Detecting Communities in Professional Social Networks

Group Name: Network Labs

Group Number: 07

Group Members:

JAYAKUMAR ALAGAPPAN MEENAKSHI (A0134431U)

PUNEET GOYAL (A0123688W)

SEQUEIRA IRVING SAVIO (A0126747X)

1.1 Problem Description

With the increase in various social networks, including professional networks, telecommunication networks, office networks, academic networks, etc., the use of social computing is also increasing. The combination of the social networks with graph mining techniques has opened vast opportunities for social computing. Detecting communities of users can be of great importance in a social network. For example, understanding the evolution of a community may help in identifying trends and patterns, and in inferring how their properties change over time. Moreover, the identified communities may be used to derive business insights.

LinkedIn is the most prominently used professional social network platform. LinkedIn allows professionals to connect in various forms, follow groups based on common interests, create professional network by connecting with people and share their profile. The users on LinkedIn share their work experience, current job status, educational background and much other such information. We aim to **leverage this information** and **detect communities in LinkedIn network**. We will test and apply the existing community detection algorithms - **Infomap, Spectral Clustering and Girvan-Newman** on our dataset and **answer the business scenarios**.

1.2 Proposed Contribution

Many community discovery algorithms have been proposed so far, but **none of them have been subjected to strict tests** to **evaluate their performance in professional social networks**. The greater part of the sporadic tests performed so far included minuscule networks with Kenned community structure or synthetic graphs with a simplified structure which is extremely uncommon in real professional networks.

In this project we test some of the **popular community detection algorithms** for **directed/undirected** graphs - **Infomap, Spectral Clustering and Girvan-Newman** in the **LinkedIn dataset** and find the **best performing algorithm** for **each of the considered edge weights** (i.e. **number of employees switching** from one organization to another and **mean duration of the individuals switching**) with reverence to the **internal/external evaluation performance metrics**. Then the **identified best algorithm** with respect to each edge weight would be **run against the US/SG - IT data subsets** and the **trend pattern would be analyzed**.

1.3 Project Objectives

For a **particular geography** we aim to analyze the **movement of employees from one company to another within a particular community**. The movement within a community will help us to identify the **most prominent/liked companies in that industry**. Another use case that will be interesting to extract from the analysis could be the movement of employees who have just 5 years of work experience and the movement of employees who have more than 25 years of experience. The idea behind this analysis is to understand how the movement of people with low work experience differs from those who are savvier and have rich work experience. Similar to this we will try to explore more movements within the communities formed and answer interesting and relevant business cases.

1.4 Assumptions

Current location is considered for gathering individual profiles under the same region for all the past businesses/ employments i.e. all the past employments are considered to have occurred in the same geographical area.

2.1 Data Collection

A basic parser was written in Python to scrape information from the public profiles of LinkedIn.

API	LinkedIn REST API
Language	Python 2.7
Framework	Scrapy 0.2.4

Sample Request

The LinkedIn REST API requires the public profile url of the users to parse for their competences, education and other individual information (i.e. roles/designations held in organizations, start date - end date in a particular role, number of connections in the network and et el.).

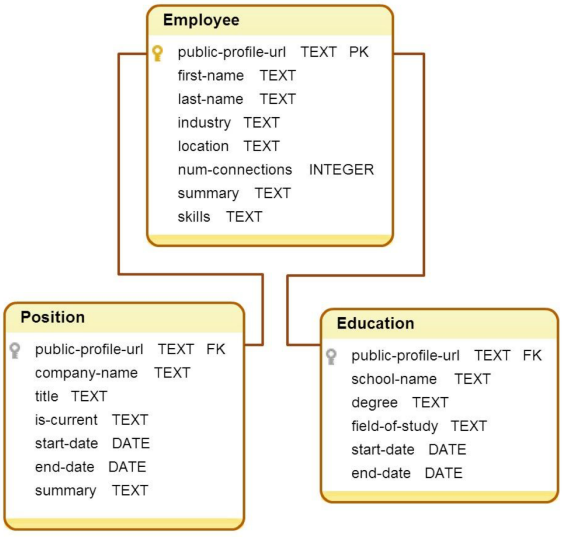
```
localhost/parse?url=<public_profile_url>&format=json
```

Sample Output

Unless otherwise designated, the greater part of LinkedIn's APIs will give back the data in the XML data format. In the event that it is more convenient for one to work with data in JSON format, one can ask for the APIs return in JSON data format.

