Fake Profile Detection

Jake Garrison, James Goin, November 22, 2014
EE 418 Network Security
Dept. of Electrical Engineering, University of Washington

Question 1

1. For the data set **Fake_data1.mat**

   (a) What percentage of fake users are correctly detected as fake based on the:

      i. Attribute similarity metric?
      0%

      ii. Friend similarity metric?
      79.23%

      iii. Basic profile similarity metric?
      34.06%

   (b) What percentage of real users are incorrectly identified as being fake based on the metrics (i)-(iii) from Part (a)?

      i. Attribute similarity metric?
      0%

      ii. Friend similarity metric?
      0%

      iii. Basic profile similarity metric?
      0% mean, 0.065% max

   (c) How are the answers to (a) and (b) affected by varying the parameters [1]:

      i. The attribute similarity thresholds ϵ and δ?

      Attribute similarity detection was entirely dependent on the value of ϵ since the number of shared attributes is constant for all profiles in this data set. If $\epsilon \leq 2$, then all outputs of the attribute metric would scale to the value of δ, and if ϵ was > 2 then the attribute similarity metric equation would be used. If δ is set too high, the detection rate will be 100% no matter what profile is being observed [Figure 1]. This case is discussed later. Additionally since all cloned profiles and their respective victim shared two attributes, they would all share the same attribute similarity value, it would either be δ, or whatever the metric evaluated to. So in this case, the attribute similarity was a binary metric, and thus not very useful in our analysis.

      The δ limit is used to provide extra weight for accounts with a small number of attributes. Since all profiles had an identical number of attributes, its purpose in this context is questionable. For false positive detection, δ had the same effect on detection rate as in the fake profile detection tests since it is simply a lower limit. ϵ however resulted in zero false positives no matter what it's value based on the 30 randomized tests. This is because the likely hood of two random accounts sharing identical attributes is small [Figure 1].

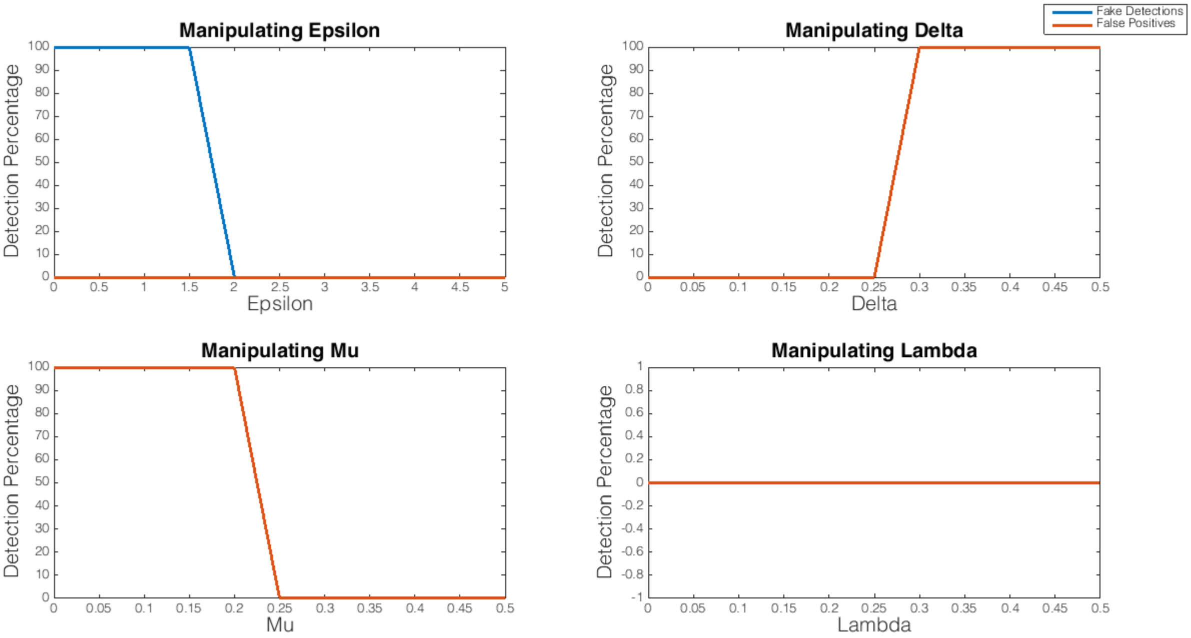

Figure 1: Attribute Similarity Metric

Note: Figures 1 to 3 show how manipulating a single variable and holding the others to their default value changes the overall detection rate for the attribute, friend and basic profile metrics

ii. The friend similarity threshold λ and μ?

