Summary

We model the problem as classifying similar nodes in weighted directed graph, in which the topics are assigned particular weight and the employees participating the network are treated as nodes with specific attribute vectors. Our goal is to assign weights to topics, calculating the attribute vectors of the nodes based on the structure of the graph and the weights of the topics. We then classify those vectors into clusters to identify potential conspirators.

Our model – the "Eagle Eye" is a flow of operations including weight assignment, nodes proximity estimation over graphs and the classification process. Each of them is based on mature existing general purpose algorithms. In the main line of the flow, we use Analytical Hierarchical Process for weights assignment. We generate the attribute vectors based on the Cycle-Free Effective Conductance with the known (non)conspirators and then use k-Means in the classification step based on those vectors.

We've evaluated our model thoroughly and carefully by conducting a series of experiments including quest for the answer, sensitivity analysis, optimization approach evaluation and comparison with other similar purpose models. In the EZ case, we successfully identified Ellen and eliminated the possibility of Carol of being part of the conspiracy. While in the test involving 83 people, we've successfully prioritized the employees and among the leaders, we point our suspicion towards Delores. In the sensitivity evaluation, we've tested our model over noisy data sets with randomized (non)conspirators and obtained stable error rate. The sensitivity analysis also indicates that our model is relatively sensitive to the weights assigned to each topic. A bunch of methods of adapting our model to larger data sets are reasoned in the paper, apart from the technique to limit the depth of the search tree that is already implemented. In order to verify our results, implemented the Crime Syndicate Recognition System [10] to compare the results of our model and theirs and our results proved to be satisfactory. We also discussed the possibility of our model for applications in other fields such as social network recommendation and infection detection.

Eagle Eye over the Net: An Effective Way of Detecting Conspiracy Using Weighted Network

ICM Contest Question C

Team # 14198

Contents

1 Introduction **2**

2 Problem Formulation **2**
2.1 Problem Restatement . . . 2
2.2 Terminology, Definitions and Data . . . 3

3 Preliminaries **4**
3.1 The Cycle-Free Effective Conductance . . . 4
3.2 Cluster Analysis . . . 4
3.3 Analytic Hierarchy Process . . . 5

4 The Solution **6**
4.1 Model Overview . . . 6
4.2 Assigning Weight to the Topics . . . 6
4.2.1 Assigning weight with simple function . . . 6
4.2.2 Assigning weight with Analytic Hierarchy Process . . . 7
4.3 Obtaining the Attribute Vector . . . 8
4.4 Classifying the employees . . . 9

5 Evaluations **10**
5.1 Model Results . . . 10
5.2 Compare the methods for Assigning Topic Weight . . . 12
5.3 Comparison with Existing Solution . . . 12
5.3.1 Crime Syndicate Recognition System Overview . . . 12
5.4 Sensitivity Analysis . . . 14
5.5 Optimizations . . . 17
5.5.1 Limiting the Depth of the BFS Search Tree . . . 17
5.6 Future Optimizations . . . 18
5.6.1 Natural Language Processing for Message Analysis . . . 18
5.6.2 Reducing Dimension of Attribute Vector with Principal Component Analysis . . . 18
5.6.3 Updating the Weights with Genetic Algorithm . . . 18

5.7 Potential Applications 19

6 Conclusion **19**

1 Introduction

Advances in technology, communications, transportation, and economies enable a highly adaptive criminal element to hide in plain sight. They provide conspiracy, i.e. criminal networks, with more opportunities to threaten the public profits and national security in the dark. Knowledge about the structure and organization of criminal networks is important for both crime investigation and the development of effective strategies to prevent crimes. However, criminal network analysis remains a primarily tough process.

Generally speaking, the most effective weapon against the conspiratorial crimes is the ability to integrate information from multiple crime incidents and information flow in the criminal networks. Usually, we can get a large set of contact messages, where the vital clues lie around, as in the case of our problem in hand.

However, due to the incompleteness, incorrectness, and inconsistency, that exist in the criminal network, it is challenging to acquire valuable information, as is mentioned in [17], which also demonstrates that problems specific to the criminal network analysis lie in data transformation, fuzzy boundaries, and network dynamics.

To untangle and disrupt criminal networks, we need both reliable data and sophisticated techniques, which are all indispensable. In [18], the authors developed a system called *CrimeNet Explorer* that incorporates several advanced techniques: a concept space method, hierarchical clustering, social network analysis methods, and multidimensional scaling. These techniques serve as good heuristics for the development of our method.

In this work, we present "Eagle Eye" — the conspiracy detection system over the message network. Our model consists of the proximity measuring technique using Cycle-Free Effective Conductance, weighing the messages with Analytic Hierarchy Process and classifying the possible (non)conspirators using k-Means Clustering. Finally we evaluate our model in various aspects and compare our model with another known implementation.

The residual of this paper will be arranged as follows: Section 2 formulates the problem, giving our key terminology, definitions and assumptions. Section 3 gives the preliminary techniques. The ideas of the whole model is presented in Section 4. Section 5 is devoted to the evaluations of our model, in which we offer the whole detailed process involving the data from both a smaller EZ case and a larger one. We also analyze its sensitivity, compare it with another known technique, propose and reason about some possible optimizations. Also presented in the section is the discussion on other potential applications of our model.

2 Problem Formulation

2.1 Problem Restatement

At the moment, we are supposed to investigate and identify a conspiracy. There is a group of 83 people in a rapidly growing company, which is making a name for itself in developing and marketing computer software. Among these people, it is believed that there is a conspiracy

threatening the profits of company. The investigators have retrieved a set of messages from these people, including the source and the destination, to assist the investigation; so far, they believed that they have identified several conspirators as well as innocent employees.

The set of messages is divided into 15 categories, 3 of which are suspicious. The detailed information, including the source, destination, and the topic category, is attached to each message.