Relevant fields were extracted from the collected data using Python scripts and were split in to table- organized CSVs.


```
{
  "positions":[
    {
      "summary":"Antique and antique restoration company.",
      "title":"Owner",
      "start-date":"2005-06-01",
      "is-current":true,
      "company-name":"Salt Point Restorations"
    },
  ],
  "public-profile-url":"/pub/a/44/932/9b3",
  "location":"Greater New York City Area",
  "first-name":"Gina",
  "num-connections":"60",
  "educations":[
    {
      "school-name":"Rollins College & Denver University"
    }
  ],
  "last-name":"Benjamin",
  "industry":"Furniture"
}
```

2.2 Data Preprocessing

Remove Outliers

In the first phase, the outliers are abstracted. Very often, there subsists data objects that do not comply with the general demeanor or model of the data. Such data objects, which are grossly different from or inconsistently erratic with the remaining set of data are termed as outliers.

Identified outliers:

- HTML/JS content in the scraped data.
- Invalid Organization names (Special/Non ASCII characters in the names)
- NULL values in the required fields - Organization name, location, industry and start date - end date of employment.

Identification of data subsets

For a specific topography (i.e. USA and Singapore) we plan to identify the movement of individuals from one organization to another within a particular industry (i.e. Information Technology (IT & Services, Computer H/W and S/W, Telecom, Networking, Security and Internet)). Hence the gathered data is filtered to fetch geospatial - industrial subsets.

For location based filtering of US data, only prominent Software industry locations were considered. Number of areas considered were 74. These locations were chosen on the view of employee concentration.

Graph Construction

The **nodes** of the graph represent the **organizations** in the network. The **directed edges** represent the **movement of employees** between the organizations. The below designated information are acclimated to construct the **edge weights**.

1) Mean duration in a categorical role/designation held by employees in an organization.

2) Total number of employees moving from one organization to another.

Say there exists an edge between organizations A and B,

Directed Graph - The number of switches from A to B/mean duration in a role held by individuals switching from A to B is assigned as the edge weight.

Undirected Graph - The total number of switches from A to B - B to A/mean duration in a role held by individuals switching from A to B - B to A is assigned as the edge weight.

Lot of disconnected components were found in the US - Information Technology dataset. Only the edges with weight of at least 2 were considered to model strong pairwise relationships between organizations in the network (i.e. at the least 2 switches were made from the former organization to the latter).

Dataset	Nodes	Edges
US-IT	7229	26555 (Number of switches > 1)
Singapore - IT	6672	12287

Normalization

Normalization is characterized as the procedure of organizing data for **more efficient access**. Min-max normalization technique is sought for normalizing the edge weight values. It performs a linear transformation on the pristine data values.

Assume that **min_U** and **max_U** are the minimum and maximum of **feature X**, the interval **[min_U, max_U]** is mapped into a incipient interval **[$newMin_U$, $newMax_U$]**. So as to limit to professional networks, the scope of the incipient interval is considered as **[0, 1][14]**. Subsequently, every value **V** from the pristine interval will be mapped into value **new(V)** utilizing the following formula:

$$\mathrm{new(V)} = \frac{\mathrm{V} - \mathrm{min_U}}{\mathrm{max_U} - \mathrm{min_U}} * (\mathrm{newMax_U} - \mathrm{newMin_U} + \mathrm{newMin_U})$$

Sampling:

The extraction of data from LinkedIn ensures that complete data to be crawled and that it is the representation of the actual IT industry of US and Singapore. While extracting the data we ensured that people who have moved from one company to another due to merger & acquisition are treated as one company. We also restricted our analysis to the number of switches for a company to be at least 10 in case of US and 5 in case of Singapore to make sure that the probability of a person from a particular company to have an account on LinkedIn increases.

3.1 Community Detection

Community detection is a consequential part of network analysis. The objective is to detect how nodes in the graph ought to be assembled into communities. It is the problem of classifying the nodes of the graph into subsets **$C_i \subseteq V$, 0 < i=K** , such that nodes belonging to a subset are all loosely related and **K** refers to the number of communities **[1].**

Some algorithms completely partition the nodes while others allow for some nodes to be considered as community-less. We'll visually perceive in this work that community detection can withal be quite utilizable to derive features for relegation tasks where the data includes interactions that are not facilely translatable into features.

