Honeycomb: Visual Analysis of Large Scale Social Networks

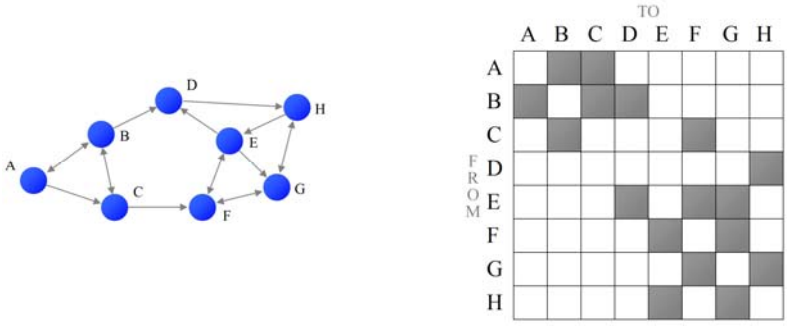
Frank van Ham[1], Hans-Jörg Schulz[2] and Joan M. Dimicco[1]

[1] IBM TJ Watson Research Center, Cambridge, MA 02142, USA
fvanham@us.ibm.com, joan.dimicco@us.ibm.com
[2] University of Rostock, 18051 Rostock, Germany
hjschulz@informatik.uni-rostock.de

Abstract. The rise in the use of social network sites allows us to collect large amounts of user reported data on social structures and analysis of this data could provide useful insights for many of the social sciences. This analysis is typically the domain of Social Network Analysis, and visualization of these structures often proves invaluable in understanding them. However, currently available visual analysis tools are not very well suited to handle the massive scale of this network data, and often resolve to displaying small ego networks or heavily abstracted networks. In this paper, we present Honeycomb, a visualization tool that is able to deal with much larger scale data (with millions of connections), which we illustrate by using a large scale corporate social networking site as an example. Additionally, we introduce a new probability based network metric to guide users to potentially interesting or anomalous patterns and discuss lessons learned during design and implementation.

1 Introduction

Social networks are structures that model the interconnections between people. Analyzing these social networks provides insight into complex phenomena such as organizational behavior, social organization, and remote collaboration. Typically, social network analysts use a combination of metrics and visualizations to determine central actors, important ties and clusters in relatively small networks of at most a few hundred nodes. With the advent of social network sites on the internet, we now have access to user generated networks that are orders of magnitude larger. For example, Facebook.com currently claims over 100 million active users, which generate billions of network connections. Traditional methods of social network analysis would break down at a fraction of this scale. Non-visual approaches have been proposed to address this problem of scale for general network analysis, such as (hierarchical) clustering, sampling, modeling or approximation. All of these methods reduce the amount of data by creating abstractions of the network. Although these abstractions might give us partial insight into higher level structures, many of these approaches abstract the data to the point where it is very hard to relate high level features to real world phenomena. Another complimentary tool in social network analysis is visualization, which uses interactive graphics to depict the structure of the social network. These diagrams can help scientists understand the over



(a) **(b)**

Fig. 1. Social network visualizations. **(a)** A node link representation of a very simple 8 node network **(b)** An adjacency matrix representation of the same network, note that the number of grey cells is equal to the number of links in **(a)**

all structure of a particular network and expose patterns that they might not have been aware of previously, or might not have even considered as a possibility. In effect, visualization allows users to perform exploratory analysis on the network and quickly generate and verify hypotheses. One of the strengths of visualizing social networks in particular is that it supports hypothesis generation about an organization's structure and interactions, something of great interest to the organizational behavior and CSCW communities [14]. But typical graph visualizations are limited in their ability to support end users in interpretation of the information and the sheer scale of the networks makes a visual mapping challenging. Social scientists have noted that "…it would seem useful for management to map the social capital ties that are relevant to the various tasks the organization faces. This mapping poses a considerable challenge: from a purely technical point of view it is far easier to map a small number of ego networks than to generate an intelligible sociocentric, whole-network map of a large, complex organisation" [2]. In this paper we present a matrix based tool for exploring very large social networks in their entirety. In contrast with other existing tools, it has the following capabilities:

- **scalability**: Although the social applications we analyzed are accessible to company employees only, and their size pales in comparison to the many millions of members of some online sites, the network we are dealing with in this paper has over 37,000 users and over 400,000 connections. Our tool handles up to a million connections with ease and we have been able to navigate synthetic datasets of up to a few million edges.
- **ability to perform temporal analysis**: Social network analysis is often challenging, but grasping the changes in a social network over time is especially difficult. These dynamic aspects may give us insight in the spread of a network over time and allow us to create better network models and predictions.
- **displaying information that is absent**: While most traditional information visualization tools can show individual information elements and the patterns emanating from them, it is much harder to show the absence of information where it was expected. Concretely, in social network analysis it might be desirable to see between which groups connections are absent or very minimal.

