A Statistical Analysis of a Time Series of Twitter Graphs

David J. Marchette

October 24, 2014

Acknowledgment: This work funded in part by the NSWC

In-House Laboratory Independent Research (ILIR) program.

NSWCDD-PN-14-00532 Distribution A: Approved for Public Release

Topics

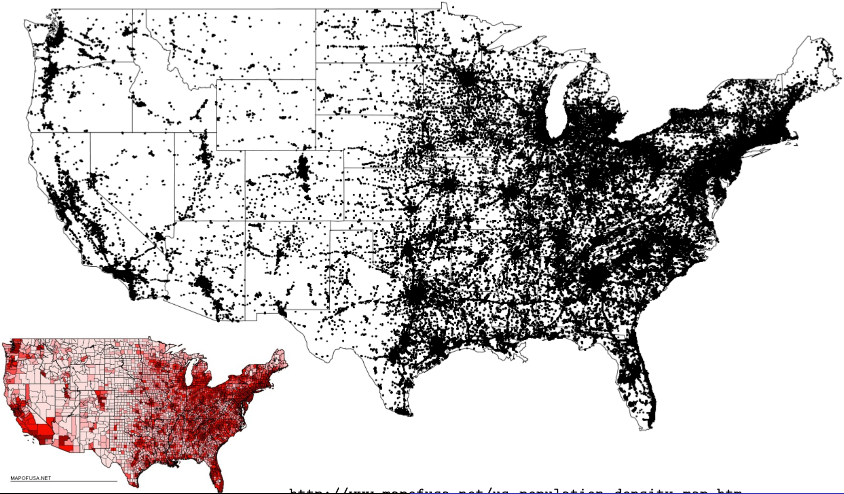
Collaborators

- Elizabeth Hohman, NSWCDD.
- Stephen Davies, University of Mary Washington.
- Ethan Novak, Michigan Technological University.

Obtaining the Data

- The Twitter API (`https://dev.twitter.com/docs/api/streaming`) provides access to (a subset) of all tweets matching a query.
- For this project we placed a rectangle around the continental United States (lower 48), and collected all tweets with a geo-location in the rectangle. Caveats:
  - Twitter puts an upper limit on the number of tweets.
  - Experiments where we tweeted out from randomly chosen locations indicated that we rarely hit the limit. Further experiments bears this out.
  - The geo-locations are only available if the device is set to provide the location. A small fraction of individuals do so.
  - As our experiment indicated, this location can be spoofed.
  - Power and network outages occur with annoying regularity.
- The data collection started on Jan 02, 2013, and is ongoing.

May 2, 2013, 1.7 Million Tweets



http://www.mapofusa.net/us-population-density-map.htm

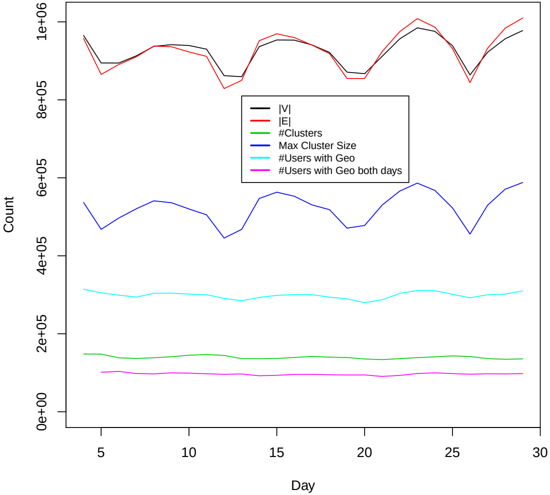
The Mentions Graph

- There are (at least) two graphs one might mean when discussing "the Twitter graph":
  - The Friends/Followers graph: a directed edge from A to B if A follows B.
  - The mentions graph: a directed edge from A to B if A mentions B in a tweet.
- In either case the graphs are dynamic.
  - People start/stop following other people.
  - Who you mention in one time period may be different than in other time periods.
- Note that in either case there is a time interval defining the graph, or alternatively the graph is dynamic, with edges appearing (and presumably disappearing after some time).
- We will be concerned only with the mentions graphs.

