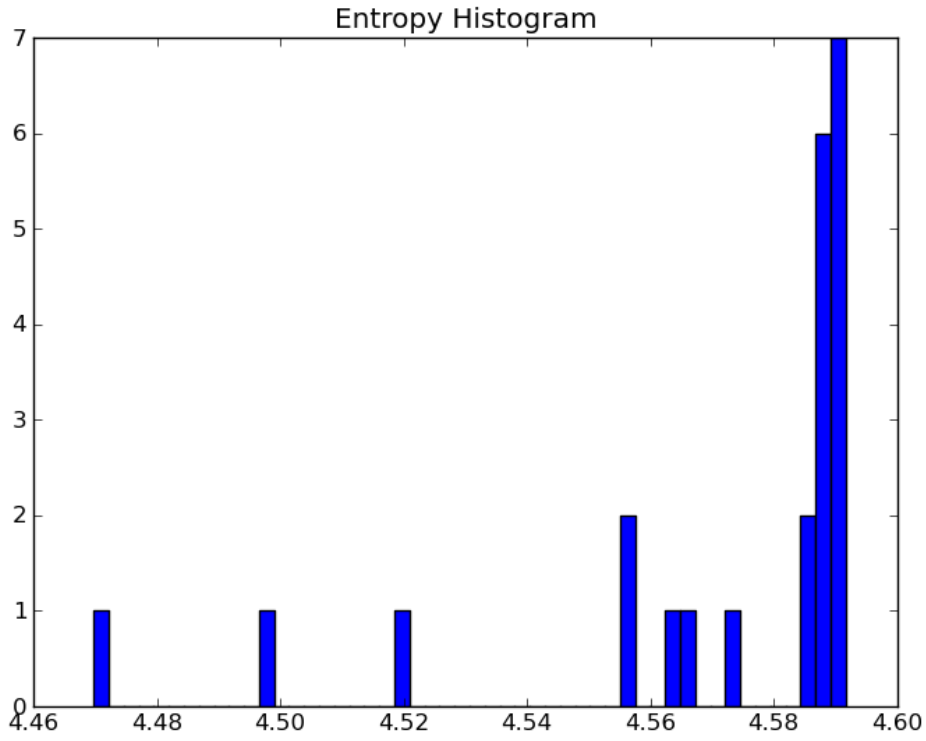
- In data collection, we ran up against the Twitter rate limit very quickly, so we had to implement authentication and caching.
- It was hard to determine the way to do snowball. If we collect every friend of an particular account, it'll be too many. So we set a variable to only collect the first N friends.
- There was special content that made our data inconsistent, for example, the carriage return character, '/r', that we had to detect and remove.

Current Result

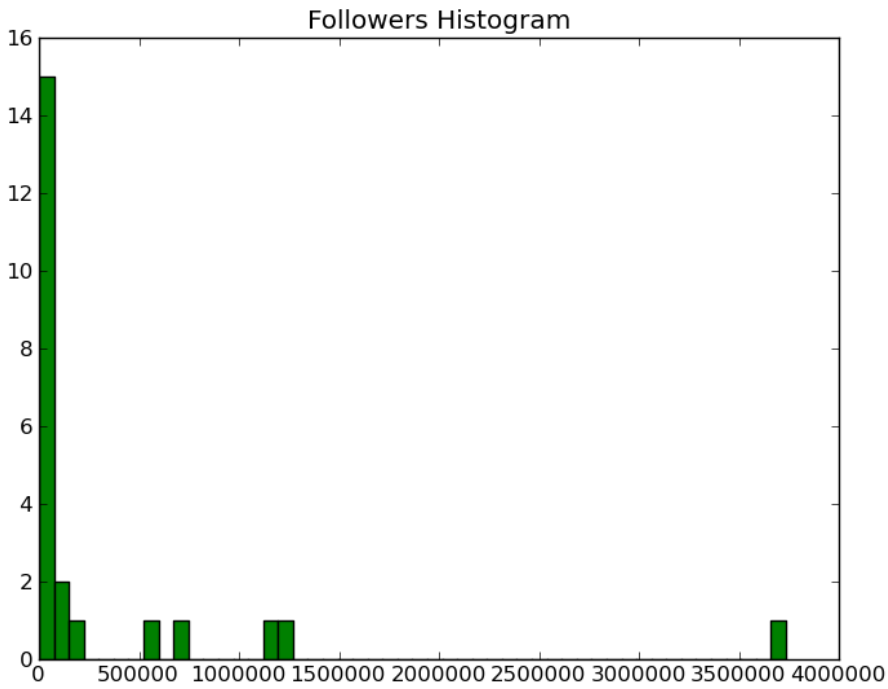
At the culmination of the process described above, we had a matrix of users and their topic distributions available as a numpy array.

```
[[ 0.00796326  0.00884478  0.00938496 ...,  0.00778552  0.0069295
   0.00826579]
 [ 1.64016913  1.69416177  1.6761334  ...,  1.42299259  1.53918118
   1.59262247]
 [ 0.06006956  0.06988641  0.07079396 ...,  0.05872886  0.05543881
   0.06235165]
 ...,
 [ 1.73257577  2.08378136  2.00727876 ...,  1.75601979  1.5550272
   1.77695223]
 [ 0.38401505  0.32146591  1.11315765 ...,  0.65818287  0.23100552
   0.31756009]
 [ 0.98929768  1.04338489  1.03878116 ...,  0.94086357  0.82182116
   1.01169468]]
```

