Finding Conspirators in the Network via Machine Learning

Fangjian Guo
Jiang Su
Jian Gao

Web Sciences Center
University of Electronic Science and Technology of China
Chengdu, Sichuan, China

Advisor: Tao Zhou

KEY CONCEPTS

Machine learning

Logistic regression

Semantic diffusion

Bipartite graph

Resource-allocation dynamics

Kendall's tau

Problem Clarification: A conspiracy network is embedded in a network of employees of a company, with each edge representing a message sent from one employee (node) to another and categorized by topics. Given a few known criminals, a few known non-criminals, and suspicious topics, we seek to estimate the probability of criminal involvement for other individuals and to determine the leader of the conspirators.

Assumptions

- Conspirators and non-conspirators are linearly separable in the space spanned by local features (necessary for machine learning).
- A conspirator is reluctant to mention to an outsider topics related to crime.
- Conspirators tend not to talk frequently with each other about irrelevant topics.
- The leader of the conspiracy tries to minimize risk by restricting direct contacts.
- A non-conspirator has no idea of who are conspirators, hence treats conspirators and non-conspirators equally.

The UMAP Journal 33 (3) (2012) 275–292.

KEY TECHNIQUES

Gradient Descent

Revised LeaderRank

Model Design and Justification For an unidentified node (an employee not identified as a conspirator or non-conspirator), we model the probability of conspiracy as a sigmoid function of a linear combination of the node's features (logistic regression). Those features are formulated from local topological measures and the node's semantic messaging patterns. Parameters of the model are trained on a subset of identified conspirators and non-conspirators. The performance of the model is enhanced by discovering potential similarities among topics via topic-word diffusion dynamics on a bipartite graph. We also perform resource-allocation dynamics to identify the leader of the conspirators; the identification is supported by empirical evidence in criminal network research.

Strengths and Weaknesses The combination of topological properties and semantic affinity among individuals leads to good performance. The time complexity of the algorithm is linear, so the method is suitable for large amounts of data. However, our model requires assistance from semantic network analysis to form an expert dictionary. Also, intrinsic differences among networks may hinder portability of the model's features.

Introduction

As shown in **Figure 1**, criminals and conspirators tend to form organizational patterns, interconnected with one another for collaboration, while still maintaining social ties with the outside, thus providing a natural context for description and analysis via networks [Baker and Faulkner 1993].

Criminal networks can be captured from various information, resulting in different types of networks, where each node represents a person, and an edge is present when two nodes collaborate in the same task, share the same family name, or (as in this case) exchange messages [Krebs 2002].

Since the nodes in the graph can be a mixture of both criminals and non-criminals, it is desirable to determine suspected criminals from topological properties of the network and other prior knowledge, which includes known criminals, known non-criminals, and information related to their interactions. Moreover, we desire a priority list of descending criminal likelihood so as to identify the primary leader of the organization.

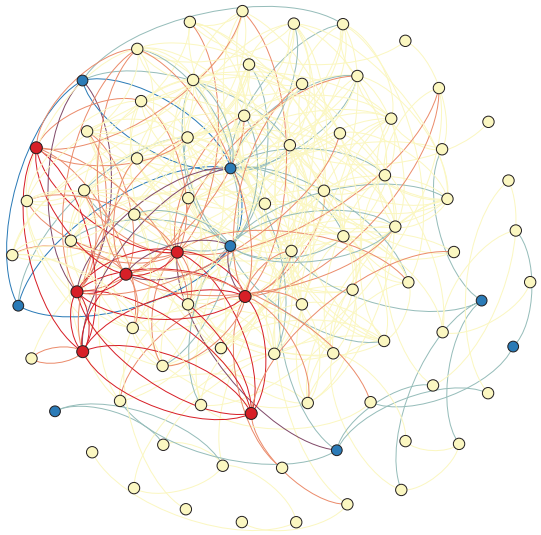

Figure 1. The 83-employee network. Red (darker gray) nodes are known conspirators and the blue (lighter gray) nodes are known non-conspirators.

Many authors have adopted centrality measures of the graph for analyzing the characteristics of criminals. Criminals with high betweenness-centrality are usually brokers, while those with high degree-centrality enjoy better profit by taking higher risk [Krebs 2002]. Morselli [2010] proposed that leaders of a criminal organization tend to balance profit and risk by making a careful trade-off between degree-centrality and betweenness-centrality.

However, centrality approaches, which utilize local properties, tend to overlook the complex topology of the whole network. Therefore, social network analysis (SNA) methods, including subgroup detection and block-modeling, have been introduced, which try to discover the hidden topological patterns by partitioning the big network into small closely-connected cliques [Xu and Chen 2005]. Despite the light that they shed on the internal structures of criminal networks, these methods still suffer from intimidating complexity with large databases [Wheat 2007].

We carefully combine the local-feature-based methods with approaches related to global topology of conspiracy networks. We propose a machine learning scheme to leverage local features, so as to estimate each node's likelihood of conspiracy involvement. We adopt dynamics-based methods, which are less computationally expensive than most other topology-based approaches, to help identify the lead conspirator and to discover semantic connections between topics.

We start with the formulation of useful local features of a node in the network, which then lead to the machine learning scheme. We feed a subset of known conspirators and non-conspirators as a training samples into the learning algorithm. We then use the algorithm to estimate the probability of being a conspirator for every other individual in the network.

