Internal Report 2011–10 August 2011

Universiteit Leiden
Opleiding Informatica

Identifying Prominent Actors
in Social Networks

Iris Hupkens

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Bachelor Thesis
Identifying Prominent Actors in Social Networks

Iris Hupkens

August 9, 2011

Abstract

In this thesis we investigate how to automatically identify the prominent actors in social networks, so that they can for instance be targeted for viral marketing. To do this, we look at characteristics of the friendship graph and apply data mining methods from the Weka toolkit. We compare the set of nodes resulting from these methods with a set of nodes that are known to represent celebrities. When only a group of people equal in size to the group of celebrities is considered to possibly be prominent, using only the degree centrality turns out to result in an accuracy larger than methods with a higher time complexity. The HITS algorithm is found to work better on social networks than looking at degree centrality when a larger group is considered to possibly be prominent. A Bayesian network using various qualities, such as the average degree of someone's friends, can be used to nominate a set of nodes which contains even more celebrities. This implies that the method of looking at degree centrality to determine which actors are prominent is good enough when only a small group of people is to be selected, but that it starts losing some of its effectivity when the goal is to target the largest amount of prominent actors possible. It also implies that techniques which were developed for the web can be effective on social networks too.

1 Introduction

A *social network* consists of a group of people and the connections between those people. In the last decade many online social networks have appeared [2], both because websites such as Myspace became popular and because existing websites started integrating tools where users could link themselves to other users (an example of this is Youtube, where people got the ability to follow other people's channels). When these social networks are viewed as graphs, where each person is a node and each connection is an edge between two nodes, these social networks have certain characteristics in common. They are *small-world networks*, meaning that nodes do not have many nodes which they are directly connected to, but most pairs of nodes are connected by a short path, and the distribution of degrees follows a power law. Most of the nodes are *connected*,

creating one component consisting of a dense core of comparatively high-degree nodes surrounded by a fringe of low-degree nodes [10].

It can be useful to know who the prominent actors in a social network are. We think that a potential application could be in viral marketing [9], to target those people who are most likely to influence other people due to their connections. Several centrality measures [13] have already been proposed to identify the most important nodes in a graph, and some of them have been used to find out which websites are considered authorities based on the network of websites linking to each other. In a real-life situation we consider people important if they are famous, for example because they appear on the television a lot, or because they have written a popular series of novels. Most people add others on social networks because they have some pre-existing connection to them, which can include being a fan of their work. It is therefore interesting to see if the prominent actors in a social network as determined by running an algorithm designed for determining important nodes in a graph are the same as the people we would consider important. Is it possible to pick out important people in a reasonable amount of time purely by looking at which friends someone has within an online social network?

In Section 2 we describe some of the terminology used in this paper. In Section 3 we mention related work which has been done. Section 4 details the testset on which all methods discussed in this paper have been tested. Section 5 contains descriptions of each of the methods which are used to try and determine which nodes are important, and in Section 6 the results of the experiments are described and interpreted. In Section 7 we discuss the final result of the experiments, after which we describe potential future work.

2 Terminology

A graph G is defined as $G = (V, E)$, where V is a set of n vertices or nodes, and E is a set of m edges between these nodes, where

$$E \subseteq \{\{u, v\} : u, v \in V, u \neq v\}$$

The *friendship graph* refers to the graph of the connections between people. This graph is undirected (all edges are symmetrical) and there cannot be an edge from a node to itself. The word *friend* will be used to refer to any person which a person has a connection to, meaning an edge between them exists in the friendship graph. $\mathit{Friends}(u)$, the set of friends of u, is defined as:

$$\mathit{Friends}(u) = \{v : \{u, v\} \in E\}$$

A *follower* is someone who has a one-sided connection to someone. If a is a follower of b, then a receives any content which b broadcasts, but the reverse is not true unless b is also a follower of a. Not all online social networks support that possibility. Orkut, for example, does not, while Twitter does. Because in

our case the friendship graph is undirected, this will not be represented any differently from a friend in that graph.

A set of nodes is *connected* if there is a path from every node to every other node, and we call the set of connected nodes and the edges between them the *connected component*.

The word *celebrity* will be used to refer to someone who has been manually labelled as being a celebrity in real-life. There is no formal definition for whether someone is a celebrity or not, but celebrities are considered to be prominent actors. A *non-celebrity* is someone who is not a celebrity. So we have:

$$Celebrities \subseteq V$$

$$Noncelebrities = V \setminus Celebrities$$

Accuracy will be used to refer to the percentage of true positives, meaning nodes which are identified by the algorithm as prominent actors and were indeed marked beforehand as being celebrities. If there are ℓ celebrities in the dataset, ℓ people will be selected and nominated as prominent actors. The accuracy is then equal to the percentage of correct nominations. We have:

$$ProminentActors \subseteq V$$

$$|ProminentActors| = |Celebrities|$$

$$TruePositives = ProminentActors \cap Celebrities$$

$$FalsePositives = ProminentActors \cap Noncelebrities$$

$$Accuracy = \frac{|TruePositives|}{|ProminentActors|}$$

The *quality* of a node is a numeric measure of some characteristic which is being tested. The basic methods that are used to potentially find celebrities will return a quality value f for each node. What this quality represents depends on the method. It is hoped that there are differences in the quality of celebrities and non-celebrities, and that the average quality will be higher for celebrities than for non-celebrities. Qualities are not normalized but they are always positive real values.

3 Related work

Online social networks have not existed for very long and are still an active area of research. In [2] their history is described, starting in 1997 when the first online social network, SixDegrees.com, was launched.