The main contribution of this paper is to illustrate how we can use a matrix based visualization in combination with algorithmic metrics to help social scientists grasp large social network datasets and potentially guide them to interesting features. We illustrate the usefulness of this approach by providing real world samples. In the next section we describe related work in this area. We then give a brief description of our sample dataset, followed by an outline of the visualization method. The main part of this paper describes the visualization and edge metrics as well as some of the insights we have gained in applying them. Finally, we discuss implications for the design of these types of tools.

2 Related Work

The analysis of social networks with statistical and graph theoretical means has had a major influence on social sciences ever since it was first proposed [7]. From the beginning, sociograms and sociomatrices were used as representations of the analyzed social networks. Sociomatrices cross-tabulate a particular connection metric over a number of actors (nodes on the network) and sociograms are traditional node-link representations of the network. Both of these representations have been used as a basis for interactive analysis tools.

The node-link representations in sociograms (Fig. 1a) have been echoed in many social network visualization tools from different research communities. From social network analysis, tools like UCINET/Netdraw and Krackplot [13] offer support for advanced metrics and offer basic node-link visualizations. From the graph drawing community, tools like Visone [4] offer a combination of metrics and advanced layout algorithms. From the visualization/HCI community tools like Vizster [9], the more advanced SocialAction [17] allows users to interactively examine visual representations of (social) networks. All of the above approaches can generate very readable diagrams of social networks of a few thousand nodes at most but have the disadvantage that they cannot display very large networks, especially if these networks are dense. Among the most advanced workbenches for performing social network analysis on large networks are Netminer and Pajek [3] which both offer a large selection of analysis algorithms as well as matrix and node-link visualizations. Users can use the supplied algorithms to extract and visualize meaningful subgraphs from a large network. However, integrating these different views on the network in a single mental model can be a daunting task.

Alternatively, sociomatrices can very easily be transformed into adjacency matrices, where each cell *(i,j)* stores the strength of the connection between actors *i* and *j*. We can then render this matrix by coloring a cell if there is a connection present and, optionally, visually mapping the strength of the connection (Fig. 1b). Note that any colored cells on the diagonal indicate connections of nodes to themselves. Reorderable matrices have been used as an analysis tool since the early 80's and have a number of distinct advantages over node link diagrams, especially if the network is dense [8]. They are impervious to clutter and overlap, scale much better and allow quick verification of the existence of a connection. When the columns are reordered properly [16] they can also be used to identify clusters in the network. On the downside, matrices are not great at

visualizing paths between multiple nodes and are not nearly as intuitive as node-link diagrams, which may explain why they are currently underrepresented in social network analysis. Recently, interactive adjacency matrices have been advocated for medium scale social network visualization in [10] and a sophisticated hybrid approach has been proposed in [11].

Outside the social networks analysis community, matrix visualizations have been used to analyze different types of networks, however the problem of scalability remains. One potential solution is to group the nodes in the network into clusters and render the aggregated network instead. This can then be repeated at multiple levels of scale, depending on the size of the input graph. This hierarchical grouping can then be used to create a single interactive visualization that allows the user to browse the network at different levels of scale [18, 1, 6]. The major problem with these prior approaches is that, without methods of automatically highlighting potentially interesting or anomalous data points, the user can spend a significant amount of time browsing these representations at multiple levels of scale without learning anything new.

3 Data

Our main data set is taken from a social network site running internally at a large multinational company. The site is an opt-in site that people use for connecting with other employees, through the sharing of photos, lists and events [5]. Part of the process of connecting is to directly link to other employees, as is typical on any social network site. When a user on the site adds someone as a connection, the other user is not required to reciprocate the connection, so employees can connect to anyone inside the company, without the need for the other person to join the site first or reciprocate back afterwards. The data set we use as a sample is a snapshot that was taken mid-July 2008. At that time, 37,000 employees had joined the site and they had formed approximately 300,000 connections.