In the literature, many algorithms have been developed to discover communities. They can broadly be divided into three main categories **[2]**

Graph Theoretic methods, based on **graph structures** to model pairwise relations between objects from a certain accumulation like **random walk methods [3]** and **partition-based methods** like **spectral methods [4]**. They are among the exemplary strategies for community discovery. They assign nodes to communities based on **computed similarity** matrix values.

Divisive algorithms, variant of hierarchical clustering algorithms like '**Betweenness**' algorithms of **Girvan and Newman [5]**, **Tyler algorithm [6]** and **Radicchi algorithm [7]** in which they divide the network into smaller subsections . They start with the entire network as one community, and at each step, optate a certain community and split in to parts.

Agglomerative algorithms or **model based algorithms** like **Modularity-based algorithms [8]** which form communities by joining nodes together. They consider each node in the network as a community, and at each step merge communities that are reckoned to be sufficiently homogeneous, perpetuating until either the desired number of communities are found or the remaining communities are found to be too dissimilar to merge any further.

3.2 Existing Algorithms: A Comparative Study

There are several graph theoretic methodologies which are utilized for identifying the presence of subsidiary communities in a network. However most of the methodologies do not include the edge weights and directedness as fundamentals in detecting communities. Incorporation of edge weights as a fundamental is of great paramountcy as it is a quantitative measure which implicatively insinuates the strength of the connected nodes of the graph [9].

Spectral Clustering

Spectral clustering algorithms depend on the estimation of eigenvalues of a similarity matrix to discover ideal partitions, given a foreordained number of partitions. It unwinds the complex issue of minimizing cut ratio over every possible k-partitions to discover the k-most minuscule eigenvalues and cognate eigenvectors of the Laplacian graph.

Normalized spectral clustering according to Ng, Jordan, and Weiss (2002) [10]

Input: Similarity matrix $\mathbf{S} \in \mathbf{R}^{n \times n}$, number **k** of clusters to construct.

• Formulate a similarity graph (based on the normalized weights). Let **W** be its weighted adjacency matrix.

• Compute the normalized **Laplacian** L_{sym}.

• Compute the first **k** eigenvectors $\mathbf{u_1, \ldots, u_k}$ of L_{sym}.

• Let $\mathbf{U} \in \mathbf{R}^{n \times k}$ be the matrix containing the vectors $\mathbf{u_1, \ldots, u_k}$ as columns.

• Form the matrix $\mathbf{T} \in \mathbf{R}^{n \times k}$ from **U** by normalizing the rows to norm 1, that is set $t_{ij} = u_{ij} / \left(\sum u^2_{ik} \right)^{1/2}$.

• For $\mathbf{i = 1, \ldots, n}$, let $y_i \in \mathbf{R}^k$ be the vector corresponding to the **i-th** row of **T**.

• Cluster the points $(y_i)_{i=1,\ldots,n}$ with the k-means algorithm into clusters $\mathbf{C_1, \ldots, C_k}$.

Output: Clusters **A1, . . . , Ak** with $\mathbf{Ai} = \{j \mid y_j \in C_i\}$.

Infomap

Infomap improves the map equation, which exploits the information-theoretic duality between the issue of compressing data, and the issue of detecting and extracting critical patterns or structures within those information [11].

It endeavors to find partitions that yield the minimum description length of an illimitable arbitrary walk on the network. It decides the community and network structure by investigating the flow of information, proxied through desultory walk calculations, among different groups of nodes.

The hierarchical map equation measures the per step average code length obligatory to describe an arbitrary walker's movements on a network, given a progressive system partition.

With this calculation, a genuinely good clustering of the network can be found in a brief while.

Girvan - Newman

It is a hierarchical deterioration process (i.e. divisive) where edges are removed in the diminishing order of their edge betweenness scores [12]. This is spurred by the fact that edges connecting distinct groups are more prone to be contained in multiple shortest paths just by the grounds that in many cases they are the only choice to go from one group to another. This technique yields good results but is very slow because of the computational complexity of edge betweenness calculations (the betweenness scores have to be re-ascertained after every edge removal).