Data gathered from the social networks Flickr, YouTube, LiveJournal and Orkut is analyzed in [10]. The analysis confirms that these social networks are scale-free, small-world and that the distribution of degrees follows a power law, and shows that these networks consist of a densely connected core of high-degree

nodes which connects smaller groups of low-degree nodes at the fringe to each other.

Several algorithms for studying the importance of nodes in networks relative to a set of nodes in that network are described in [13]. These could be used in various networks including online social networks.

There have been a few other studies where the characteristics of a social network graph were specifically used to try to determine certain qualities of the people in it. In [5] it is investigated which methods are most effective in predicting how influential users in an online social network are. In this case, the social network being studied was the social news aggregator Digg, and people were considered more influential if the news articles they posted got more votes from other people in the social network. A proposed method (NodeRanking) to determine the reputation of participants in a social network is described in [12]. It uses information about their position in that social network. The social network investigated was a research community, and participants in that network were considered to have a high reputation if their works were cited often.

Related research has also been done on the *webgraph* (graph of links between websites). Several techniques have been proposed for determining if a website has authority value, such as Pagerank [11] and HITS [7]. Because online social networks share certain characteristics with the webgraph, techniques which can be used to find authorative websites might also be useful in determining which users in a social network are authorative. For example, like a social network, the graph of the internet is scale-free. In this paper, the HITS algorithm will be investigated. HITS stands for Hypertext-Induced Topic Search and is an iterative method of ranking nodes based on the nodes it is connected to.

4 The testset

All techniques described in this paper have been tested on a random sample taken from an actual online social network, in which a small group of people is marked as important. The sample consists of approximately 500,000 nodes, of which 0.06% were labelled as celebrities. These celebrities were labelled by the administrators of the social network, who looked at whether they were famous in the real world in order to make their decision. It is a group consisting of politicians, musicians, TV personalities, writers, athletes, etc. All nodes in the sample form a connected component. Table 1 gives a more detailed description of this testset.

5 Measures for importance

The following subsections detail potential ways of determining which nodes in the friendship graph are important.

Nodes	498,061
Celebrities	313
Average degree (all nodes)	51
Average degree (celebrities)	855
Number of edges	12,949,922
Connected components	1

Table 1: Statistics of the testset.

5.1 Degree centrality

Common measures for centrality in a network include degree centrality, betweenness centrality and closeness centrality [4]. Of these, *degree centrality* is the simplest to calculate, because only information about a node and its friends is needed, and this is information which is often readily available in social networks. Degree centrality is the idea that nodes with higher degree are more important than nodes with lower degree.

In a directed graph, the indegree of a node is the amount of incoming connections it has. The outdegree of a node is the amount of outgoing connections it has. In undirected graphs like the friendship graph (which is the graph we are considering), these are the same, and the word degree is used to describe the amount of connections that a node has. We define the degree centrality f_{deg} of a node u as follows:

$$f_{deg}(u) = |Friends(u)|$$

When the amount of friends each node has is known, determining degree centrality can be done in linear time.

5.2 Nominating friends

For almost every person in a social network, that person's friends have more friends than they do. This is known as the friendship paradox [3]. A method which can be used in determining whether epidemics are about to start [1], is to take a random group of people, ask them to nominate a few of their friends, and then to see if any of these friends show signs of illness. Because people who have more friends are more likely to be named as friends by random people, the group created with this method consists of people who have more friends on average than a group consisting of random people would, and they are also more likely to be exposed to the disease early.

To try and use this effect to detect prominent actors, the method used is as follows: for every node in the graph, a random node is selected from the nodes which it had a connection to. The quality for that node is incremented by 1 every time it is chosen. Algorithm 1 shows the pseudocode implementation of this method. The function SelectPerson randomly selects one of the nodes in $Friends(u)$, with equal probability for each node. We define $f_{randnom}(u)$ as the

number of times u was selected when randomly selecting a friend for each node in the graph, with the chance of each friend of a given node being selected being equal. Because there is a random component in determining it, this quality value is not the same across different runs of the algorithm.

Algorithm 1 Nominating friends

```
for all u ∈ V do
    u.quality ← 0
end for
for all u ∈ V do
    v = SelectPerson(Friends(u))
    v.quality ← v.quality + 1
end for
f_randnom(u) = u.quality
```

A modified version of this is to select the most popular friend for each person in the social network, instead of a random friend. The most popular friend, in this case, is the friend with the highest outdegree. In this method each node's quality was equal to the amount of friends for which it is the most popular friend. So we define:

$$f_{popnom}(u) = |\{v \in Friends(u), \forall w \in Friends(v) : f_{deg}(u) \geq f_{deg}(w)\}|$$

Note that this means that if several of a node's friends are equally popular, then they are all selected. Algorithm 1 can again be used, but the function SelectPerson will be replaced by one which instead returns the subset of nodes with the highest degree.

Randomly nominating a friend for each node can be done in linear time. When every friend of a node must be checked to see if it is the most popular friend, nominating the most popular friend for each node can be done with complexity $O(m)$ where m is the amount of edges. Because the average degree in social networks is small compared to the amount of nodes, this will scale well.

It is also possible to nominate more than one person. In the most extreme case, everyone would nominate all their friends, and the resulting quality would be equal to degree centrality. When nominating a node's x most popular friends (without skipping over anyone when several of their friends are equally popular) the resulting quality value $f_{xpopnom}$ is defined as:

$$PopularityRank(u, v) = |\{w \in Friends(v), f_{deg}(w) > f_{deg}(u)\}|$$
$$f_{xpopnom}(u) = |\{v \in Friends(u), PopularityRank(u, v) < x\}|$$

5.3 Triadic closure

Friends are often part of groups which are tightly connected, and in social networks many new connections are made between friends of friends [8]. The cre-

ation of connections between two nodes which are both connected to the same node is known as *triadic closure*. This is a phenomenom which occurs over time, but even though the dataset used is static, it is still possible to look at how many of these connections already exist.