4 Visualization

Given the scale of our target dataset rendering a single 37,000 by 37,000 matrix is impractical from both a resource and user interaction perspective. We therefore follow the approach taken by [18] and use a hierarchy on the nodeset to reduce the size of the adjacency matrix. Fig. 2 shows how we can collapse an 8 by 8 to a 3 by 3 matrix by using a predefined hierarchy to aggregate cells. The resulting collapsed matrix color codes the total number of edges for each submatrix below (darker colors indicating more edges) and still maintains many of the features of the original network. For example from Fig. 2b we can still easily see that there are no connections from groups *J* and *K* to group *I* and that there are relatively many connection among nodes in group *I*. As a final step, we normalize the cell sizes so that we can repeat the same process with the collapsed matrix. Note that, even though the relative difference in cell sizes in Fig. 2b accurately

portrays the difference in the number of leaf nodes, we found adjacency matrices with irregular cell sizes much harder to comprehend when they grow larger, especially when

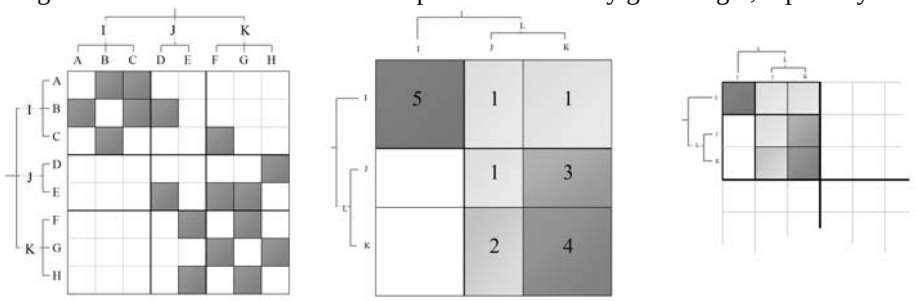


Fig. 2. Collapsing an 8 by 8 adjacency matrix to a smaller 3 by 3 matrix **(a)** original matrix with node hierarchy on both sides **(b)** collapsed version of the matrix with lowest level of the hierarchy eliminated and edge counts aggregated **(c)** This collapsed version itself forms a small section of a higher level adjacency matrix.

the difference in cell sizes is large. For that reason we will use a regular subdivision in the samples in this paper, even though our visualization tool allows users to choose either option. The actual hierarchy used to drive the above process is variable and depends on the interest of the users of the visualization. In the samples in this paper we have used two distinct hierarchies. One uses the management hierarchy to correlate connection behavior with organizational structure, while the other one uses a geographical hierarchy based on the user's working location (i.e. continent - country - state - city - building) to correlate connection behavior with geographical location. In practice, we can also use different hierarchies or construct a hierarchy ourselves by using other node attribute information.

In terms of interactivity our tool is very similar to its predecessor described in [18] but it is more memory efficient and allows for pluggable metrics. The user is initially presented with an adjacency matrix that displays connections at the highest level of abstraction (e.g. in the case of the geographical hierarchy connections between employees in different continents). By left clicking on a cell *(X,Y)* the user can indicate he or she wants to examine that particular connection in more detail and the visualization then displays the matrix that shows the connections between the direct children of *X* and *Y*. A simple right click brings the user back to the cell he or she came from. The transition between these two matrices is animated to help the user understand the relationship between the two representations. Dynamic labels help the user understand what relation they are looking at and a popup menu provides details on demand.

To deal with the issue of visual scalability we have used the hierarchy to reduce the matrix to a more manageable size. Computational scalability is obtained by using a semi-external memory approach, that is, we keep the entire nodeset and the hierarchy of the network in RAM while a relational database stores the actual connections between the nodes in the network. When a user requests a higher level view of the network, aggregation of edges in the database is done on the fly using a fast lookup algorithm. Our current prototype is implemented in Java and uses OpenGL for graphics output. We

have successfully loaded and navigated synthetic graphs up to 5 million edges using only 200MB of RAM.

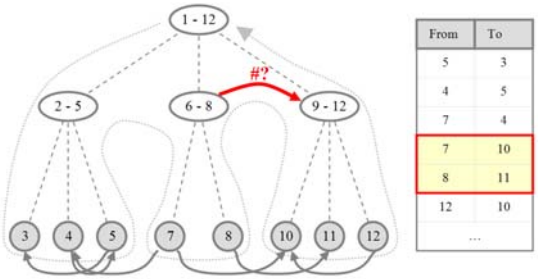


Fig. 3. Schematic representation of the network (grey nodes and curved connections) and the aggregation hierarchy. By numbering the nodes in the aggregation hierarchy in a depth first manner (dotted line) and keeping track of the minimum and maximum values encountered during this traversal we can determine the number of edges connecting groups (6-8) and (9-12) by running the query: SELECT COUNT * FROM EDGES WHERE 6 'From' 8 AND 9 'To' 12;