Steps

1. The betweenness of all subsisting edges in the network is computed first.
2. The edge with the highest betweenness is abstracted.
3. The betweenness of all edges influenced by the abstraction is recalculated.
4. Steps 2 and 3 are reiterated until no edges remain.

Algorithm	Time Complexity (n - Number of vertices m - Number of edges)	Type
Spectral Clustering	O(md + nd log n + nd2) (Full edge weight matrix of G is reduced from n × n to d × n)	Weighted / Directed
Infomap	$O(n^2 \log(n))$	Weighted / Directed
Girvan–Newman	O(nm2)	Weighted / undirected

3.3 Community Discovery Algorithms: Performance Metrics

To begin with, we characterize the community 'goodness' metrics that formalize the intuition that 'good' communities are both compact and well-connected internally while being relatively well-disunited from the rest of the network [13].

Given a set of nodes **S**, we consider a function **f(S)** that describes the connectivity of nodes in **S**. **G(V, E)** denotes the graph with **n = |V|** nodes and **m = |E|** edges. n_S is the number of nodes in **S**, $n_S = |S|$.

Metrics based on Internal Connectivity:

Internal Density: It is defined as the number of edges - m_s in subset **S** divided by the total number of possible edges between the nodes. $f(S) = 2 * m_S/n_S * (n_S - 1)$

Triangle Participation Ratio (TPR): TPR is the best measure for density, cohesiveness and clustering in the list of goodness metrics. It is the fraction of nodes in **S** that belong to a triad.

$$f(S) = \frac{|\{u : u \in S, \{(v,w) : v,w \in S, (u,v) \in E, (u,w) \in E, (v,w) \in E\} \neq \varnothing\}|}{n_S}$$

Metrics based on External Connectivity:

Cut Ratio: It is the portion of existing edges (out of every single conceivable edge) leaving the group - $f(S) = c_S/n_S (n - n_S)$

Metrics based on Internal/External Connectivity:

Normalized Cut:

Depicts how well the subset **S** is separated from graph **G**. It sums up the fraction of external edges over other edges in the subset **S** (i.e. conductance) with the fraction of external edges over all non-community edges.

$$f(S) = \frac{c_S}{2m_S + c_S} + \frac{c_S}{2(m - m_S) + c_S}$$

Metrics based on the network model:

Modularity:

$f(S) = 1/4 (m_S - E(m_S))$ is the distinguishment between m_S, the quantity of edges between nodes in **S** and $E(m_S)$, the expected number of such edges in a desultory graph with identical degree sequence.

4 Challenges Faced

Monitoring the script runs: Scripts failed multiple times while cleansing due to inconsistent data.

Identifying geospatial subsets: It wasn't an easier task to map places with larger countries. For instance, consider USA, the location attribute was not consistent. The employee's location was set to his area of employment (**Eg.** Greater Boston Area, Greater Seattle Area) instead of his country (USA). The mapping of areas to countries had to be done manually.

Grouping Nodes: It was arduous to find identical organizations with diverse names/labels and group them as the same association. **For eg.** facebook, facebook Inc. and www.facebook.com - all come under the same label facebook.

Model pairwise relationships: Construction of graph was difficult for individuals with concurrent positions/ designations. Special edge cases were added to the scripts to handle them separately.

Dataset	LinkedIn dataset scraped from public profile ~15 GB
Libraries	igraph 0.7.0
Languages	Python 2.7 R 3.1.3
Tools	Cytoscape 3.2.1 Gephi 0.8.2 NodeXL Tableau
RDBMS	SQLite 3.8.8.3

5.1 Evaluation and Results

We ran the algorithms - Spectral Clustering, Infomap and Girvan - Newman were run against the US - IT graph consisting of 7229 nodes and 26555 edges. The algorithms were run for two different edge weights i.e. number of switches and mean duration in a role held by individuals. **Table 1** shows the comparison results.