Two different types of triadic closure will be investigated: triadic closure between someone's friends, and triadic closure with a friend of a friend. It seems plausible that triadic closure between friends would be less common for celebrities than it is for non-celebrities. If someone is known by people from many different social groups then there will be proportionally less friendships between their friends.

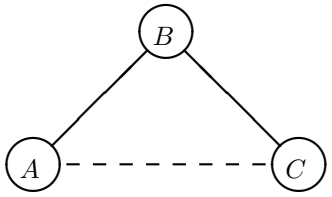

Figure 1: Three nodes, A, B, and C

The first type of triadic closure that will be investigated is the situation which is seen in Figure 1 with node B. B is a friend of both A and C, who might or might not be friends with each other. The method based on the existence of connections between two friends of a node, in this case A and C, will be known as *triadic closure between friends*. The quality of a node u will be equal to the amount of possible connections between pairs of its friends divided by the amount of these connections which actually exist:

$$PotentialConnections'(u) = |\{\{v, w\} : v, w \in Friends(u), v \neq w\}|$$

$$TrueConnections'(u) = |\{\{v, w\} \in E : v, w \in Friends(u)\}|$$

$$f_{tcbf}(u) = \frac{PotentialConnections'(u)}{TrueConnections'(u)}$$

The result is a quality which is higher if a smaller percentage of a node's friends are friends with eachother. This can be measured in $O(n(k_{max})^2)$ where n is the amount of nodes and k_{max} is the highest degree present in the graph. Determining the amount of potential connections for each node can be done in linear time, since it only depends on the degree of the node, which is presumed to already be known. For each node the set of potential connections can be determined in linear time on the amount of friends that node has. Its size increases quadratically with the degree of the node investigated, and the existence of each of these potential connections must be tested.

The second type of triadic closure that will be investigated is the situation where the node being investigated is node A in Figure 1. A is a friend of B and B is a friend of C, C might or might not be a friend of A. The quality of a node u will be equal to the proportion of friends of each of its friends who are also friends of u:

$$PotentialConnections(u) = |\{\{u, w\} : (\exists v : \{u, v\} \in E, \{v, w\} \in E)\}|$$

$$TrueConnections(u) = |\{\{u, w\} \in E : (\exists v : \{u, v\} \in E, \{v, w\} \in E)\}|$$

$$f_{tcffof}(u) = \frac{TrueConnections(u)}{PotentialConnections(u)}$$

The result is a quality which is higher when a node is friends with a larger percentage of its friends' friends. Like the previous type of triadic closure, and for the same reasons, this can be measured in $O(n(k_{max})^2)$. The time complexity of both of these methods does not lend itself well to investigating the amount of nodes which is likely to exist within a social network, but because the maximum degree is usually a lot smaller than n it is still possible for a powerful computer to calculate these values within a reasonable amount of time. It might be possible to reduce the complexity further by not checking edges in the graph multiple times. The complexity of checking every edge in the graph once is $O(m)$, which would scale better.

5.4 Average degree of friends

There might be differences in the average amount of friends a celebrity's friends have. To investigate this, each node's quality was set to the average degree of its friends:

$$f_{adof}(u) = \frac{\sum_{v \in Friends(u)} f_{deg}(v)}{f_{deg}(u)}$$

The range of this quality is more limited when a node's degree is higher. The highest possible quality can only occur when someone is only friends with the people in the graph who have the highest degree, because any average which is calculated using at least one value that is lower than the maximum will be lower than one calculated using only the maximum. This means that if x people are tied for highest degree, the people with the highest quality under this method can only have a degree of m. Similarly, if m people are tied for lowest degree, the people with the lowest quality under this method can only have a degree of x. Although this will not be taken into account during the actual calculation of the quality, it will be good to remember when trying to use this measure of quality to find the celebrities, to avoid looking only at nodes of low degree. The complexity of calculating these averages is $O(m)$.

5.5 HITS

One method which is investigated is a modified version of the HITS algorithm. HITS stands for Hyperlink-Induced Topic Search. It is a method which iteratively ranks webpages by assigning hub and authority scores to them [7]. During each iteration, the hub scores of all pages which link to a page are added to its authority score, and the authority score of all pages which the page links to are added to its hub score. At the end of each iteration the results are then normalized by dividing them by the square root of the squares of all authority scores or hub scores.

In an undirected graph, the hub and authority scores would be the same, so to test the effectivity of this algorithm in finding celebrities only a single score was used. This score becomes the quality for each node. The resulting modified HITS algorithm is as seen in Algorithm 2.

Algorithm 2 Modified HITS (k is the number of iterations)

for all $u \in V$ **do**
  $u.f_{hits} \leftarrow 1$
  for $i = 1 \rightarrow k$ **do**
    for all $u \in V$ **do**
      $u.f_{hits} \leftarrow \sum_{v \in Friends(u)} v.f_{hits}$
    end for
    $norm = \sqrt{\sum_{u \in V} (u.f_{hits})^2}$
    for all $u \in V$ **do**
      $u.f_{hits} \leftarrow u.f_{hits}/norm$
    end for
  end for
end for

The new quality of each node must be calculated from the quality of each individual friend of that node. Therefore, each iteration of this algorithm can be done in $O(m)$ time.

5.6 Combinations

By considering several factors together, it might be possible to reach a higher accuracy than what would be possible by only looking at the degree of nodes. The qualities generated by the preceeding methods will be fed to a datamining tool, WEKA [6], in order to see if any patterns emerge. The goal is to find a simple classifier. The first classifier which will be tried is a Naive Bayes classifier. The second is the C4.5 algorithm, which generates a decision tree.