The μ threshold played a large part in overall detection since it was the threshold at which the metric identified an account as a clone or not. It is designed to be used to balance the amount of false positives with successfully detected clones. If μ is set too low (< 0.1) Nearly everything will be detected as a clone, so there are many false positives. If set too, high nothing will be detected as a clone. Ideally there would be an optimized value of μ that would be an ideal compromise between false positives and successful detections, but since these spaces did not overlap using the default variables with the given data set, no such optimization was possible. It should be set so that all profiles are not trivially detected as clones or original. In our results this optimal range is between 0.2 and 0.35, but in this range there are no false positives based of the 30 randomized subset tests. Further testing on the complete set would likely yield some false positives in this range of μ [Figure 2].

λ is used to give more weight to accounts with a relatively small friends list. Since the lower bound of friend similarity metric is set by λ, raising λ will effectively raise the clone detection rate. As with δ, since λ is also a lower bound threshold, if it is too high the detection percentage will always be 100%. In testing for false positives, λ had a minimal impact since most profiles compared didn't have any mutual friends. This is because the likelihood of two random accounts sharing identical attributes is slim [Figure 2].

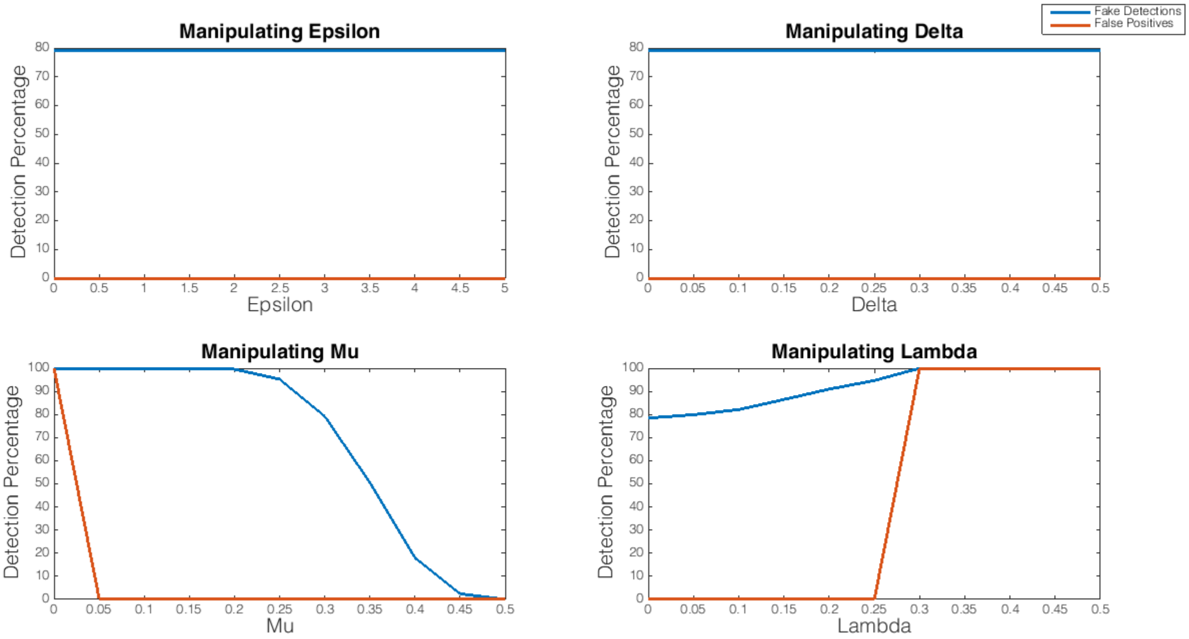

Figure 2: Friend Similarity Metric

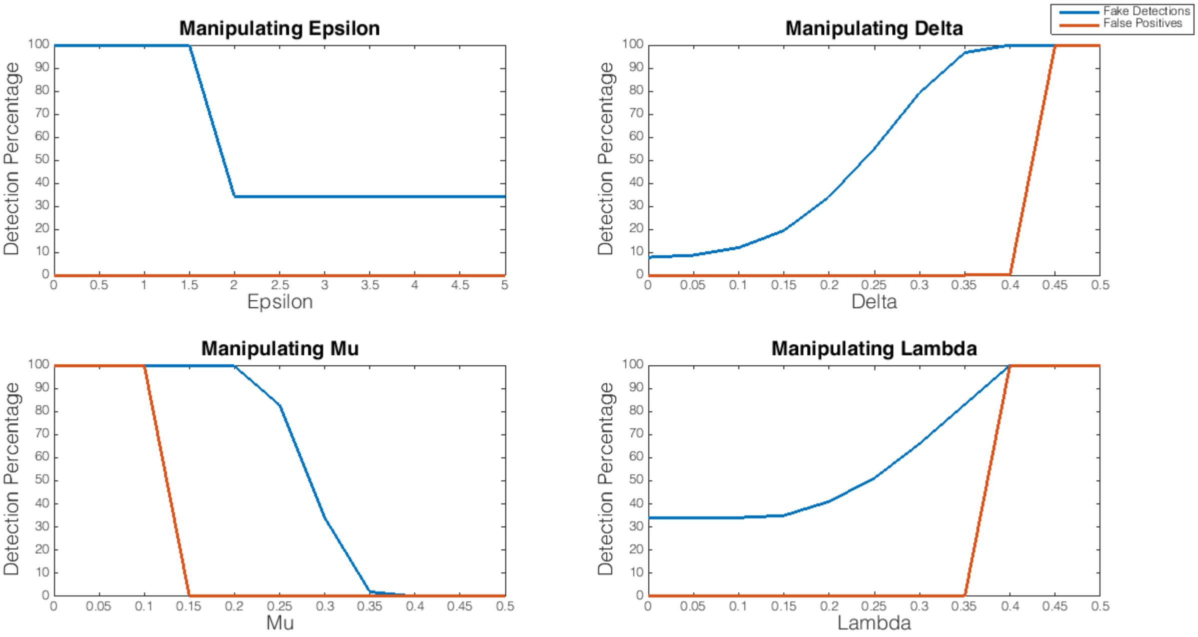

Figure 3: Basic Profile Similarity Metric

