Combining Machine Learning with Shortest Path Methods

Discovering, Visualizing, and Analyzing Hollywood's Power Clusters to Go From Six Degrees of Kevin Bacon to Knowing Colin Firth

Armand Prieditis
Neustar R&D
San Francisco, CA USA
email: armand.prieditis@neustar.biz

Chris Lee
Neustar Labs
Mountain View, CA
email: chris.lee@neustar.biz

***Abstract*—This paper describes a method to model, discover, and visualize communities in social networks. It makes use of a novel method based on the "Six Degrees of Kevin Bacon" principle: find the shortest path between entities in a social graph and then discover communities based on clustering with those shortest-path distances. We have applied this idea to find Hollywood's power clusters based on IMDB (Internet Movie Database), which links actors to movies. Using this method, we found roughly three clusters of Hollywood elite actors, the largest of which contained many of Hollywood's best-known actors. For living actors, we found Colin Firth (who played *Pride and Prejudice*'s Mr. Darcy), Javier Bardem (who played a psychopathic killer in *No Country for Old Men)*, and Joaquin Phoenix (who played Johnny Cash and a Roman Emperor in *Gladiator*) to be some of the most well-connected actors in Hollywood. This suggests that analyzing a social network using our method can lead to some surprising results**.

Keywords-Social networks; modeling; disovery; visualization; clustering; influence analysis; machine learning.

I. INTRODUCTION AND MOTIVATION

What is a **social network?** Typically, names such as Facebook, Twitter, or Google+ spring to mind when one thinks of a social network because that is the moniker these websites adopt. While these websites are not the only type of social networks, they are good examples of networks that are "social." This is because they comprise: a set of **entities** that participate in the network. In social networks such as Facebook, Twitter, and Google+, the entities are people. In general, entities do not have to be people. They also comprise a set of **relations** between the entities. For example, in Facebook, the relations are called "friends." In Twitter, they are follower and followee relationships. Finally, social networks comprise **weights** on those relations. For example, the higher the weight, the stronger the relationship. While most social networks such as Facebook, Twitter, and Google+ are all or none weights (i.e., you are either a friend or not; either a weight of a 1 or a 0), other social networks could have the degree of the relationship expressed as a weight. This degree might not be explicit. For example, how often someone reads the postings of someone they follow could be could be used to determine the weight.

Most real-world networks are not random and exhibit locality. That is, a randomly constructed network rarely looks like a real-world network and the intuition behind locality is that the relationships among entities tend to cluster somehow. For example, if X knows both Y and Z, then Y and Z probably know each other. One reason that Y and Z might know each other is that that they both comment on X's postings and hence they eventually discover and befriend each other based on those postings. Or, X could have introduced Y to Z other either online or in the real-world, based on X as a mutual friend.

This paper considers methods by which Y and Z could (or should) get to know each other by the modeling, discovery, and visualization of local communities that they share. More generally, this article touches on social influence in the sense of how a community influences the individuals in the community. Social influence is an active area of research because it aims to understand how information, memes, ideas, knowledge, experience, and innovation spread in a social network. Thus, analyzing and mining social networks can provide insights into how people interact and why certain ideas, memes, and opinions spread in the network and others do not. Although this paper describes a specific clustering method, it is not about clustering. That is, many different clustering methods could be used and we would expect comparable results. This paper is about how shortest path methods can improve upon clustering in social networks.

Discovery of communities can also be viewed as **link prediction**. Clearly, social networks are dynamic and constantly evolving and methods that can *anticipate* future links, such as link prediction, are important. As the network evolves, two unconnected nodes in the same community may eventually form a link between them. The intuition is that if future links can be predicted, the growth of a social network can be facilitated. Moreover, the relationships of the entities might be more satisfying from discovering other like-minded people faster. Thus, link prediction can be used to model how a social network evolves over time.

A. Social Networks are Ubiquitous

While social networks such as Facebook, Twitter, and Google+ capture the mindshare of the term "social network," social networks go beyond mere friend networks. In fact, the entities do not even have to be people to be

 ISBN: 978-1-61208-423-7

considered a social network. That is, a social network does not necessarily have to be in a social context. For example,

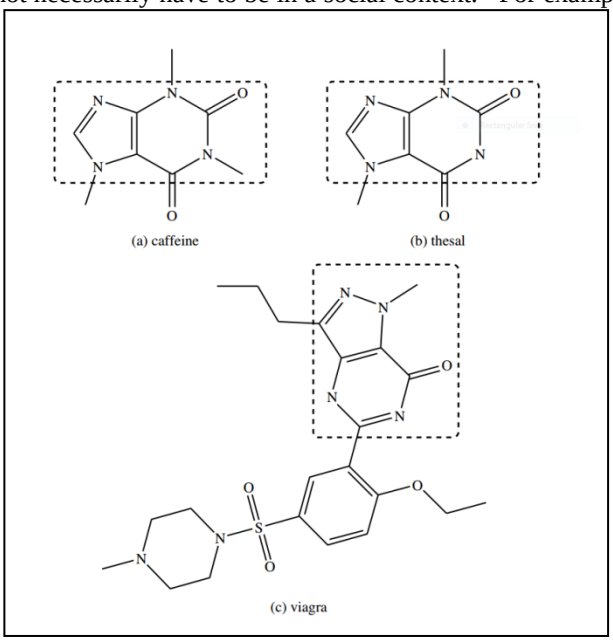

Figure 1. A "Social' Network in Chemistry

social networks *that* are non-social in context include electrical power grids, telephone call graphs, the spread of computer viruses, the World Wide Web, and co-authorship and citation networks of researchers. In a citation network, the entities might represent individuals who have published research papers and the relations between the entities might be researchers who jointly co-authored one or more papers. Weights might include the number of joint publications—the higher the weight, the more joint publications. The communities one might be able to discover in this network might include researchers working in the same area. Other such social networks might be possible to construct. For example, two Wikipedia editors can be related if they've edited the same article. Alternatively, the articles themselves can be the entities, which are linked if they have been edited by the same person.