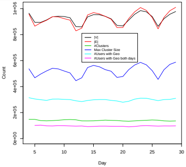
Day Graphs

- For each day in April, 2014, we construct a graph consisting of all users who mention another user at any time in the day. Due to power outages, we only consider April 4-29.
- There is a directed edge from each user to each of the users they mention.
- We do not keep track of whether the mention is a single or multiple: “hey @buddymine wanna go 2 lunch” vs “hey @friend1 @friend2 @friend3 lets get together”
- Just like with emails, one could construct a hypergraph containing this information. We do not.
- We also do not retain edge multiplicity in the work I will describe. One mention in a day is the same as 20 mentions – although I will discuss an exploitation task that utilizes this information.

Statistics


|V|
|E|
#Clusters
Max Cluster Size
#Users with Geo
#Users with Geo both days
Count
0e+00
2e+05
4e+05
6e+05
8e+05
1e+06
Day
5
10
15
20
25
30

Comments

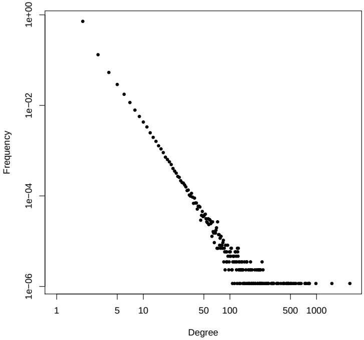


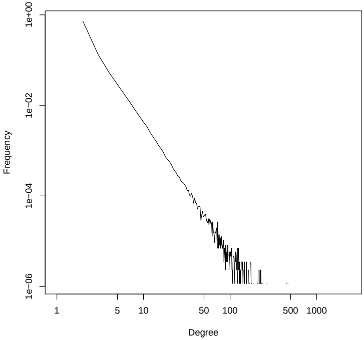
- Note the seasonality (weekly-ality?).
- The graphs are extremely sparse (roughly one edge per vertex).
- There's a huge connected component, and tons of smaller ones.
- The numbers of vertices and edges don't show the full picture.

The Degree Distribution

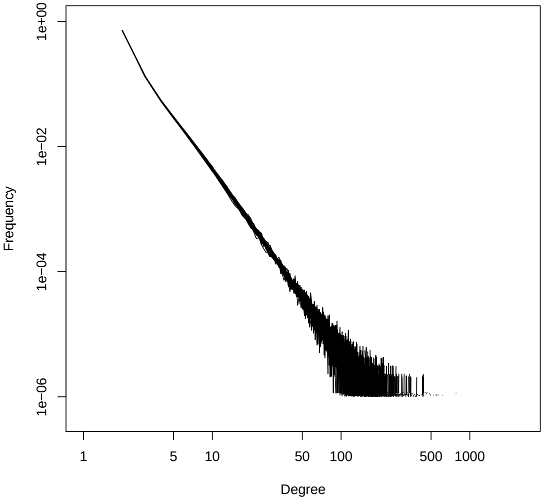
- Degree distributions are used to get a feel for the gross structure of a graph.
- As a visualization tool, one plots the degree versus the number of vertices with that degree, usually on a log-log scale.
- If the plot is linear (generally ignoring the two end-points of very low and very high degree vertices) we say that the graph is a power-law graph.

Degree Distribution for April 13





Degree Distributions for All Days

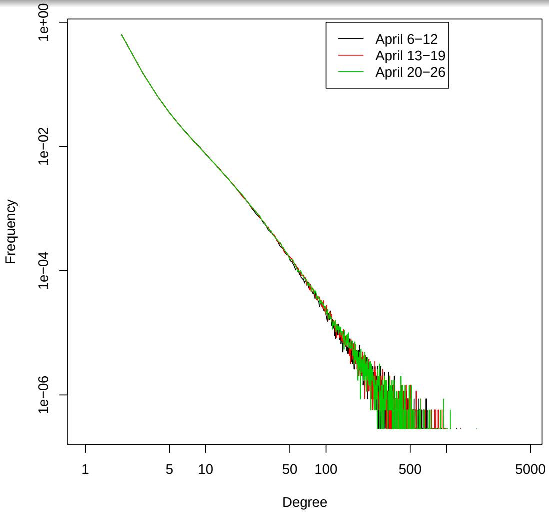


Week Graphs

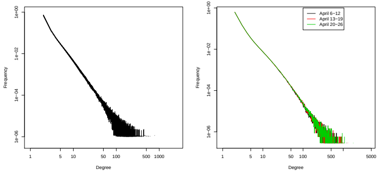
- There are three weeks in April, 2014 for which we have complete data (no sensor outages):
  - April 6–12.
  - April 13–19.
  - April 20–26.
- We construct the mentions graph for each of the weeks.

