# Incorporation of Geographic Coordinates in Random Dot Product Graphs

David J. Marchette

Naval Surface Warfare Center

This work was supported by the NSWCDD ILIR Program

Interface 2011


# Outline

# Outline

# The Model: RDPG

- We consider in this work attributed simple random graphs with directed edges:
  - $G = (V, E; X, Y)$ where
    - $|V| = n$, $E \subset V \times V \setminus \{(v, v)\}$.
    - $X, Y \in \mathrm{II}^n \Delta^d$ (or $D^d \cap \mathbb{R}^{+d}$).
  - We assume the vertices and edges are perfectly observed, but the attributes may be completely unobserved (latent) or imperfectly observed.
  - In particular, we may observe some $Z$ which is related to $X, Y$ (correlated) and we'll use $Z$ to improve our information about $X, Y$.
- We assume the edge probabilities are given by:

$$P[uv \in E] = X_u^T Y_v.$$

- We call $X$ the "out vectors" and $Y$ the "in vectors".

# Estimating the Latent Attributes

- We can estimate the latent vectors $X$, $Y$ by making the observation that if we could augment the adjacency matrix $A$ of $G$ with $X_u^T Y_u$ (call this $\widetilde{A}$) the singular value decomposition provides the optimal solution $X$, $Y$ to minimizing
$$\|\widetilde{A} - XY^T\|.$$
(Note: we assume the rows of $X$ and $Y$ correspond to the vertices.)
- It turns out, a good estimate of $X_u^T Y_u$ is $\frac{d_u}{n-1}$, where $d_u$ is the degree of $u$. This comes from:
$$E[d_u] = E[\sum_{v \neq u} X_u^T Y_v] = (n-1)E[X_u^T Y_u]$$
when $X$, $Y$ are independent.

# Combining External Information

- Now suppose we have some information about the $X$, $Y$.
- For example, suppose we have noisy observations of them.
- We could:
  1. Use the observed (noisy) values and ignore the observed graph.
  2. Estimate $X$, $Y$ using the graph, and then factor in the observations.
  3. Incorporate the observations into the model fit.
- We'll look at one way to do this third approach.

# Outline

# Joint Embedding of Dissimilarity Data

- Consider data from two or more different sensors:
  - Given inter-point distance matrices $D_0$ and $D_1$ (with rows/columns matched – corresponding to the same real-world entity) and $0 \leq \lambda \leq 1$, let $D_\lambda = \lambda D_0 + (1-\lambda) D_1$ and form
$$D = \begin{pmatrix} D_0 & D_\lambda \\ D_\lambda & D_1 \end{pmatrix}$$
  - Then using $D$ for the embedding (multidimensional scaling) jointly optimizes fidelity (maintaining the separate inter-point dissimilarities $D_i$) and commensurability (matching up the paired entries so they are close in the embedding).

# A Similarity Version

- Instead of dissimilarities, think of a graph as a representation of similarities. Given two similarity matrices (e.g. an adjacency matrix and measurements of entity similarity), $A$ and $S$ form:
  - $S_\lambda = \lambda A + (1 - \lambda) S$,

$$D = \begin{pmatrix} A & S_\lambda \\ S_\lambda & S \end{pmatrix}$$

    - Embed using spectral theory rather than multidimensional scaling.
- This fuses graph information ($A$) and entity attributes (associated with $S$).

