Abstract

While several papers have analyzed the dataset using simpler centrality metrics and ad hoc clustering methods, none have done so using more current metrics and clustering methods. Motivated by this lack of rigorous analyses, in this paper we give a brief overview of similar works, identify and propose dataset filtering parameters, identify meaningful analytic measures such as PageRank, and finally, present the results of our analyses and clustering outputs. In this regard, we also seek to define a methodology by which to approach the analysis of social and organizational communication datasets, since the methods and evaluation metrics of such approaches suffer two primary problems: firstly, a lack of comprehensive datasets, and secondly, subjective notions of performance.

Introduction

The amount of electronic communications data has increased dramatically with both the proliferation and integration of electronic communications, however most such data is not available due to confidentiality and privacy concerns. This leaves us with only a handful of publicized analyses of real world social and organizational communications, and limited knowledge of the recurrent characteristics they may contain. However, the spectacular collapse of the energy trading company Enron in 2001 forced the release of thousands of internal emails, inadvertently providing one of the best organizational communication datasets to date. In this context, and in the absence of any rigorous analysis including PageRank, we decided to examine a real-world communication network to automate the discovery of important individuals, hierarchical relationships, and communities. Similar work was performed by (Palus et al, 2011), from whom we obtained organizational charts to asses our community detection schemes.

It is important to note that although the Enron dataset is in the public domain, it contains the names of real individuals, and hence this report as well. We ask that others respect the privacy of such individuals given the information discovered from their emails; though only email metadata was analyzed, not email content.

Problem Definition

The problem definition may be defined, albeit nebulously, as one of attempting to discover workflows for communication-network analysis that could ultimately lead to automated discovery of communities, hierarchy, and other organizational characteristics. The reason for the breadth of this definition is that such observations inform many valuable uses-cases: to the analysis of an organizational hierarchy, fraud detection, inefficient or duplicate work processes, spotting work silos, improving productivity, and so on. Communication networks also differ significantly from more observable or deterministic network data, since the data is subject to significant noise, and further, the information content of communications cannot be inferred from purely from who-talks-to-whom networks. Therefore, these networks typically require careful filtering, observation, and analysis in order to be sufficiently well-defined to be used as input to other network analysis methods. Their organic nature requires significant manual transformation and analysis, and thus careful identification of the assumptions and steps required to generate results in a reproducible fashion.

For this reason, we divided our approach to this task into three parallel, iterative steps:

1. Identify network construction parameters: These include both filtering and structural characteristics.
2. Analyze PageRank and similar centrality metrics, both theoretically and applied to this dataset.
3. Analyze the output network in terms of node centralities, global characteristics, communities, and community hierarchy.

We deliberately chose these steps to analyze the data before making assumptions about the output, results, and artifacts generated. The intention was not to presume the necessity of a result based on hidden assumptions about network structure, as we often found in the literature. Instead, these steps were intended to fully capture how modifying our assumptions and network representations affected the “necessity” of any particular result. In many regards, we derived this methodology from the work by [CHAPANOND, 2005], in which the authors conscientiously described how modifying their representations of the same network data drastically affected the community clusters they were able to detect and characterize. Since the ultimate goal would be to automate these tasks, doing so requires fully capturing how to select the parameters required to do so.

-might fill out step 1-3 in further detail: filters? Which filters? Edge, node, email counts, reflexive, directed/undirected, external, duplicate-email address resolution, etc

Models, Algorithms, Measures (this could be more PageRank centric)

We wished to analyze the network using all of the primary methods covered in the course, to discover the hierarchical importance of nodes, and to compare this information with the communities generated by various community detection algorithms. The goal was to have centrality metrics for correlating communities with the centrality indices of their nodes. For instance, the Walktrap community detection uses random walks to detect communities, based on the assumption that communities contain intra-edge densities than edges to other communities. Hence, it is reasonable to hypothesize that betweenness metrics of nodes within such a community should correlate with the communities generated by this method. Our goal was to assess any such correlations between node-centralities and detected communities, and further, to assess how well these communities correspond with the known corporate hierarchy of Enron.