Week	$\|V\|$	$\|E\|$	# Clusters	Max Cluster Size
April 6–12	3,470,349	5,301,605	161,969	3,024,976
April 13–19	3,461,504	5,359,283	159,312	3,025,135
April 20–26	3,494,342	5,421,333	159,482	3,057,264

Degree Distributions – 3 Week Graphs



Week Graphs



- Note the consistency, and how much the weeks and the days look the same.
- The graphs aren't exactly power law – note the slight curve – but are probably as "power law" as one could reasonably expect.
- This curve is consistent across multiple graphs, and this must say something about the structure of the graphs.

Random Dot Product Model

- The random dot product (RDP) model is a type of latent position model.
- The idea is that each vertex is assigned a vector, and the probability of an edge between two vertices is the dot product of their vectors.
- Obviously the vectors must be constrained so that all dot products are in $[0, 1]$.
- For directed graphs, each vertex is assigned two vectors: and "in" vector and an "out" vector. The probability of a directed edge is the product of the source's out-vector with the destination's in-vector.

Estimating the Random Dot Product Model

- Note that the RDP vectors can (almost) be read off from the singular value decomposition.
- If A is the adjacency matrix, then:

$$A = UDV^t$$

where U and V are the matrices of (left/right) singular vectors and D is the diagonal vector of (non-negative) singular values.

Estimating the Random Dot Product Model

- Of course, we want to find the d-dimensional vectors of the model (I will always assume we know or can guess d).
- In Frobenius norm, the singular value decomposition gives the best low-rank approximation:

$$A \approx U_d D_d V_d^t$$

- The problem is, this isn't what we want: We don't care what the diagonal of A is, and we certainly don't want our algorithm to try to give us vectors of zero length!

Spectral Embedding

- The approach we take is to augment the diagonal of the adjacency matrix.
- There are many possibilities. Looking at the expectation of the degree of a vertex (under a few simplifying assumptions) we choose:

$$a_{ii} = \frac{\text{degree}(v_i)}{n-1}.$$

- We call this "spectral embedding" although the term generally refers to any embedding that uses eigen/singular vectors of (a possibly transformed) adjacency matrix.
- Note that to obtain our estimate of the vectors we scale by $\sqrt{D_d}$ (this is ok since D is non-negative).
- Further, note that our estimate of the vectors is only "correct" up to rotations.

Rationale for RDPG

- Intuitively, it seems that people might have a set of latent variables that describe their interests and affinities.
- People whose vectors are close might be likely to be friends, and hence might mention each other.
- So, at first blush, it seems that the RDPG is a reasonable model for modeling these graphs.

Theory

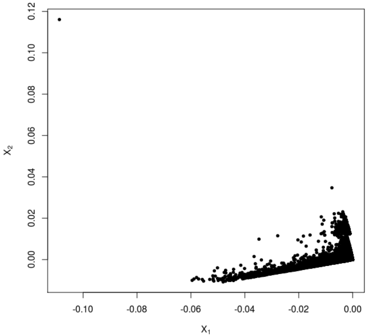
- Sussman et al at JHU proved the following:

Theorem

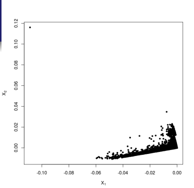
If a random graph is a block-model (probability of edges is constant within and between blocks) then asymptotically the spectral embedding is distributed as a mixture of normal distributions.

- Thus the natural grouping of the block model is reflected in a natural grouping of the embedding.
- Similarly, if the vectors cluster, then the graph will tend to cluster, in the sense that like vectors will have similar connection patterns.