5 Metrics

Since every edge in the network is represented by a cell in the matrix, we can use the space in this cell to communicate information about the connection it represents. Although some structural edge metrics exist [15] they are mainly focused at identifying individual edges that connect different communities and rarely take edge weights into account. This means that although they are able to identify communities at the level of individual actors, running them on aggregated data produces less reliable results and in some cases might be too computationally expensive. Previous approaches [18, 1] have exclusively mapped the number of aggregated edges, or the *connection count*. We can express the connection count *CC* for an edge *(X,Y)* as:

$$CC(X,Y) = \|(x,y) \in E : x \in desc(X)\ and\ y \in desc(Y)\|$$

with *desc(X)* indicating the set of descendant leaves of a node in the hierarchy. To be able to compute connection counts between groups efficiently we employ a special numbering scheme on the nodes in the network. We traverse the hierarchy on the nodes in the network in a depth first traversal starting at the root, and incrementally number the encountered nodes. For each node in the hierarchy we keep track of the minimum and maximum value encountered for that node. We can then use these Depth First Search numbers to query the edgelist stored in a database whenever we need the total number of connections between two groups (see Fig. 3 for details).

Combined with caching of the top level matrices this keeps our tool memory-efficient and fast. Alternatively, a solution where all edge counts are precomputed and stored in a disk based index can be used [6], although this makes it harder to switch between multiple hierarchies.

Although the connection count is probably the most straightforward metric one can think of, it has a number of drawbacks. Firstly, at higher abstraction levels the adjacency matrices grow increasingly dense, making it harder to detect patterns based on the presence of connections. The use of a color scale to indicate the number of connections

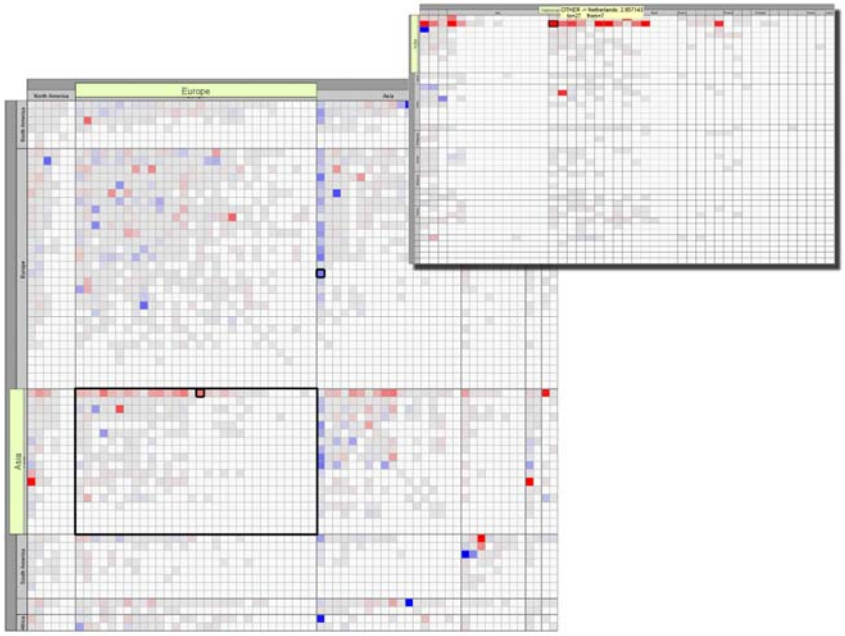


Fig. 4. shows the difference between incoming and outgoing connections, with India showing up as an outlier. The horizontal red band and vertical blue bands indicate there are many unreciprocated connections from Indian employees. The inset shows a zoomed view of the outlined cell with the connections between Asia and Europe, the red horizontal band represents connections from mobile employees (with location OTHER) in India to people in Europe.

might help somewhat, but the large number of connections on the diagonal in social networks often drown out subtler off-diagonal patterns.

Finally, one key problem in all previous matrix-oriented visualization approaches is that it is very hard for a user to identify what connection patterns are potentially interesting to examine further. In many instances the user is simply presented with a matrix of aggregated connection counts and is expected to identify anomalous patterns by interactively browsing the data. Although the user can now potentially look at interesting patterns, this ability is useless if the user has to spend a large amount of time inspecting abstract matrices at different levels of detail to find these interesting patterns. There is a definitive need for edge-oriented metrics that go beyond simple edge counts, yet can still be computed efficiently. In the next two sections we will propose two of these metrics and use our sample dataset to highlight their usefulness.

5.1 Asymmetry

Since connection behavior does not need to be reciprocated (i.e. *X* can connect to *Y* but *Y* does not have to connect back to *X*) it might be interesting to look for patterns that involve asymmetric connections. For example, for each edge *(X,Y)* we can define the asymmetry as

$$ASYM(X,Y) = CC(X,Y)/CC(Y,X)$$

Fig. 4 shows the asymmetry in connections, using the geographical hierarchy to aggregate. Small imbalances in asymmetries can distract from detecting large imbalances, so we only plot asymmetry values above a small cutoff. One pattern that immediately emerges from this image is the large horizontal (red) and vertical (blue) bars in the row and column representing India. This indicates India overall has more connections going out to another country than it has coming back. The team that had developed the social networking application was aware of the fact that a lot of US users were getting friend requests from users in India they did not know personally, but did not know that this pattern persisted over different countries. To determine where this behavior was coming from, we drilled down on the connections from India to Europe (inset) and observed that most of the people responsible for these asymmetric connections report their work location as 'OTHER', which in practice means 'non-traditional office'. Employees working in non-traditional offices (typically drop-in stations or from home) may be using the social network site with a particular focus on meeting new people, because they do not have regular face-to-face contact with coworkers.