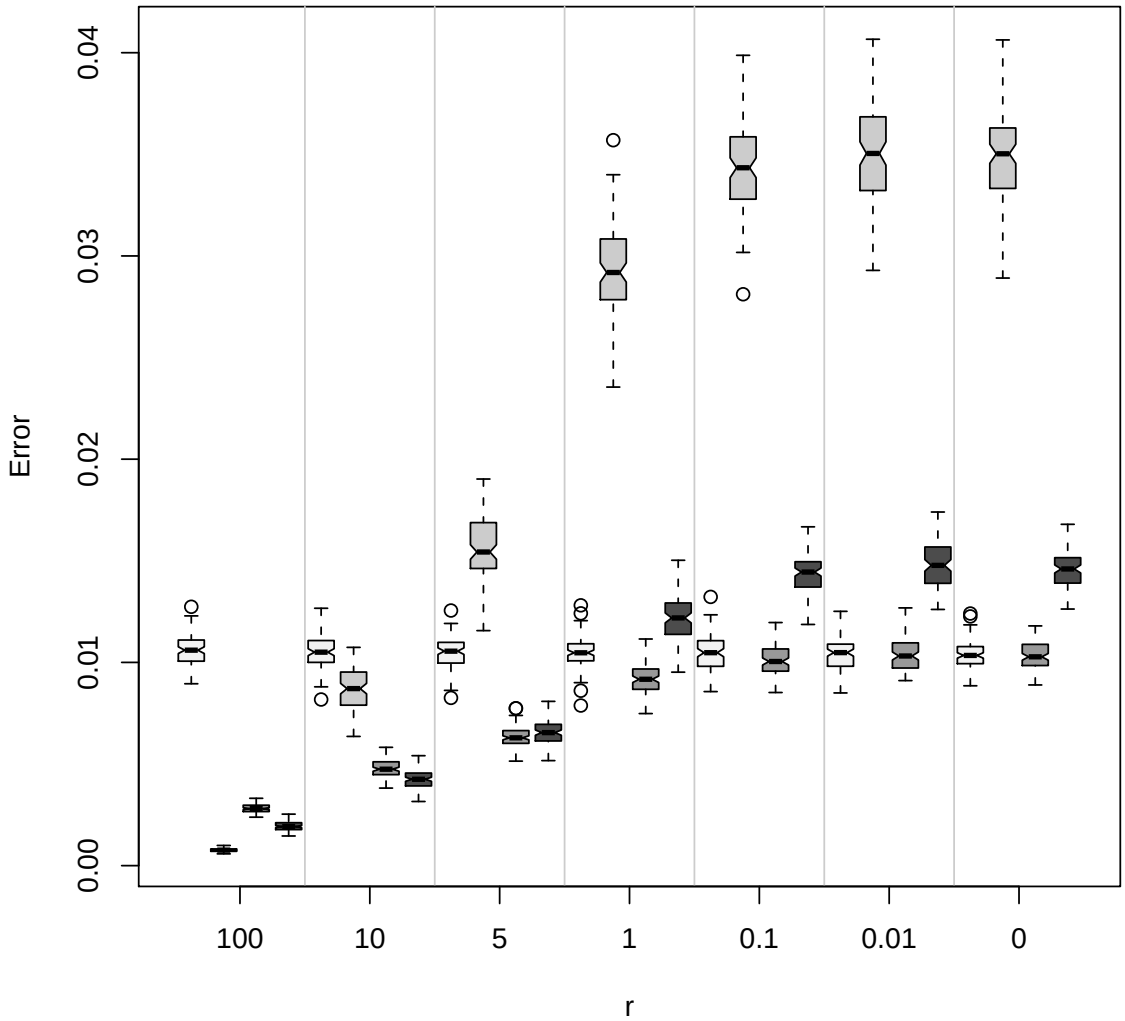
# Joint Embedding of Graph and Attribute Observations

- Given observations $\widetilde{X}$, $\widetilde{Y}$.
- Form $S = \widetilde{X}\widetilde{Y}^T$.
- Use the embedding procedure described above.
- If instead we are given $S$ (for example, a graph defined by geographic nearness), we can use this in the algorithm.

# Example

- The following plot shows the results of an experiment in which a graph is fused with noisy information about the attributes that define the edge probabilities.
- In the following plot, the x-axis should be thought of as increasing noise.
- The fusion result is the third darkest box. Consider the middle ($r = 1$) case.
- The y-axis is the error in estimating the attributes.
  - Fusion beats the estimate from the graph alone (first, lightest), and from the attributes alone (second, next lightest).
- Ignore the darkest box in these plots.

# Example



# Geo-coordinates

- We will incorporate geo-coordinates into the RDPG model using the following assumption:
  - In communication graphs, people communicate more frequently with those close to them than far away.
  - Thus, geo-location can be viewed as “related” to the attributes, and may provide useful information in the fitting of the attribute vectors.
- This assumption may not be valid for some forms of communication.
- We will consider Twitter data, where we will first look at the extent to which the assumption is valid.