Spectral Embedding

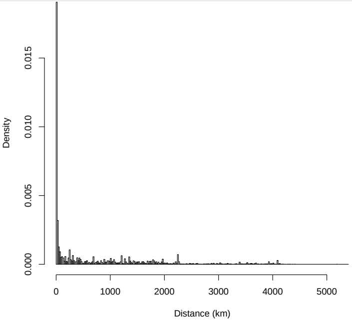


Hmmmm



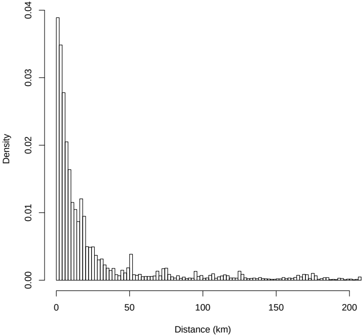
- This doesn't look like a mixture of Gaussians.
- Not to worry, we shouldn't have expected that, because of the degree distribution, and we don't think a block model is correct.
- Here's another reason to think that there might be something useful in the RDP model: people tend to tweet locally – you are more likely to mention someone who is geographically close to you than someone far away. Geography is an important part of "social space".

Support for the "Tweet Locally" Claim



- 10,000 users selected at random from those with geolocation during the week of April 6–12.
- Distance between each of a user's tweets and each of the tweets of their neighbors in the graph. 42,209,108 distances.

Let's Zoom In

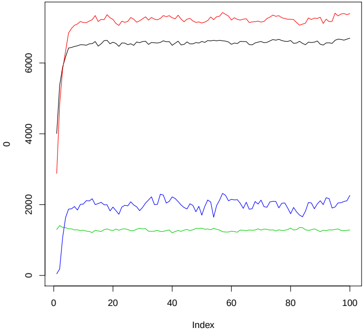


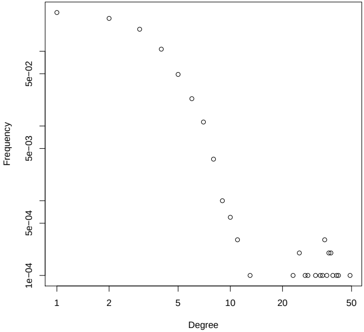
- There are 8,962 distances that are exactly 0km. How can this be? We'll address this in a bit.

One Model

- Set $|V| = n$, $p_l, p_o, p_c, p_i \in [0,1]$, $n_c, n_v, d < n$.
- Generate n_v vectors uniformly on the d dimensional simplex.
- Assign each vertex a county using the county population estimates.
- For each vertex v_i:
  - Choose edges from v_i from the previous graph with probability p_l.
  - Choose vertices within v_i's county with probability p_i, and connect edges using the vectors (RDP).
  - Choose counties different from v_i's county with probability p_o and connect edges using the vectors (RDP).
  - With probability p_c connect to a uniformly chosen celebrity.
- Drop isolated vertices. Return the simplified graph.

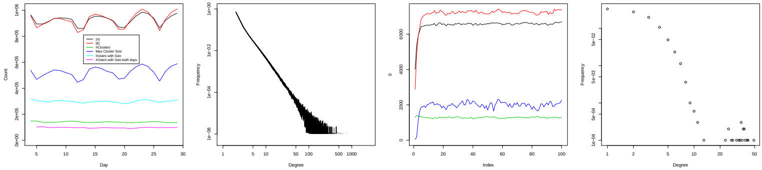
$n = 10000$





Black: n, red: $|E|$, green: # Components, blue: max component.

Comments on Modeling



- The model still needs some work, but it does have some of the characteristics of the observed graphs.
- The weekly pattern should probably be modeled separately.

Geo-Inferencing

- Recall that at best 3% of tweets have a geolocation.
- Since people tend to "tweet locally" can we use the graph to infer the geographic location of a user who does not report location?
- Other approaches:
  - Use the location that the user reports in their profile.
  - Look for tweets by the user that mention a (uniquely) locatable place.
- We will investigate locating a user by the locations of the people the user mentions.

