Determining Gender in the Enron Email Corpus via Information Clustering Techniques

Mike Corbin
Patrick Trinkle
(corbin2|tri1)@umbc.edu

University of Maryland Baltimore County
1000 Hilltop Circle
Baltimore, MD 21250

Problem

- Given a corpus of emails written in a mixed gender environment can the frequency of usage of certain words determine gender?

Previous Work

- A similar problem is authorship attribution
  - There has been work that uses statistics of stop words to prove the author of a document with other works of theirs as input
  - The author of the 15th book of the Wizard of Oz has been statistically determined in this method
  - The author attribution problem is typically performed on larger works

Background

- People use certain phrases more frequently than other, etc. However, this information is easily forged.
- Words people use unconsciously such as stop words are harder to mimic. The stop words vary based on not just what is said but how it is said.
- Given enough documents from a particular author or set of authors it is feasible to separate the authors by stop word usage

Background (cont.)

- Using the vector space model of corpus representation where the dimension is how many terms are examine, the documents are vectors in the hyperspace
- If each user's documents are mapped into hyperspace they should plot near each other based on word usage

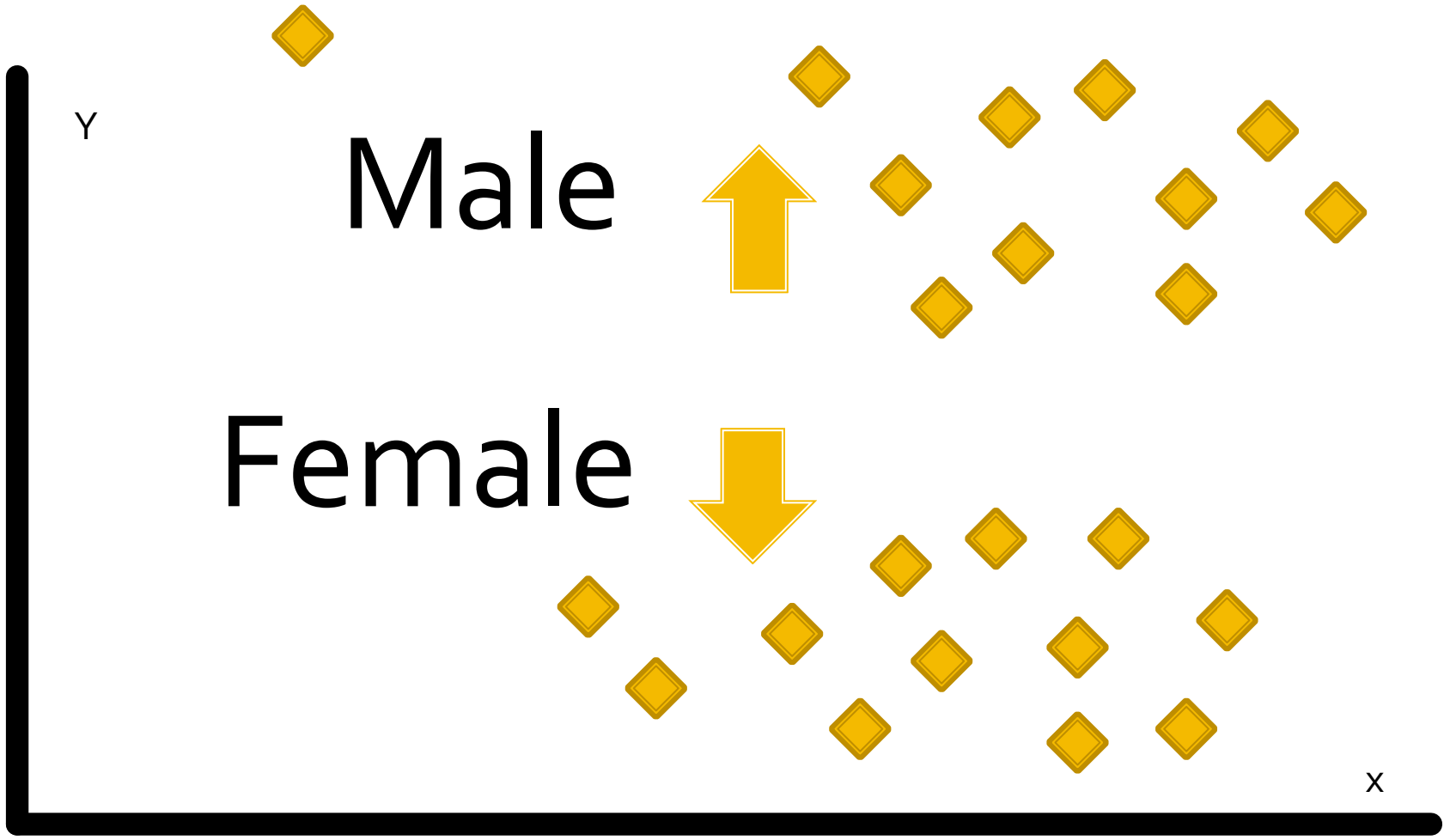
Background (cont.)