Given all the 83 staff, 7 identified conspirators, 8 known innocent employees, and 400 messages transmitted among them, we are expected to find out other unidentified co-conspirators and unknown leaders, and to address the following goals:

- Goal 1: Prioritize the 83 employees by likelihood of being part of the conspiracy with our model and algorithm.
- Goal 2: The model must be able to adapt to different conspiracy condition, thus can be a valuable general solution on problem this type.
- Goal 3: Evaluate the model thoroughly, analyze its sensitivity and the probability of its future optimization and application.

2.2 Terminology, Definitions and Data

We model the employee communication network as directed graph $G = (V, E)$. Each individual is represented as a vertex $v \in V$, and each message transmitted between node i and j, is denoted as a directed edge $e_{ij} \in E$.

- topic weight: a metric defined to represent the suspicion scale of each topic, thus each edge carries a weight $w_{e_{ij}}$.
- node proximity: a metric defined to describe the similarity of two nodes, and the node proximity between node i and j is represented by Cycle-Free Effective Conductance as np_{ij}, see Section 3.
- conspiracy tendency: a metric to describe the likelihood to be conspirators.

We also make following assumptions to correct some errors and irregularities in the attachment.

1. There are two employees named Elsie. We use the 37th node as Elsie.
2. There are two employees named Gretchen. We use the 4th node as Gretchen.
3. There is no employee named Delores. We use the 10th node named Dolores as Delores.
4. There is a message with topic category 18. We omit it.

3 Preliminaries

3.1 The Cycle-Free Effective Conductance

In this work, we adopt the algorithm called Cycle-Free Effective Conductance (CFEC) presented originally in [8]. It is an improved version of Effective Conductance (EC) [1,4].

The Effective Conductance is an intuitive (yet practical and widely used) approach for measuring centrality in social networks [2] as well as for study of connection in sub-graphs [5]. The EC of graph G between nodes s and t is denoted by $EC(s,t)$. Let $P_{esc}(s \to t)$ denotes the *escape probability*, the probability that a random walk starting at s reaches t before returning to s, and deg_i denotes the *degree* of node i. According to [1], $EC(s,t)$ can be expressed as:

$$EC(s,t) = deg_s \cdot P_{esc}(s \to t) = deg_t \cdot P_{esc}(t \to s) \tag{1}$$

Although EC has such advantage that it takes in both distance and number of alternative paths, its monotonicity property might get undesirable in cases involving degree-1 nodes.

Thus, the more effective CFEC measurement has been proposed in [8]. The probability of transition from node i to j is $p_{ij} = \frac{w_{ij}}{deg_i}$. Given a path $P = v_1 \to v_2 \to v_3 \to \cdots \to v_r$, the probability that a random walk starting at v_1 will follow this path is given by:

$$Prob(P) = \prod_{i=1}^{r-1} \frac{w_{v_i v_{i+1}}}{deg_{v_i}} \tag{2}$$

Also, the *weight of path P* is defined as:

$$Wgt(P) = deg_{v_1} \cdot Prob(P) \tag{3}$$

The *cycle-free escape probability* $P_{cf.esc}(s \to t)$ from s to t is the probability that a random walk starting from s will reach t without visiting any node more than once.

Accordingly, the *cycle free effective conductance* is defined as:

$$EC_{cf}(s,t) = deg_s \cdot P_{cf.esc}(s \to t) \tag{4}$$

Additionally, let $\mathcal{R}$ be the set of simple paths from s to t:

$$P_{cf.esc}(s \to t) = \sum_{R \in \mathcal{R}} Prob(R) \tag{5}$$

Multiplying the degree:

$$CFEC(s,t) = \sum_{R \in \mathcal{R}} Wgt(R) \tag{6}$$

3.2 Cluster Analysis

Cluster analysis creates groups, or clusters, of data, which can be used to identify the co-conspirators and innocent employees with their attribute matrix in hand.

Currently, there are several clustering methods, such as Hierarchical Clustering, k-Means Clustering, and Gaussian Mixture Models. According to the introduction and comparison in [12] , the k-Means Clustering operates on the actual observations (rather than the larger set of

dissimilarity measures in Hierarchical Clustering), and creates a single level of clusters; so it is more suitable than Hierarchical Clustering for larger amounts of data. When compared with Gaussian Mixture Models, which form cluster by representing the probability density function of observed variables as a mixture of multivariate normal densities, the k-Means Clustering can be more rapidly-implemented and easy to handle. So, we choose the k-Means Clustering to do our classification.

Given a set of observations $(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})$, where each observation is a d-dimensional real vector, k-Means Clustering aims to partition the n observations into $k(k \leq n)$ sets $\mathbf{S} = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares (WCSS):

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{x_j \in S_i} \|\mathbf{x_j} - \mu_i\|^2 \tag{7}$$

where μ_i is the mean value of the points in S_i.

Algorithm 1 *k-Means Clustering* Algorithm

Require: choose initial positions for the cluster centroids
  repeat
    for all points **do**
      calculate its distance from each cluster centroid
      assign the point to the nearest cluster
    end for
    recalculate the positions of the cluster centroids
  until within-cluster sum is minimized

3.3 Analytic Hierarchy Process

The Analytic Hierarchy Process (AHP) [6] is a structured technique for organizing and analyzing complex decisions. In our paper, Analytic Hierarchy Process is one of the methods that is used to assign weight for each topic. The procedure is as follows:

1. Define the *Comparison Matrix* $\mathbf{A}$, with element a_{ij},of topic pairs.

   Compare topic i (tp_i) and topic j (tp_j), with our common sense, then we assign a value in the range of $1 - 9$ as a_{ij} representing priority. The greater this value be, the more suspicious the tp_i is, compared with tp_j. Repeat this process to evaluate all $a_{ij}, (i > j)$, then evaluate the $a_{ij} = \frac{1}{a_{ji}}(i < j)$, and $a_{ij} = 1(i = j)$.