# Outline

# The Africa Twitter Data

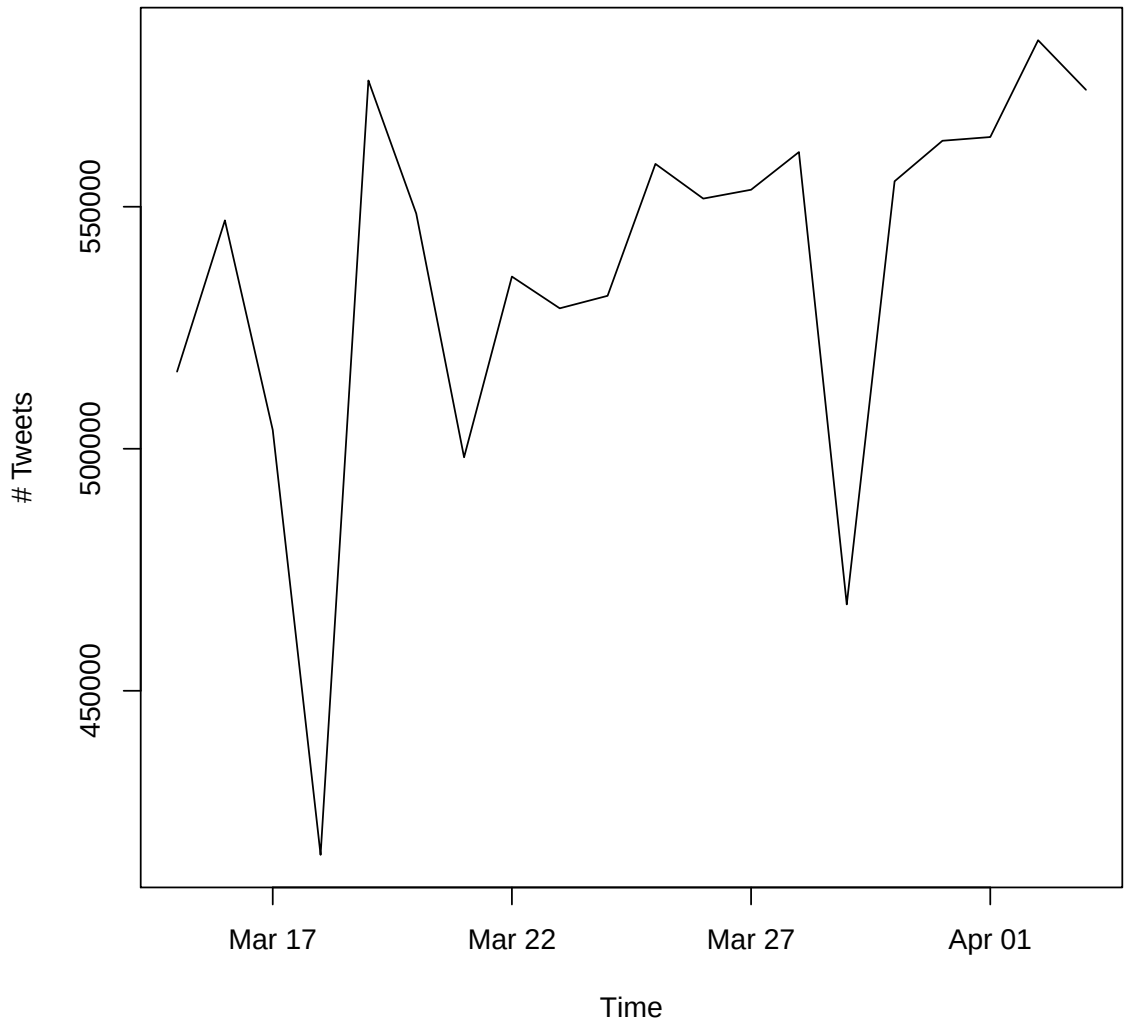
- Twitter data collected from Africa from 3/15/2011 through 4/03/2011.
- Data downloaded every 2 minutes.
- Download defined by a 3500 kilometer circle centered in Bangui, Central African Republic (on Avenue Barthelemy Boganda). This position is totally arbitrary.
- Roughly 11-million distinct tweets in this 20 day data set.
- $336,500$ distinct tweeters.