6 Experiments

All of the methods described above were tested on the testset described in Section 4. They were implemented in C++ and the resulting quality for each node was used to try to determine who the celebrities were. For all the methods considered, the average quality of the nodes labelled as celebrities was found to be higher than for non-celebrities. The nodes were sorted on quality in descending order. When the top ℓ were selected from this sorted list of nodes, that already resulted in a non-zero accuracy for some of the methods. Table 2 shows these results.

Method	Accuracy
Degree centrality	60.7%
Triadic closure between friends	0%
Triadic closure with friends of friends	0%
Average degree of friends	0%
Nominate 1 random friend	36.1%
Nominate highest-degree friend	31.6%
HITS (10 iterations)	51.4%

Table 2: Results of sorting on quality and taking the top ℓ ($\ell = |Celebrities|$).

A method of generating qualities which has an initial accuracy of zero can still be useful in determining which actors are prominent when it is combined with other methods. An attempt was done to improve on these results by requiring prominent actors to have a certain minimum degree. The idea is that if someone has only a small amount of connections they are unlikely to truly be prominent. This was realized by ignoring nodes which had a degree that was lower than a cutoff point k. (The implementation of this was to have the sorting function sort nodes with a degree smaller than k behind those nodes which did have a degree of at least k. The reason it was done this way was to always allow ℓ prominent actors to be nominated even if k was chosen to be so high, that less than ℓ nodes in the friendship graph had a degree of at least k.) Figure 2 shows the results. If the cutoff point for degree coincides with what the cutoff point for degree would be when taking the ℓ celebrities with the highest degree, then all the methods will result in an accuracy equal to that of sorting on degree centrality (though the order of the top ℓ elements might be different). This is why all methods reach the exact accuracy of degree centrality at one point in the graph. HITS and triadic closure between friends both gave a better result than sorting on degree for some cutoff value. Selecting a random friend did as well, but because of the randomness inherent in this system and because the difference was small this is likely to be the result of random chance.

The results from these experiments are examined in more detail in the subsections below. Note that not all non-celebrities in the testset will be shown in the graphs which represent each node as a point, because doing so would make

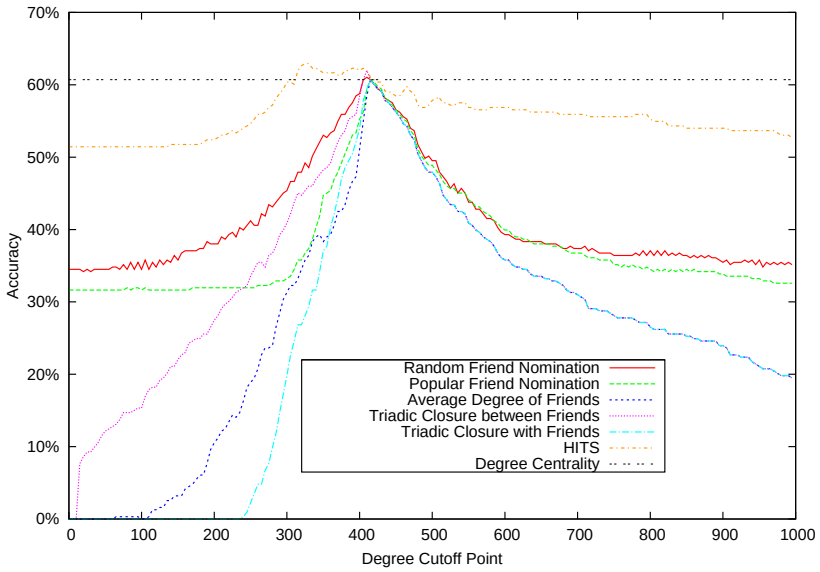

Figure 2: The relation between minimum required degree and accuracy. The horizontal axis shows the minimum degree required for a node to be a celebrity. The vertical axis shows the accuracy.

it difficult to see differences in density due to the large amount of data points which would be necessary to represent them. Instead, a random sample of 2% of the non-celebrities will be shown.

6.1 Degree centrality

When only the quality of each node was considered and nothing else, the accuracy for sorting on quality f_{deg} as described in Section 5.1 was the highest of any of the methods investigated. As can be seen in Table 2, approximately 60% of the celebrities in the testset were found by sorting the people by degree and naming the top ℓ as being prominent actors. The chance of finding a celebrity by selecting a random person out of the entire testset is a thousand times as small as the chance of finding one by picking a random person out of this top ℓ. However, this method leaves 40% unaccounted for, and it might very well be possible to do better.

Celebrities in this testset tend to have a larger amount of friends than non-celebrities, which is consistent with the way authorities are determined in the network graph of the internet by algorithms such as PageRank. Sites which

Total iterations	10
Best accuracy	36.1%
Worst accuracy	31.9%
Average accuracy	33.8%
Mean	34.1%
Standard Deviation	1.68%

Table 3: Statistics of $f_{randnom}$

are linked to by a lot of different sites tend to be more important, while if a website just has a lot of links to other sites, that does not say anything about the authority of that site.

6.2 Nominating friends

Both of the methods described in Section 5.2 performed similarly, but the accuracy of sorting on $f_{randnom}$ was higher than that of sorting on f_{popnom}. Determining the quality for each node by randomly choosing a friend for each node could be used to find (on average) slightly more than a third of the celebrities. When the method was used 10 times, the results were as shown in Table 3. Although there was a random component to its accuracy, in all 10 cases the accuracy was higher for the top ℓ nodes of $f_{randnom}$ than for f_{popnom}. Neither method performed better than simply sorting on degree.

When nominating x of each person's most popular friends and sorting on the resulting quality values $f_{xpopnom}$, the accuracy gradually moved to that of degree centrality, but logically never became better. This can be seen in Figure 3. For a high enough x, the accuracy was the same as for degree centrality.