Since highly suspicious topics are essential to the performance of machine learning, we then try to discover similarities between topics, by performing simple source-allocation dynamics on the bipartite semantic network made up of topics and sensitive words. Those findings expand our knowledge on suspicious topics, thus enhancing the accuracy of our machine learning model.

To find the leader of the conspirators, we apply a dynamics-based ranking algorithm on a subgraph extracted from the network. Our findings are in agreement with empirical knowledge about the centrality balance of criminal leaders.

Finally, we perform sensitivity analysis to test the robustness of our approach.

A Machine Learning Solution

We use machine learning mainly because of its adaptiveness and reorganization, which simulate humans' actions to obtain fresh knowledge.

We describe the construction of our machine learning framework in detail, including feature formulation, core learning methods, and experimental results. Through statistical analysis on the results, we propose an enhancement based on semantic diffusion.

We commence with several necessary assumptions:

- All the data and information about the EZ case network and the 83-node network are relatively stable over a long period.
- The contents of the communication among conspirators tends to be relevant about suspicious topics or some formal issues, rather than gossip.
- The two networks feature similar core mechanisms for communication transmission.

Feature formulation

- **Centrality**

  We exploit three types of centrality—degree centrality, betweenness centrality, and closeness centrality—to determine the center of the suspicious network from different aspects:

  - *Degree centrality.* Degree centrality [Freeman 1979] indicates activeness of a member, and a member who tends to have more links to others may be the leader. However, as explained in Xu and Chen [2003], degree centrality is not quite reliable to indicate the team leader in a criminal network. For a graph $G(V, E)$, the normalized degree centrality of node i is

$$C_D(i) = \frac{\sum_{j=1}^{|V|} \nu(i,j)}{|V|-1}, \qquad i \neq j, \tag{1}$$

where ν is a binary indicator showing whether there exists a link between two nodes. Since our graph is directed, we calculate separately the in-degree and out-degree of every node.

- *Betweenness centrality.* Betweenness centrality [Freeman 1979] describes how much a node tends to be on the shortest path between other nodes. A node with large betweenness centrality does not necessarily have large degree but illustrates the role of "gatekeeper"—someone who is more likely to be a intermediary when two other members exchange information. The normalized betweenness centrality is

$$C_B(i) = \frac{\sum_{j=1}^{|V|} \sum_{k<j}^{|V|} \omega_{j,k}(i)}{|V|-1}, \qquad k \neq i, \tag{2}$$

where $\omega_{j,k}(i)$ indicates whether the shortest path between node j and node k passes through node i.

- *Closeness centrality.* Closeness centrality [Sabidussi 1966] is usually utilized to measure how far away one node is from the others. Closeness of a node is defined as the inverse of the sum of its distances to all other nodes and can be treated as a measure of efficiency when spreading information from itself to all other nodes sequentially. It indicates how easily an individual connects with other members. The normalized closeness centrality is

$$C_c(i) = \frac{\sum_{j=1}^{|V|} \rho(i,j) - C_{c\min}}{C_{c\max} - C_{c\min}}, \qquad i \neq j, \tag{3}$$

where $\rho(i,j)$ is the length of the shortest path connecting nodes i and j. $C_{c\min}$ and $C_{c\max}$ are the minimum and maximum lengths of the shortest paths.f

- **Number of known neighboring conspirators**
We consider as a significant feature the number of known neighboring conspirators of a node. The interaction among conspirators in a message network suggests a much stronger connectivity than that among non-conspirators: A conspirator is more likely to communicate with an accomplice. As shown in **Figure 2**, we calculate the ratio of known conspirators among one's adjacent neighbors, which measures proximity with known accomplices: The value is 1 if the individual connects with all the known conspirators, and 0 means that no conspirators connect to the individual. The known suspicious clique obviously represents a more compact connectivity. Therefore, the more known conspirators among

an individual's neighbors, the greater the possibility that the individual is an accomplice.

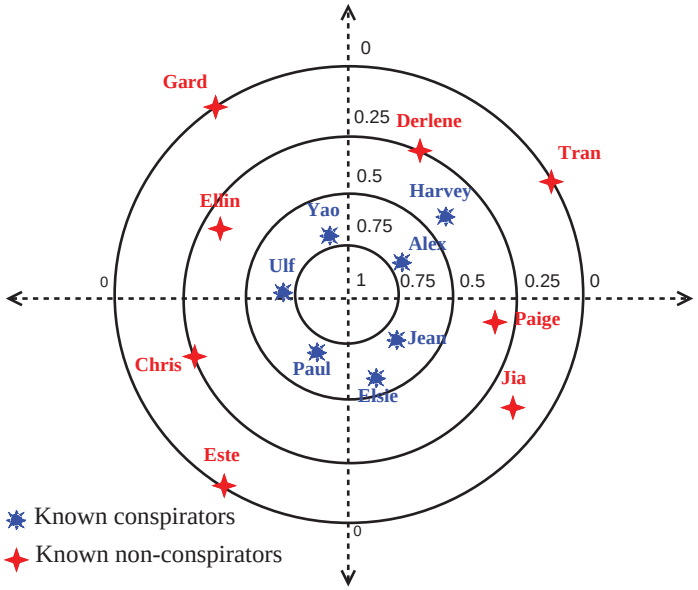

Figure 2. Ratio of known conspirators among adjacent neighbors. To avoid the overlapping of names with a linear scale, we adopt a topographic map type of diagram, with a peak at the center and symmetric contour circles around it. The closer a person is to the center, the more likely that the person is a conspirator.