In a separate survey of 2000 users of this social network service, we asked about different reasons why they were motivated to use the site. Comparing the responses between countries, users in India reported to a significantly higher degree than other countries that they were using the site for getting to know people they would not otherwise meet at the company, using the site to find experts and using the site to discover people with similar interests. All three of these activities involve reaching out and connecting across organizational and cultural boundaries, and these motivations offer a partial explanation of why this particular cluster of outward links may exist. Asymmetry-type metrics can also be useful in wider contexts, especially if the connection's weights can be significantly different for each direction. Examples of such networks include trade, financial networks or communication networks.

5.2 Deviation from expected

One of the major disadvantages of rendering absolute metrics, whether they are absolute connection counts or the absolute number of unreciprocated connections, is that groups with the largest number of users are very likely to have the highest number of connections. Indeed absolute connection counts allow us to make observations such as 'Users in the US have 3024 connections to users in India' but there is no way to estimate how valuable that observation actually is. Is 3024 connections a lot, given the characteristics of the network and the size of the two countries, or did we expect to see more?

One way of estimating the usefulness of an observation is to try and determine what part of this observation might be explained by pure chance. In other words, if we had distributed the *E* edges in the graph completely arbitrarily, how many edges can we expect to fall between the US and India? This problem is similar to a chi-square analysis of a set of observations, using the matrix *M* of connection counts as the contingency matrix. Note that the total number of connections in a row *X* of *M* is equal to the total number of edges X_{out} that have their startpoint in *X*, and the total number of connections in a column *Y* of *M* is equal to the number of edges Y_{in} that have their endpoint in Y. The probability of having an edge connecting *X* and *Y* if the data were randomly distributed (i.e. the choice of start and endpoint of an edge are independent) is then equal to $P(X,Y) = \frac{X_{out}}{E} * \frac{Y_{in}}{E}$ with expectation:

$$EXP(X,Y) = E * P(X,Y) = \frac{X_{out} \cdot Y_{in}}{E}$$

We can then compute the chi square metric for the total adjacency matrix, which tells us if there are significant correlations between the clusters we have defined on the nodeset. If that is the case (and in our particular case it was) we can look at the value of the individual cells and try to establish where these correlations might originate from.

As an alternative derivation, we can look at the problem of obtaining a graph with exactly *E* edges, of which *k* connect nodes *X* and *Y* as a discrete probability problem: Suppose we have a 2 dimensional *R* by *C* grid *M* (our adjacency matrix) in which we have to assign edges to cells. In order to be able to generate all possible assignments we have a jar of $R \times C - W$ black and *W* white edges. White edges are to be placed in a specially designated subgrid of *M* of size *X* by *Y*. Black edges have to be placed outside of this subgrid. If we sample *E* edges from the jar *without replacement* and the probability of pulling a white edge from the jar is *P(X,Y)* as defined above, what is the probability that exactly *k* edges will be white? This problem can be modeled as a hypergeometric distribution which has the same expectation value $E * \frac{RC*P(X,Y)}{RC} = EXP(X,Y)$ and variance:

$$VAR(X,Y) = EXP(X,Y) * (1 - P(X,Y)) * \frac{R*C - E}{R*C - 1}$$

The remaining free variables in the above expression are the values for *R* and *C*. These determine in relation to what part of the matrix we compute the variance and expectancy. Typically $R \times C = N^2$ if we want to look at the distribution of the edges over the whole graph, but we can also compute the expected values given the characteristics of a particular subcell (typically an ancestor of the cell we are looking at) in the matrix. The metric outlined above is in effect a more advanced version of a basic density metric, which estimates the number of expected edges in a matrix cell solely based on cell size, i.e. $P_{density}(X,Y) = \frac{X}{R}\frac{Y}{C}$ The above metric instead uses the in- and out degree distribution of the graph under consideration to estimate the number of expected edges, similar to the p_1 network model [12].