Note: In observing that λ and δ rescale the friend and attribute similarity respectfully, if one of them is ≥ 0.42 (using default parameters) no matter what the other is you will have a misleading 100% detection for basic profile similarity for $\mu = 0.3$. This is because the minimum similarity metric result for setting λ or δ to 0.42, μ to 0.3 and κ and χ to 1 is 0.3 since:

$$\frac{0.42}{\sqrt{2}} \geq \mu$$

As a result all profiles with δ or $\lambda \geq 0.42$ will be falsely detected fake as shown in the plots.

iii. Overall, which of the metrics (i)–(iii) provided the most effective (highest detection probability and lowest false-alarm probability) mechanism for identifying fake users? Illustrate your answers to (a)–(d) through graphs and tables as needed.

In our analysis the friend similarity metric proved to have the highest detection rate (79.23%). Though it would seem basic profile similarity would yield better overall results, the detection rate dropped dramatically since attributes are being considered as well ($\kappa \neq 0$) [Figure 3]. Because all of the attributes are $\leq \epsilon$, they get rescaled to δ, which is a fairly small number and contributes little to the numerator of the friend similarity metric. The denominator however is larger $\sqrt{2}$ since $\kappa = \chi = 1$. As a result, the basic profile metric decreases when attributes are considered. A solution to this for this particular data set would be to optimize κ and χ for optimal clone detection. This optimization yields a non-surprising result of setting κ to one and χ to zero. This optimization results in the friend similarity metric being equivalent to the basic profile similarity, which is an overall higher percentage. The reason we opted to use the non-optimized κ and χ and keep the attribute data in our basic similarity test was because it enabled us to vary ϵ and δ in later analysis with the hopes of identifying an ideal value for μ.