Accordingly, our measures included both global metrics and full node centrality enumerations, in addition to several community detection models. For the sake of analyzing the basic network structure, our global metrics included degree distribution, path-length distribution, diameter, the number of connected components, global clustering coefficient, and average local clustering coefficient. Node centrality metrics included betweenness, Kleinberg’s hub and authority scores, PageRank, eigenvalue centrality, and raw degrees (distinguishing indegree and outdegree for directed graphs). Lastly, we leveraged all of these following existing api’s for community detection:

1. Fast-greedy: Groups nodes by maximizing the modularity score of the resulting network until no further progress can be made (*Clauset, et al*).
2. InfoMap: Combines random walks with information theoretic graph properties to generate communities (*Rosvall, et al*).
3. Leading Eigenvector: Uses an eigenvector-based approach highly similar to fast-greedy, but using the eigenspectrum of matrices to maximize modularity (*Newman*).
4. Label Propagation: Assigns labels to nodes, then updates label assignments based on the dominant node within each node’s neighborhood (Raghavan).
5. Multilevel: Similar to fast-greedy, a bottom-up approach to assigning nodes to communities to maximize a modularity until no further improvement can be made (*Blondel, et al*).
6. Edge-betweenness: Removes high-betweenness edges to separate communities, based on the observation that communities typically have such edges connecting them (*Girvan, et al*).
7. Walktrap: Similar to InfoMap, uses random walks to define communities, based on the intuition that random walks will remain local to a community (*Newman*).
8. Spinglass: Uses an alternative formulation of modularity from statistical physics, minimizing the energy of particle spin-states of nodes until community structures are found (*Reichardt, et al*).

All of the listed implementations are part of the python-igraph api, so calling them to generate communities was trivial. Having so many community detection implementations under a single api made *ad hoc* observations extremely accessible, for which the igraph developers and authors of these methods deserve enormous credit.

[could probably characterize at least one of these algorithms, or generalize all of them in terms of general strategy; or fill-in with pagerank]

Implementation/Analysis

Since we were analyzing a dataset from a real-world human context, we respected the fact that this analysis would require significant iteration. Many network representations of the email data were possible, so we wished to foreground any assumptions or filtering methods when constructing various representations. This placed great importance on writing well-parameterized scripts for generating networks with different characteristics, analyzing them, and reporting the results of each analysis in a structured format. This led to the following workflows, which is relevant to any future network analysis of social or organizational network data.

[Insert workflow graphic]

As previously described, we wished to test various representations of the same network dataset, to analyze such representations, observe results, and to repeat as many times as needed to be confident in our conclusions. The first requirement was to prepare a parameterized build-script capable of consuming the Enron dataset (a very large directory structure of emails) and outputting a network representation in some portable format. As such, we composed a Python build-script capable of consuming the sent email of all Enron employees, extracting their metadata, and finally building a network according to some input parameters. Only sent emails were included, as a method for removing external email and removing duplicate emails.

For greatest flexibility, the following parameters were included: directed/undirected, weighted/unweighted, reflexive/non-reflexive, edge-filtering, node-filter, external-filtering. The first three options simply denote the desired structural properties of the output graph: whether or not it should be directed, weighted, or should allow reflexive loops (emails to oneself). The last three options were important de-noising parameters found in the literature. The edge-filter defines how many emails two individuals must exchange to have an edge, while the node-filter similarly describes how many emails some node must exchange with any others to be included in the network. Implementing these parameters was also important for consistency with other literature, for which an edge filter of 30 and node filter of 5 were suggested in both (*Chapanond*) and [TYLER 2003]. Lastly, the external-filtering parameter describes whether or not to exclude addresses outside of the “@enron” address space. This filter was not used, since many employees used personal email addresses, and likewise, the company was heavily integrated with outsiders in the course of its trading operations and government relations.