- **Number of current non-suspicious messages from known conspirators**
  Table 1 shows the topics mentioned between known conspirators.[1] A known conspirator rarely talks with accomplices about topics irrelevant to their conspiracy, though a very small proportion of unknown topics appear. If most of the information received from a known conspirator is irrelevant, the receiver is probably not a conspirator.

Table 1.

Topics among known conspirators. Known conspiratorial topics have an asterisk and are highlighted in blue (light gray).

	Jean	Alex	Elsie	Poul	Ulf	Yao	Harvey
Jean		**11***			**8**		**14**
Alex			**1**	**13***	**11***	**3**, **7***	
Elsie		**11***			**13***		
Poul	**11***		**7***		**7***		**4**
Ulf		**7***, **11***, **13***				**13***	
Yao	**13***	**7***, **11***, **13***	**7***, **9**		**13***		**2**, **7***
Harvey						**13***	

[1]Topic 16 in the raw data is regarded as wrong and thus discarded.

Methods

We use logistic regression to model the probability of a node being involved in the conspiracy. We obtain the parameters of the model by using a gradient descent algorithm to solve an optimization problem on a training set.

Logistic Regression

We consider a training set $\{(x^{(i)}, y^{(i)})\}$ of size m, where $x^{(i)}$ is an n-dimensional feature vector and $y^{(i)}$ indicates the classification of the node, i.e., $y^{(i)} = 1$ for conspirators and $y^{(i)} = 0$ for non-conspirators. The nodes in the training set are drawn from the 15 known conspirators and non-conspirators.

As a specialization of a generalized linear model for Bernoulli distribution, logistic regression estimates the probability of being a conspirator as

$$P(y = 1|x; \theta) = h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}, \tag{4}$$

where $\theta \in \mathbb{R}^n$ is the parameter vector.

Under the framework of the generalized linear model, the *maximum a posteriori* (MAP) estimate of the parameter θ is

$$\min_\theta J(\theta), \tag{5}$$

where the cost function is given by

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left[-y^{(i)} \log \left(h_\theta(x)^{(i)} \right) - \left(1 - y^{(i)} \right) \log \left(1 - h_\theta(x^{(i)}) \right) \right] + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2, \tag{6}$$

with λ being a regularization parameter.

Gradient Descent

The cost function $J(\theta)$ is minimized by gradient descent, which drives θ down the locally steepest slope, in hope of reaching the global minimum of the cost function.

At every iteration before convergence, a new θ replaces the old θ via

$$\theta := \theta - \alpha \nabla_\theta J(\theta), \tag{7}$$

where α is a small positive constant.

Leave-One-Out Cross Validation

Since we are informed of the correct classification of only N_0 nodes ($N_0 = 15$ in our case), in a given round we only use ($N_0 - 1$) of them as the training set, while leaving one out for cross validation (C-V). At every round, the next correctly classified node is left out and the others serve as the training set; then the trained hypothesis is tested on the left-out node. In this way, by averaging N_0 rounds without overlapping, the errors for both the training set and the cross validation set can be evaluated.

Suppose, for example, that in the j-th round sample $(x^{(j)}, y^{(j)})$ is left out and the training set is given by

$$S_j = \{(x^{(l)}, y^{(l)}) \mid l = 1, 2, \cdots, j-1, j+1, \cdots, N_0\}. \tag{8}$$

Using this training set, parameter vector $\theta^{(j)}$ is obtained, and the corresponding hypothesis is tested on both S_j and the left-out $(x^{(j)}, y^{(j)})$, obtaining this round's training error ε_{S_j} and C-V error ε_j.

Hence, by averaging over j, the training error and C-V error are

$$\varepsilon_S = \frac{1}{N_0}\sum_{j=1}^{N_0} \varepsilon_{S_j}, \qquad \varepsilon = \frac{1}{N_0}\sum_{j=1}^{N_0} \varepsilon_j. \tag{9}$$

Setting the Regularization Parameter

The regularization parameter $\lambda > 0$ is selected to minimize the cross validation error, i.e.,

$$\lambda = \arg\min_{\lambda > 0} \varepsilon. \tag{10}$$

Results

By training the logistic regression model with our leave-one-out cross validation strategy, λ is optimally set to 1.9 and the overall C-V error is $\varepsilon = 0.27$ (with training error $\varepsilon_S = 0$). Then, while fixing the chosen λ, we retrain the hypothesis on the maximum training set, making full use of known conspirators and non-conspirators.

Table 2.

Top 10 in the priority list (known conspirators excluded).

Name	Dolores*	Crystal	Jerome*	Sherri	Neal	Christina	Jerome	William	Dwight	Beth
Node No.	10	20	34	3	17	47	16	50	28	38
Probability of conspiracy	.56	.51	.39	.32	.30	.27	.25	.25	.24	.23

The trained hypothesis gives the estimated probability for node i being a conspirator, resulting in a priority list of suspects, ranked in descent order of criminal likelihood. The top 10 suspects are shown in **Table 2**, with managers marked by an asterisk.