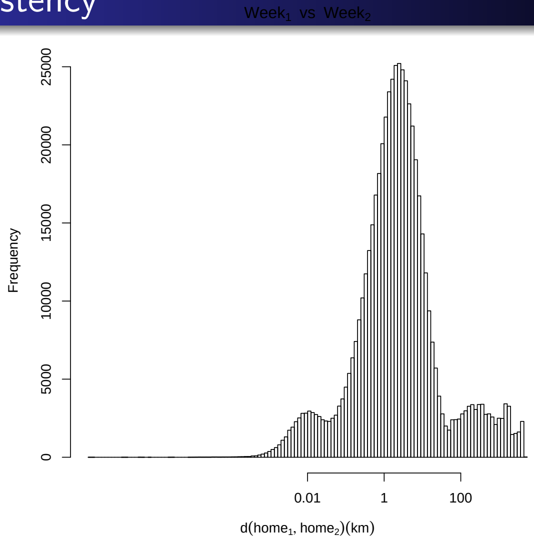
The Problem

- Given a user and the mentions graph, determine the location of the user at the time of the tweet.
- Caveat: As we saw above (and will see more of below) an accuracy between 1km and 50km is as good as we should expect, except for particularly sedentary tweeters.
- This accuracy may be sufficient for many aggregation tasks:
  - Looking for disease outbreaks at the city/county/state/country level.
  - Tracking large storms and power outages.
  - Looking at large events such as sporting events, conventions, parades, marathons.
- It may be possible to improve the accuracy using the text in the tweet in some cases. We will not address this.

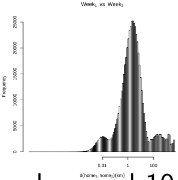
Geo-Consistency

- First, how much do people move around?
- Consider the question of how much a user has moved from one week to the next.
- We compare the position of a user who appears (with a geolocation) in two consecutive weeks.
- The following plot is a histogram of the distance between the "home" position of the user in two consecutive weeks.
- Here "home position" is defined by fitting a 2D kernel estimator to the positions and using the highest density point.

Geo-Consistency



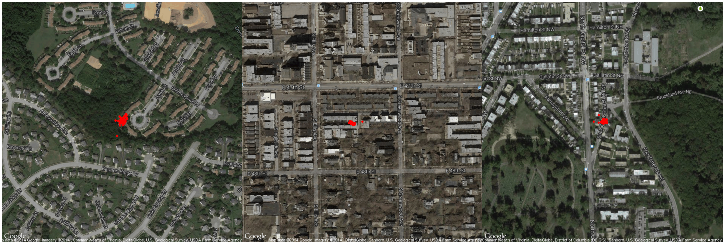
Geo-Consistency Comments



- Note the bump centered around 10m. The accuracy of commercial grade GPS is reported to be around 8m. These are people who didn't move, but were using GPS to report position.
- Note that most everyone sticks around near 1km of where they were last week.
- Some people travel very far away (more than 100km).

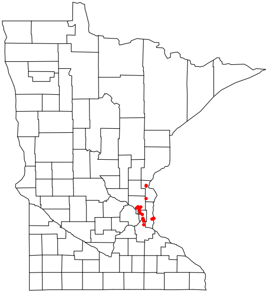
Geo-Consistency Comments

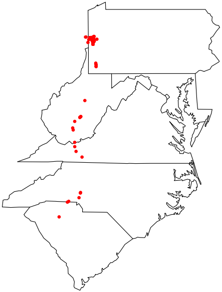
- Some people stick around a small area:



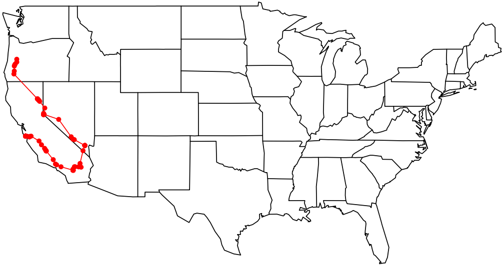
Geo-Consistency Comments

- Some people travel very far away (more than 100km).