- For our purposes it is too noisy to view the information in hyperspace therefore Principal Component Analysis is performed on the vectors to provide the 2 more important terms
- With this reduction from hyperspace to Euclidian 2-space the vectors can be visually checked for any grouping

Background (cont.)

- Grouping in the gender problem will hopefully indicate than the vectors representing male and female users will appear together
- Example on next slide

Background (cont.)



Task

- Enron corpus ~1.5 GB of emails; organized as the mailboxes of a small subset of users.
- Most have gender obvious names, John, Mary, Scott, Susan (as opposed to Leslie or Logan)
- Build list of top 100 terms in corpus

Task (cont.)

- Determine the top 50 most frequently used function words (stop words: “to,” “the,” “and”)
- Determine term frequencies for each of the top 100 users (users with the largest quantity of emails)

Task (cont.)

- Use Principal Component Analysis to determine weight of terms in dataset
- Plot the data on a graph (gnuplot) and examine in 2-dimensional space if there is a clear margin between users flagged as men versus women

Approach

- Use a Python script to sort all the emails into folders by sender (excluding duplicates)
- Use Python script to generate statistics about the corpus—how many emails per user, as well as determine top 100 terms

Approach (cont.)

- Use parallel computing to rapidly process the 100 users; using MPI to handle the I/O.
  - 2 servers, >= 100 clients, 1 blade process (strictly for pca)
- Use C code to tokenize each user's email as though all the emails in a folder were one document
  - Tokenizer breaks on "from:" or "to:" in email file, therefore missing any complicated threads

Information

- Sorted by user there were 6,321 users in the system. However, multiple email addresses could refer to the same individual (j.adams@enron.com == jadams@enron.com)
- Many of these accounts are non-user (4oenron@enron.com, techsupport@enron.com)

Information (cont.)

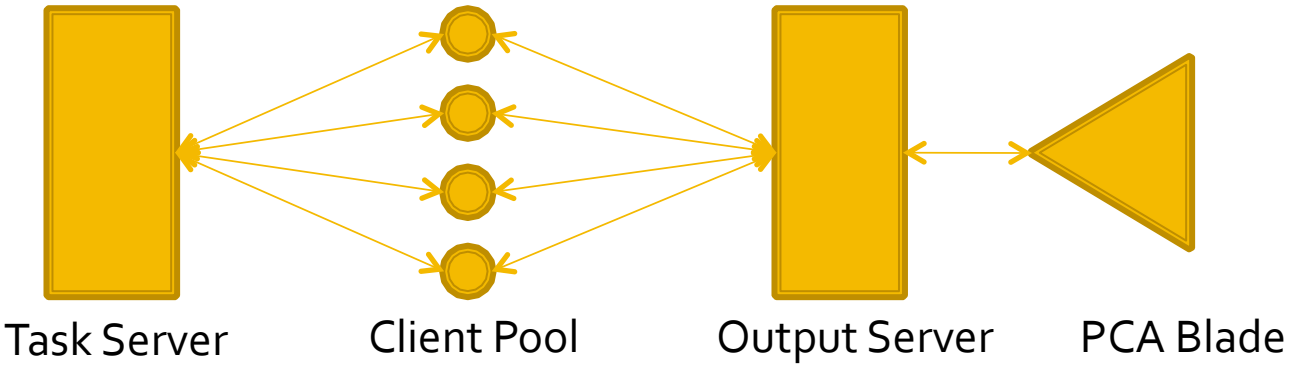
- Most users have under 100 emails; however, the top 31 have over 1,000 emails
  - Only the top 6 have over 3,000 emails
- For our top 100 users we merged all their accounts with varying name formations
- The top terms were stop words, as expected

Information (cont.)