Figure 3 illustrates the probability of criminal involvement estimated by $h_\theta(x)$ versus the corresponding rank in the priority list, where three managers (Jerome, Dolores, and Gretchen)[2] are marked by circles.

Dolores (manager) is indeed the person deserving highest suspicion, and Jerome (manager) is also likely to be involved in conspiracy.

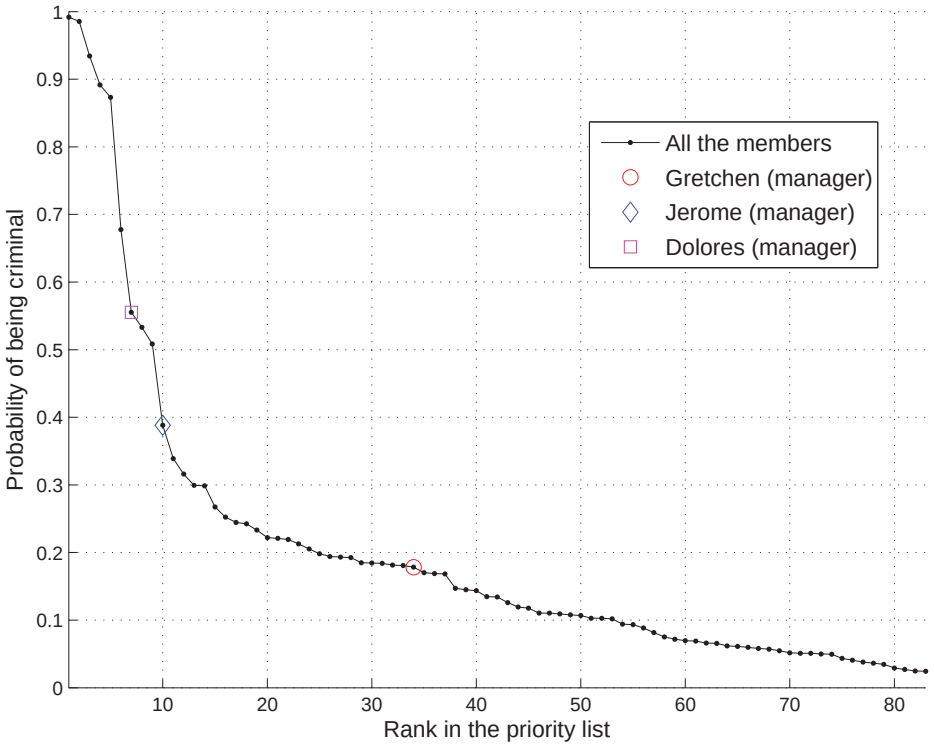

Figure 3. Probability of conspiracy vs. corresponding rank in the priority list

Semantic Model Enhancement

Semantic information is more important to humans than the complicated topology structure. For example, similar text information in their messages motivates us to conclude in the EZ case that Inez is similar to George, who is definitely a conspirator (for instance,"tired" when describing Inez and "stressed" when describing George). Similar cases can be also found in the 83-people network case: The word "Spanish" from known conspiratorial topic 7 is highly suspicious and appears repeatedly in other unknown topics (e.g., topic 2 and 12). The contents about "computer security," which is

[2]Since several nodes are named either Gretchen or Jerome, we select those with bigger out-degrees to be managers, that is, Node 32 is Gretchen (manager) and Node 34 is Jerome (manager).

treated as part of the key in the whole conspiracy, is also active in many other unknown topics, such as 5 and 15. Hence, it is natural to train a computer to measure similarities among topics so as to reveal potential information.

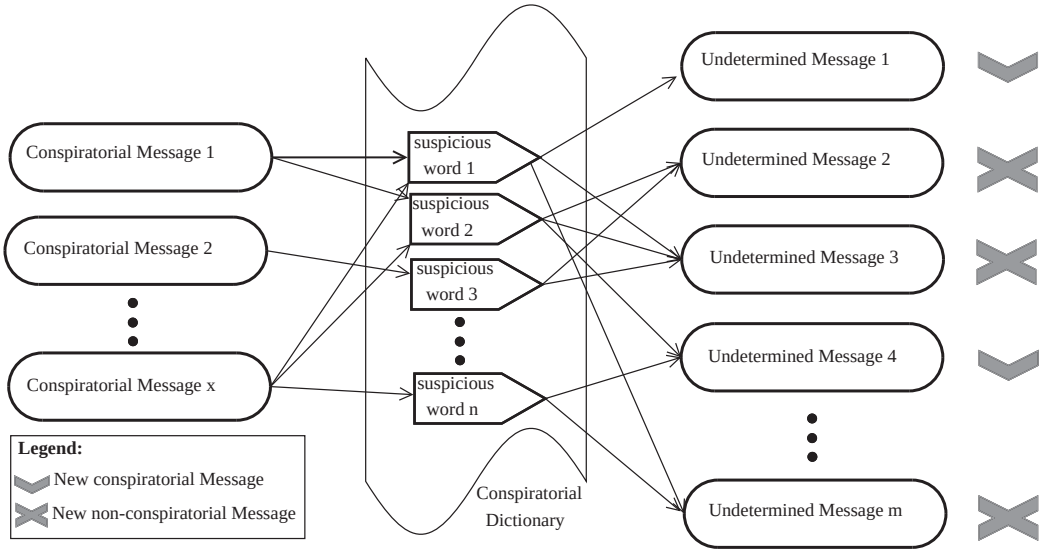

Figure 4. Framework of topic semantic diffusion.