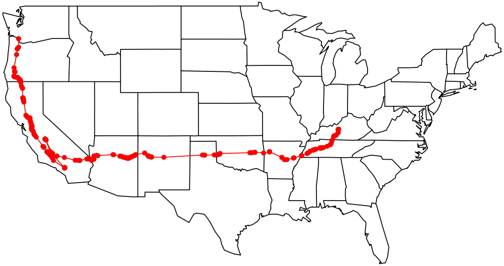




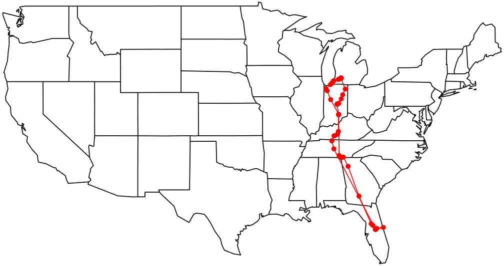
Bunny Trails



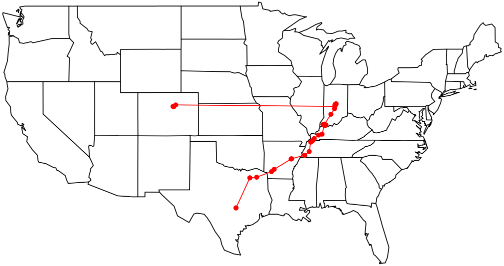
Bunny Trails



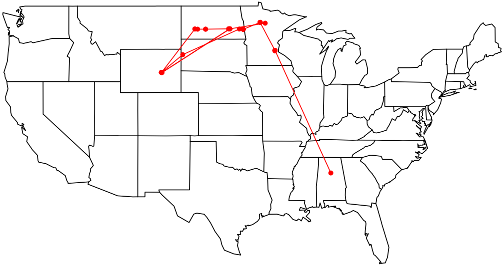
Bunny Trails



Bunny Trails



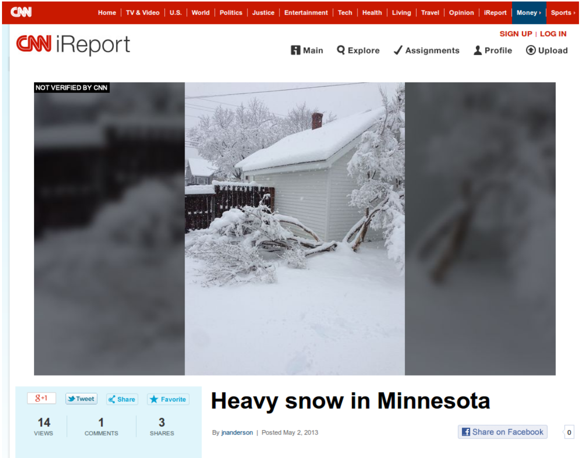
Bunny Trails



Geo-Consistency Comments

- Recall the 8,962 0km distances we saw. These are probably not stationary people. They are most likely people who report a constant location.
- You (or your app) can put any latitude and longitude onto a tweet.
- Researchers use this to test things (e.g. what percentage of geolocated tweets do we collect?).
- Next is another example of how to report a “fake” location.

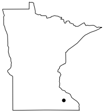
Clicking on "The Bird"



Clicking on "The Bird"

Here's what happens when you click on the bird:

```
<link rel="canonical" href="http://ireport.cnn.com/docs/DOC-966912"/>
<meta property="og:url" content="http://ireport.cnn.com/docs/DOC-966912"/>
<link rel="image_src" href="http://i.cdn.turner.com/ireport/sm/prod/2013/05/02/WE00946115/2465834/imagejpg-2465834_lg.jpg"/>
<meta property="og:image" content="http://i.cdn.turner.com/ireport/sm/prod/2013/05/02/WE00946115/2465834/imagejpg-2465834_lg.jpg"/>
<meta property="og:title" content="Heavy snow in Minnesota"/>
<meta property="og:type" content="cnn-social:story"/>
<meta property="og:site_name" content="CNN iReport"/>
<meta property="fb:app_id" content="80401312489"/>
<meta property="fb:page_id" content="129343697106537"/>
<meta property="og:latitude" content="44.03394898869472"/>
<meta property="og:longitude" content="-92.44855944075849"/>          <!-- twitter card implementation -->
```



Hence the 2D kernel estimator (or other methods) to estimate the user's "home" location.

How Well Can We Geo-Inference?

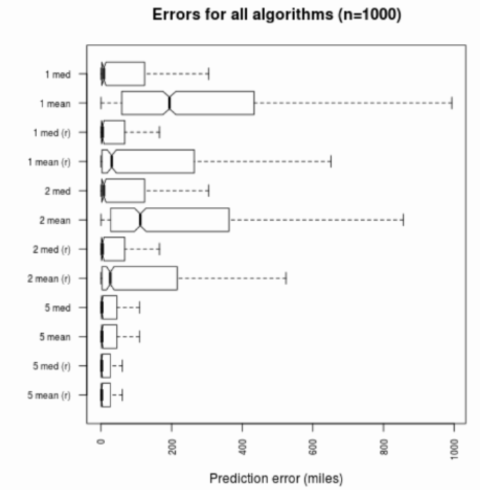
- Our “bunny trails” indicate that using only the graph, we are going to err on people who are traveling.
- Even if we used the previous values of a user’s location to predict the current location, our consistency result shows we shouldn’t expect better than 1km for most people.
- The “click on the birdy” and other types of “spoofing” means that we need to think about what we mean by “a user’s location”: not the location of a single tweet, but the location from which the user tweeted the most during that time period.
- Not everyone tweets locally. Using only the graph can only give us so much.