More generally, social networks and their characteristics can often be generalized to networks found in a diversity of fields such as biology, chemistry economics, mathematics, and physics. For example, Figure 1 shows a social network in chemistry, where the entities are atoms and the relations are bonds. In the figure, (a) shows the caffeine molecule, (b) shows the thesal molecule, and (c) shows the Viagra molecule. All of these molecules are biologically and pharmaceutically important and hence their network analysis of activity is important.

Social networks can also include collaborative filters, where recommendations are based on customer preferences. Such networks can be viewed from the point of view of the customers as entities and the relations expressing customers who bought the same products. Such networks can also be viewed from the point of view of the products as the entities and the relations expressing products that were bought by the same customer.

Determining the entity vs. the relation can get complicated. For example, **users** can place **tags** on **websites** on social tagging sites, such as deli.cio.us. Users can be connected to other users based on tags they place on the same website. Alternatively, users can be connected to other users based on the *type* of tags they use. Both of these, can, of course, be flipped: websites can be connected based on the same users; tags can be connected based on the same users or websites.

Biological networks include epidemiological models, cellular and metabolic networks, food webs, and neuronal connections. The exchange of email or communication messages can also form social networks within corporations, newsgroups, chat rooms, friendships, dating sites, and corporate control (i.e., who serves on what boards). The entities in an email network represent individuals and a relation between entities can include an email exchange in any direction between two individuals. A weight might mean the number of emails between two individuals in a given period. This view distinguishes normal emailers from spammers: a normal emailer has higher frequency communication with a small set of individuals whereas a spammer has low frequency communication with a large set of individuals.

In a telephone network, the nodes might represent the phone numbers and relations might include two phones that have been connected over some period of time. Weights might include the number of calls.

Thus, many different networks bear similarities in terms of how social networks can be explicitly or implicitly derived from them: for paper citation networks entities might be papers or people and a relation exists if one paper cites another or the same paper was co-authored by two people. For collaboration **networks** entities might be people

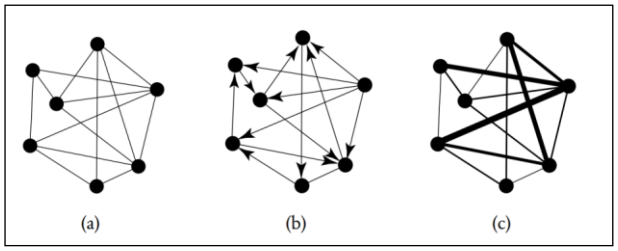

Figure 2. Social Networks as Graphs: Undirected (a), Directed (b), and Weighted (c)

and a relation expresses one person working with another. For semantic word graphs, such as in a dictionary or a thesaurus, entities might be words and a relation exists between two words if they are associated with each other. For biological networks, entities might be processes and a relation exists if two processes are related (e.g., protein or drug interactions). For news networks entities might be events, people, or words and relations might be causal links or people in common.

 ISBN: 978-1-61208-423-7

B. *Social Networks as Graphs*

A reasonable way to model a social network is as an undirected or directed or undirected graph. In an undirected graph, the entities are modeled as nodes and the relations are modeled as edges. The weight is represented by a labeled edge. Typically, the relations require a directed graph because the distinction between a follower and a followee is important. That is, if X follows Y then there is a directed edge between X and Y, but not necessarily vice versa. Informally, one can say that X "points to" Y. Note that this relationship could have been modelled the other way, with Y pointing to X, but the in-pointers to a node are typically more important than the out-pointers. That is, the people who follow you are a stronger sign of a relationship than the people who you follow because you have control over who you follow but not vice versa. For example, one could follow Lady Gaga, but that means little to most people. But if Lady Gaga follows you, that means a lot. In short, a directed graph can capture relationships that are one way, but not the other. Social relationships modeled as directed graphs are common in the real-world, so common that phrases such as "unrequited love" have been invented in order to capture them.

Figure 2 illustrates undirected, directed, and weighted graphs. For example, (a) shows an undirected graph. The nodes are the dark circles and the undirected edges are the lines. Graph (b) illustrates a directed graph. The nodes are the same, except the lines are now directed. Graph (c) illustrates an undirected weighted graph, where the thickness of the edge is proportional to the weight of the edge.

C. *Discovering Communities in a Social Network*

Figure 3 shows a social network represented as a graph with nodes A, B, C, D, E, F, and G and undirected edges as the relations between nodes. Visually, nodes A, B, and C seem more closely related to each other than to the other

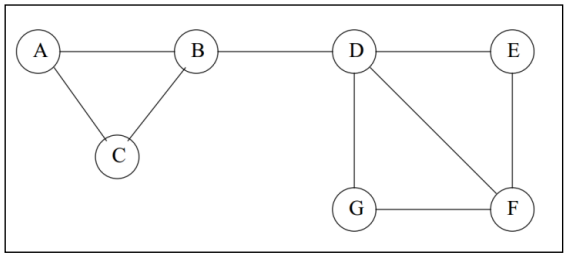

Figure 3. Visual Discovery of Communities by Distance

nodes. Similarly, nodes D, E, F, and G seem more closely related to each other than to the other nodes. Thus, one way in which the nodes can be clustered or groups is in terms of distance to each other. More specifically, the act of clustering can be viewed as discovering a **community** in a social network. The intuition is that nodes A, B, and C might have something in common, at least more in common than with nodes D, E, F, and G. In short, one way to discover communities is to group by distance in terms of relations. An important aspect of a social network is that it can be implicit, by virtue of liking the same things, visiting the same sites, or having similar attributes. One important task is to discover *homophily*, which can be viewed as discovering communities in a social network.