Question 2

1. For the data set **Fake_data2.mat**

Note: For (a) and (b) values used are for accumulation metric with d = 3, α (rate cutoff) = 1, and final cutoff of 88%. For node metric an α cutoff of 44% was used. The value for final cutoff of 88% was calculated off of the parameters given and figure 1, while the node final cutoff was calculated from Figure 7. These are based off of the fact that the intersection of these plots show the point where both detections are optimized without sacrificing from each other. More explanation of this is given in part (c). All of the following graphs are relating a specific adjustable metric variable to the percent of profiles correctly identified, real as real and fake as fake in order to find these optimization's. Also the percent of real users who are incorrectly identified as being fake is equivelent to one minus the percent identified correctly.

(a) What percentage of fake users are correctly detected as fake based on the:

i. Accumulation rate metric?
63.33%

ii. Node metric?
90.00%

(b) What percentage of real users are incorrectly identified as being fake based on the metrics (i)-(iii) from Part (a)?

i. Accumulation rate metric?
49.3%

ii. Node metric?
79.45%

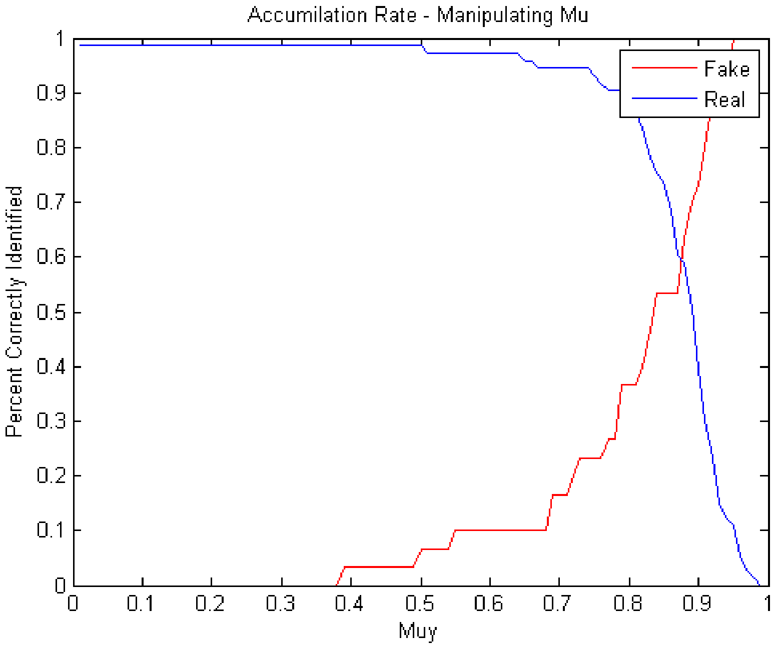

Figure 4: Accumulation Rate Metric Manipulating Mu, Constant d=3 and Cutoff=1

(c) How are the answers to (a) and (b) affected by varying the parameters [2]:

i. The sample interval d?

As the sample interval d increases the percent of fake profiles correctly identified increases and percent of real profiles correctly identified decreases as shown in figure. Based on how accumulation rate is calculated it can be shown that the sample interval d does not effect the mean, but as d increases the standard deviation decreases.

$$\alpha_p = \frac{|\mu_p - \mu|}{\sigma}$$

This leads to larger values of α_p. Since a profile is considered fake if $\alpha_p \geq \alpha$ (α is set as 1 in this case), profiles are more likely to be flagged as fake. As d increases the percentage of fake users correctly detected as fake will also increase, however the number of real users incorrectly identified as fake will also increase. As shown in the Figure 5, by comparing the percent of users correctly identified as fake or real per profile for different values of d, the intersect represents the point where both are maximized without sacrificing from each other. From this point optimization is up to the social network administrator to decide whats more important, detecting all fake profiles (d increasing), or minimizing false positives (d decreasing).

ii. The rate of accumulation threshold, α?