A good test case for our deviation metric would be to see if we can correlate the company's business units to their respective geographies. As an example, we would expect employees of the German business unit to have most of their contacts with em-

ployees in Germany. Other business units span different geographies however, and our metric should also point this out. Although up to now we have only shown matrices that use the same hierarchy on both sides, the visualization can also support different hierarchies on the two sides of the matrix. Fig. 5a shows part of the matrix that displays the links from the managerial hierarchy (rows) to the geographical hierarchy (columns) with manual annotations. Color represents the deviation from the expected value with blue cells indicating fewer connections than expected and red cells indicating more connections than expected. The size of the glyph in each cell gives an indication of the

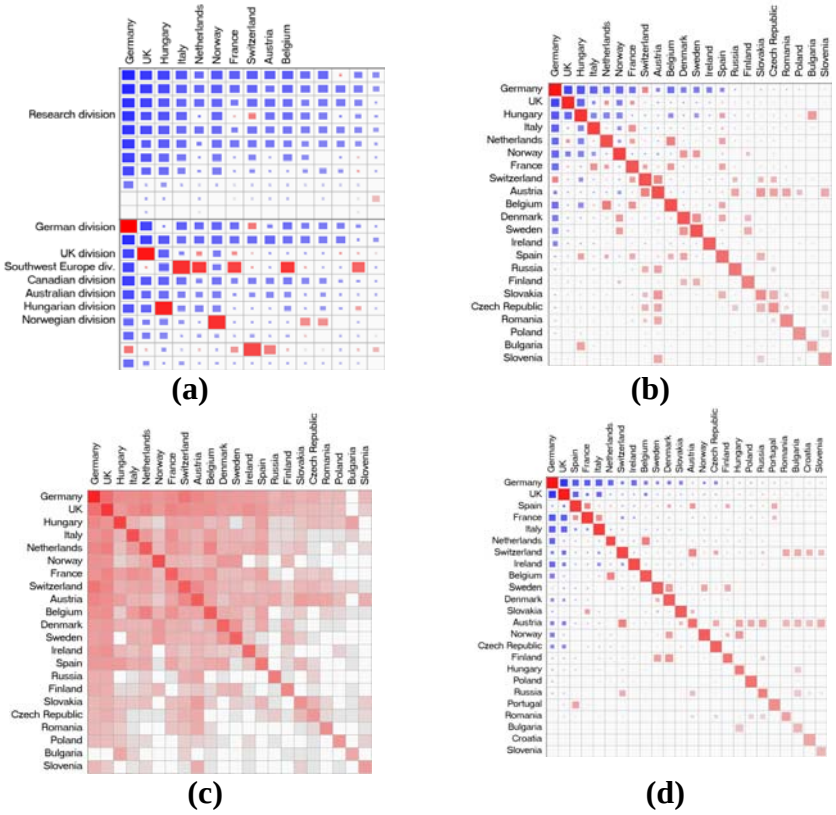


Fig. 5. Mapping the difference between the observed number of connections and the number of connections we would expect if the distribution of the edges were random. Red cells indicate a higher number of connections than expected, blue cells indicate a less than expected number of connections. The size of each square indicates the significance of the deviation. **(a)** connections between divisions and countries **(b)** connections among different European countries. **(c)** same view as **(b)** but using the connection count metric. **(d)** filesharing activity between different European countries.

significance of the deviation from the expected number as cells that deviate only in a small amount from the expected value are within the bounds of normality and should not be visually highlighted. The expected patterns show up very clearly, but we also observed connections that might not have been so obvious initially, with employees in

research connecting to employees in Switzerland. This is most likely due to a large corporate research laboratory in Switzerland, but the ability to isolate this fact is encouraging.

We also applied the same metric to look at connections within Europe (Fig. 5b) and found a number of interesting patterns. One fairly obvious one is the tendency to connect to people in the same country, indicated by the bright red diagonal. Other observed patterns have to do with language and geographical proximity (for example, Switzerland connects more than expected to Germany, Austria and France) or with organizational structures (Sweden, Norway, Finland and Denmark are all part of a single organizational structure). One pattern we did not expect to find were the larger than expected number of connections between Austria and many of the countries in Eastern Europe. We were subsequently able to verify that the company, like many other multi-national firms, handles a significant part of its Eastern European business from Vienna.

As a second check on the validity of these observations we explored a different set of connections. The company also uses internal online document sharing services where people can upload a single document and share it with multiple colleagues, which saves the user from having to mass-mail the document. We visualized a set of 75,000 document sharing relationships, where two actors are related if one of them has shared a document with the other. Fig. 5d shows document sharing relationships over different countries in Europe. Again, Austria appears to connect significantly more to Eastern Europe than other countries. The asymmetry is expected since sharing a file is an activity that is usually not reciprocated by the receiving user. Still, the rough similarities between the images in Fig. 5b and Fig. 5d verify that the patterns seen in both the social network service and the file-sharing service reflect meaningful organizational patterns in the company.