Table - 1

Algorithm - Weight	Internal Density	TPR	Cut Ratio	Normalized Cut	Modularity	Number of Communities	Average size of Communities
Infomap - Switches	***0.85***	***0.068***	***0.0012***	***0.637***	***0.67***	***476***	***15.187***
Spectral - Switches	0.75	0.026	0.00038	0.29	0.587	512	14.11
Girvan - Newman - Switches	0.71	0.04	0.00015	0.4	0.564	556	13.001
Infomap - Duration	0.62	0.073	0.00011	0.15	0.442	645	11.2
Spectral - Duration	***0.64***	***0.09***	***0.00028***	***0.18***	***0.48***	***609***	***11.87***
Girvan - Newman - Duration	0.56	0.066	0.00004	0.08	0.41	745	9.703

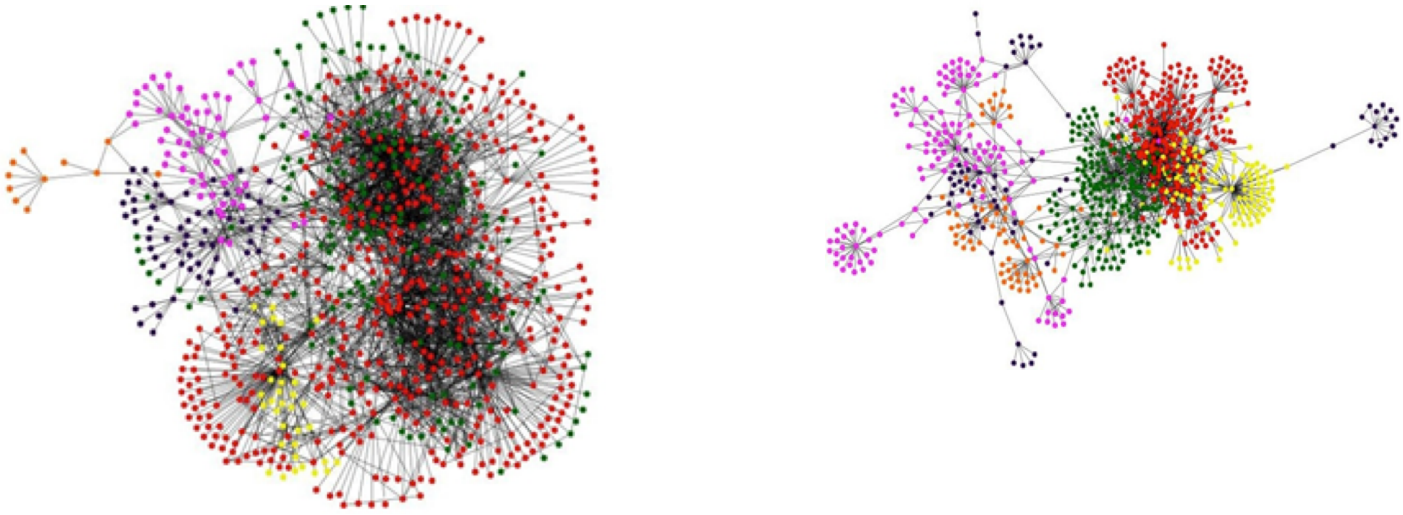
The high scores of these metrics indicate the better quality of the formed communities. High **modularity** score denotes that the composed communities have dense or close-knit associations between the nodes within community but meager associations between nodes in other distinctive communities. Maximal value of **internal density** connotes higher number of connections per node in a community. It in turn portrays the pre-eminence structure of the composed communities. Higher the scores of **normalized cut** and **cut ratio**, relatively lower the number of connections in the boundary of the community than the number of connections with in the community. Incremented **TPR** scores denote that the discovered communities are predominant or profoundly compelling in nature.

Predicated on the obtained results, we can reason that **Infomap** and **Spectral clustering** algorithms fit well for the professional networks with edge weights - number of switches and average duration individually.

Table 2 depicts the distribution of communities for United States/Singapore – IT data set with reverence to the number of switches and average duration.