The rate of accumulation threshold, α, is used to determine how many standard deviations from the mean is considered acceptable. If an interval leads to an α_p greater than α then it is considered fake. This value has a large influence on the detection rate since it is primarily what determines if a certain interval is considered to have a normal growth rate. The profile is then tagged with a percent of these rates that are considered fake, which is then compared

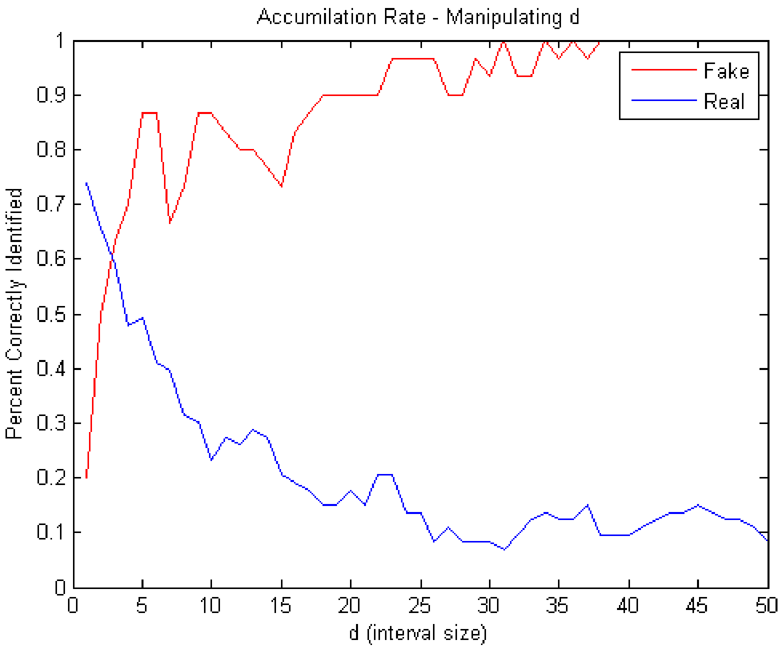

Figure 5: Accumulation Rate Metric Manipulating d, Constant Mu and Cutoff

to another threshold shown before part (a).

This can be shown in Figure 6 which compares the percentage of real and fake profiles being correctly identified (fake as fake, real as real) as α is varied. The intersection of these plots shows the point where both detections are maximized without sacrificing from the other. As the cutoff α increases the number of fake profiles correctly identified drops at approximately linear rate, and as α decreases the percentage of real users identified as fake decreases (real users identified correctly increases).

(d) Overall, which of the metrics (iv)-(v) provided the most effective (highest detection probability and lowest false-alarm probability) mechanism for identify fake users?

The node metric had a best case 90% detection rate, however it also incorrectly identified 79.45% of the real users as fake. As you vary alpha for the node metric, there is an immense change in detection characteristics for fake profiles. A strong metric would have a high percent of fake profiles and real profiles correctly identified with the same alpha. Based on Figure 6 it can be shown that there is no alpha that has a high percent of correctly identified profiles with overlap between fake and real. This is due to an immense drop in correctly identified fake profiles around alpha of 0.5 going from about 90% to 20%, while real profiles detection rate grows approximately linear. This shows that based on the data set we were given that the node metric algorithm is not a suitable metric, even under optimization, for profile detection due to the high percent of real profiles incorrectly identified as fake, even though it has a high fake profile detection rate.

The accumulation metric had a 63.3% fake users identified as fake and 49.3% real users incorrectly identified as fake. This was with a relatively optimized value of the final cutoff for given values of $d = 3$ and $\alpha = 1$, however it would still be possible to optimize these values better based on manipulating d (interval size) and α to still maximize fake users identified and lower real users incorrectly identified. This can be shown in part (c). It may not be possible to make the detection method perfect, however if the social network administrator cared more about whether all profiles were detected or just minimizing real users being incorrectly identified, they could modify the parameters to better match those goals without as drastic of a change in the other, unlike the node metric.