2. Check the consistency of the judgments.

   Consistency is satisfied $\iff$ $\forall i, j, k, such\ that\ a_{ij}, a_{jk}, a_{ik} \in \mathbf{A},\ we\ have\ a_{ij} \cdot a_{jk} = a_{ik}$. Generally speaking, strict consistency is hard to satisfy. Instead we define a *Consistency Ratio* $CR = \frac{CI}{RI}$, where $CI = \frac{\lambda(A)-n}{n-1}$ is the Consistency Index; n is the dimension of matrix $\mathbf{A}$; $\lambda(A)$ is the largest eigenvalue of matrix $\mathbf{A}$; RI is the average value of CI for random matrices. RI is only concerned with the dimensions of the square matrix. Table 1 from [19] lists the empirical value of RI for comparison matrix with less than 15 dimensions.

n	1	2	3	4	5	6	7	8
RI	0	0	0.52	0.89	1.12	1.26	1.36	1.41
n	9	10	11	12	13	14	15	
RI	1.46	1.49	1.52	1.52	1.56	1.58	1.59	

Table 1: The average value of Consistency Index for random matrices

3. Revise the Comparison Matrix.

   Although we have the standard of "perfect" consistency, unfortunately, it is hard to achieve. With help of *Consistency Ratio* we can tell to which degree a comparison matrix is consistent. In most cases, we regard the comparison matrix as consistent when its $CR < 1$, in which case we calculate the eigenvector corresponding to the largest eigenvalue as the normalized topic weight. Otherwise, we need to revise the comparison matrix using minimum element method. We notice that when $\mathbf{A}$ is a consistent matrix, there exists vectors $(\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_n})$, $\forall i, j$, $a_{ij} = \frac{w_i}{w_j}$. When A needs to be revised, we extract the first $n-1$ minimum elements except the main diagonal, with which can determine a set of vectors $(\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_n})$, to construct the new comparison matrix $\mathbf{A}'$. Repeat this process, until the $\mathbf{A}'$ has satisfied the consistency requirement.

4 The Solution

4.1 Model Overview

In our case, it is clear that the topics one employee talks and the target he or she talks to will contribute to the evaluation of his or her conspiracy tendency. So, our job is concentrated on two fields — the topics mining and the social network analysis.

There are 15 topics in total: some (3) topics are regarded as suspicious for sure, while others not.Different topics vary in their content, leading to various weights.

Each contact, a message from one employee to another, acts as a link to connect different individuals together into a large network. The more links exits between two nodes, the closer their relationship is.

The combination of the topic weights and the network analysis constitute the foundation of our model.

4.2 Assigning Weight to the Topics

We propose 2 methods to evaluate the *topic weight*. We implement both of them. Later in Section 5.2 we will compare the result of different approach.

4.2.1 Assigning weight with simple function

The simplest (yet quite intuitive) way is to judge each topic based on its frequency of use by known conspirators or non-conspirators. Let $F_{m,cons}$ denote the frequency of topic m used by conspirators and $F_{m,non}$ denote its frequency used by non-conspirators. Now the weight of

topic, denoted as w_m is defined as:

$$w_m = \frac{F_{m,cons}}{F_{m,non}} \tag{8}$$

This measure of weighing messages largely depends on the initial division of conspirators and non-conspirators. It does not take other factors such as message semantics into account and thus it only serves as a reference for our actual approach.

4.2.2 Assigning weight with Analytic Hierarchy Process

The second approach is to use the Analytic Hierarchy Process method described in section 3.3. We manually make a semantic analysis on the 15 topics to get our initial comparison matrix. Then we adopt the AHP method to make it consistent, in order to make the topic weight vector reasonable.

In our case, we have some background knowledge on this problem that it is a conspiracy about embezzling funds from the company and using the Internet fraud to steal funds from credit cards of those doing business with the company. In order to set a weight vector for the total 15 topics, we have to view these topics first and find out some keywords. We conclude the keywords as follows:

- Finance: may relate to funds embezzled
- Spanish: some topics, including the suspicious ones, contain Spanish words
- Paige: the controversial manager who was paid much attention to recently
- Computer security: relate to the funds from credit cards stolen
- Private meeting: any secret is suspicious

Table 2 shows the keywords appearing in each topic.

Topic	Keywords
1	finance
2	finance, Spanish
3	
4	Paige
5	computer security
6	Paige
7	Spanish, Paige, private meetings
8	
9	
10	finance
11	finance, computer security
12	Spanish
13	Paige, computer security
14	finance
15	Paige, computer security

Table 2: Keywords semantic analysis (manual) on 15 topics

After we have a preliminary analysis on different kinds of topics, we can prioritize them with scale from 0 to 10, and then calculate a comparison matrix from it. Table 3 gives a demonstration of the comparison matrix calculated for the EZ case.

	1	2	3	4	5
1	1	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{3}$	2
2	4	1	$\frac{1}{4}$	3	5
3	8	4	1	3	9
4	3	$\frac{1}{3}$	$\frac{1}{3}$	1	3
5	$\frac{1}{2}$	$\frac{1}{5}$	$\frac{1}{9}$	$\frac{1}{3}$	1

Table 3: Pairwise Comparison Matrix for EZ case

Finally, the Analytic Hierarchy Process will give us the normalized weight for each topic, shown in Table 4

Topic Index	1	2	3	4	5	6	7	8
Topic Weight	0.0276	0.0741	0.0160	0.0419	0.0724	0.0520	0.1776	0.0127
Topic Index	9	10	11	12	13	14	15	
Topic Weight	0.0152	0.0214	0.1232	0.0842	0.1817	0.0301	0.0700	

Table 4: Normalized Topic Weight for 15 topics, given in *Analytic Hierarchy Analysis*

4.3 Obtaining the Attribute Vector

The attribute vector serves as a key "identity" for classifying the employees. We model the attribute vector v_i for person i as a k dimensional vector $(\mathbf{x_{i1}}, \mathbf{x_{i2}}, \ldots, \mathbf{x_{ik}})$, whereas k is the total number of known conspirators and non-conspirators. The value x_{im} represents the proximity between employee i and known the conspirator or non-conspirator m.

In our method, x_{im} is calculated mainly based on the approach of CFEC, but we've modified its probability calculation methods to tailored it into our needs.