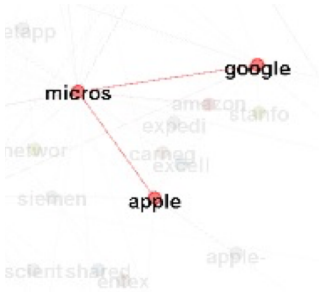
Table - 2

Algorithm - Weight	Data set	Number of Communities	Average size of Communities
Infomap - Switches	US - IT Nodes -7229 Edges - 26555	476	15.187
Spectral - Duration		609	11.87
Infomap - Switches	SG - IT Nodes - 6672 Edges - 12287	342	19.5
Spectral - Duration		346	19.28



Identified communities using Infomap and Spectral Clustering algorithms respectively (Partial Singapore - IT subset)

Success Measure:



The success of our analysis is defined by the communities formed using our best fitted algorithm. The communities consist of the companies that are part of the same industry, for instance the **community 1** in US data set consists of 74% the companies from **telecommunication industry** and like wise for the **Singapore data set** the **community 1** consists of 82% companies from the **financial services** industry. Along these lines when we analyzed the movement of employees within a community, it was not surprising to see that employees in a particular company tend to move to other companies within the same community. Hence this helped us to answer the case and recommend the shortest path for an employee to reach the desired company of her choice. An example of such a use case is as follows:

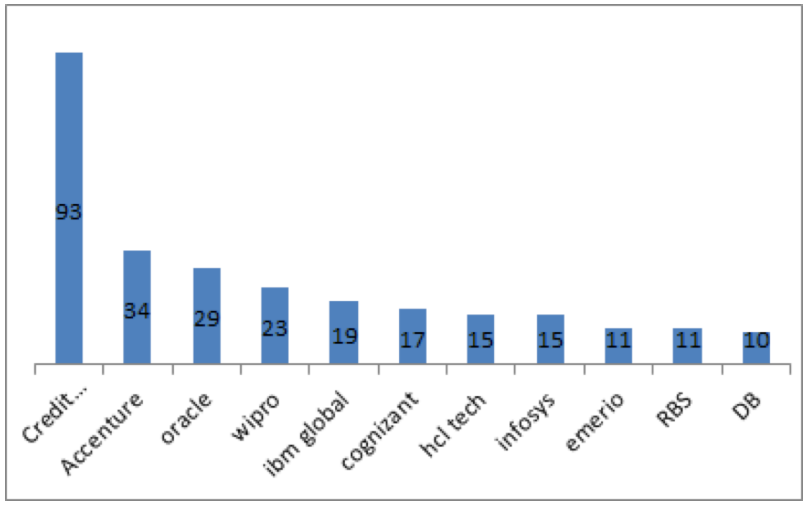
5.2 Community Analysis

Based on the best suited algorithm (**Infomap - Switches and Spectral - Duration**) we took top two communities from both Singapore and US data sets. We define top 2 with reverence to the number of companies in a particular community. Following is the analysis for each of the community.

Singapore Data Set:

Community 1 using number of switches as weight:

The top community of Singapore is made up of Financial service companies with **Citi bank** and **Credit Suisse** having the highest In degree and **Accenture** with the highest out degree. The maximum betweenness centrality is for **Accenture**, followed by **Cognizant**, **Citibank** and **Barclays**.



Community 1 using number of years of experience as weight:

When we analyzed the community using number of years of experience as weight, for people with less than **10 years** of experience we found out that **Credit Suisse** is the most desired company followed by **Accenture** and **Oracle**.

US Data Set:

Community 1 using number of switches as weight:

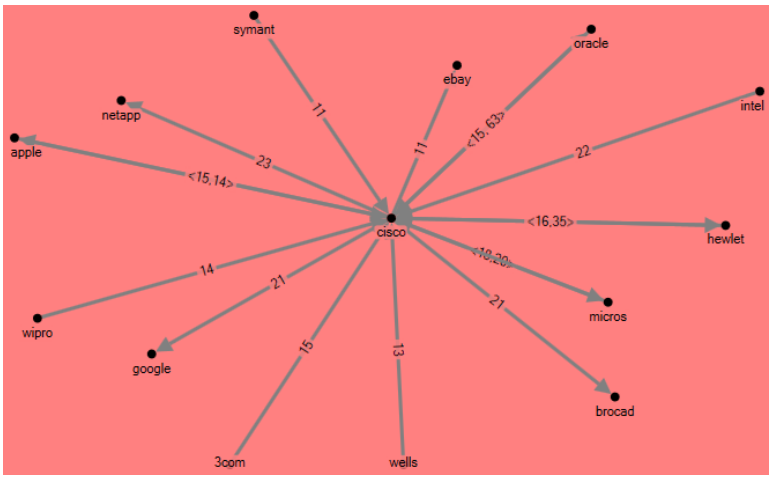
This community consists of **internet**, **service** and **product** based technology companies dominated by **Microsoft** in terms of **degree centrality**. After **Microsoft** the company with **highest centrality** is **Cisco** followed by **Google**, **Oracle** and **HP** each with a centrality of 12.