The practice of computing statistics for different network decompositions is an established procedure in the social network analysis community. However, the results are often displayed in simple statistical tables, which makes interpretation not very intuitive. The added value here is that we can display the resulting deviations from the expected values in context, use a comprehensive display to look for patterns and easily compare them with other network metrics at different levels of granularity to investigate. Within the context of a corporation, social network analysis is often touted as a mechanism for revealing the invisible power structures outside of the management hierarchy. One of the unique strengths of the visualization is that the structure of the social network is overlaid onto the formal hierarchy of the company (divisional or geographical) allowing comparison of the formal and informal hierarchies in the company.

5.3 Time varying Networks

Another distinct advantage that matrix based visualizations have over node link based ones is their ability to deal with data that varies over time. Many other approaches have used animated node-link diagrams to display changing networks but this approach has a major drawback. If node positions are kept static over time, the topology of the network will not accurately reflect the data at that point in time. Yet if we animate the node positions over time, the user is confronted with an animation that both changes the geometry

and the connectivity pattern simultaneously, which in practice is very hard to understand. In matrix visualizations, each edge (whether realized or not) is already allocated a section of screen space, which allows us to animate the growth of connections in a stable visual representation. Depending on user interest we could for example choose to display the cumulative edge count over time, a heatmap of the new connections made in the last month or highlight the most volatile connections. In our prototype we have implemented a slider that lets us visualize any time interval between the launch of the site and the date of the snapshot and apply any of the metrics to the connections that were made in that time interval.

6 Lessons learned

In this paper we have shown that interactive adjacency matrices are a viable alternative and, in many cases, preferable to node-link diagrams when it comes to analyzing large scale social networks. Large scale formal evaluation on these types of tools is difficult because exploratory analysis inherently involves tasks which are ill-defined. That is, the user does not know what they are looking for exactly (yet). We have illustrated their usefulness by performing exploratory analysis on real-world large social networks obtained from an internal company research project. This section discusses some lessons learned during design and implementation.

6.1 Integrate multiple metrics in a single mental model

Scientists in the area of social network analysis have come up with a number of useful metrics to analyze the connection patterns in social networks. However, understanding the structure of a social network requires analysts to understand how the different metrics in the social networks interrelate. Many existing tools allow the user to run an analysis and then visualize the results [3] but integrating all these separate perspectives into a single, coherent mental model is often left up to the user. Having a single consistent mental model of a complex data structure allows user to incrementally build up knowledge by allowing easier correlation of newly observed facts with previously observed facts. As an example, we observed that Austria has strong ties to other Eastern European countries. When we looked to verify this observation using a different network of the same individuals, we could easily identify a similar pattern because connections from Austria were visualization in the same row and formed the same horizontal pattern. Had we tried to do this analysis with node-link diagrams the layouts of the two networks would have varied wildly, making direct comparison much harder.

6.2 Use concrete hierarchies to drive the analysis

One of the prerequisites of this visualization technique is a suitable hierarchy on the set of nodes in the network. Here, we have used readily available hierarchical decompositions such as the management hierarchy. This allows us to interpret connections between

higher level aggregations directly, because they relate to a concrete and meaningful grouping. Previous approaches [1, 6] have used network clustering or matching algorithms to automatically generate a hierarchy on a network when none was available. Although these approaches offer the similar scalability and corresponding interaction model, interpretation of the resulting visualization is often problematic because it is much harder to put a meaningful label on an individual cluster. Recursive clustering algorithms are more problematic as we do not know what the common attributes of nodes in a group might be. This makes it much harder to ground the interpretation of the resulting structure in existing knowledge. If no natural hierarchy is available for the network under consideration we recommend using ordinal node attributes to create a hierarchical partitioning if the number of categories. That is, we can recursively aggregate nodes in the network by grouping them if they have the same value for a particular attribute. This approach is in a sense a hierarchical version of the idea behind PivotGraph [19].

6.3 Absent data is also information

One of the chief advantages matrices have over node-link diagrams is their ability to highlight missing connections. For denser networks, the absence of a connection where one was expected might be just as informative as the presence of a connection where it was not expected. Matrix views explicitly represent absent connections as an empty cell, which allows us to look for patterns that involve empty cells. Sample patterns may include sparsely filled rows or columns in an otherwise dense matrix or (almost) empty aggregate cells. Similarly, computing the difference between a statistically expected value and the observed value allows us to examine a higher order of absence: given a suitable baseline model we can determine how many connections we would have expected and highlight the difference. Many current information techniques only focus on highlighting patterns that are present, often neglecting patterns that are notably absent.