Instead of modeling the random walk probability, we model the probability of increasing the conspiracy tendency of the t when s has said something. This modeling makes sense in that it naturally models the possibility of the impact that a communication or message will have on the receiver. Thus, the increasing tendency probability of two adjacent nodes v_i and v_j is defined as:

$$P_{i \to j} = \frac{w_{ij}}{\sum_{e \in \mathcal{E}_i} w_e} \tag{9}$$

Here the $\mathcal{E}_i$ represents the set of all the edges going outbound from i. Respectively, for a path p starting at s and ending at t, the tendency probability is:

$$Prob_{tend}(p) = \prod_{i,j \in p} P_{i \to j} \tag{10}$$

We care more on the actual tendency s has upon t. Thus it's defined in Equation 11 given that P_{st} represents the set of all simple paths from s to t:

$$EC_{s,t} = (\sum_{e \in \mathcal{E}_s} w_e) \cdot \{\sum_{p \in P_{st}} Prob_{tend}(p)\} \tag{11}$$

In reality, we consider the influence on tendency to be symmetric, namely SEC, thus:

$$SEC_{s,t} = EC_{s,t} + EC_{t,s} \tag{12}$$

In order to find all paths between two points, a Breadth-First-Search is launched, traversing all the nodes to find the paths. Given those concepts, the actual algorithm is analyzed in Algorithm 2

Algorithm 2 Calculating the tendency vector for each point in graph G

INPUT:
Directed graph G with each edge $e_i (i = 0, 1, 2, \dots)$ carrying "tendency" weight W_{e_i}
OUTPUT:
An $m \times n$ matrix $\mathcal{M}$ with r^{th} row vector $(x_{r1}, x_{r2}, \dots x_{rm})$ representing the expected increment $SEC_{rk} (1 \leq k \leq m)$ between r^{th} person and the known (non)conspirator m.

```
V ← All points
CIV ← All conspirators and non-conspirators
for (v_i, cv_j) ∈ V × CIV (1 ≤ i ≤ ||V||, 1 ≤ j ≤ ||CIV||) do
    P_ij ← paths between v_i and cv_j
    for each p_k in P_ij do
        Calculate Prob_{p_k} as in equation 10
    end for
    EC_ij ← (Σ_{e∈E_{v_i}} W_e) · Σ_{p_k∈P_ij} Prob_{p_k}
    EC_ji is calculated as EC_ij
    M_ij ← EC_ij + EC_ji
end for
```

4.4 Classifying the employees

Once we have the $m \cdot n$ attribute matrix $\mathcal{M}$ of conspiracy tendency for the total m employees to the n known (non)conspirators. We can use the clustering analysis method to classify all the staff according to their attribute vector.

Here we use the k-Means Clustering method mentioned in Section 3.2, which needs two parameters: one is the attribute matrix, the other is the number of clusters we want. We have varied the number of clusters to check the result with the 15 known employees, and finally found that dividing the network into 2 or 3 clusters is seasonable. The details of clustering result will be shown in the next section.

Having 3 clusters in hand, we can recognize one as conspiracy cluster, one as innocent cluster, and the last one as uncertain cluster. For the uncertain cluster and the conspiracy cluster, we rank them according to their distance to the centroid of the conspiracy cluster: the closer one will rank higher, thus we get R_1 as the first part of final ranking sequence. For the innocent cluster, we rank them according to their distance to the centroid of the innocent cluster: the closer one will rank lower. And in this way we can obtain R_2 as the second part of final ranking sequence. Finally we concatenate R_2 at the tail of R_1 to get the final ranking sequence.

5 Evaluations

We've implemented most of our model in Steel Bank Common Lisp(SBCL) [3]. The rest part of the model is done with Matlab [12]. The model is analyzed thoroughly and carefully. We also implemented others' approach and compared our model with theirs.

5.1 Model Results

Given the initial condition that 3 topics are suspicious, 8 known innocent employees, and 7 identified conspirators, we can get the following ranking corresponding to the different topic weight generation methods.

The conspiracy tendency ranking, given in Table 5, uses the AHP to generate topic weights. In this ranking, Delores ranks 8th, who is involved in the conspiracy. Gretchen and Jerome are supposed to be innocent. Using the simple weight method, we get ranking in Table 6. Still, Delores has high tendency to be conspirator, while the other two senior managers are innocent.

Based on the result above, we are confident to claim Delores as a conspirator and identify other 2 senior managers as innocent.

1	2	3	4	5	6	7	8	9
ELSIE	PAUL	SEENI	JEAN	ULF	HARVEY	PAIGE	DOLORES	LARS
10	11	12	13	14	15	16	17	18
CRYSTAL	DWIGHT	NEAL	CHRIS	ERIC	STEPHANIE	SHERRI	ELSIE	KRISTINA
19	20	21	22	23	24	25	26	27
PRISCILLA	MALCOLM	WAYNE	BETH	YAO	ALEX	GRETCHEN	FRANKLIN	MARION
28	29	30	31	32	33	34	35	36
SHELLEY	KRISTINE	DOUGLAS	GERRY	GRETCHEN	WESLEY	FANTI	RENI	JEROME
37	38	39	40	41	42	43	44	45
PATRICIA	DARLENE	JULIA	NEAL	MARCIA	BARIOL	COLE	PATRICK	SANDY
46	47	48	49	50	51	52	53	54
ELLIN	LOIS	BETH	LE	HAZEL	JEROME	OLINA	MELIA	KAREN
55	56	57	58	59	60	61	62	63
HAN	KIM	WILLIAM	FRANCIS	DONALD	CORY	HARK	CHA	VIND
64	65	66	67	68	69	70	71	72
SHENG	DAROL	GARD	QUAN	DAYI	CHARA	ESTE	LAO	PHILLE
73	74	75	76	77	78	79	80	81
ANDRA	MAI	CARINA	TRAN	KATHERINE	LOUIS	MARIAN	JIA	CLAIRE
82	83	84	85	86	87	88	89	90
CHRISTINA	ERICA							