6.3 Triadic closure

Neither of the two quality values described in Section 5.3 had an accuracy higher than zero, however there were still statistical differences between celebrities and non-celebrities in the quality values calculated.

When comparing the percentage of possible connections which exist between the friends of a person (triadic closure between friends), there was a negative correlation with being a celebrity. It did not make a difference for the result unless people with low degree were ignored, because the celebrities still had a larger percentage of triadic closure than those non-celebrities who happened to have only a few friends, none of which were connected. When people with low degree were ignored, it was possible to find slightly more celebrities using this method than by using degree centrality, but only if the cutoff point was very close to the minimum degree found in celebrities nominated by looking at the nodes with the highest degree. The graphs of Figure 4 and Figure 5 show the relation between the degree of a node and the percentage of possible connections which exist between its friends.

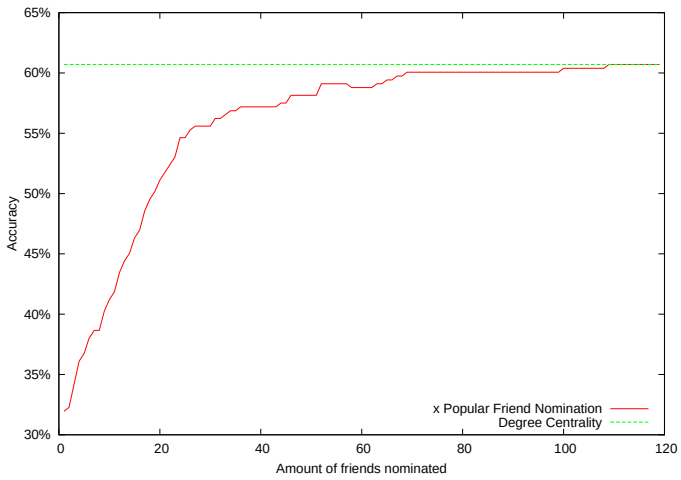

Figure 3: The relation between accuracy and the amount of most popular friends nominated per node. The vertical axis shows the accuracy (in percentage of celebrities found in the top ℓ, the horizontal axis shows the value of x.

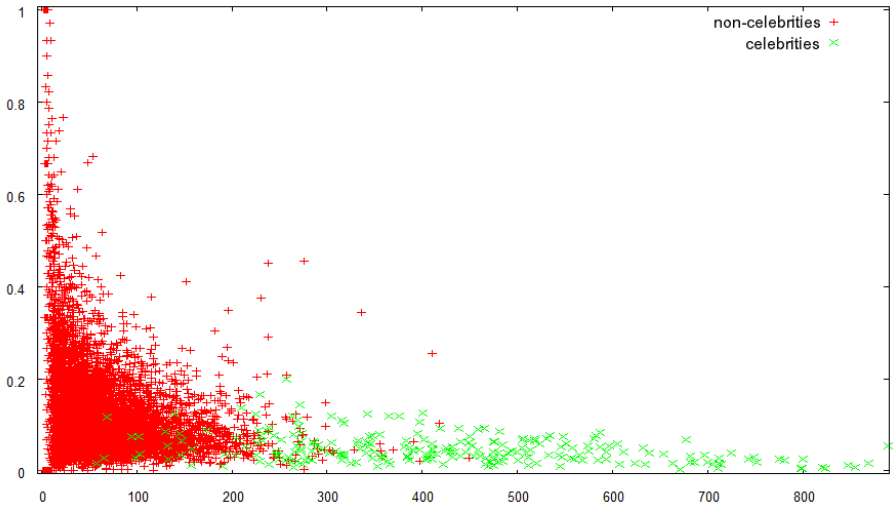

Figure 4: The relation between degree and triadic closure between friends. The horizontal axis represents degree. The vertical axis represents the percentage of possible connections which exist between someone's friends.

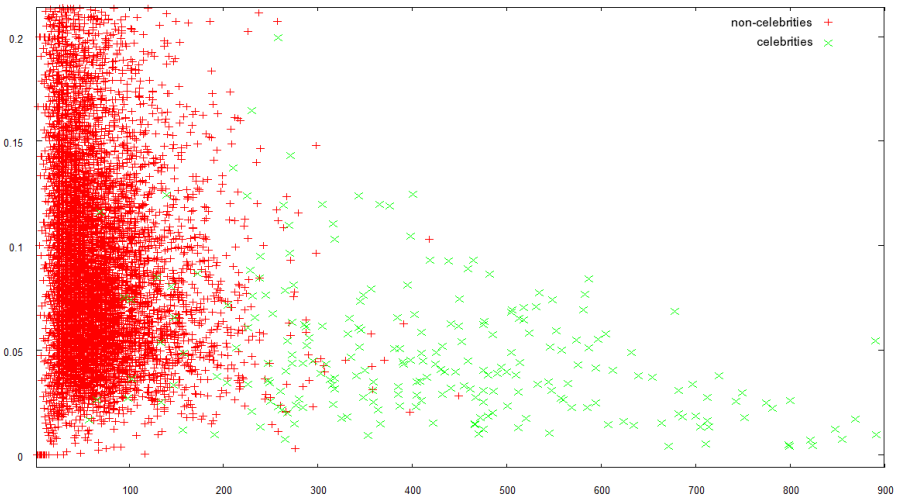

Figure 5: A close-up of the bottom left part of Figure 4.