Another way to discover communities is to form groups based on common attributes. For example, Figure 4 shows a graph coloring based on interest in music, sports, and cooking. In this case, the nodes A, B, C, and D form one cluster, nodes C, H, I, and J form another cluster, and nodes D, E, F, G form a third cluster. Note that in this case, the nodes might have been grouped similarly by their relations instead of their attributes. Thus, it might be likely that nodes sharing relations are interested in some of the same things (i.e., have the same attributes). Otherwise, such nodes would have little basis for interacting with each other.

Although the concerns are different in different fields, the idea of community discovery can be treated similarly, as described here. Indeed, one way that complex networks and complex systems can be understood is by discovering structures in the form of communities in them. Human cognition often prevents analyzing the network as a whole; finding communities is a way to simplify a network into a small set of communities, so that human cognition can then take over. In short, this paper recognizes that modeling,

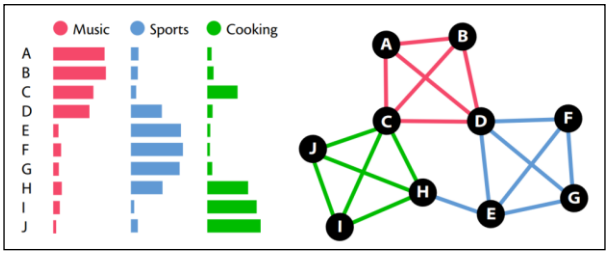

Figure 4. Communities Discovered by Common Interests

discovery, and visualization of communities in networks is a general methodology applicable to most real-world networks. It also recognizes that finding an appropriate division of labor between humans and machines is important to combine the unique cognitive strengths of humans with the tireless computational abilities of machines.

D. *Modeling, discovering, and visualizing communities in Hollywood actor networks*

We wanted to test our ideas for modeling, discovering, and visualizing communities on a large enough data set to produce interesting results. For this reason, we choose the IMDB (Internet Movie Database) [1], which is publicly available. The database contains hundreds of thousands of movies, many of which are obscure, and thousands of actors, most of whom are obscure bit players. This database can be viewed as a bipartite graph where each node either corresponds to an actor or to a movie. In this graph, an edge between an actor and a movie means that the actor appeared

 ISBN: 978-1-61208-423-7

in the movie. The task is to model this data somehow to make it easy to discover and visualize communities.

E. Organization of the Rest of this Paper

The rest of this paper is organized as follows. Section II describes related work. Section III describes our approach to modeling, discovering, and visualizing communities in a social network. Section IV summarizes our results. Finally, Section V presents our conclusions and several promising areas for research.

II. RELATED WORK

Discovering communities in a social network can be viewed as clustering. As such, researchers have used two general approaches to clustering: hierarchical or agglomerative [2] and divisive [3]. Both approaches require a distance metric. When the edges have weights, those weights can be used as a distance measure. The difficulty arises when the edges are unlabeled, as in most online social networks: the "friends" network. It's possible to use a weight of 1 or 0 for a direct edge and a large weight for those without a direct "friends" relationship, but these measures violate the triangle inequality principle of a distance metric, which generally causes anomalies in clustering.

Assuming a suitable distance measure can be found, researchers have defined the distance between clusters as the minimum distance between two nodes of each cluster. Hierarchical clustering first combines two nodes connected by an edge. It then chooses at random edges that are not between the two nodes in the cluster to combine the clusters to which each of the two nodes below. This agglomeration continues until an appropriate criterion is met. Divisive clustering proceeds in the opposite direction: starting with one giant cluster, it successively seeks edges that break the cluster into smaller and smaller parts.

These standard clustering methods have produced somewhat unsatisfactory results in social networks. As a result, researchers have developed specialized clustering methods aimed specifically at finding communities in social networks. One method, a divisive one, is based on finding an edge that is least likely to be in cluster and then removing it. This method uses the Girvan-Newman (GN) algorithm [4] to calculate the number of shortest paths running between every pair of nodes. An edge with a high GN score is a candidate for removal. The GN algorithm essentially conducts a breadth-first search of the graph and counts the number of times the same edge is encountered for all pairs of nodes.

Thus, by using the GN-based score, this specialized clustering method removes edges, which has the effect of decomposing the graph into subcomponents. The process beings with the initial graph and then each time it removes that edge with the highest GN score until the graph is decomposed into an appropriate number of connected components.

Another approach uses matrix theory (i.e., spectral methods as in [5]) to partition a graph such that the number of edges that connect different components is minimized. But such "cut-based" methods are unstable because cuts are not desirable that break the two components into unequal size.

In general, the approaches to finding communities in social networks have been somewhat unsatisfactory, often relying on arbitrary distance measures.

Social network analysis is an active area of research and this paper can be considered part of that work. For example, a Google search reveals nearly three hundred conferences on or related to social network analysis. A recent book [6] describes some of the network relational structures described here. Moreover, the Web Science conferences have been publishing leading work in social network analysis since 2009. Related work in these conferences includes research on six degrees of separation in social networks [7], clustering users on social discussion forums based on roles [8], topic-author networks [9], influence detection in networks [10], status evaluation [11], four (not six) degrees of separation [12], the spread of misinformation in a social network [13], and social graph annotation based on activities [14]. All of these results are consistent with the results presented here. For example, we found, just as in [12], that much less than six degrees separate most actors.