- The matrix given to the PCA code is 50 rows x 100 columns of floats ~19kB
- PCA maps the 50-dimensional space into 2-dimensions, something easy to view in gnuplot
- Term frequencies are normalized by the sum of the counts of all tokens in the emails
  - This includes all tokens—not just our 50

Solution

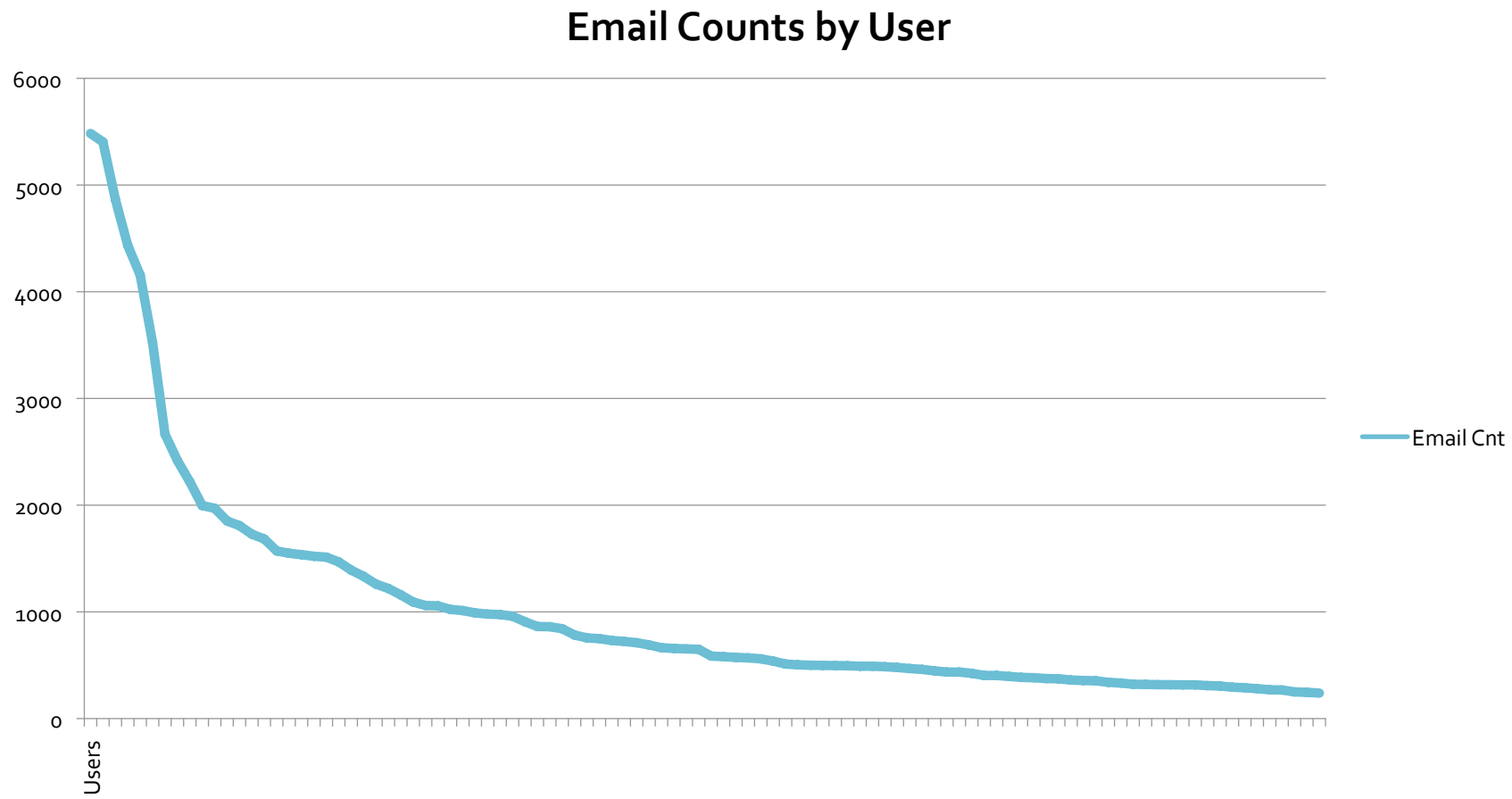
- 1 Server tasks clients which request user folders
- 1 Server listens for output from the clients
- 1 Blade waits to process output