In parallel to this work, we also implemented a PageRank analysis as follows…

Results

A summary of our analysis follows, for an undirected graph generated with an edge-filter of 30 emails and a node-filter of 5. The appendix contains complete global data for a subset of the networks we generated. The global structural data of the networks we generated was generally consistent for all of the filters and structure parameters passed to the build script. The raw numerical metrics are shown below, along with the path-length and degree distribution. The path-length and degree distribution show characteristic properties for a social network: high clustering (many sub-communities) gave rise to an exponentially-distributed node-degree, and an overall bell-curve path-length distribution.

[raw data] [degree distribution] [path length distribution]

The centrality indices of all nodes were also compiled. For every centrality measure, all nodes’ values were calculated, placed in a list sorted by these values, and output to a file for that measure. This gave a complete ordering of nodes under each centrality measure, and the top fifteen for each measure were also placed in the global network stats file, giving the following top nodes. The top six for each list are shown here:

Pagerank betweenness Eigenvector Degree Hubs Authority

[insert table of top five nodes per metric]

The community detection methods performed superbly, visually identifying organizational communities in a highly distinct manner:

[fast greedy] [betweenness] [eigvector]

Full-size images are contained in the deliverables for this report, and should be examined separately. However, even the compressed images here clearly show how well the community detection methods worked in terms of separating out the hierarchical groupings of the organization. Note both how strongly each method separated the visually-apparent sub-groups within the company, and further how consistently each community detection method detected these groups. Only three community detection methods are shown here. The remaining outputs are contained in our deliverables and show the same characteristics.

Our initial intention was to somehow numerically analyze the correlation of the communities. However, the plots above provide such an accessible way to visually-analyze the performance of the community detection methods that doing so was not necessary, and would not have been productive given its lack of definition. Further, the communities were surprisingly insensitive to changes in the filtering parameters used to construct each network representation. This reaffirmed to us that no loss of generality was experienced as a result of filtering, as claimed by [CITE SVD FILTERING]. This is also good news, since it means that the network size could be compressed greatly, giving better performance with no loss of community information.

Our primary intention was to perform hierarchical clustering, not simply community detection. However, the visuals of the detected communities, combined with node centrality information, provided a far more organic way to determine hierarchies within the organization. For instance, note that hierarchical clustering could simply be a matter of ordering communities by their “leader” node under any given centrality metric. Visually inspecting the communities and the top-k nodes under various centrality measures showed the organizational structure in a much more natural way. For instance, “jeff.dasovich” was apparently a top government relations official, a logistical communications role that is clearly evident in the high-degree of this node, the size of its community, and the other centrality measures assigned to this dominant node. The isolation of other communities shows the autonomy and specialization of their roles. For instance “kay.mann”, a recurring node of high centrality under multiple measures, provided general legal counsel. Thus, we noted a direct correspondence between many centrality results and the compact communities neatly surrounding these nodes.

Other interesting properties are shown as well. According to reports, “vince.kaminsky” was something of a whistleblower over the company’s accounting practices, and is shown here operating as the central node of an isolated community as a research director. The isolation of this group could have many real implication with regard to company disaffection or other negative relationships with the overall organization. In fact, all of the communities shown above directly correspond with specific operational groups within the company: government relations, legal counsel, transportation, energy-trading networks, and more. Both the size, cohesion, and autonomy of various groups provides important insights into the potential significance of each groups.

A most interesting observation is that the primary communities shown above correspond with individuals whose corporate roles were highly visible, but not necessarily top-tier. We actually expected this result, under which mid-level staff are clearly shown carrying the greatest operational importance, if measured in communications volume. By the Pareto rule, 20% of an employee-based provide 80% of productivity, which is reflected directly in the degree distribution for the graph, and also the detected communities, which are centered on such employees.