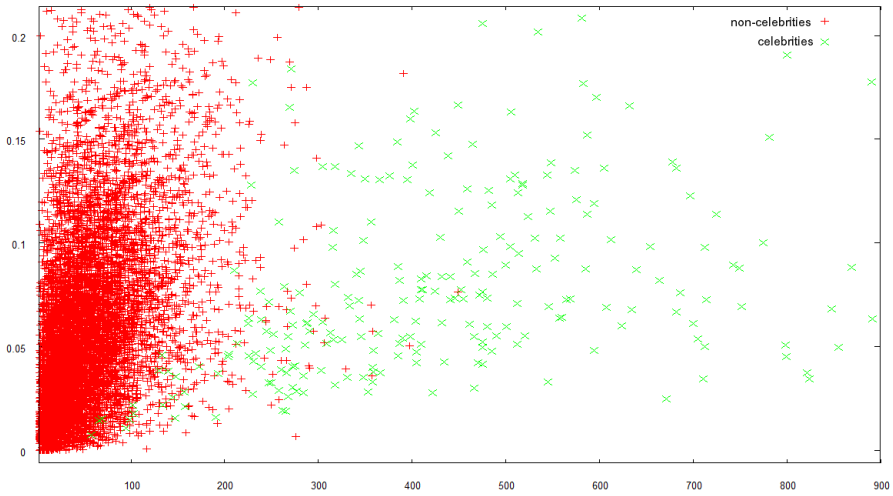

Figure 6: The relation between degree and triadic closure with friends of friends (close-up). The horizontal axis represents degree. The vertical axis represents the percentage of possible connections which exist between the friends of a person's friends and that person. Both axes are cut off in order to show more detail in the most dense area of the graph.

When comparing the percentage of possible connections which exist between the friends of a person's friends and that person, there was a positive correllation with being a celebrity. Figure 6 shows the relation between the degree of a node

and the percentage of possible connections which exist between the friends of a person's friends and that person. The full version of the graph in Figure 6 has been omitted because it did not show a lot of extra information. The top part contained only non-celebrities, all with low degree, and the right part contained only celebrities with an average quality similar to that of the celebrities shown in the zoomed-in version. Although there are statistical differences, they are not clear-cut, there merely appears to be a larger chance for a celebrity to have certain values than there is for a non-celebrity. Because the amount of non-celebrities in the testset is large enough to make up for statistical differences, this alone does not result in a high accuracy.

6.4 Average degree of friends

Using the method described in Section 5.4 also did not directly lead to a non-zero accuracy when only the quality value was considered, although there were statistical differences. The graphs in Figure 7 and Figure 8 show the relation between the degree of a node and the average degree of its friends. The graph in Figure 7 shows a zoomed-out view of the entire graph. Figure 8 is a view which zooms in on that part which contains the most data points.

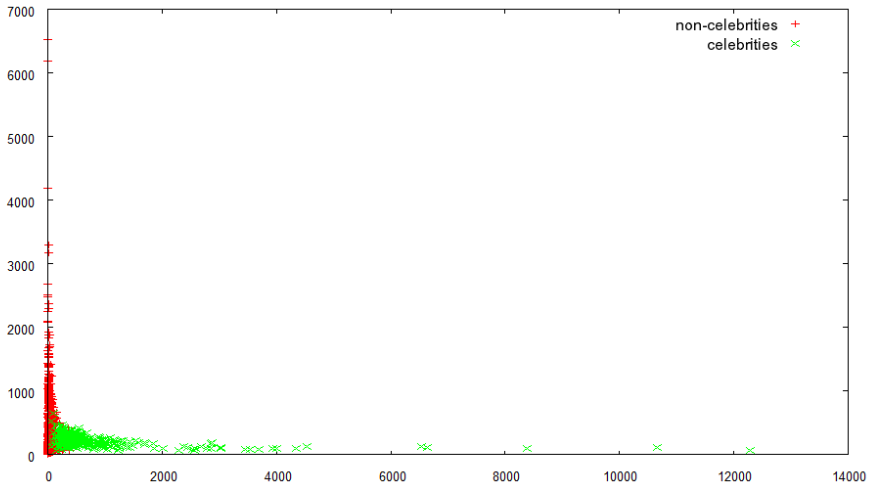

Figure 7: The relation between degree and average degree of friends. The horizontal axis represents degree. The vertical axis represents average degree of friends.

On average the celebrities have friends with higher degree than non-celebrities, but there are a lot of non-celebrities (even ones with more friends than some celebrities) whose friends have higher degree than any of the celebrities. The highest average degrees of friends are only attained by people who have very few friends, but who happen to have one friend with very high degree (a celebrity in most cases). Because only 2% of the non-celebrities are shown in this figure

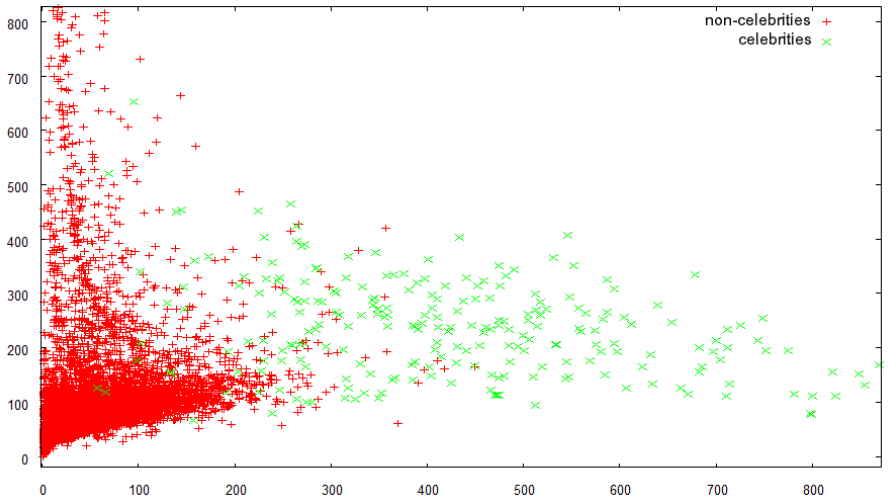

Figure 8: A close-up of the bottom left part of Figure 7.

the actual ability to separate the datapoints shown in Figure 8 into celebrities and non-celebrities is not as good as it seems, so that no cutoff point for minimum degree was found where this method resulted in a higher accuracy.