III. OUR APPROACH TO MODELING, DISCOVERING, AND VISUALIZI'G COMMUNITIES IN SOCIAL NETWORKS

We began our process with the IMDB [1], which links actors to movies. Next, we converted this bipartite graph into a social network where actors are the entities and a relation between one actor and another means that the two actors have appeared in the same movie.

Even though we built our graph with the entire IMDB, we focused on the top 100 actors of all time (based on IMDB, 62 of which were all in the same connected component, which we focused on for computational efficiency and presentation brevity): Jack Nicholson, Marlon Brando, Al Pacino, Daniel Day-Lewis, Dustin Hoffman, Tom Hanks, Anthony Hopkins, Denzel Washington, Spencer Tracy, Laurence Olivier, Jack Lemmon, Gene Hackman, Sean Penn, Johnny Depp, Jeff Bridges, Gregory Peck, Ben Kingsley, Leonardo DiCaprio, Tommy Lee Jones, Alec Guinness, Kevin Spacey, Javier Bardem, Humphrey Bogart, Clark Gable, George C. Scott, Jason Robards, Peter Finch, Charles Chaplin, James Cagney, Burt Lancaster, Cary Grant, Sidney Poitier, Alan Arkin, Samuel L. Jackson, Sean Connery, Christopher Walken, Heath Ledger, Jamie Foxx, Colin Firth, Joaquin Phoenix, Jeremy Irons, George Clooney, Tom Cruise, Matt Damon, John Hurt, Brad Pitt, Nicolas Cage, John Travolta, Clint Eastwood, Orson Welles, Charlton Heston, Henry

 ISBN: 978-1-61208-423-7

Fonda, Ian McKellen, Liam Neeson, Woody Allen, John Malkovich, Mickey Rourke, Danny DeVito, Robert Mitchum, Buster Keaton, Harvey Keitel, and Martin Sheen.

We also explored the top 250 and top 1000 actors (as ranked by IMDB) and obtained similar results. However, we found that the top 100 list adequately captured the core ideas well while making the results convenient for presentation here. Another reason we focused on this top 100 list of well-known actors is that when we ran our system on the largest set of actors (i.e., the entire set of actors in the IMDB movie database) we found that the cluster centers were comprised of these relatively unknown actors: Stéphanie Sokolinski, Olivier Rittano, Magid Bouali, David Luraschi, Simon Muterthies, David Vincent, Stéphanie Blanc, Anne Comte, Juliette Goudot, and Anne Nissile. Since no one among our associates could recognize even a single actor in these clusters, we felt that the interested reader would get a better feel for our system if the actors were "well-known" even though the clustering is on *all* the actors in the IMDB movie database. We simply ignore the less-known actors even though they are behind the scenes in the clustering. Note that the appearance of these less-known actors near the cluster centers does not mean that our approach does not work. It merely means these less-known actors happened to locate near the center because they greatly outnumber well-known actors. That is, because of their large numbers, a less-known actor is more likely to appear near a center than a well-known actor. Indeed, watching the credits roll by at the end of any typical modern movie confirms that only a few actors in that roll are well-known.

Next, we added an edge between each actor in the same movie. For example, Danny DeVito and Jack Nicholson were in *One Flew Over the Cuckoo's Nest* and hence they are connected with a single link. Thus, the initial graph we built contains only *direct* social relationships between actors. In our social network the entities are the actors and the relations that link them are joint appearance in a movie, but we could have just as easily built a social network where the entities are the movies and the relations that link them are joint appearance of actors in both movies. We choose the former because we were more interested in finding out the "Hollywood power clusters." That is, we were interested in discovering which well-known actors would turn out to be at the center of the largest clusters. We were also interested in finding out which actors were central to multiple clusters—which actors act as articulators in multiple clusters. Note that this type of analysis is unrelated to clustering, but is a post-clustering analysis.

As a result, we wanted to link *all* the actors together somehow. The problem, as with the distance measures that we mentioned, is that actors either have a link (i.e., a weight of 1) or they do not (i.e., a weight of 0). A high weight, as assigned in the previously mentioned research, is clearly unacceptable because there could just be a few actors separating any two actors. For example, the game of "Six Degrees of Kevin Bacon," assumes that any actor can be linked through his or her film roles to Kevin Bacon within six steps. (Sadly, Kevin Bacon does not make an appearance in this paper even though the title mentions his name. This is because he is not a member of the Hollywood power clusters we found.)

To combat what we call the "binary problem" of edges (i.e., either a 1 or nothing), we ran a shortest-path algorithm between all pairs of actors in all connected components, one such algorithm per connected component. We then focused on the largest connected component, which contained roughly 5000 actors. Here is an example of the edge weights between a few selected pairs of actors, emanating from Buster Keaton, silent movie star of the 1920's: Humphrey Bogart:1, Daniel Day-Lewis: 2, Matt Damon: 2, Javier Bardem: 2, Jamie Foxx: 2, Joaquin Phoenix:2, Henry Fonda:2, and Johnny Depp:2.

This example illustrates Buster Keaton's connection to both modern and old-time movie stars. For example, he's directly connected to Humphrey Bogart (having starred in the same film), but is only two connections away from Matt Damon. That is, he starred in a film with someone who starred in a film with Matt Damon, a modern movie star. We were not surprised that the "Six Degrees of Kevin Bacon" holds true, but we were surprised at how few steps away an actor of the 1920's was from actors of the new millennium, over 80 years later. Going the other way, from recent actors to old-time actors, we see that Jamie Foxx is similarly connected to both old and new actors: Humphrey Bogart: 2, Daniel Day-Lewis: 2, Matt Damon: 2, Javier Bardem: 1, Johnny Depp: 1, and Charles Chaplin: 2.