Lexical ambiguity exists widely among words, which can have different meanings depending on context. So it is not wise to abandon human intelligence and depend only on algorithms. Therefore, we draw the problem of topic semantic diffusion into a topic-similarity measurement task based on an expert dictionary. We form the bipartite network illustrated in **Figure 4**, between the conspiratorial dictionary constructed from the conspiratorial messages about known suspicious topics, and all of the information in the message traffic. We exploit a resource allocation mechanism to extract the hidden information of networks [Zhou et al. 2007] and unfold the similarity among different topics.

The bipartite network is modeled as the bipartite graph $G = (D, T, E)$, where

- $D = \{d_i\}$ is the dictionary of suspicious words, shown in the middle column in **Figure 4**;
- $T = \{t_l\}$ is the message set, which is divided into two categories:
  - messages with known status (left column in **Figure 4**), and
  - undetermined messages (right column in **Figure 4**);
- E is an edge set, with $(d_i, t_l) \in E$ indicating that word d_i in the conspiratorial dictionary D occurred in message t_l of the message set T;

We initially give 1 unit of resource to each known conspiratorial message in T and 0 to the remaining messages. The process of semantic diffusion

includes two steps, namely the redistribution of resource from message topics to keywords, and that from keywords back to topics.

We commence with the first allocation from set T to set D:

$$f(d_i) = \sum_{i=1}^{n} \frac{a_{il} f(t_l)}{K(t_l)}. \tag{11}$$

Equation **(11)** expresses the calculation of the resource held by t_l after the first step, where $K(t_l)$ denotes the degree of the node t_l, $f(x)$ denotes the resource carried by x, and a_{il} is defined as

$$a_{il} = \begin{cases} 1, & (d_i, t_l) \in E; \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

The intuitive explanation of Step 1 is simply the process of redistributing resource from T to D, with the transferred amount regulated by the degree of nodes in T.

This is followed by Step 2, which is to reflect the resource flow back to T from D obeying the same rule but in the inverse direction, as shown from the middle column to the right column in **Figure 4**. So the resource finally locates on t_i and satisfies

$$f^{'}(t_i) = \sum_{l=1}^{m} \frac{a_{il} f(d_l)}{K(d_l)} = \sum_{l=1}^{m} \frac{a_{il}}{K(d_l)} \sum_{j=1}^{n} \frac{a_{ji} f(t_j)}{K(t_j)}. \tag{13}$$

After this two-fold process, the amount of resource held by every element in T can be interpreted as a score of similarity. The rank of a topic according to its score represents the degree of its similarity to the information in the dictionary—that is, the higher the score, the more likely the topic is a newly-found conspiratorial topic.

We set $D = \{$'Spanish', 'system', 'network', 'computer', 'meeting'$\}$ as the conspiratorial dictionary, and **Table 3** illustrates the final result for all 15 topics in the 83-people network case. The known suspicious topics are 7, 11, and 13. They are naturally the top three, and topic 5 is more suspicious than other unknown topics. Topics 2, 12, and 15 are among the group with the second highest possibility in unknowns; and the remaining topics tend to be irrelevant to the conspiracy.

We then append topic 5 to the set of known conspiratorial topics and train the model again; the overall C-V error decreases from 0.27 to 0.13. Since Since topics 2, 12, and 15 are less similar to known suspicious topics, as shown in **Table 3**, appending them to model training does not evidently influence the correctness. The enhanced correctness here indicate that with enough keywords in the conspiratorial dictionary and enough topics with abundant contents, such a method is likely to perform very well.

On the other hand, if we utilize the speaker instead of the keywords to construct a bipartite graph with the topics, we will also get similarity

Table 1.
Rank of all topics based on similarity to known suspicious topics (known conspiratorial topics have an asterisk and are highlighted in blue).

Rank	Topic Number	Similarity to known suspicious topics
1	**11***	0.750
2	**7***	0.667
3	**13***	0.667
4	5	0.417
5	2	0.167
6	12	0.167
7	15	0.167
8	1,3,4,6,8,9,10,14	0

among topics based on the transmitting speaker. However, the determination of the relationship between differing results under these two standards is definitely beyond this paper.

The resource allocation method is also highly efficient: Its time complexity of computation is linear in the number of edges of the graph, which enables good performance with huge amounts of data.

Identifying the Leader of the Conspiracy

Our machine learning scheme tries to estimate the likelihood of a node committing conspiracy. However, the likelihood does not proportionally indicate leadership inside the network, because the identification of leaders is further complicated by the network's topology.

We adopt LeaderRank, a node-ranking algorithm closely related to network topology, to find the leader. We extract from the network the subgraph connected by known suspicious topics. Because of its robustness against random noise, LeaderRank is appropriate for addressing criminal network problems, which usually suffer from incompleteness and incorrectness.

LeaderRank

The LeaderRank algorithm is a state-of-the-art achievement on node ranking that is more tolerant of noisy data and robust against manipulations than traditional algorithms such as HITS and PageRank [Lü et al. 2011]. This method is mathematically equivalent to a random-walk mechanism on the directed network with adaptive probability, leading to a parameter-free algorithm readily applicable to any type of graph.

We insert a ground node, which connects with every node through newly-added bidirectional links, in order to make the entire network strongly connected, so that the random walk will definitely converge.