Geo-Inferencing Algorithms

- We first looked at several variants of a graph-based geo-inferencing algorithm.
- These were tested on a small subset of vertices in one graph to get a feel for performance.
- In this we only looked at the mutual mention graph – both users must mention the other.
- The numbers in the next plot refer to the minimum number of neighbors a user must have to allow an estimate.

1. Use the "home coordinates" of the neighbors as the home for the user.
2. Weight the coordinates by the number of mentions.
3. Exponentially weight the coordinates (count more recent mentions more).

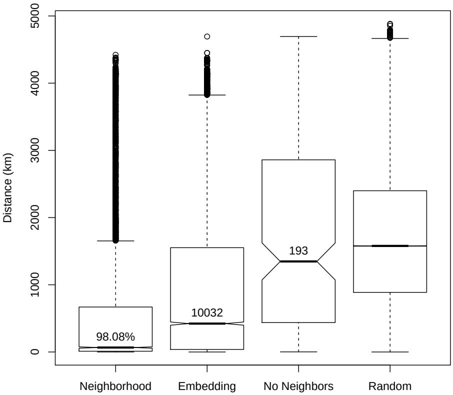
Initial Experiment: Representative Sample of Runs



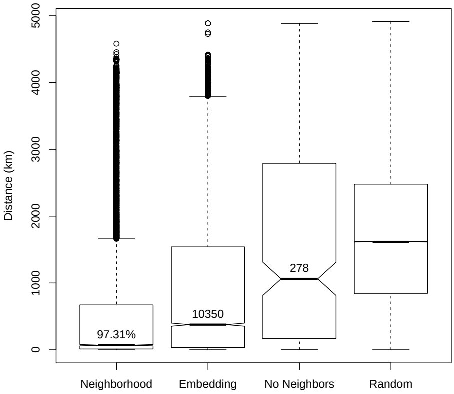
A More Extensive Experiment

- We used each of the one week graphs.
- For each week, we do four estimates:
    1. Neighbors using the 2D kernel on the mutual mention graph.
    2. Embed the graph (per connected component) and use the nearest neighbor in the embedding. We use the fast nearest neighbor algorithm in the `FNN` R package.
    3. Only compute the error for those in the first experiment that had to be eliminated because they had no neighbors with coordinates.
    4. Pick coordinates at random from all coordinates.

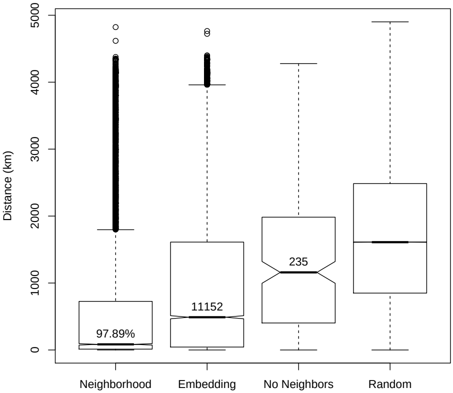
Results: April 6-12



Results: April 13-19



Results: April 20-26



Comments

- Only about 1–3% of tweets contain geo-location.
- This limits the utility of the neighborhood approach rather considerably.
- Embedding is considerably better than chance, but much worse than neighborhood. This leaves open the possibility that it can take the place of the neighborhood approach for those whose neighbors don't have geolocation.
- As we see, about 2% of the (supposedly all geolocated users) don't actually have geolocations. The embedding approach for these is (maybe) marginally better than chance, but is basically useless.

Conclusions

- There is a lot of interesting structure in the Twitter mentions graphs.
- They are basically one huge connected component and a bunch of tiny ones.
- Some of the reason for this is the very high number of mentions of celebrities, and sites such as 4-square.
- We have seen that the structure of the graphs can be used to infer things about the users (such as their position).
- The spectral embedding approach to inference is disappointing. Perhaps one reason for this is the very large degree individuals that artificially connect the graph.
- Certainly there is much more work to be done.