We would not have guessed that Jamie Foxx, who recently appeared in Tarantino's *Django Unchained*, is a mere two steps away from Charles Chaplin, silent movie star of the 1920's. Conducting a shortest-path analysis reveals such connections between actors. Thus, the motivation behind the shortest-path analysis is to compute *indirect* relationships, which we believe are as important as direct relationships in clustering and in discovering communities.

Next, we applied a clustering algorithm to find how the actors clustered based on these shortest-path distances in the largest connected component of actor relations. We choose **K-Means** clustering as the method to cluster the actors. K-Means clustering partitions the data points into K clusters such that each data point belongs to the cluster with the nearest mean [15]. Thus, each cluster's mean serves as a summary of the data points in the cluster. The resulting partition can be viewed as a set of Voronoi cells. We used Lloyd's algorithm to find the K means [16]. This algorithm begins with an initial random set of K means. Next, it assigns each data point to the nearest mean of the K means. It then recalculates the K means for each cluster and repeats the assignment. This continues until the assignments no longer change. Although there is no guarantee that a globally optimum set of assignments can be obtained (i.e.,

 ISBN: 978-1-61208-423-7

those that minimize the sum of a least squares fit between the data points and their closest clusters), multiple random restarts can increase the confidence that a globally optimum set of assignments can be founds. To start with good initial parameters, we used the K means ++ assignment algorithm [17], which is an effective way to ensure faster convergence by choosing better initial values. We choose the K-Means clustering method both because of its simplicity and because of its ability to deal with numerical values through a straightforward distance measure, which is consistent with the distance measure in our application.

IV. SUMMARY OF RESULTS

Using the standard estimate of the mean-squared error over all the data points, we obtained the following results for K-means clustering: K = 5: 32790, K = 25: 24957, K = 50: 21781. After K = 50, the train-and-test error rate began climb, so we stopped with K=50 and used that as the baseline K for all the results described here.

The largest cluster contained the following actors: Jack Nicholson, Marlon Brando, Al Pacino, Daniel Day-Lewis, Dustin Hoffman, Tom Hanks, Anthony Hopkins, Denzel Washington, Laurence Olivier, Jack Lemmon, Gene Hackman, Johnny Depp, Jeff Bridges, Gregory Peck, Ben Kingsley, Leonardo DiCaprio, Tommy Lee Jones, Alec Guinness, Kevin Spacey, George C. Scott, Jason Robards, James Cagney, Burt Lancaster, Cary Grant, Sidney Poitier, Samuel L. Jackson, Sean Connery, Christopher Walken, Heath Ledger, Colin Firth, Jeremy Irons, Tom Cruise, John Hurt, Brad Pitt, Nicolas Cage, John Travolta, Clint Eastwood, Orson Welles, Charlton Heston, Henry Fonda, Ian McKellen, Liam Neeson, Woody Allen, John Malkovich, Mickey Rourke, Danny DeVito, Robert Mitchum, Buster Keaton, Harvey Keitel, and Martin Sheen. Based on our cluster analysis, this largest cluster can be viewed as Hollywood's true power brokers in terms of their connections. In other words, cluster analysis shows this to be the true "A-list" of actors. Actors on this list tend to be tightly connected to each other.

The next largest cluster contained Spencer Tracy, Humphrey Bogart, Clark Gable, Peter Finch, Charles Chaplin, Jamie Foxx, and Joaquin Phoenix. These are also well-known power-brokers, but nothing like the first list. Finally, the next largest cluster contained the remaining actors: Sean Penn, Javier Bardem, Alan Arkin, George Clooney, and Matt Damon. The rest of the clusters (i.e., the other 47) contain nearly all unknown actors and hence we will not discuss them here. This suggests that it is difficult to break into the top three Hollywood power clusters.

Next, we "softened" the notion of cluster membership in K-means and found the list of the 10 closest actors to each cluster's center. Membership is "soft" because these actors might not necessarily be in the cluster. Names such as Jamie Foxx, Javier Bardem, and Spencer Tracy appear on many of clusters. Subsequently, we counted the number of times each actor appeared in the top 10 closest actors in each cluster and obtained the following results: Peter Finch (50), Spencer Tracy (49), Colin Firth (49), Charles Chaplain (48), Javier Bardem (48), Heath Ledger (48), and Joaquin Phoenix (48). After a big gap, Matt Damon comes in at 30. The rest of the actors do not appear in as many clusters as these. Among dead actors, Peter Finch, Spencer Tracy, and Heath Ledger would have been the ones to get to know to make Hollywood connections. Among living actors, Colin Firth, Javier Bardem, and Joaquin Phoenix appear to be the go-to guys to make connections. These actors can be viewed as major "articulators" who are well-connected to nearly everyone. Intuitively, this means that if you get to know these actors they might help you unlock the doors to the most power clusters in Hollywood. Colin Firth was a surprise to us. But then, upon closer examination, we found out that Colin Firth's films have earned more than $936 million and that he's had over 42 movie releases worldwide. Based on our analysis, our advice to a young actor interested in Hollywood social climbing is to get to know Colin Firth.

For the largest cluster (the "Jack Nicholson" one), Figure 5 shows a visualization of the ten closest actors in that cluster and their distance apart. We used the NetworkX Python facility [18] to produce a planar graph, given the inter-node distances. What is interesting about this visualization is that James Cagney appears to be the prototypical actor in this largest cluster. That is, he is most like the average member of this cluster than anyone else. For the next largest cluster (the "Spencer Tracy" one), Figure 6 shows a visualization of the ten closest actors in that cluster and their distance apart. Spencer Tracy sits comfortably in the middle of this cluster, even though he died over half-century ago. This visualization vividly demonstrates the temporal reach of good actors: they can die, but they never really leave Hollywood. For the third-largest cluster, Figure 7 shows that Colin Firth, who we have already said is worth getting to know for social climbing, is at the center of this web of actors.