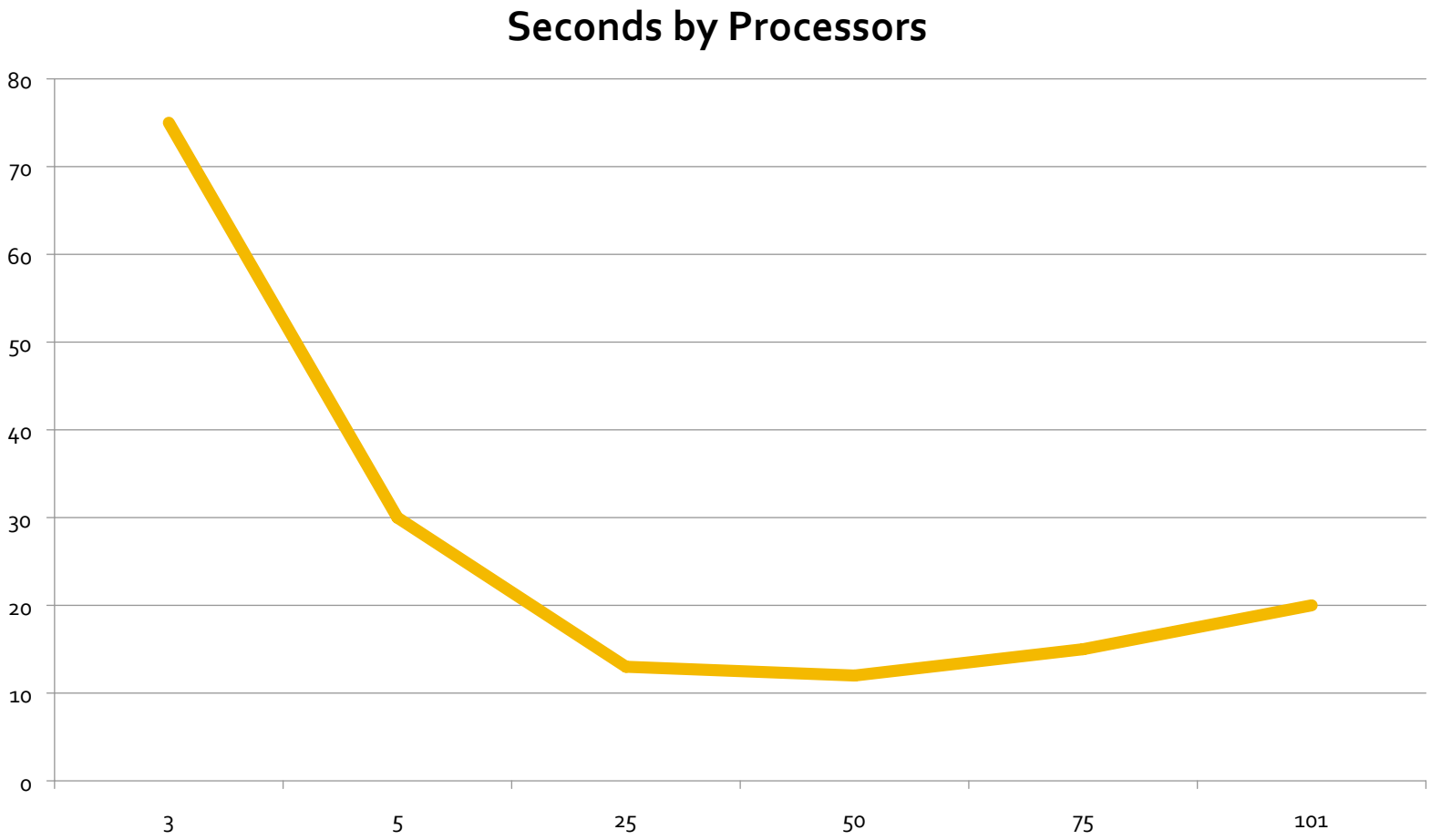
Results

- The processing of the emails is extremely fast
- When instead of the top 100 users, the top 6-10 users are processed there is a significantly different graph than with all 100
  - This detail is because the top 6-10 users have dramatically more emails than the top 100—the graph has a clear drop-off