Table 5: Conspiracy tendency ranking with AHP

1	2	3	4	5	6	7	8	9
ELSIE	JEAN	PAUL	SEENI	ULF	HARVEY	PAIGE	DWIGHT	DOLORES
10	11	12	13	14	15	16	17	18
CRYSTAL	LARS	ERIC	DARLENE	STEPHANIE	CHRIS	WAYNE	JULIA	LOIS
19	20	21	22	23	24	25	26	27
MARCIA	ELSIE	BETH	NEAL	FRANKLIN	MARION	YAO	KRISTINE	SHELLEY
28	29	30	31	32	33	34	35	36
JEROME	GRETCHEN	ALEX	GERRY	DOUGLAS	GRETCHEN	FANTI	RENI	BARIOL
37	38	39	40	41	42	43	44	45
COLE	JEROME	PRISCILLA	WESLEY	PATRICIA	PATRICK	BETH	FRANCIS	KIM
46	47	48	49	50	51	52	53	54
WILLIAM	HAZEL	NEAL	HAN	KRISTINA	KAREN	VIND	MALCOLM	CORY
55	56	57	58	59	60	61	62	63
DONALD	SHERRI	LOUIS	HARK	CHA	QUAN	SHENG	DAROL	LE
64	65	66	67	68	69	70	71	72
ESTE	OLINA	KATHERINE	GARD	MAI	ANDRA	LAO	MELIA	SANDY
73	74	75	76	77	78	79	80	81
CARINA	MARIAN	CHARA	DAYI	TRAN	PHILLE	ELLIN	JIA	CHRISTINA
82	83	84	85	86	87	88	89	90
ERICA	CLAIRE							

Table 6: Conspiracy tendency ranking with Simple-weight

Next, we give the new priority when the new information comes to light that Topic 1 is connected to conspiracy and that Chris is one of the conspirators.

Table 7 is the new ranking. Still, we are confident to claim Delores as a conspirator and identify other 2 senior managers as innocent.

1	2	3	4	5	6	7	8	9
ELSIE	PAUL	JEAN	SEENI	ULF	HARVEY	CRYSTAL	PAIGE	CHRIS
10	11	12	13	14	15	16	17	18
DOLORES	DWIGHT	ERIC	LARS	DARLENE	STEPHANIE	JEROME	LOIS	JULIA
19	20	21	22	23	24	25	26	27
YAO	WAYNE	MARCIA	FRANKLIN	ELSIE	GRETCHEN	NEAL	SHELLEY	BETH
28	29	30	31	32	33	34	35	36
KRISTINE	MARION	DOUGLAS	ALEX	GERRY	GRETCHEN	FANTI	RENI	BARIOL
37	38	39	40	41	42	43	44	45
JEROME	COLE	WESLEY	PATRICIA	BETH	PRISCILLA	HAN	KRISTINA	HAZEL
46	47	48	49	50	51	52	53	54
WILLIAM	NEAL	SHERRI	PATRICK	VIND	KAREN	MAI	MELIA	FRANCIS
55	56	57	58	59	60	61	62	63
ELLIN	KIM	LOUIS	CORY	SANDY	CHA	DONALD	SHENG	DAROL
64	65	66	67	68	69	70	71	72
ESTE	LE	KATHERINE	OLINA	QUAN	LAO	GARD	HARK	TRAN
73	74	75	76	77	78	79	80	81
CHARA	MALCOLM	DAYI	PHILLE	ANDRA	MARIAN	CARINA	CHRISTINA	CLAIRE
82	83	84	85	86	87	88	89	90
JIA	ERICA							

Table 7: Consipracy tendency rankings with Simple-weight(Chris as conspirator)

Intuitively, the ranking doesn't changes so much. We count every pairs of nodes to check if there is a position alternate in rankings, and discover that there are totally 235 pair changes. And the total number of possible ranking alternation should be $C_{83}^{2} = \frac{83*82}{2} = 3403$, i.e. only a 6.91% change.

5.2 Compare the methods for Assigning Topic Weight

We use the different two methods to get the topic weight for each topic. Then we use this weight vector to evaluate the attribute matrix for the EZ case, and classify the employees. Finally we test the accuracy of the weight vectors.

First we test the EZ case with 2 and 3 clusters, and we will list all the conspirators. Then we will test the larger case with 2 and 3 clusters, and calculate the false positive (unfounded innocent) and false negative (unfounded conspirator). These two concepts are defined for test purpose. The false positive rate represents among the known innocent employees(say n), what percentage of them are not classified as innocent. That is to say, if 2 of the n members are not correctly classified as innocent members, the false positive rate is $\frac{2}{n}$. The false negative rate is defined vice versa.

Clusters	Method	Conspirators
2	Simple	Dave Ellen George Harry
	AHP	Dave Ellen George Harry
3	Simple	Dave Ellen George Harry
	AHP	Dave Ellen George Harry

Table 8: Comparason result over the EZ case

Table 9: caption

Clusters	Method	Size	Unfounded Innocent	Unfounded Conspirator
2	Simple	(76, 7)	0	0.2865
	AHP	(75, 8)	0	0.1429
3	Simple	(76, 4, 3)	0	0.5714
	AHP	(64, 13, 6)	0.25	0.4286

Table 10: Comparason result over the complicated network

5.3 Comparison with Existing Solution

We evaluate our model's reliability by bringing in a *Crime Syndicate Recognition System* from [10]. The main function of his system is to break up the entire complicate social network into several groups containing less people with stronger relationships, and then classifies these groups with the known conspirators and innocent employees, thus the scope of the conspirators will be quickly narrowed. It doesn't take the edge weights into consideration while do the decomposition, which makes it less practical for our problem. Even though, the classification result of this system still provides a good approximation to the facts.

5.3.1 Crime Syndicate Recognition System Overview

Based on several kinds of association decomposition algorithm for social network and cluster algorithm for complex network, the method is carried out as follows.