V. CONCLUSIONS AND FUTURE WORK

Evaluating the quality of these clusters is difficult as there is no standard grouping of actors against which we can compare our results. It may be possible to borrow evaluation ideas from the research focused on "power" users in social networks [19], but this work lacks a clustering component.

Short of such an evaluation, these results can be viewed as the discovery of power communities among Hollywood actors. We believe that the process of **Modeling**, **Discovery**, and **Visualization of Communities,** as we have presented it, is a powerful way to analyze social networks. **Modeling** comprises **Choosing Entities and Relations → Building a Social Graph Based on Relations → Calculating an All-Pairs Shortest-Path Metric**. **Discovery** comprises **Finding the Parameters of a Piece-wise Linear Function** (i.e., this is what K-Means clustering discovers). **Visualization** comprises **Laying out the Nodes and Their Relations In the Discovered Communities on a Planar**

Graph such that the layout preserves the distance metric between nodes.

We believe this process is an appropriate division of labor between machines, which are good at mind-numbing calculations, and humans, who are good at detecting visual patterns. It is difficult to perceive visual patterns in a large multi-dimensional space such as that produced after the all-pairs shortest-path metric is calculated. However, once the discovery process is completed and the resulting communities are displayed on a planar graph, the human visual system, with all its virtues, can take over and unlock patterns difficult for machines to see. Without this discovery, these patterns are nearly impossible to visually unlock. We believe this type of discovery and analysis might be important in determining how to spend advertising dollars on the Internet: find those nodes that are most influential and spend the most money there. The scientific contribution of this paper is a way to combine shortest-path methods with clustering to yield better results.

Based on our results, our conclusion is that when it comes to well-known actors, there are only three Hollywood power clusters, with one cluster dominating the other two in terms of size. Some actors are more well-connected than others, namely Colin Firth, Javier Bardem, and Joaquin Phoenix.

Could similar results be expected for other types of networks? We have applied the same idea to geo-locating the world's routers [20]. This work builds a map of directly connected internet routers based on time delays between routers (as returned by the trace command), calculates the shortest path time-delay between all pairs of routers based on this map, and then clusters the results. In this application, the time delay is analogous to the degree of separation between actors.

We are currently investigating several promising directions for future work including a more sophisticated clustering algorithm (e.g., EM), adding attributes for additional clustering knowledge (e.g., when and where each actor was born and the types of roles for which they are known), and applying our idea to predicting the geographic location of the world's routers (the entities) based on the round-trip transit time between the routers (the relations).

Working with large social networks can be computationally difficult. We believe our method can be extended to networks will millions of nodes by making use of frameworks, such as Hadoop and Spark, which we are currently investigating. An important advantage of our method is that every step in the process we have described can easily be parallelized to make it scalable.

REFERENCES

[1] J. J. Jung. (2012). "Attribute selection-based recommendation framework for short-head user group: An empirical study by MovieLens and IMDB." Expert Systems with Applications, 39(4), 4049-4054.

[2] K. C. Gowda and G. Krishna. (1978). "Agglomerative clustering using the concept of mutual nearest neighbourhood." Pattern Recognition, 10(2), 105-112.

[3] S. M. Savaresi, D. Bolev, S. Bittanti, and G. Gazzaniga. (2002, April). "Cluster Selection in Divisive Clustering Algorithms." In SDM.

[4] M. E. Newman and M. Girvan. (2004). "Finding and evaluating community structure in networks." Physical review E, 69(2), 026113.

[5] U. Von Luxburg. (2007). "A tutorial on spectral clustering." Statistics and computing, 17(4), 395-416.

[6] J. Scott. (2012). Social network analysis. Sage.

[7] L. Zhang and W. Tu. (2009) "Six Degrees of Separation in Online Society." In: Proceedings of the WebSci'09: Society On-Line, 18-20 March 2009, Athens, Greece.

[8] J. Chan, C. Hayes, and E. Daly. (2010) "Decomposing Discussion Forums using Common User Roles." In: Proceedings of the WebSci10: Extending the Frontiers of Society On-Line, April 26-27th, 2010, Raleigh, NC: US.

[9] N. Naveed, S. Sizov, S and S. Staab. (2011) "ATT: Analyzing Temporal Dynamics of Topics and Authors in Social Media." In: Proceedings of the ACM WebSci'11, June 14-17, Koblenz, Germany, pp. 1-7.

[10] Chandra, P., & Kalyanasundaram, A. (2012, June). "A network pruning based approach for subset-specific influential detection." In Proceedings of the 3rd Annual ACM Web Science Conference (pp. 57-66). ACM.

[11] B. State, B. Abrahao, and K. Cook. (2012, June). "From Power to Status in Online Exchange." In Proceedings of the 3rd Annual ACM Web Science Conference (pp. 57-66). ACM.

[12] L. Backstrom, P. Boldi, M. Rosa, J. Ugander, and S. Vigna. (2012, June). "Four degrees of separation." In Proceedings of the 4th Annual ACM Web Science Conference (pp. 33-42). ACM.

[13] N. P. Nguyen, G. Yan, M. T. Thai, and S. Eidenbenz. (2012, June). "Containment of misinformation spread in online social networks." In Proceedings of the 4th Annual ACM Web Science Conference (pp. 213-222). ACM.