Since celebrities tend to have a lot of friends, and they are not friends with themselves, it seems like the highest averages would be most likely to be reached by people with a low degree who were connected to a few very high-degree nodes.

6.5 HITS

The modified HITS algorithm described in Section 5.5 gave the most promising results of all the methods tested. Table 4 shows the accuracy when sorting on f_{hits} after each iteration of the HITS algorithm. Even after more than 20 extra iterations, the accuracy did not change any further. Because of this, the HITS algorithm was allowed to run for 10 iterations in all further experiments requiring the HITS quality values (such as the test described in Figure 2).

After one iteration the accuracy was exactly equal to that of the degree centrality. This is because the quality of each node was determined by taking the sum of qualities of a node's friends and normalizing the resulting qualities, and because these qualities all start at 1 each node's quality after one iteration was proportional to its degree.

HITS worked better than any other method when nodes with low degree were ignored when selecting prominent actors. If the cutoff point was in a range not too far below the minimum degree, it resulted in better accuracy than the method based purely on each node's own degree. It outperformed or equalled all the other methods (except degree centrality) with any cutoff point for nodes to be ignored, too.

Iteration	Accuracy
1	60.7%
2	48.9%
3	56.9%
4	50.2%
5	53.7%
6	50.7%
7	52.1%
8	51.1%
9	51.4%
10	51.4%

Table 4: Accuracy of modified HITS

6.6 Discussion

Of all the methods investigated in this paper, HITS with a minimum required degree performed best in nominating prominent actors. Looking at triadic closure (between friends) with a minimum required degree was also promising. Because the time complexity of determining degree centrality in a social network is linear in the amount of nodes, the other methods also do not outperform degree centrality in speed.

To better compare the ability of the methods to distinguish between regular people and prominent actors, the *ranking* of each celebrity in the sorted list was calculated for all methods. This ranking is the node's place in the list sorted on the quality generated by the method. To prevent inaccuracy resulting from a quality measure rating many nodes as having the same quality, this ranking was made equal to the highest position of a node with equal quality in the sorted list. So if celebrity x was in the nth spot in the sorted list, and $n-1$ to $n-k$ had the same quality as x, the ranking of x is set equal to $n-k$.

The results are shown in Figure 9. Interestingly, HITS starts outperforming degree centrality as a measure of importance of nodes when more than a thousand nodes are considered to be prominent actors, instead of just a number equal to the amount of pre-labelled celebrities. It is also shown that even for the methods which do not directly result in a non-zero accuracy, the ranking for the majority of celebrities is better than average. There are approximately 500,000 nodes in the testset, so any node with better-than-average quality would have a ranking lower than approximately 250,000. There is a slight bias towards lower ranking values due to the way they are calculated, but this only really seems to be a problem for the list created by sorting on $f_{randnom}$. There, it can be seen that the nodes are grouped into clusters which have the same quality value.

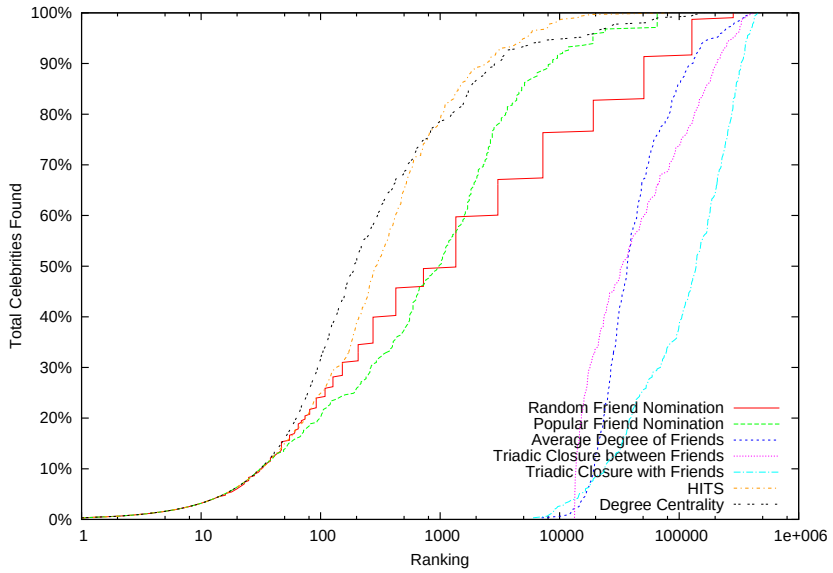

Figure 9: The relation between ranking in the sorted quality list and amount of celebrities encountered by that point. The vertical axis is logarithmic in order to better display differences in the first few hundred nodes of a sorted list.