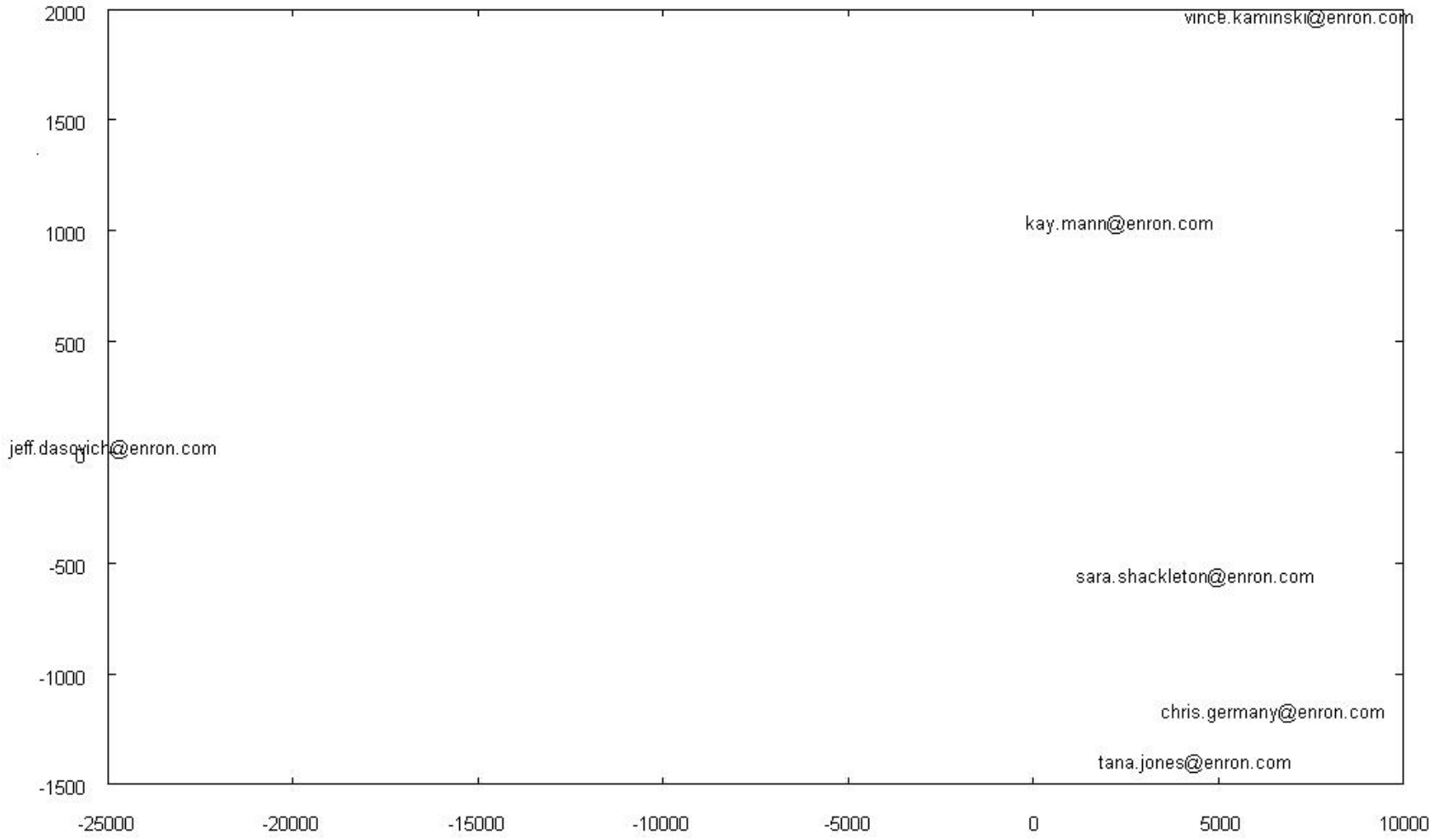
Results (cont.)