For a graph $G = (V, E)$, every node in the graph obtains 1 unit of resource except the ground node. After the beginning of the voting process, node i at step t will get an adaptive voting score $\nu(t)$ according to the voting from its neighbors:

$$\nu_i(t+1) = \sum_{j=1}^{|V|+1} \frac{\mu_{ij}}{D_{\text{out}}(j)} \nu_i(t), \tag{14}$$

where μ_{ij} is a binary indicator with value 1 if node i points to j and 0 otherwise. $D_{\text{out}}(j)$ denotes the out-degree of node j. The quotient of the above two arguments could be considered as the probability that a random walker at i goes to j in the next step. Finally, the leadership score of node i is $\nu_i(T_c) + \nu_{\text{gn}}(T_c)/|V|$, where $\nu_{\text{gn}}(T_c)$ is the score of the ground node at steady state.

Suspicious Topic Subnetwork Extraction

We extract from the network of company employees the subnetwork G_{T_S} connected by suspicious topics only, so as to minimize the coupling of the company's hierarchical structure to the conspiracy relations.

Suppose that T_{ij} denotes the set of topics mentioned by messages from node i to node j, and T_S denotes the set of known suspicious topics ($T_S = \{7, 11, 13\}$). Then G_{T_S} is the maximum subgraph of the original graph G, whereas

$$T_{ij} \subseteq T_S, \text{for all } (i, j) \subseteq E_{T_S}. \tag{15}$$

Edge Reverse

The original LeaderRank algorithm deals with finding leaders in Internet social networks, where the direction of an edge has a dissimilar meaning from our case: If A points to (follows) B in Twitter, then B is considered to be a leader of A. However, in our communication network, an edge from A to B suggests that A has sent B a message. Therefore, assuming that a leader in a criminal network tends to be a sender rather than a receiver, each edge in G_{T_S} has to be reversed to be compatible with LeaderRank's design. We denote by G'_{T_S} the reversed subnetwork induced by suspicious topics.

Results

By running LeaderRank on G'_{T_S}, a ranking score is assigned to every node in this subgraph, which generates a list of possible leaders ranked in descent order, as shown in **Table 4**.

Yao (node number 67) is ranked as the chief leader of the conspiracy.

Table 4.
Partial results of LeaderRank on G'_{T_S}.

Name	LeaderRank score
Yao	2.67
Alex	2.21
Paul	1.92
Elsie	1.62

Empirical Support

Empirical analysis of criminal networks finds that a leader of a criminal organization tends to carefully balance degree-centrality and betweenness-centrality. It has been proposed that the leader usually maintains a high betweenness-centrality but a relatively low degree-centrality, for enhancing efficiency while ensuring safety [Morselli 2010].

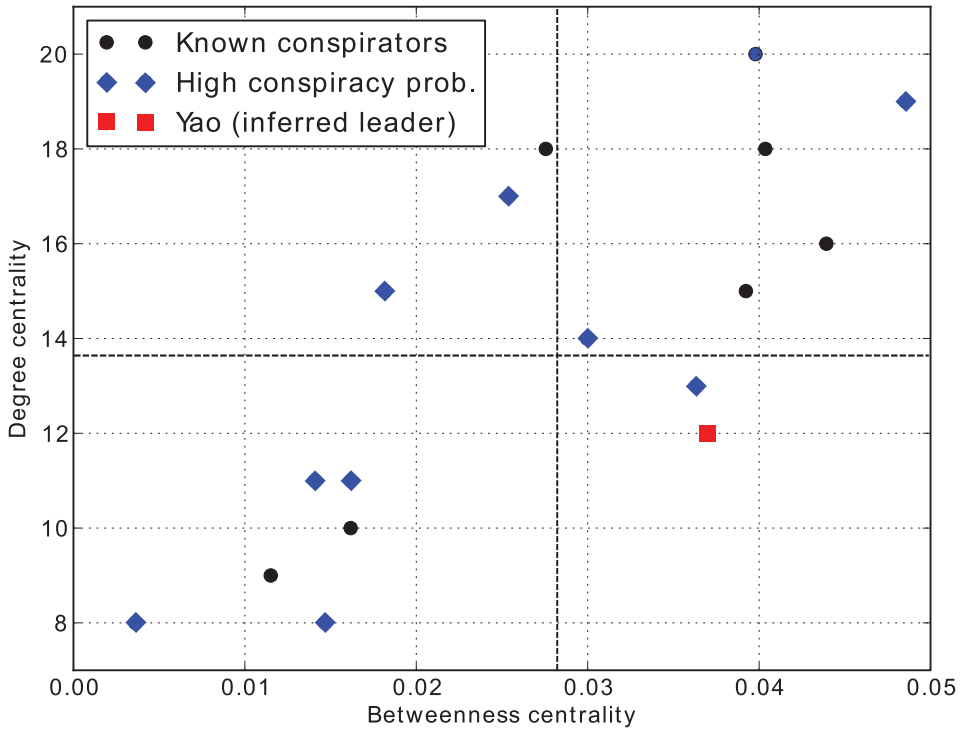

Figure 5. The joint distribution of betweenness centrality and degree centrality. Yao is at the lower right.