[14] A. V. Sathanur and V. Jandhyala. (2014, June). "An activity-based information-theoretic annotation of social graphs." In Proceedings of the 2014 ACM conference on Web science (pp. 187-191). ACM.

[15] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. (2002). "An efficient k-means clustering algorithm: Analysis and implementation." Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24(7), 881-892.

[16] C. N. Vasconcelos, A. Sá, P. C. Carvalho, and M. Gattass. (2008). "Lloyd's algorithm on GPU." In Advances in Visual Computing (pp. 953-964). Springer Berlin Heidelberg.

[17] S. Agarwal, S. Yadav, and K. Singh. (2012, March). "K-means versus k-means++ clustering technique." In Engineering and Systems (SCES), 2012 Students Conference on (pp. 1-6). IEEE.

[18] A. Hagberg, P. Swart, and D. Chult. (2008). "Exploring network structure, dynamics, and function using NetworkX." (No. LA-UR-08-5495). Los Alamos National Laboratory (LANL).

[19] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao. (2009, April). "User interactions in social networks and their implications." In Proceedings of the 4th ACM European conference on Computer systems (pp. 205-218). ACM.

 ISBN: 978-1-61208-423-7

[20] A. Prieditis and G. Chen. (2013). "Mapping the Internet: Geolocating Routers by Using Machine Learning." In Computing for Geospatial Research and Application (COM.Geo), 2013 Fourth International Conference on Computing for Geospatial Research and Application (pp. 101-105). IEEE.