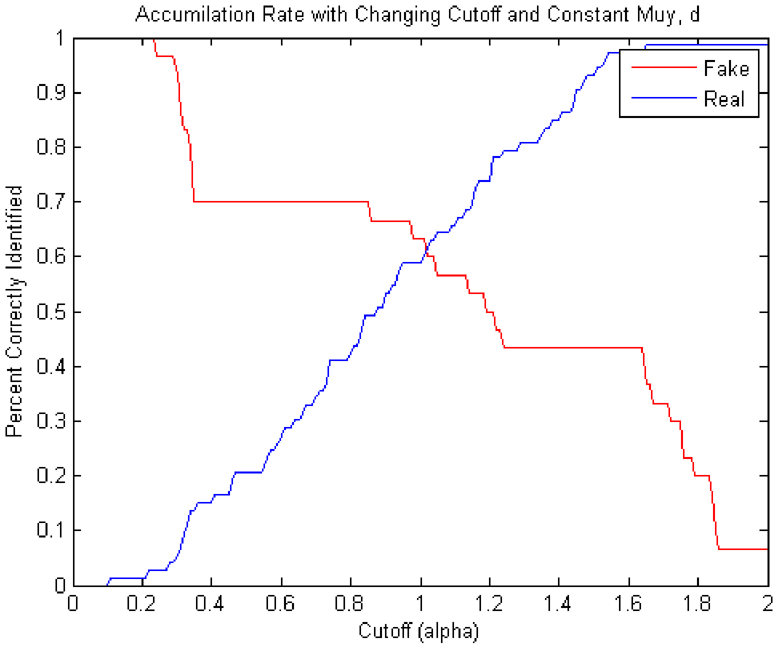

Figure 6: Accumulation Rate Metric Manipulating Cutoff, Constant Mu and d

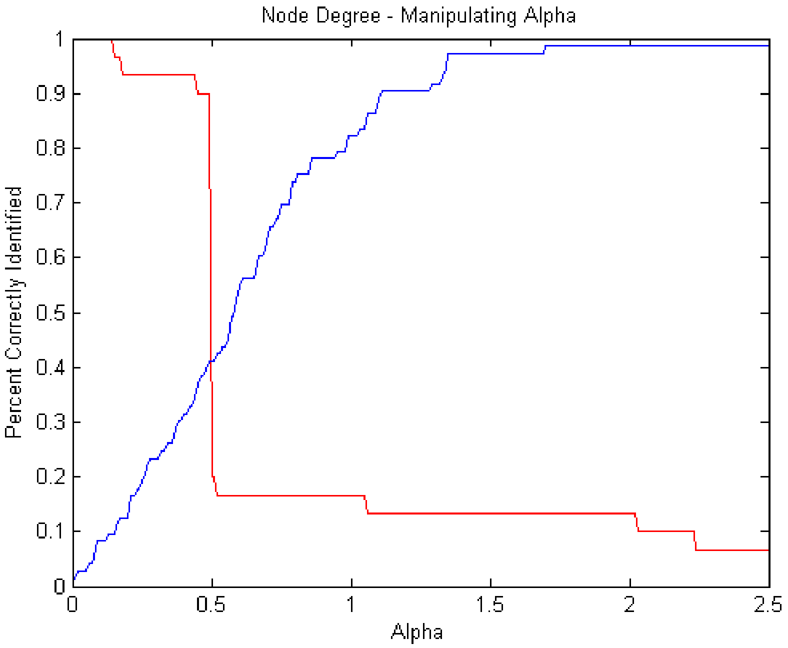

Figure 7: Node Rate Metric Manipulating Alpha

Question 3

In [1], a detection method for multiple faked identities profile similarity is introduced, including two types of attack with corresponding metrics. What are the two types of attack that are considered? How do the attacks differ from those considered in the remainder of [1]? How is the detection method modified to address each attack?

The multiple faked identities metric is a refined notion of friend similarity metric that considers fake identities forging the victim and clone. It provides a more accurate detection process by addressing the following attacks:

1. The first type of attack is when the faked profile builds a network of fake friends that are clones of the victim's friend. The more fake friends the clone obtains, the more confusing the network gets and the smaller the likely-hood of a real user correctly identifying the fake network.
2. The second type of attack considers similar identities in the friend network of the victim, but also mutual friends with the faked profile. With enough overlap between the networks of the victim and clone, it will be hard for others to tell which one is fake, and much easier for the clone to build his fake network.

This differs from basic friend similarity because the basic version doesn't consider the fake identity creating multiple fake accounts to clone the victim's friends. Another shortcoming in the basic metric is it doesn't look for similar friends in a profiles friends list. Due to the increased complexity of this more advanced method, and the sparse, unreliable nature of the given data set, we argue that using the multiple faked identity scheme would not yield a vastly different result.