Figure 5 illustrates the joint distribution of betweenness centrality C_B and degree centrality ($D_{\text{in}} + D_{\text{out}}$) for the 7 known conspirators and 10 other nodes with high conspiracy likelihood, where two dashed lines mark average values of the displayed nodes. Yao's high betweenness-centrality with relatively low degree-centrality accord with the identity of a leader. Our conclusion that Yao is the leader is thus empirically supported.

Discussion

We identify the leader of the criminal network by performing the LeaderRank algorithm on the extracted, edge-reversed, suspicious-topic-connected subgraph; and our findings are strengthened by empirical research results.

Evaluating the Model

Sensitivity Analysis

Considering the usual incompleteness, imprecision, and even inconsistency in mapping criminal social networks [Xu and Chen 2005], the method for deducing criminality should be robust enough to cope with minor alternations of the network. Otherwise, there could be mistaken accusations. Therefore, we perform a sensitivity analysis on our approach.

Requirement 2 of the problem statement provides an appropriate scenario for such a test: While other conditions remain unchanged, new information indicates that Topic 1 is also connected to criminal activity, and Chris, who was considered innocent before, is now proven guilty.

Priority List

By applying our methods to these altered conditions, we find that among the top 10 of the previous priority list (the 7 known conspirators excluded), 7 of them are still in the new top 10, while the remaining 3 find their new places at 12th, 14th, and 16th.

A more sophisticated measurement of the sensitivity of the priority list is *Kendall's tau* coefficient τ [Sen 1968]. Given two priority lists $\{p_k\} = \{p_1, p_2, \cdots, p_n\}$ and $\{q_k\} = \{q_1, q_2, \cdots, q_n\}$—for example, $p_2 = 5$ means node 2 is ranked 5th in the $\{p_k\}$ list—then

- (i, j) (for $i \neq j$) is a *concordant pair* if their relative rankings agree in the two lists, i.e., $p_i > p_j$ and $q_i > q_j$, or $p_i < p_j$ and $q_i < q_j$;
- otherwise, if $p_i < p_j$ but $q_i > q_j$, or $p_i > p_j$ but $q_i < q_j$ (i, j) is a *discordant pair*.

Kendall's tau is defined as

$$\tau = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{\frac{1}{2}n(n-1)}, \quad \textbf{(16)}$$

which lies in $[-1, 1]$, with 1 for perfect ranking agreement and -1 for utter disagreement.

The value of *Kendall's tau* for the two priority lists of Requirement 1 and Requirement 2 is $\tau = 0.86$, justifying the robustness of the machine learning approach.

Let us assume that known conspirators and non-conspirators are independently wrongly classified with the same specific probability. **Figure 6** depicts the expected Kendall's tau vs. the misclassification probability, separately for conspirators and non-conspirators. Even if the misclassification probability is as high as 0.5, Kendall's tau does not drop below 0.8, substantially proving the inherent stability of our methods.

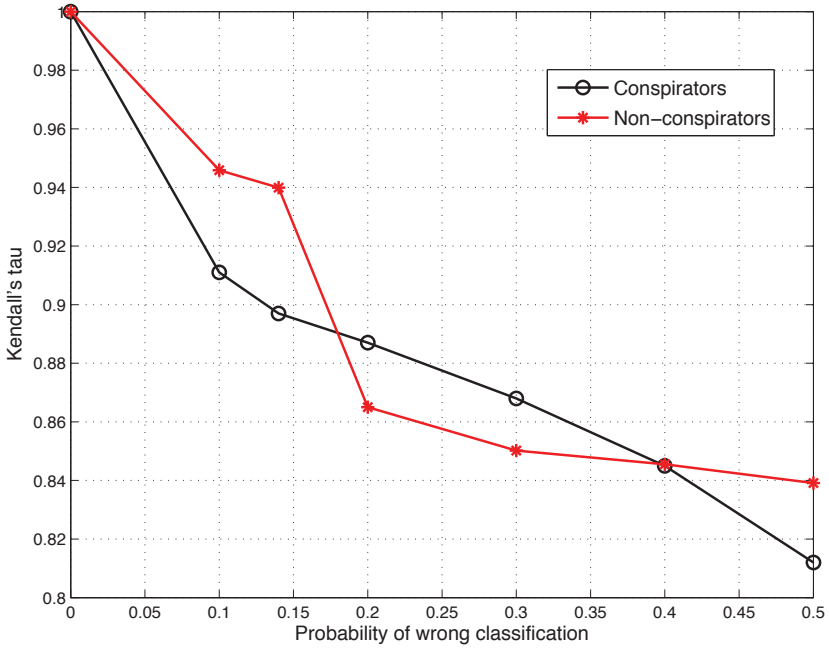

Figure 6. The expected Kendall's tau declines as misclassification probability increases.

Probability Inflation

Figure 7 illustrates the change of estimated conspiracy probability due to the modified conditions of Requirement 2, with the previous value as x-axis, and the new as y-axis. Generally, most nodes exhibit a small "inflation" in criminal probability, as indicated by the distribution of dots skewed from the diagonal line. The augmented probability is compatible with the new information that expands both the set of suspicious topics and known conspirators.

The analysis suggests that our machine learning method is insensitive to minor alterations and can still produce reasonable results with new information.

References

Baker, Wayne E., and Robert R. Faulkner. 1993. The social organization of conspiracy: Illegal networks in the heavy electrical equipment industry.