It defined a clustering coefficient for the edge belongs to a social network:

$$C_{ij} = \frac{z_{ij} + 1}{\min[ki - l, kj - l]} \tag{13}$$

z_{ij} represents the number of the triangles contain the edge e_{ij}. k_i is the degree of the node i. l represents the number of edges connecting the nodes i and j.

The clustering coefficient reflects the position of the edge in a social network, center or margin. An edge with large C_{ij} should be close to the center, while edges with small C_{ij} will be close to the margin.

With the clustering coefficient calculated, they make use of the fast decomposition to simplify the entire social network. The steps as follows.

1. Calculate clustering coefficient for all edges belonging to the network.
2. Remove the edge with the minimal clustering coefficient.
3. Repeat step 2 if there is no subgraph produced, otherwise we get two subgroups with stronger connection.
4. Keep decomposing the new subgraph until all the subgraphs have a strong enough relationship or contain the minimal or less number of people.

The result of the *Crime Syndicate Recognition System* is shown in figure 1.

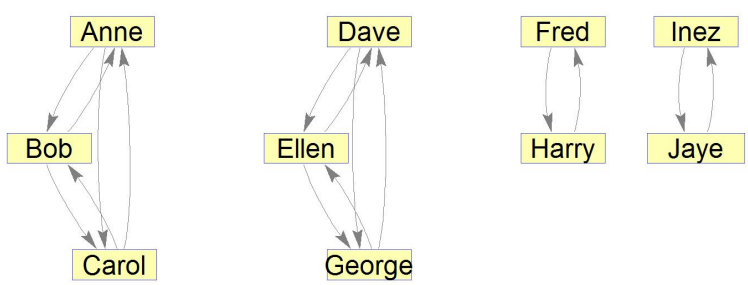

Figure 1: Recognition results for the EZ case using *Crime Syndicate Recognition System*

***Crime Syndicate Recognition System* result**

In figure 1, the edges denotes the relation of two nodes, but not the messages from one to the other. As for the EZ case only involving 10 nodes, and based on our assumption, we can say that Ellen is a conspirator, because of the strong relation with the two identified conspirators, Dave and George. Other people can't be justified with greater confidence. The reason for misidentification of Bob and Inez should be the limitation of the sample size.

We also use the *Crime Syndicate Recognition System* to discompose the large social network in current case, and end up with 11 separated subgroups. Because there are so many nodes in this case that we omit the demo graph. Among the 11 subgroups, there is surprisingly a group of 7, denoted as G_1, contains 5 known conspirators, which promotes the conspiracy tendency of

the other 2. The other 1 known conspirators are located in the largest subgroup, denoted as G_2, which contains nearly 30 nodes, and there is one identified innocent people in this group, too. We adopt a strategy to classify these nodes into 3 categories: all people in G_1 and the people has direct connection with conspirator in G_2 fall into the conspirator category. The residual people in G_2 are in the uncertain category and all the other groups of people are regarded innocent.

***Crime Syndicate Recognition System* comparison**

Comparing two results from both model, although technically they use different method, the result can be considered consistent. When it comes to classifying the original group into 11 subgroups, like the Recognition System of [10], our model will at most put the conspirators into two categories.

Generally speaking, our model and the method from [10] produce consistent results. But the pitfall for the method in [10] is that they doesn't take the topic weights into consideration.

5.4 Sensitivity Analysis

The model consists of a proximity estimating algorithm and a classification algorithm, making it difficult to analyze its sensitivity by merely algebra. Thus, we analyze it via experiments.

We've set up totally 4 scenarios. They are:

- Randomly pick up people as additional conspirators while the weight stays the same(generated by algorithms that are not largely determined by the number of conspirators). Measure the false-negative value of the results.
- Randomly pick up people as additional conspirators while the weight changes during this course. Measure the false-negative value as the results
- Randomly pick up people as additional non-conspirators while the weight stays the same. Measure the false-positive value of the results.
- Randomly pick up people as additional non-conspirators while the weight changes. Measure the false-positive value as the results.

For each type of the experiments, we add 1 to 5 people as additional members. In order to eliminate uncertainty caused by randomness, each experiment is performed 6 times repeatedly. The results are shown in Table 11, 12, 13 and 14.

Additional	Test1	Test 2	Test 3	Test 4	Test 5	Test 6	Mean
1	0.5000	0.6250	0.5000	0.5000	0.5000	0.5000	0.5208
2	0.5556	0.5556	0.5556	0.6667	0.5556	0.5556	0.5741
3	0.5000	0.6000	0.5000	0.6000	0.5000	0.6000	0.5500
4	0.6364	0.5455	0.4545	0.6364	0.5455	0.6364	0.5758
5	0.5833	0.5000	0.5833	0.5833	0.5833	0.5833	0.5694

Table 11: Measuring false negative rate sensitivity with fixed message weights

A boxplot of all four tables is also presented in this paper, in Figure 2.

Additional	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Mean
1	0.6250	0.3750	0.5000	0.3750	0.3750	0.5000	0.4583
2	0.4444	0.4444	0.5556	0.4444	0.4444	0.4444	0.4630
3	0.5000	0.6000	0.5000	0.5000	0.5000	0.5000	0.5167
4	0.5450	0.5450	0.5455	0.5455	0.5455	0.5455	0.5455
5	0.5000	0.5000	0.5833	0.5833	0.5833	0.5833	0.5556

Table 12: Measuring false negative rate sensitivity with dynamic message weights

Additional	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Mean
1	0.2222	0.2222	0.2222	0.3333	0.0000	0.3333	0.2222
2	0.2000	0.1000	0.3000	0.5000	0.3000	0.4000	0.3000
3	0.3636	0.2727	0.2727	0.3636	0.2727	0.2727	0.3030
4	0.4167	0.3333	0.3333	0.0833	0.5000	0.4167	0.3472
5	0.3846	0.4615	0.2308	0.4615	0.2308	0.2308	0.3333