7 Conclusions and future work

We have shown how matrix based visualization tools have significant advantages over node link diagrams when it comes to analyzing very large networks in general and large social networks in specific. They deal better with denser networks, offer ample screen-space to display metrics for each edge and can more easily display change over time. We have also shown how we can use statistical measures to estimate the usefulness of a particular observation. The one major disadvantage that matrices have over traditional node-link diagrams is that they are unable to display structural features such as shortest or multi-segments paths in the network in an intuitive manner. However, aggregating networks greatly increases their density and in most cases the high level overviews generated here are too dense to display in node-link form.

The approach outlined here opens up a number of viable routes for further work. Having a common, consistent ’data space’ available may offer possibilities for collaborative analytics where multiple users analyze a potentially large network dataset. One

can think of highlighting areas that have already been examined to guide users to previously unexplored sections of the matrix or allowing users to collaboratively annotate the visualization for example. In terms of metrics, our expectancy metric provides promising initial results, but still overemphasizes the tendency of actors to connect within their own community. Better random network models that take this into account may provide a better assessment of what datapoints (e.g. connections) are interesting and which are not. Alternatively, secondary network data (for example, obtained from email exchanges or other known organizational patterns) might be overlaid to see where users' connections deviate. In terms of visualization one could imagine implementing a flexible hierarchy based on node attributes, where users can quickly try out partitions to see where correlations with structure might lie. Also, our current visualization does not allow users to view connections between groups at different levels in the hierarchy simultaneously and asymmetric zooming might be helpful here. Finally, application of this type of visualization techniques to graphs from different application areas may give researchers insight in much larger networks than previously possible.

References

1. J. Abello and F. van Ham. Matrix zoom: A visual interface to semi-external graphs. In *Proceedings of the IEEE InfoVis 2004*, pages 183–190, 2004.
2. P. S. Adler and S.-W. Kwon. Social capital: Prospects for a new concept. *The Academy of Management Review*, 27(1):17, 2002.
3. V. Batagelj and A. Mrvar. Pajek - program for large network analysis. *Connections*, 21:47–57, 1998.
4. U. Brandes and D. Wagner. visone – analysis and visualization of social networks. In M. Jünger and P. Mutzel, editors, *Graph Drawing Software*, pages 321–340. Springer, 2004.
5. J. M. DiMicco, D. R. Millen, W. Geyer, C. Dugan, B. Brownholtz, and M. Muller. Motivations for social networking at work. In *Proceedings CSCW '08*, 2008.
6. N. Elmqvist, T.-N. Do, H. Goodell, N. Henry, and J.-D. Fekete. Zame: Interactive large-scale graph visualization. In *Proceedings of the IEEE Pacific Visualization Symposium 2008*, pages 215–222, 2008.
7. L. C. Freeman. *The Development of Social Network Analysis: A Study in the Sociology of Science*. Empirical Press, 2004.
8. M. Ghoniem, J.-D. Fekete, and P. Castagliola. On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information Visualization*, 4(2):114—135, 2005.
9. J. Heer and D. Boyd. Vizster: Visualizing online social networks. In *Proceedings of the IEEE InfoVis 2005*, pages 32–39, 2005.
10. N. Henry and J.-D. Fekete. Matlink: Enhanced matrix visualization for analyzing social networks. In *Proceedings of INTERACT 2007*, pages 288–302, 2007.
11. N. Henry, J.-D. Fekete, and M. J. McGuffin. Nodetrix: a hybrid visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1302–1309, 2007.
12. P. Holland and S. Leinhardt. An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 77:33–50, 1981.
13. D. Krackhardt, J. Blythe, and C. McGrath. Krackplot 3.0: An improved network drawing program. *Connections*, 17(2):53–55, 1994.
14. D. McDonald and S. F. D. Fisher. Workshop on social networks for design and analysis: Using network information in CSCW. In *Proceedings of CSCW '04*, 2004.

15. G. Melançon and A. Sallaberry. Edge metrics for visual graph analytics: A comparative study. In *12th International Conference on Information Visualisation (IV08)*, pages 610–615, 2008.
16. C. Mueller, B. Martin, and A. Lumsdaine. A comparison of vertex ordering algorithms for large graph visualization. *International Asia-Pacific Symposium on Visualization*, pages 141–148, 2007.
17. A. Perer and B. Shneiderman. Integrating statistics and visualization: case studies of gaining clarity during exploratory data analysis. In *Proceedings of CHI 2008*, pages 265–274, 2008.
18. F. van Ham. Using multilevel call matrices in large software projects. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 227—232, 2003.
19. M. Wattenberg. Visual exploration of multivariate graphs. In *Proceedings CHI '06*, pages 811–819, New York, NY, USA, 2006. ACM.