# Data Collect

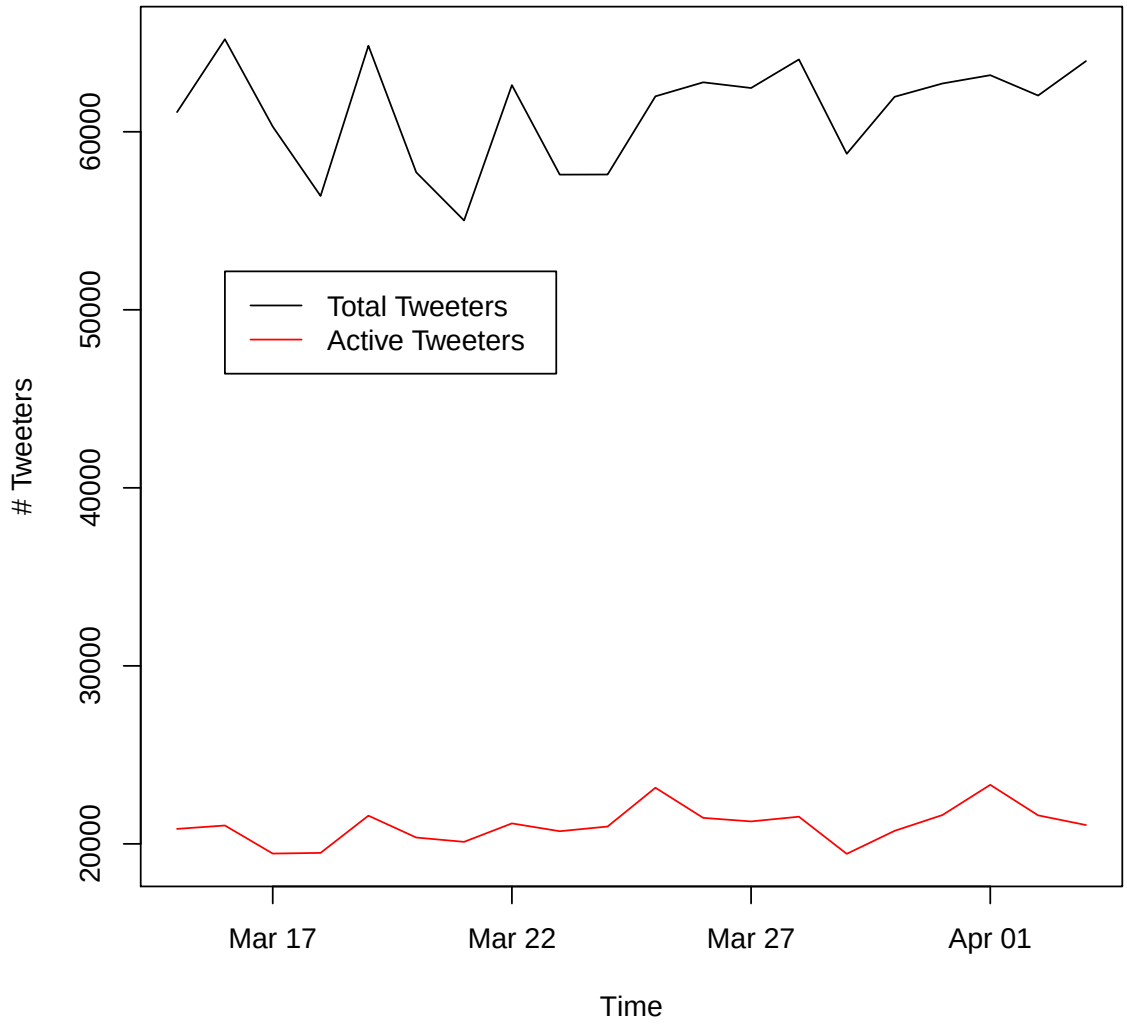
# The Africa Twitter Graph

- One graph per day.
- A user is a vertex if they tweeted during the 20 day collection period.
- An edge from vertex $u$ and $v$ if $u$ enters $v$ (as @$v$) in at least one tweet that day.
- Retweets are removed.
- Geo-coordinates are provided by the data with unknown accuracy and variable quality.
  - Some are coordinates.
  - Some are city/country.
  - Some are city (often spelled creatively) – no country.
  - Some are random – not associated with location.
- We only consider the first two as being "good" geo-locations.