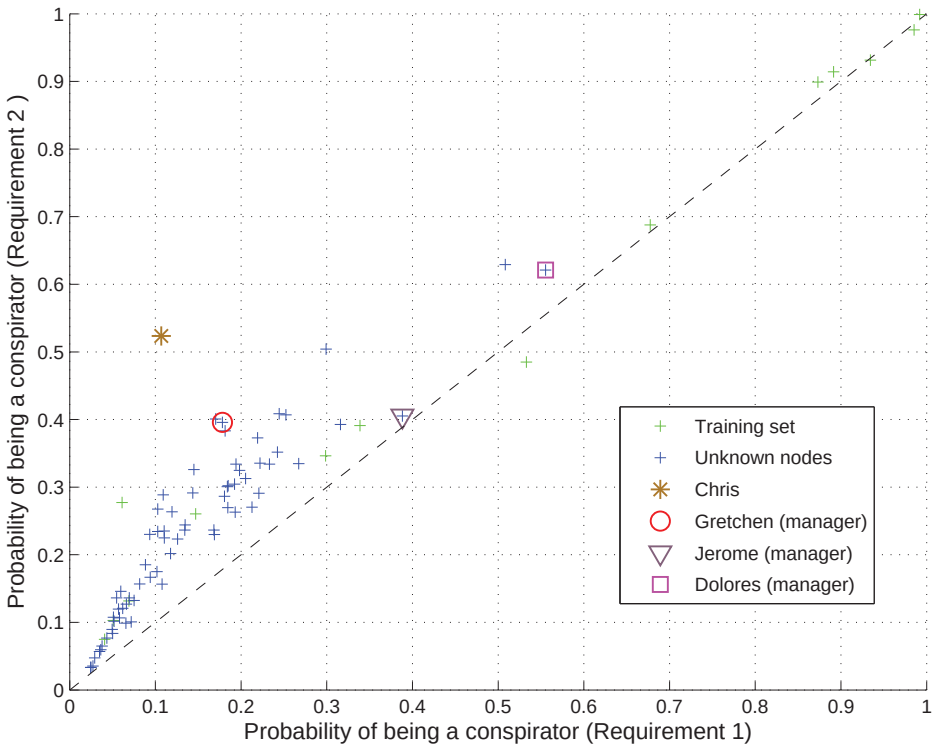

Figure 7. Criminal probabilities before and after the change of conditions.

American Sociological Review 58 (6) (December 1993): 837–860. http://webuser.bus.umich.edu/wayneb/pdfs/networks/Conspiracy.pdf.

Chen, Hao, and Burt M. Sharp. 2004. Content-rich biological network constructed by mining PubMed abstracts. *BMC Bioinformatics* 5: 147. http://www.biomedcentral.com/1471-2105/5/147, doi:10.1186/1471-2105-5-147 .

Freeman, Linton C. 1979. Centrality in social networks conceptual clarification. *Social Networks* 1 (3) (1978–1979): 215–239. http://psyonline.com.br/portal/administrator/components/com_jresearch/files/publications/freeman.pdf .

Girvan, M., and M.E.J. Newman. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99 (12) (11 June 2002): 7821–7826. http://techlab.bu.edu/members/gail/710Girvan,Newmann2002.pdf, doi:10.1073/pnas.122653799 .

Krebs, Valdis E. 2002. Mapping networks of terrorist cells. *Connections* 24 (3): 43–52. http://vlado.fmf.uni-lj.si/pub/networks/doc/Seminar/Krebs.pdf .

Lü, Linyuan, Yi-Cheng Zhang, Chi Ho Yeung, and Tao Zhou. 2011. Leaders in social networks, the *delicious* case. *PloS ONE*, 6 (6):

e21202. http://www.plosone.org/article/info:Adoi/10.1371/journal.pone.0021202, doi:10.1371/journal.pone.0021202.

Morselli, Carlo. 2010. Assessing vulnerable and strategic positions in a criminal network. *Journal of Contemporary Criminal Justice* 26 (4) (September 2010): 382–392. http://ccj.sagepub.com/content/26/4/382.short, doi:10.1177/1043986210377105.

Sabidussi, Gert. 1966. The centrality index of a graph. *Psychometrika* 31 (4): 581–603.

Sen, Kumar Pranab. 1968. Estimates of the regression coefficient based on Kendall's tau. *Journal of the American Statistical Association* 63 (December 1968): 1379–1389.

Wheat, Christopher. 2007. Algorithmic complexity and structural models of social networks. http://scripts.mit.edu/~cwheat/research/modelsel.20070416.

Xu, Jennifer, and Hsinchun Chen. 2003. Untangling criminal networks: A case study. In *Intelligence and Security Informatics: Lecture Notes in Computer Science 2665*, edited by G. Goos, J. Hartmanis, and J. van Leeuwen, 232–248. New York: Springer, 2003.

_____. 2005. Criminal network analysis and visualization. *Communications of the Association for Computing Machinery* 48 (6) (June 2005): 100–107.

Zhou, Tao, Jie Ren, Matúš Medo, and Yi-Cheng Zhang. 2007. Bipartite network projection and personal recommendation. *Physical Review E* 76 (4): 046115. http://doc.rero.ch/lm.php?url=1000,43,2,20071213113651-JT/zhang_bnp.pdf.

Jiang Su, Jian Gao, Tao Zhou (advisor), and Fangjian Guo.