Table 13: Measuring the false positive rate with fixed message weights

Additional	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Mean
1	0.3333	0.4444	0.3333	0.3333	0.0000	0.0000	0.2407
2	0.4000	0.3000	0.3000	0.4000	0.4000	0.1000	0.3167
3	0.0000	0.3636	0.0909	0.3636	0.0000	0.4545	0.2121
4	0.4167	0.0833	0.3333	0.2500	0.1667	0.4167	0.2778
5	0.3077	0.0769	0.4615	0.3846	0.2308	0.3846	0.3077

Table 14: Measuring the false positive rate with dynamic message weights

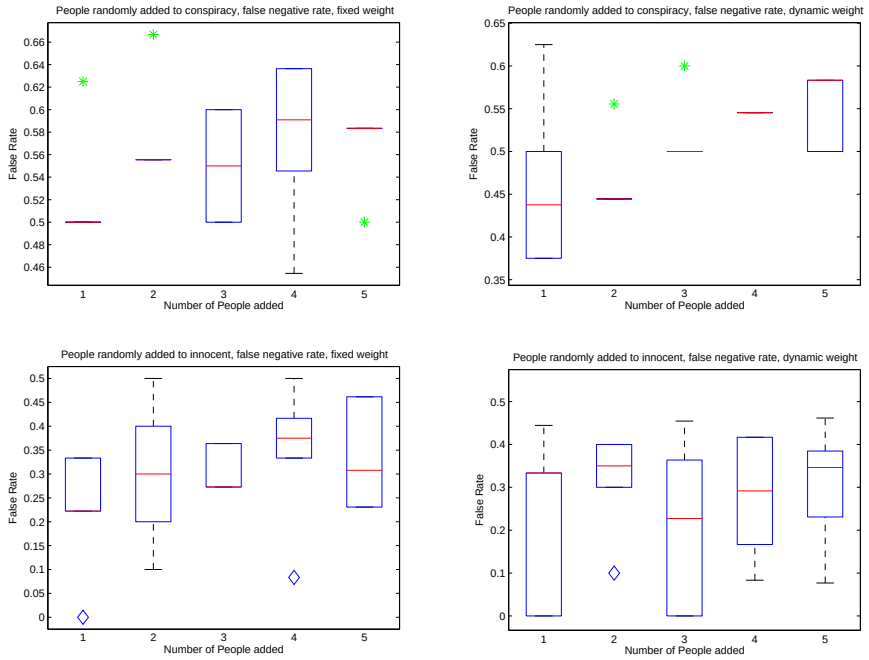

Figure 2: Measuring the sensitivity using additional members

From Figure 2, we see that false rates of both type do not change significantly with the number of additional members when weights are not changed. This indicates that our algorithm is stable when the messages are assigned proper weight, such as when detailed analysis of the topics has been accomplished. As long as the weights remain unchanged, the model itself is robust with even nearly 50% of noise.

However, if the weights are not well established, such as the case in which the weights are not pre-generated via large training sets and/or the weights themselves need to be extracted from those people participating the network, the uncertainty of this model is high.

The results indicate that our model is not sensitive to the number of (non)conspirators but the weights pre-analyzed. In order to improve the model's stability, adopting better effective weighting algorithms would be a good choice.

This leads to the conclusion that our model performs consistently with standard machine learning models. When the parameters are generated via proper training set, the model can be extremely robust and stable.

Another discovery is that, within each experiment, weighting algorithm that depends on the number of (non)conspirators tend to be more stable with random additional data. Maybe this is because of the simple weight algorithm we adopt here, since the probability of picking either a conspirator or a non-conspirator is determined by the proportion of people of those two types.

5.5 Optimizations

5.5.1 Limiting the Depth of the BFS Search Tree

During the Breadth-First-Search, each time we need to find all paths possible from one node to another. Such computation is heavy, and even if optimized by memorization, it still costs quite amount of time to complete. Thus, we need to limit the depth of the BFS search tree.

The brutal force way is to abort the search whenever it reaches a particular depth. Such optimization algorithm requires no additional computation overhead. Intuitively, this way of cutting the search tree is not recommended, and it proves to be true in many cases. However, in this case, this brutal force way is both efficient and effective.

An intuitive evidence comes from the claim that the average number of steps between any two people on Earth is six [16].

According to [8], the CFEC proximity measurement itself favors short paths. As originally demonstrated by [8](also in Equation 5), CFEC strongly discourages long paths as the probability of following it decays exponentially.

We've also conducted a series of experiments to prove our decision and it pays off.

Shown in Figure 3 are 4 experiments on selecting different depth limit(or "hops" with respect to wireless network terminology). The testbed is the EZ case with 2 conspirators and 2 innocent employees. We've analyzed the relative growth of tendency value while selecting different depth. As is depicted, when the depth reaches 4, most of the data points tend to be stable. Some data points seem to keep increasing sharply. They might get stabilized with a few more steps but their values are so small that they should not contribute more to the classification.

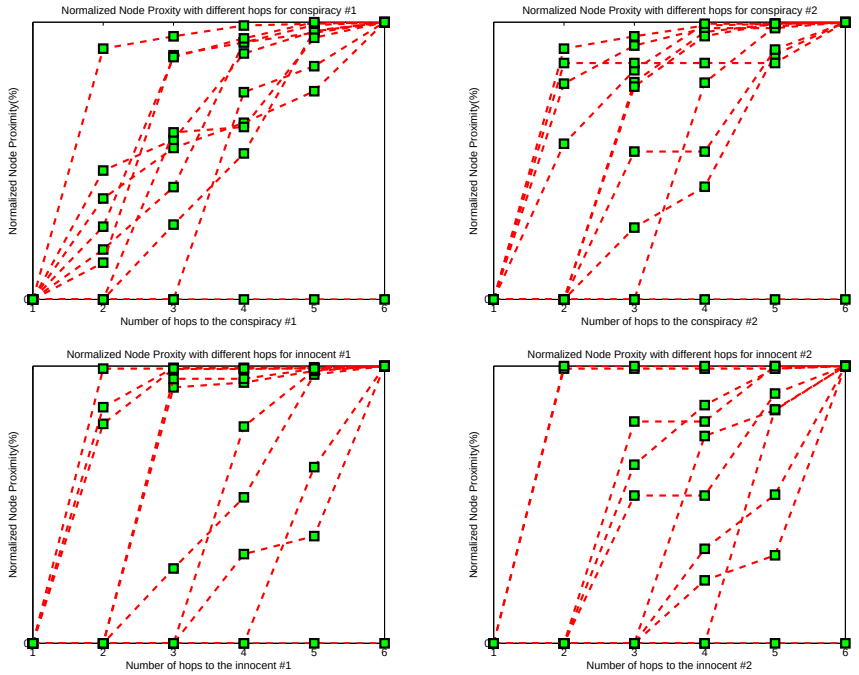

Figure 3: Percentage of Growth on Different Depth of BFS Search Tree

5.6 Future Optimizations

Although the results in the model seem to be satisfactory in the test, it needs further optimizations and/or enhancements to ensure scalability and portability.