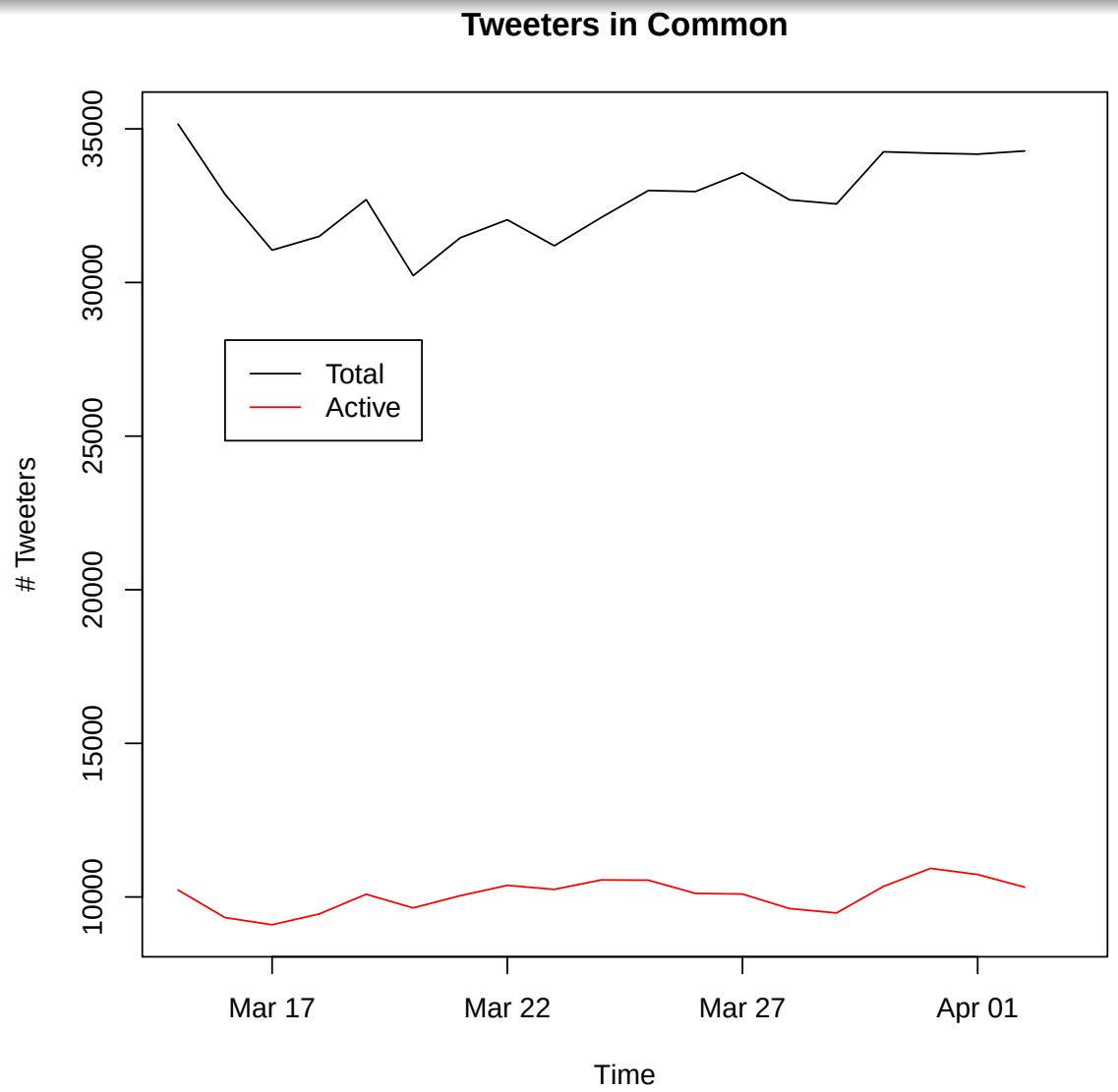
# Number of Tweets



# Number of Tweeters



# Number of Tweeters in Common



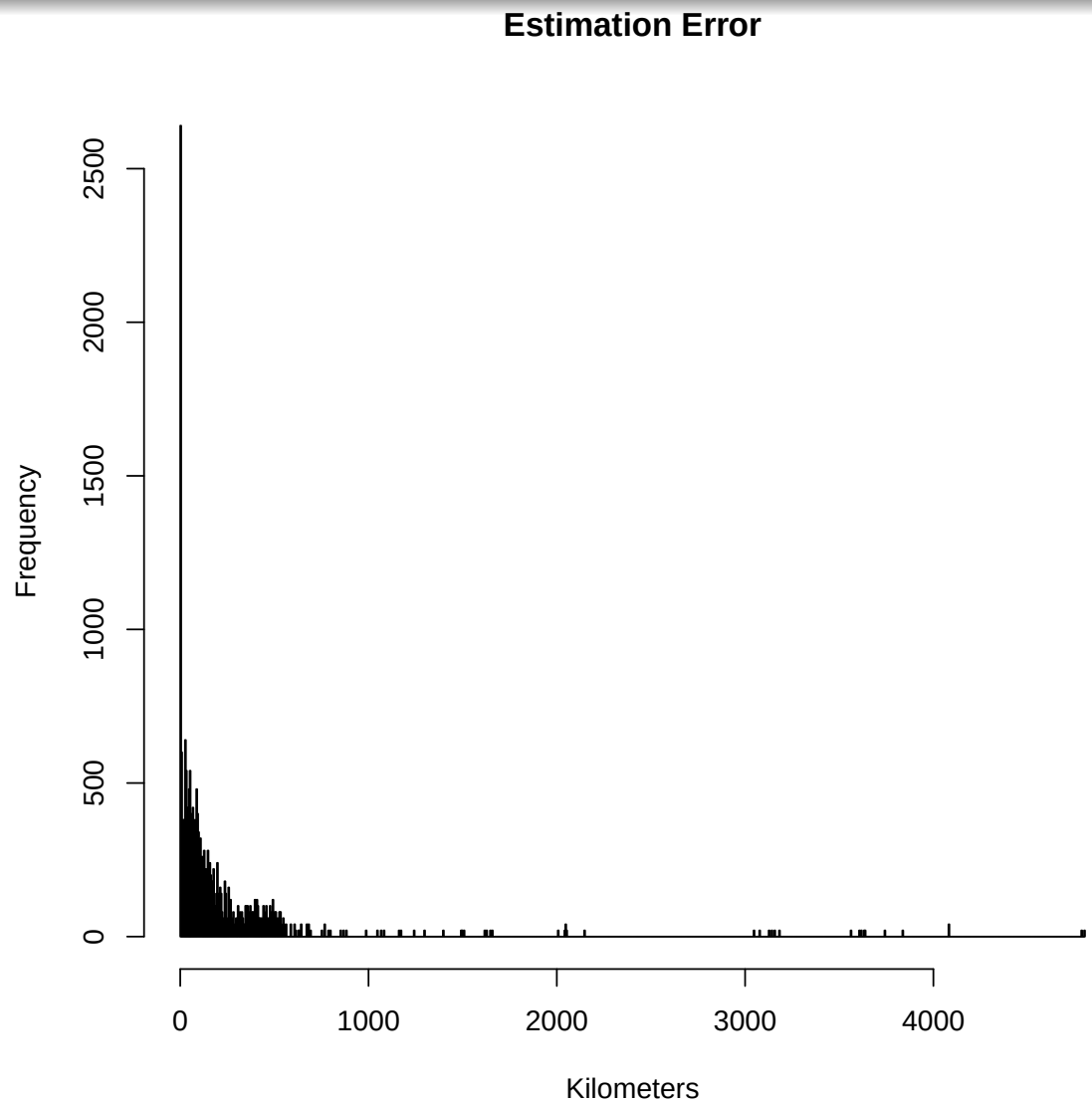
# Outline

# Estimation of Geo-Coordinates

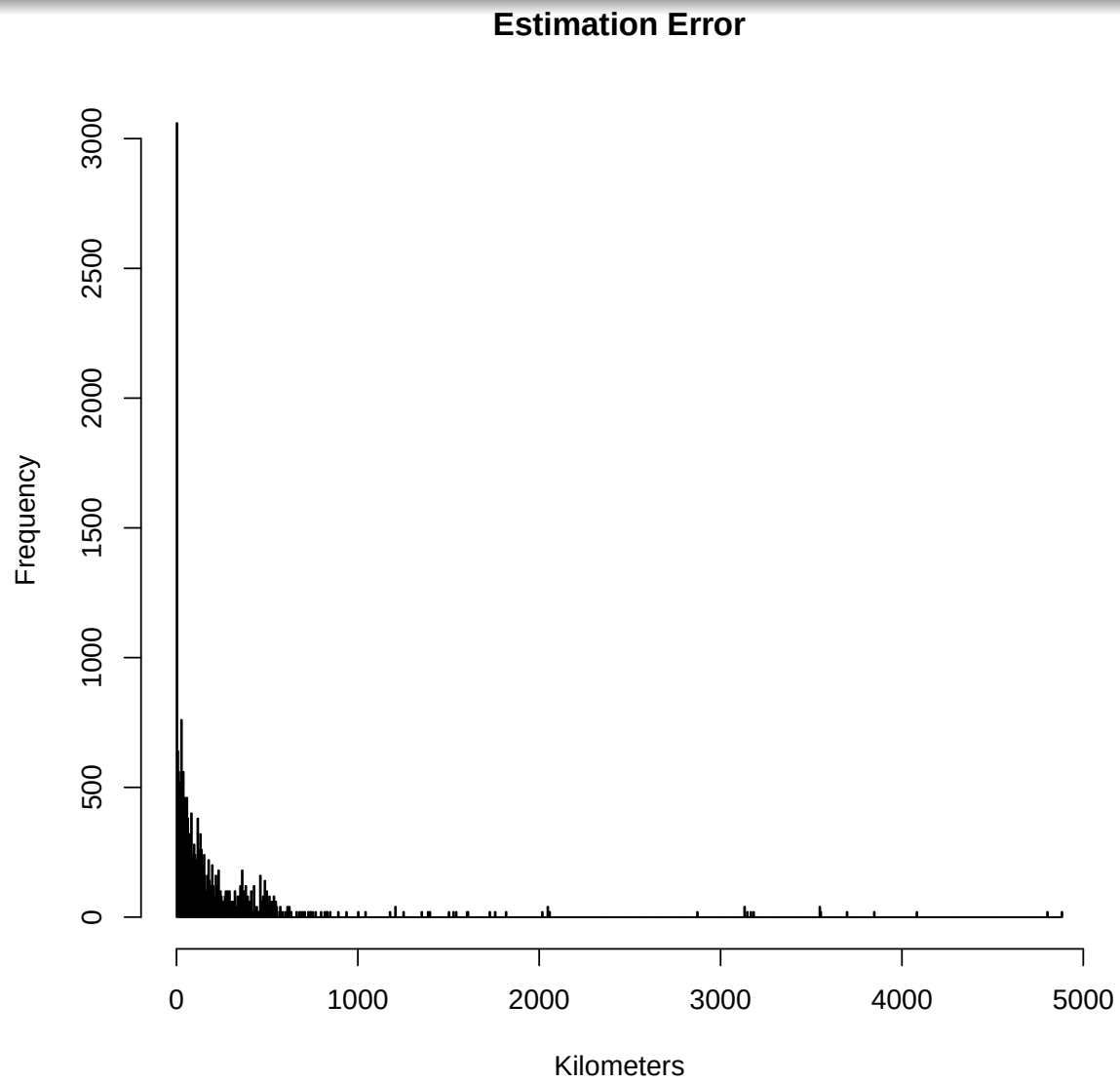
- We can estimate the geo-coordinates of a vertex by:
  - Average the geo-coordinates of those the user tweets to (is connected to). This is the “crude” version.
  - Embed using the RDPG model, and weight the average according to the distance to the vertices in the embedding.
- Many other schemes are possible.

| | Crude | RDPG |
|---|---|---|
| %< 50K | 32.2 | 37.2 |
| %< 100K | 50.6 | 52.3 |
| %< 250K | 75.1 | 75.4 |

# Estimation Error – Crude



# Estimation Error – RDPG



# Estimating Edges

- Given a graph $G_t$ for day $t$, predict the edges of the next graph.
- Three methods:
  1. Predict the edges observed in $G_t$, $E_t$.
  2. Predict the edges associated with geographic neighbors (say, within 10 kilometers – the geo-graph).
  3. Using the geo-graph, embed using $G_t$ and the geo-graph, then rank the edges according to distance between vertices in the embedding.
- Note:
  1. is pretty naive.
  2. provides a huge number of potential edges to check.
  3. is computationally challenging.