Question 4

How do the attack models of [1] and [2] differ? Why are the detection methods proposed in [1] unsuitable for detecting the adversary considered in [2]?

Attack model [1] focuses on comparing the profile specifications of the victim with those of the clone. It doesn't consider the profile's evolution over time, but instead compares the actual values of the profile's attributes and friends at one point in time. Attack model [2], however, emphasizes detecting fake profiles based off of their friend accumulation rate through time and their friend node degree. Metric [2] considers users growth rate and size of friends list, while metric [1] compares the number of similar attributes and friends in each list including excluded and recommended lists. Metric [1] is not suitable for detecting the type of adversary mentioned in [2] because it is designed mostly to find fake users trying to clone victims in order to access victim's friends and spread advertisements. Metric [2], however, is focused on detecting profiles that may be adding random people in order to grow and share information.

Question 5

Section IV of [2] observes that the connected components of the social network graph can also be used to detect fake profiles. Suggest one possible detection metric that could be used to detect fake profiles based on the connected components of the graph.

The research paper [2] discusses two possible detection metrics. The first uses the average degree of nodes in the online social network (OSN) graph and the second uses the number of singleton friends in the OSN graph. Based off this, one possible detection could be the number of friends a user and the user's friends have in common. For example most real users friends tend to also be friends with that users friends. This could easily be computed through intersecting a users friends network with each of their friends network. If a user's interconnected friend graphs have few connections then that user most likely is randomly adding people. A friend graph will likely also have interconnected groups that may be somewhat isolated from each other. These isolated groups will most likely have common similar attributes such as location, work, or relationship. By combining a graph of interconnected friends with weighted attributes, real profiles will tend to be grouped based on those attributes and have a many interconnected similar friends. An example of this is shown in the figure below base on my Facebook data. Each color represent a different attribute such as work or location [Figure 8].

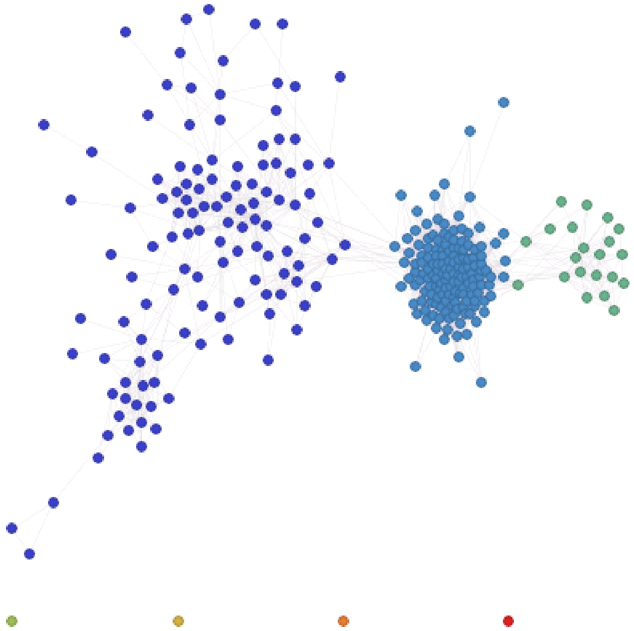

Figure 8: Example Detection Metric Using Connected Components of Graph

Question 6

Choose one of the detection metrics proposed in either [1] or [2], and describe one technique that an adversary could use to attempt to avoid this detection method.

Detection [2] would be easily avoided if the adversary had a friend growth rate that was within the bounds of α. Doing this would not flag the adversary as an outlier when compared to the average rate of friend accumulation. In addition the adversary would have to have a friend count near the average so it doesn't appear as an outlier when the total number of friends is considered. Following these simple steps, an adversary would likely fail to be detected by metric [2], and continues to haunt victims for all eternity.

References

1. Jin, L., Takabi, H., Joshi, J.: Towards active detection of identity clone attacks on online social networks.

2. Conti, M., Poovendran, R., Secchiero, M.: FakeBook: Detecting fake profiles in online social networks. IEEE/ACM International Conference on Advances in Social Networks Analysis and Data Mining (2012)