The top 5 companies in terms of betweenness centrality are as follows:

Cisco: 717 Microsoft: 613 HP : 439 Oracle: 403 Infosys: 356

The maximum movement of employees from **Microsoft** to **Amazon** (81) and **Microsoft to Google** (75) which are the third and fourth highest movements in the community respectively.

Deep dive into Cisco:



The above chart shows the movement of of employees in **Cisco**. This is a directed graph with nodes representing the companies and edges denoting the number of people moving from one firm to another. The double number on certain edges for example **<15,14>** on the edge between **Apple** and **Cisco** show that 15 people have moved from Cisco to Apple and 14 people have moved from Apple to Cisco.

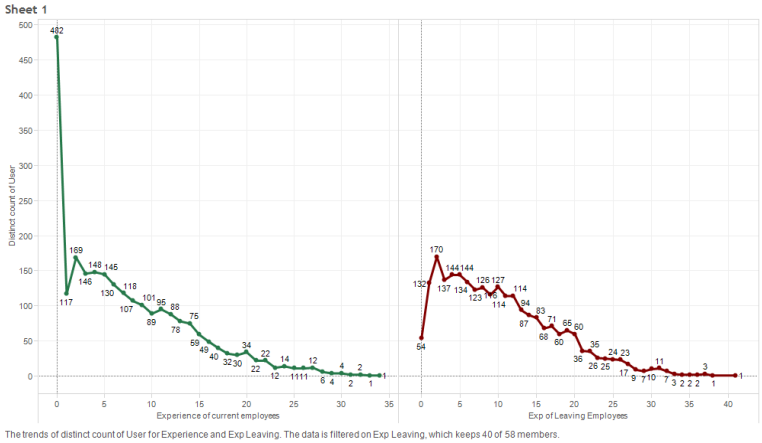
The above analysis answers the following business cases:

1. **Symantec, Wells Fargo, Wipro, Apple, 3COM, Microsoft, Intel, Oracle** and **HP** are the targeted firms from where most of the people move to Cisco, hence this helps the head hunters to target people from these companies.

2. **Ebay, Apple, Microsoft, Google, Brocade Communications, NetApp, Oracle** and **HP** are the most likable places to work for Cisco employees as most of the movement of employees from Cisco are to these firms.

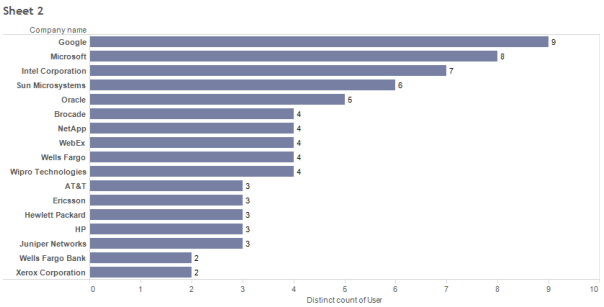
Now let's, analyze the movement of people across all communities in and out of Cisco.

From the above chart we can see that that people with **2 years** of experience have the highest attrition rate from Cisco.

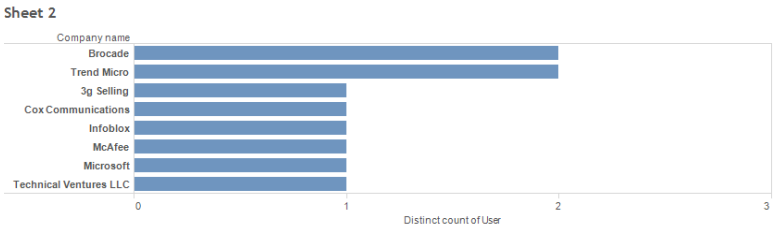


The trends of distinct count of User for Experience and Exp Leaving. The data is filtered on Exp Leaving, which keeps 40 of 58 members.