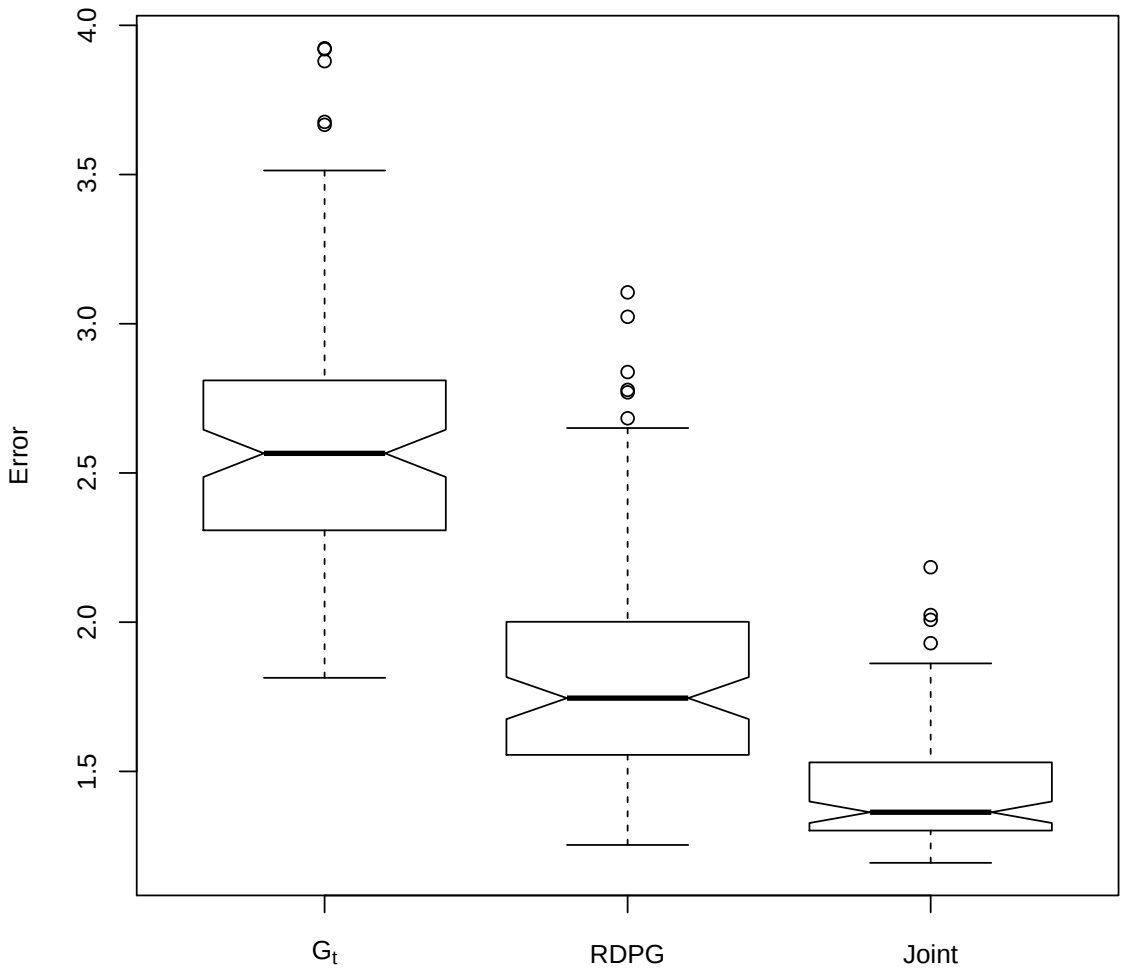
# Experiment

- Because of the large number of vertices, we sample 1000 vertices, compute error:

$$\sum (a_{ij} - zij)^2 / sum(A)$$

  where $A$ is the true adjacency matrix (that of $G_{t+1}$) and $Z$ is the estimate.
- Three methods:
  1. $Z = A_t$.
  2. $Z = XY^T$.
  3. $Z = X_j Y_j^T$ using the joint embedding.
- We repeat this for a total of 100 samples.

# Prediction Error



# Discussion

- The Twitter data is an interesting data set to investigate a number of interesting questions in social network analysis and social media.
- We have shown that to some degree the social network provides the ability to localize (at a crude level) the individuals, provided this locality information is available for enough of the individuals.
- We can improve prediction error through the RDPG model, with some (weak at this point) evidence that adding in geo-coordinates can improve this.

# Future Work

- Incorporate fast (approximate) nearest neighbors for both the geo-position graph and the prediction.
- Improve the geo-coordinate matching by incorporating the actual distance rather than a simple graph model.
- Investigate adding in content (e.g. hash tags).
- Increase memory and compute power for more extensive experiments.

Acknowledgment: This work funded in part by the Office of Naval Research In-House Laboratory Independent Research program.