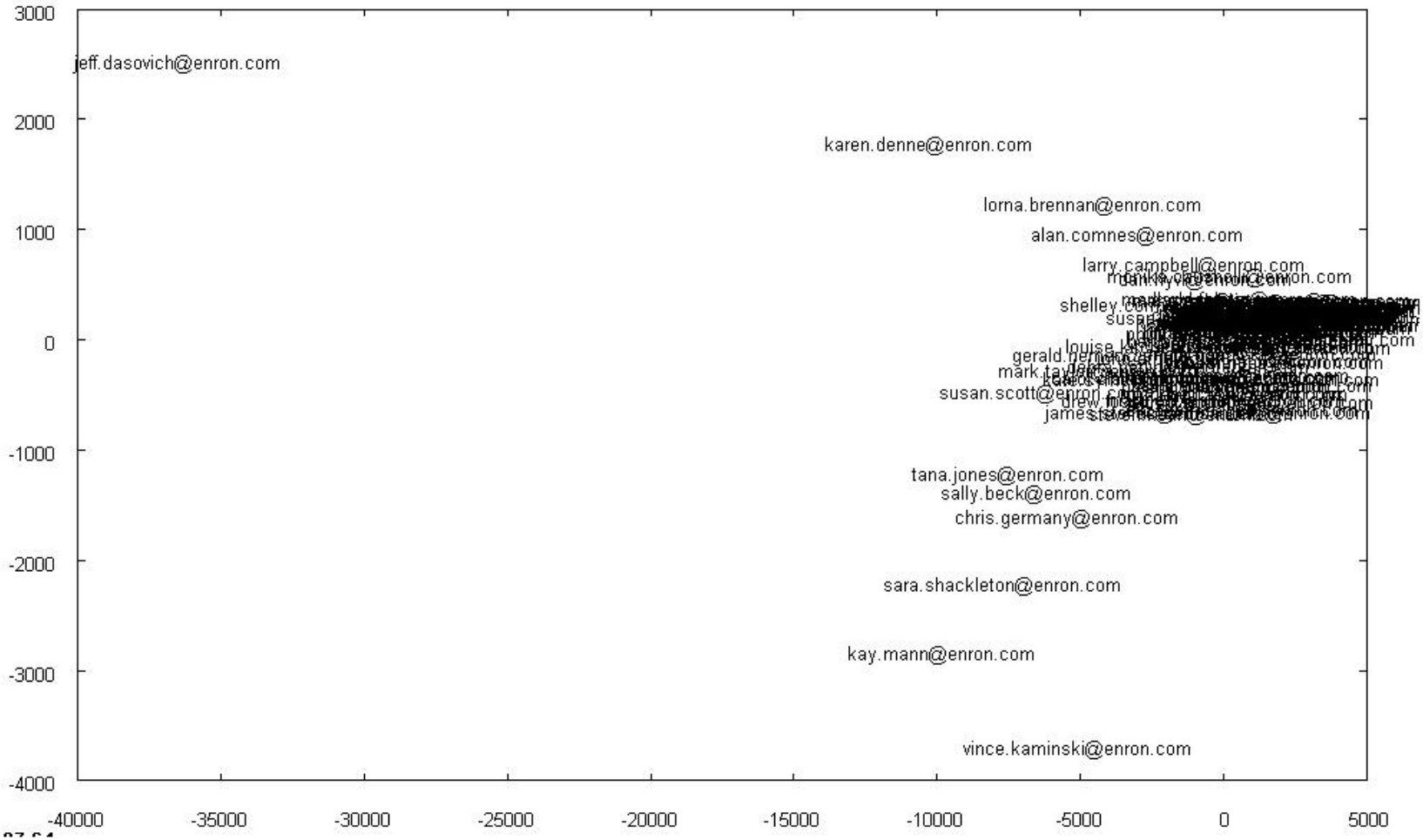
Results (cont.)



Results (cont.)



Results (cont.)



Conclusion

- Because the 100 users had so few emails overall there was insufficient information for our approach
  - More preprocessing could strip emails into individual pieces providing more data—break threads apart—our code only handled 1 message in a thread
- It would be worthwhile to reapproach the problem using centroids (geometric means) of the document vectors to group users, versus using Principal Component Analysis

Conclusion (cont.)

- Thanks for your time.
- Questions?
- Mike Corbin, Patrick Trinkle
- University of Maryland, Baltimore County
- Spring 2009