Among the people who have less than **5 years** of experience we can see that most of the people have moved to **Google** followed by **Microsoft**, **Intel** and **Sun Microsystem**. And most of the people have moved at the position of **Software engineer** which is quite evident from the chart below.



Distinct count of User for each Company name. The data is filtered on Experience, which keeps 27 of 58 members. The view is filtered on Company name, which keeps 1,087 of 1,489 members.

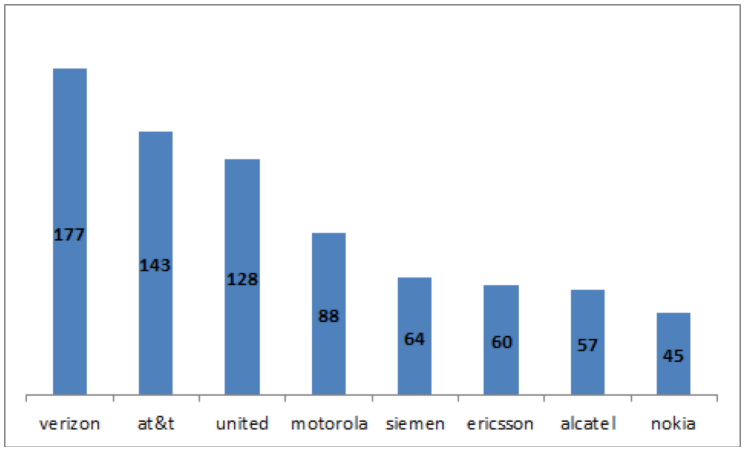


Distinct count of User for each Company name. The data is filtered on Experience, which keeps 30, 31, 32, 33 and 34. The view is filtered on Company name, which keeps 1,088 of 1,489 members.

While the above analysis is for people with less work experience, the movement of people with more than **30 years** of experience is on the chart above.

Community 2 using work experience as weight:

When we analyzed the community using the duration factor as weight, we found out that for individuals having experience less than **10 years**, **Verizon** is the most desired company. Also the top 10 most desired companies with both the weight measures turns out to be same.



6 References

[1] Hemant Balakrishnan, Narsingh Deo: Discovering communities in complex networks. ACM Southeast Regional Conference 2006: 280-285.

[2] Bo Yang, W.K. Cheung, and Jiming Liu, “Community Mining from Signed Social Networks”, IEEE / KDE, 19, No. 10, 2007.

[3] K. Steinhaeuser, N.V.Chawla, “Identifying and evaluating community structure in complex networks” Pattern Recognition Lett. (2009), doi:10.1016/j.patrec.2009.11.001

[4] Statistics and Computing, Vol. 17, No. 4. (1 December 2007), pp. 395-416, doi:10.1007/s11222-007-9033-z

[5] G. W. Flake, S. Lawrence, C. Lee Giles, “Efficient Identification of Web Communities”, ACM SIGKDD, 2000.

[6] M. Girvan and M.E.J. Newman, “Community Structure in Social and Biological Networks”, PNAS, 99, No. 12, pp. 7821-7826, 2002.

[7] M. E. J. Newman, “The Structure and Function of Complex Networks”, SIAM Review, 45, 167–256, 2003.

[8] Andreas Noack, “An Energy Model for Visual Graph Clustering”, GD 2003, Pages 425.436, Springer-Verlag, 2004.

[9] JOUR Trubin, V. A., "Strength and reinforcement of a network and tree packing", Cybernetics and Systems Analysis, 1991-03-01, Springer New York, 1060-0396, Mathematics and Statistics, http://dx.doi.org/10.1007/BF01068376.

[10] Ulrike von Luxburg, A Tutorial On Spectral Clustering.

[11] Martin Rosvall and Carl T. Bergstrom http://www.mapequation.org/assets/publications/RosvallBergstromPNAS2008Full.pdf

[12] M E J Newman, and M. Girvan, 2004, http://arxiv.org/pdf/condmat/0308217.pdfFinding and evaluating community structure in networks

[13] http://cs.stanford.edu/people/jure/pubs/comscore-icdm12.pdf

[14] http://stn.spotfire.com/spotfire_client_help/norm/norm_scale_between_0_and_1.htm