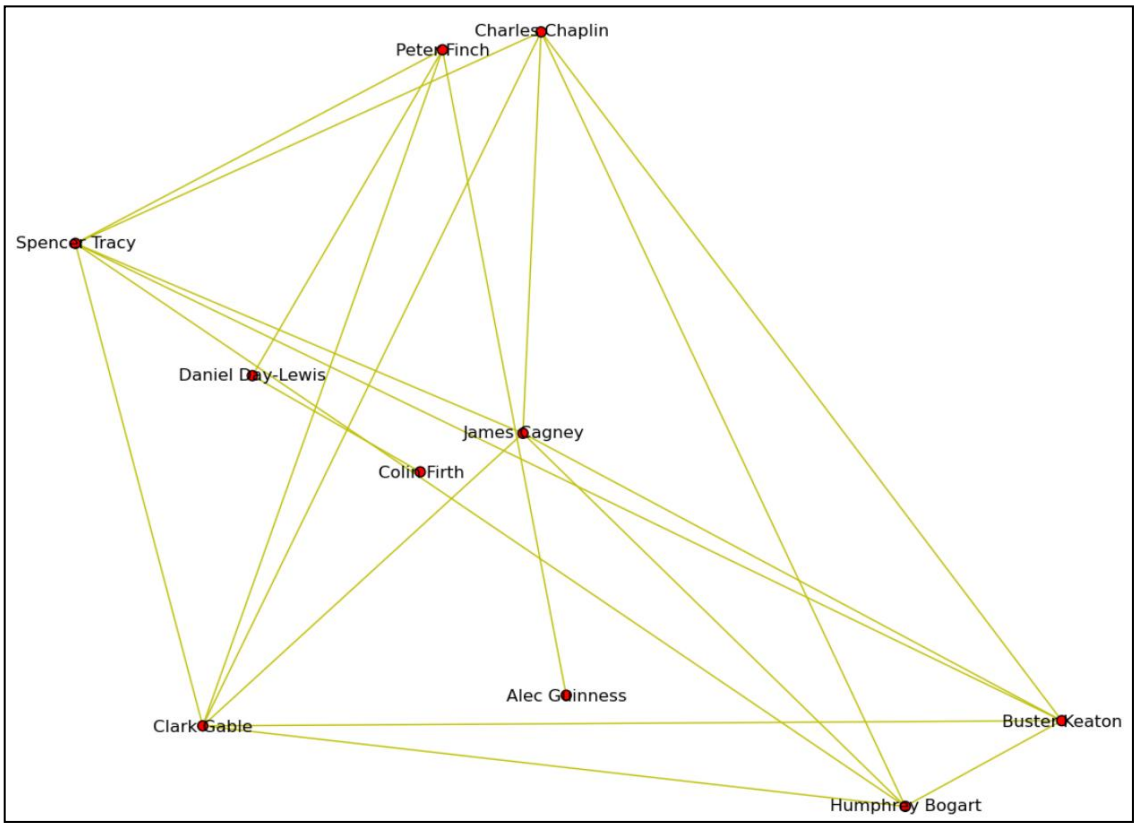

Figure 5. The Ten Actors Closest to the Center of the Largest Cluster

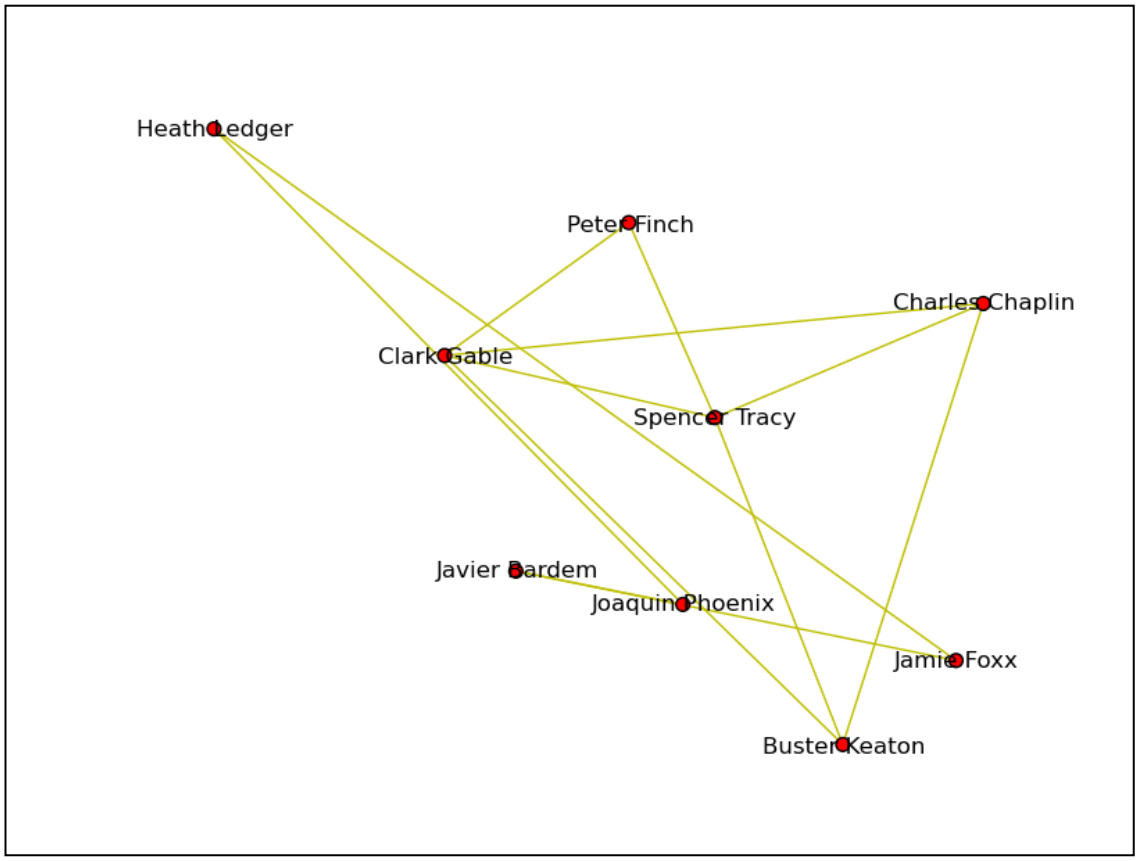

Figure 6. The Ten Actors Closest to the Center of the Next-Largest Cluster

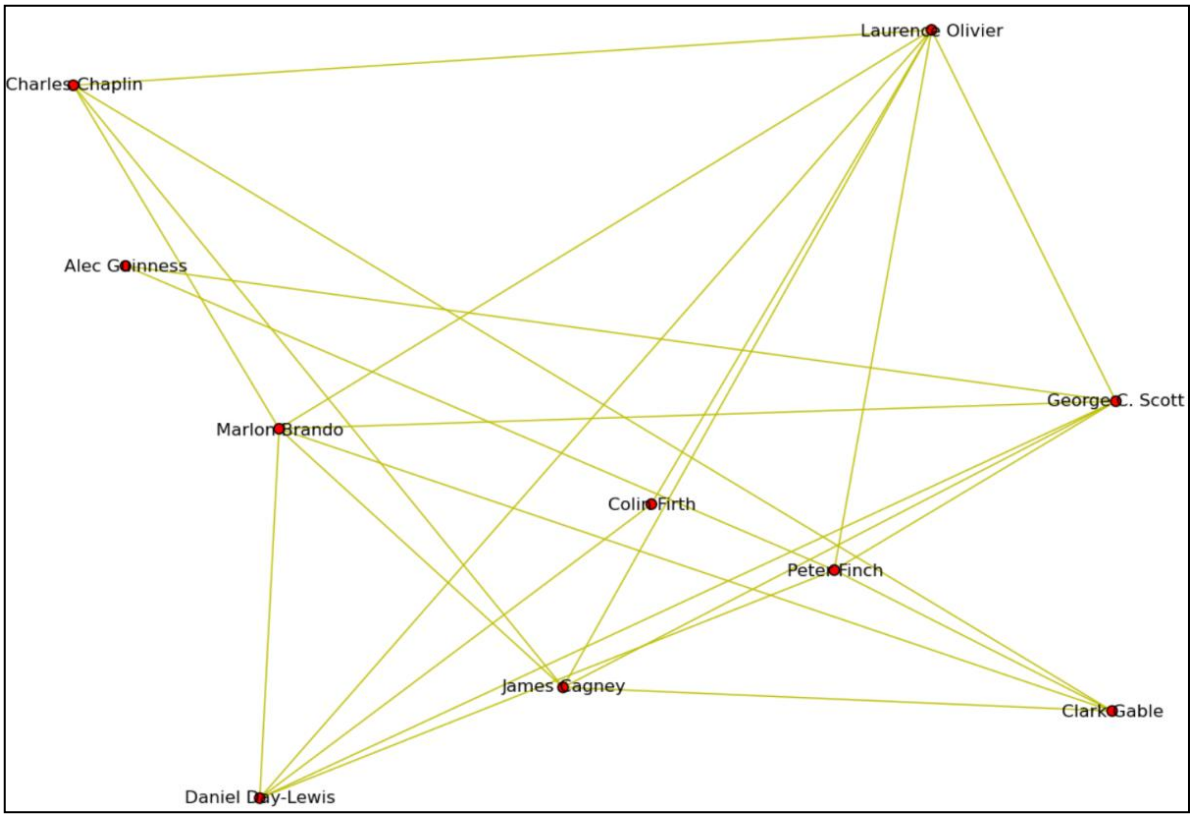

Figure 7. The Ten Actors Closest to the Center of the Third-Largest Cluster

 ISBN: 978-1-61208-423-7