5.6.1 Natural Language Processing for Message Analysis

Classifying individual message into a particular topic by hand can be cumbersome and impractical as the message traffic gets higher. Thus, natural language processing(NLP) algorithms such as Latent Semantic Analysis(LSA) [9] can offer tempting possibility. What's more, topics can be automatically divided and selected with NLP techniques, which further reduces the workload for human beings. However, allowing two or more machine learning algorithms running over a specific system maybe risky. The system must be designed to tolerate dual or triple false results generated by those algorithms in a chain. The easiest technique to avoid such risk is to perform the same calculation multiple times to reduce the error rate. Other solutions can be based upon decision tree [13] and other systems engineering [15] techniques, which are beyond the scope of our paper.

5.6.2 Reducing Dimension of Attribute Vector with Principal Component Analysis

As the known number of conspirators or non-conspirators grows larger, the dimension of the attribute vector increases dramatically. Thus, the Principal Component Analysis(PCA) [11] can be adopted here to quickly reduce the dimension of the attribute vectors before the execution of the k-Means algorithm while maintaining most of the important information.

5.6.3 Updating the Weights with Genetic Algorithm

When the conspirators are classified, the weights itself might be updated. The Genetic Algorithm(GA) [14] can be used here in the following steps:

1. Generate different weight vectors by adding random noise to one existing weight vector.
2. Calculate the false positive or false negative rate in classification with those weight vectors.
3. Eliminate those vectors that cause large false rate.
4. Updating a few vectors by adding little amount of random noise, such step is in courtesy to mutation. Generate a few new vectors by calculating the average value between each two vectors. Merge those vectors with the initial set of vectors.
5. Continue until the result becomes stable and correct.

This algorithm was originally put into our model. However, due to the limited size of the data set, we cannot measure the false rate, since in most cases, two rates differ only by including/excluding one member of the target set. Thus, this algorithm is left for future work over larger data sets.

5.7 Potential Applications

Our model contains a network proximity measuring algorithm, a weight evaluation technique and a clustering algorithm, all of which are general purpose approaches and thus, the model itself has many potential uses. Possible applications include detecting infected cells, social network recommendation [7] and so on.

- **Infected Cells Detection:** CFEC models the possibility or tendency passed between nodes. It's quite clear that such measurement is applicable to the infection possibility or expectation of other cells. And thus, this scenario has no difference from the basic scenario discussed so far.
- **Social Network Recommendation:** Our model gives clear proximity between two nodes. There is no exception for interests. By classifying the nodes based on different potential of influence upon interests, our model is applied intuitively into such scene.

6 Conclusion

We proposed the "Eagle Eye" system to detect the conspirators over the weighted network. The combination of the algorithms used in the model are simple, intuitive yet powerful. We've extensively analyzed our model and proposed further techniques and optimizations. In all, our model successfully accomplishes the mission and with a few enhancements, can be applied to various field with larger data sets and/or greater constraints on precision.

References

[1] Bollobas and Bela. *Modern Graph Theory*. Springer, corrected edition, July 1998.

[2] Ulrik Brandes and Daniel Fleischer. Centrality measures based on current flow, 2005.

[3] Alexey Dejneka. Steel bank common lisp, 2012.

[4] Peter G. Doyle and J. Laurie Snell. *Random Walks and Electric Networks*. Mathematical Association of America, Washington, DC, 1984.

[5] Christos Faloutsos, Kevin S. McCurley, and Andrew Tomkins. Fast discovery of connection subgraphs. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 118–127, New York, NY, USA, 2004. ACM.

[6] Ma Teresa Lamata Jose Antonio Alonso. Consistency in the analytic hierarchy process: A new approach. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 14:445–459, 2006.

[7] Ioannis Konstas, Vassilios Stathopoulos, and Joemon M. Jose. On social networks and collaborative recommendation. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 195–202, New York, NY, USA, 2009. ACM.

[8] Yehuda Koren, Stephen C. North, and Chris Volinsky. Measuring and extracting proximity in networks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 245–255, New York, NY, USA, 2006. ACM.

[9] Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. An Introduction to Latent Semantic Analysis. *Discourse Processes*, pages 259–284, 1998.

[10] Hong Zhang Liang Li, Jian Cao. The recognition system of crime syndicate based on social network analysis, 2008.

[11] Lindsay I. Smith. A tutorial on Principal Components Analysis, 2002.

[12] Inc. The MathWorks. Matlab r2011b documentation statistics toolbox, user's guide, 2011.

[13] wikipedia. Decision tree.

[14] wikipedia. Genetic algorithm.

[15] wikipedia. Systems engineering.

[16] wikipedia. Six degrees of seperation, 2004.

[17] Jennifer Xu and Hsinchun Chen. Criminal network analysis and visualization. *Commun. ACM*, 48:100–107, June 2005.

[18] Jennifer J. Xu and Hsinchun Chen. Crimenet explorer: a framework for criminal network knowledge discovery. *ACM Trans. Inf. Syst.*, 23:201–226, April 2005.

[19] Shubai Xu. *Analytic Hierarchy Process Principle*. Tianjin University Press, Washington, DC, 1988.