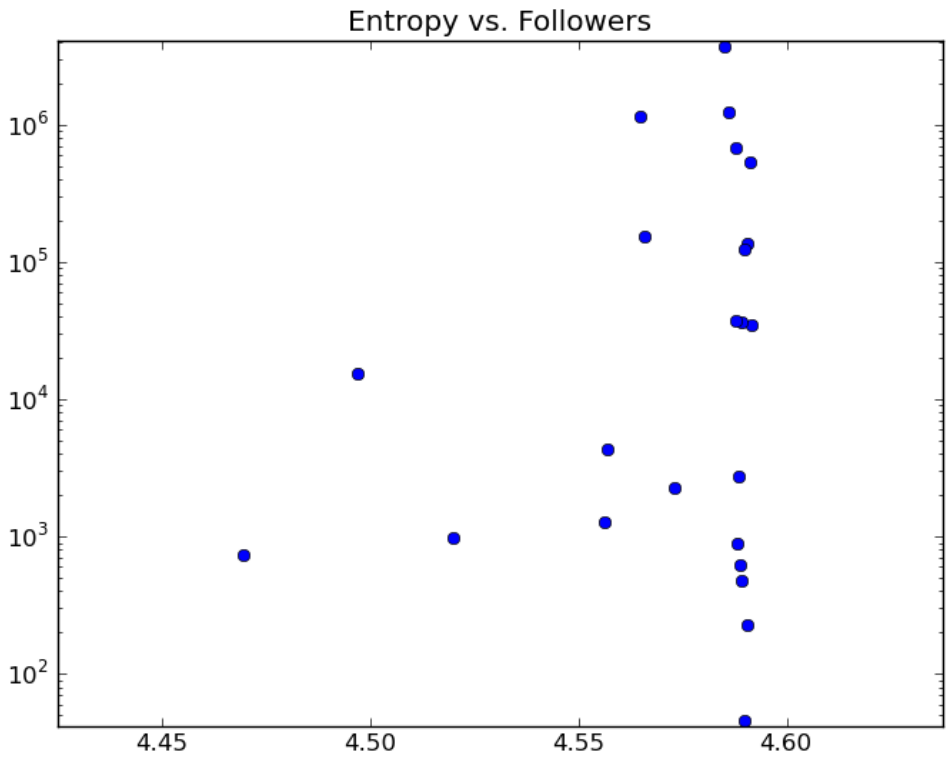
Surprisingly, we found that most of the users in the study had a similar topic entropy. We don't know why this is.



Meanwhile, the number of followers per user was predictably characterized by a heavy tail distribution.



Plotting the topic entropy and the number of followers (on a log scale) reveals...very little. The concentration of twitter users around the maximum of the topic entropy range makes it difficult to establish any relationship between number of followers and topic entropy. Further work is needed.



Analysis

- We filtered out the users with less than 200 tweets. Though this was motivated because we thought that the computed topic entropy would be more comparable across users if each user had the same number of tweets in the data set, this may hide some real effects in the data.
- Topic entropy might not represent the content well. It is a very formal measure that is entirely insensitive to the topics involved. It's possible that, for example, particular topics are more highly correlated with popularity.
- We did not look at account metadata or structural information beyond number of followers.
- These results can only be considered preliminary, due to data sparsity. Our main accomplishment for this iteration has been the development of the tool chain that we can now use for more in-depth analysis.

Next steps

- Collect more data. Our preliminary results were run for a single snowball of two hops centered on one of the reseachers. This is not a representative sample. In future iterations we will expand our data set.
- Modify the filter to people with 50 or more tweets instead of 200, and see if this has an effect.
- Add ‘frequency’ into our analysis. Did the user tweet 200 times in a single week, or were the tweets spread over a whole year? We can quantify this and include it as a factor in our analysis.
- See whether aggregating all tweets of a user into a single document before running the topic model has an effect on the results. It is possible that tweets are too short to effectively analyze with a generic topic modeling algorithm.
- Use the weight of the individual topics per user, rather than aggregating them into ‘topic entropy’.
- Run the analysis varying the number of topics inferred. This is a free parameter in the modeling algorithm.
- We have not looked at the relationships *between users* at all in the preliminary analysis. However, we could import the matrix representation of the data snowball’s connectivity into the data set under analysis.

In general, we will be using the tools developed in the first iteration to build out data sets and then explore the data with clustering (such as k-means) and anomaly detection algorithms.