On the other hand, chief-executives tend to be more distant from ground operations, and hence they were very difficult for any of the community detection methods to approximate. This shows that the greedy-heuristics inherent in these methods is very good at capturing the bulk operational communities within an organization, but not the organization of the topmost staff. In fact, many top-tier individuals were identified in our data, however their identification and coherence was not as pronounced as the operations-level communities shown above. Therefore, the failure to detect many executive communities has as much to do with realities of corporate governance as with the greedy algorithmic properties of the community detection algorithms. The greed of these algorithms clearly made them insensitive to the subtleties to detect executive board staff, let alone their structure.

Conclusions

In this report, we showed how node centrality metrics and community detection methods can be applied effectively to the task of hierarchical organizational analysis. This work has many real-world applications: detecting anomalies, finding structures of disaffection around employees/groups, improving operational efficiency, estimating work loads, and so on. Perhaps the greatest benefit of this work is simply being able to observe the current state of an organizational network in a highly parameterized manner, for whatever application this may benefit.

As we showed, developing a parameterized network-factory was as important as the raw analysis itself. This is because only by testing many different network representations could we have a high degree of confidence in our observations. This is especially so for human communications networks, since their structural properties can’t be estimated numerically without a lot of manual observation and iteration over different representations.

Future work

Algorithmically, the task of hierarchical clustering encompasses so many parameters that it would be necessary to go into far greater depth

In terms of implementation, the most notable improvement would be to use more generic api’s for some of our graph generation steps. Rather than laboriously writing code to manually extract email information and store this information in some internal graph-like structure in memory, it would be far better to use SQL or a similar data-driven api to store parsed emails in structured tables. This would be far more amenable for performing the flexible queries and generating network representations from the primitive data objects of this schema (emails). It would also make for a more portable and robust implementation, compared with home-rolled code transformations in Python.

References

Chapanond, A., Krishnamoorthy, M. S., & Yener, B. (2005). Graph Theoretic and Spectral Analysis of Enron Email Data. Computational and Mathematical Organizational Theory, 11(3), 265-281.

Clauset, A., M. E. J. Newman and C. Moore. (2004) *Finding community structure in very large networks.* Phys Rev E 70, 066111.

Cohen, William W. The Enron Email Dataset (May 7, 2015 version). Carnegie Mellon University. Retrieved from: www.cs.cmu.edu/˜./enron/.

Gleich, David F. (2015). PageRank Beyond the Web. SIAM Review, Vol. 57, No. 3, pp. 321–363.

Gleich, David F., Abraham Flaxman, Paul Constantine, Asela Gunawardana (2010). Tracking the Random Surfer: Empirically Measured Teleportation Parameters in PageRank.

Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E Phys. Rev. E,* *74*(3). doi:10.1103/physreve.74.036104

Palus, S., Bródka, P., & Kazienko, P. (2011). Evaluation of Organization Structure Based on Email Interactions. *Governance, Communication, and Innovation in a Knowledge Intensive Society,* 117-127. doi:10.4018/978-1-4666-4157-0.ch010.

Pons, P., Lapaty, M. (2006) Computing communities in large networks using random walks. J. Graph Algorithms Appl 10(2), 191-218.

Raghavan, U.N. and Albert, R. and Kumara, S. (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76:036106, 2007.

Reichardt, J., & Bornholdt, S. (2006) Statistical mechanics of community detection. *Physical Review E Phys. Rev. E,* *74*(1). doi:10.1103/physreve.74.016110

Rosvall, M. and C. T. Bergstrom. (2008) *Maps of information flow reveal community structure in complex networks*. PNAS 105, 1118.

Rowe, R., Creamer, G., Hershkop, S., & Stolfo, S. J. (2007). Automated Social Hierarchy Detection Through Email Network Analysis. Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis - WebKDD/SNA-KDD ’07.

Tyler, J.R., M.D. Wilkinson and B.A. Huberman (2003), “Email as Spectroscopy: Automated Discovery of Community Structure within Organizations,” in Proceeding of the International Conference on Communities and Technologies, Netherlands, kluwer Academic Publishers.

VD Blondel, J-L Guillaume, R Lambiotte and E Lefebvre. (2008) Fast unfolding of community hierarchies in large networks. J Stat Mech P10008.