6.7 Combinations

To determine whether the various quality values could be effective when combined with eachother, the WEKA tool [6] was used to test various classification algorithms on the testset. The input of this tool was a list of quality values as seen below. Due to the random nature of random friend nomination, quality values generated by that method were not considered for this experiment. An example of what the dataset looks like can be seen below. In this example, the values represent, in order: f_{deg},f_{adof},f_{tcbf},f_{tcffof},f_{hits},f_{popnom}, and celebrity status. The last value is 1 if the node is a celebrity and 0 if it is not.

```
107,83.0654,0.0370305,0.0472547,0.000290582,6.00029,0
323,119.898,0.0527181,0.141581,0.00158989,114.002,0
5,45.4,0.7,0.061674,2.61316e-006,2.61316e-006,0
60,84.9667,0.147458,0.102393,7.06898e-005,2.00007,0
...
```

By using the Naive Bayes classifier, the nodes were classified as seen in Table 5. A set of approximately 5,000 nodes (1% of the testset) was nominated as prominent actors. This set included 96.4% of the celebrities in the testset. The Naive Bayes classifier outperforms the other methods, but not by much: Compare Figure 9, where 94.9% of celebrities have a ranking below 5,000 when using the HITS method.

not prominent	prominent	
493,067	4,712	non-celebrities
11	302	celebrities

Table 5: Confusion matrix of the Naive Bayes classifier.

Using the C4.5 algorithm on the testset to generate a decision tree always resulted in a tree which immediately rejected everything below a certain degree as not being a celebrity, resulting in about 20% of the celebrities not being considered prominent actors. The values f_{hits} and f_{adof} (average degree of friends) were then used to partition the high-degree nodes further, but even when the tree was pruned, there was not enough generalisation for a tree generated on part of the testset to work on another part of it. Typically, less than half of the celebrities were considered to be prominent actors through this method, while only around 80% of the prominent actors nominated by the decision tree were celebrities. An example of a tree that was generated using this method is displayed below. For each node in the tree, a test is done on one of the quality values to see which node of the tree should be proceeded to. If no more tests are to be done, the test is followed by a colon and the expected celebrity status. The first number between brackets represents nodes that were classified correctly, the second number represents nodes that were classified incorrectly. Somewhat interestingly, it turned out that a low average degree of friends rather

than a high one could be used to determine which nodes were prominent after looking at f_{deg} and f_{hits}, even though f_{adof} is higher on average for celebrities than for non-celebrities.

```
degree <= 312: 0 (497222.0/71.0)
degree > 312
|   degree <= 498
|   |   hits <= 0.015068
|   |   |   hits <= 0.005728: 0 (262.0/3.0)
|   |   |   hits > 0.005728
|   |   |   |   averagefrienddegree <= 123.889: 1 (8.0/3.0)
|   |   |   |   averagefrienddegree > 123.889: 0 (245.0/17.0)
|   |   hits > 0.015068
|   |   |   averagefrienddegree <= 261.926
|   |   |   |   averagefrienddegree <= 189.211: 1 (7.0)
|   |   |   |   averagefrienddegree > 189.211
|   |   |   |   |   tcffof <= 0.053022: 1 (5.0)
|   |   |   |   |   tcffof > 0.053022: 0 (44.0/19.0)
|   |   |   averagefrienddegree > 261.926: 0 (123.0/33.0)
|   degree > 498
|   |   hits <= 0.020564
|   |   |   averagefrienddegree <= 153.023: 1 (27.0/2.0)
|   |   |   averagefrienddegree > 153.023: 0 (21.0/3.0)
|   |   hits > 0.020564 (122.0/4.0)
```

7 Conclusion and future work

We have studied ways to determine celebrity status in friendship graphs. In this paper we have shown the following:

- Looking purely at the degree of nodes in a social network is a surprisingly effective method of determining who the prominent actors are, especially when compared to methods such as picking the most popular friend for each node. By contrast, methods which are based on nominating friends of nodes turn out not to be a very good method of determining which nodes are prominent.
- Looking at the percentage of triadic closure between a node's friends can be combined with information about its degree to more accurately predict which actors are prominent.
- If a large amount of true positives is desired and the amount of false positives is considered less important, HITS seems to be more effective in determining how likely it is that someone is a prominent actor when that person does not have a high degree.

- Prominent actors are characterized by having friends who have less connections between eachother than average. Their friends are also likely to be more popular than average, although this is not guaranteed.
- Combining the knowledge about degree and degree of friends with that of whether friends are friends with eachother can be used to more effectively decide who the prominent actors are, but so far, the demonstrated gains are small. A Naive Bayes classifier can work quite well when taking more false positives for granted, while the C4.5 algorithm is less promising.

Due to the high time complexity of determining certain centrality measures such as eigenvector centrality and betweenness centrality, they were not investigated. In future work it might be interesting to investigate these centrality measures by calculating approximate values. It might also be interesting to investigate if it is possible to determine the triadic closure properties of nodes in a network with lower time complexity.

Acknowledgements

This bachelor project was done under supervision of Walter Kosters and Frank Takes. Many thanks to both of them for their help and support.

References

[1] Nicholas A. Christakis and James H. Fowler. Social network sensors for early detection of contagious outbreaks. *PLoS ONE*, 5(9):e12948, 2010.

[2] danah m. boyd and Nicole B. Ellison. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, 2008.

[3] Scott L. Feld. Why your friends have more friends than you do. *American Journal of Sociology*, 96(6):1464–1477, 1991.

[4] Linton C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, 1979.

[5] Rumi Ghosh and Kristina Lerman. Predicting influential users in online social networks. In *Proceedings of KDD workshop on Social Network Analysis*, 2010.

[6] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1):10–18, 2009.

[7] Jon M. Kleinberg. Authorative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[8] Gueorgi Kossinets and Duncan J. Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88–90, 2006.

[9] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. The dynamics of viral marketing. *ACM Transactions on the Web*, 1(1):5 pages, 2007.

[10] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, pages 29–42, 2007.

[11] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank Citation Ranking: Bringing Order to the Web*. Stanford Infolab, 1999.

[12] Josep M. Pujol, Ramon Sanguesa, and Jordi Delgado. Extracting reputation in multi agent systems by means of social network topology. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1*, pages 467–474, 2002.

[13] Scott White and Padhraic Smyth. Algorithms for estimating relative importance in networks. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 266–275, 2003.