Summary

To address the problem of conspirator detection using the provided message traffic information, we first gave a basic model based on the similarity measure between one individual and the already identified ones. Specifically, we construct a graph to encode the connection between two individuals in the social network. Then we measure the pairwise similarity by the commute distance between two nodes, which can be efficiently computed through a pseudo inverse of the graph Laplacian matrix. Finally, a logistic regression model is trained on the pairwise distance feature vectors, predicting one's probabilities of being part of the conspiracy.

We then applied our model to the data of the current problem. First, we showed the correctness and the robustness of our model by training it with only a few identified individuals, to see if it can correctly predict the identities of the rest. After such model validation, we formally calculated one's probabilities of being a conspirator with all the identified ones used in training the logistic regression. Moreover, we studied the sensitiveness of outcomes to the parameters used in the initial graph construction.

Further analysis shows that our model actually gives each individual an implicit representation that bears many properties the network, especially the local structures of the constructed graph. Based on this observation, we sought to enhance our model by incorporating this prior information into the state-of-the-art models from semantic network and text analysis, which typically learn an explicit representation for each individual. As a case study, we investigated two models, namely Probabilistic Latent Semantic Index (PLSI) and Nonnegative Matrix Factorization (NMF). We proposed a regularization framework to achieve our goal, and introduced an efficient algorithm to solve the optimization problem.

Finally, we generalized our model to other applications and put forward a few notes concerning the scalability issue.

Find Conspirators by Message Traffic

MCM/ICM Contest Question C

Team # 16857

Contents

1 Introduction **2**

2 Notations **3**

3 Assumptions **4**

4 Graph Construction in Basic Model **4**
4.1 More on the Weights 5

5 Modeling Similarity: Commute Distance **5**
5.1 Motivation 5
5.2 Definition 5
5.3 Calculations 6

6 Prioritization & Classification: Logistic Regression **6**
6.1 Motivation: Relationship with Posterior Probability 7
6.2 Maximum Likelihood for Regression 7

7 Results **9**
7.1 Prediction: Correctness and Robustness 9
7.2 Our Results 9
7.3 Parameter Sensitiveness 11

8 Model Analysis: a Revisit **12**
8.1 Explicit Representation for Commute Distance 12
8.2 Feature Extraction From a Graph 12

9 An Improved Model **13**
9.1 Topic Models 13
9.2 Nonnegative Matrix Factorization 14
9.3 Representation Enhancement 15
9.4 Generalization 16

1 Introduction

Currently, one of the risks the Intergalactic Crime Modelers (ICM) face is to solve white-collar and high-tech conspiracy crimes. Frequently, conspirators are hidden in a large social network in the company. While its complexity creates problem for investigators, from another perspective it can effectively convey relational information critical for conspirator detection. Therefore, how to extract useful information from the network and its message traffic becomes critical for creating a good model to detect conspiracy crimes.

Consider one specific social complex to investigate a certain conspiracy, our goal is pretty clear:

- Dig out important characteristics in the social complex;
- Identify people who are most likely conspirators;
- Make a priority list for ICM to investigate.

Now given the transmitted message network and topic information of the company workers, how to identify crime members still remain a tricky problem. A good model with low error rate follows three criteria below:

- Don't let guilty parties get off;
- Don't make innocents falsely accused;
- Don't let someone have the opportunity to get reduced sentences.

Meanwhile, the network modeling technique will be more appreciative if it can deal with tens of thousands of network data quickly.

Based on two factors: message amount on individuals and basic topic classification (contains two classes: one is known suspicious message topics the other is unknown ones), we construct a basic model to preliminarily analyze each persons likelihood of being part of the conspiracy. The steps of the basic model are shown as below:

Step 1: Construct a weighted graph using two factors above.

Step 2: Measure the distance between each point using commute distance.

Step 3: Use certain conspirators and innocents information to collect main features.

Step 4: Use logistic regression to satisfy the relationship between collected features and certain outcomes, then output the suspicious rank order.

Because the characteristics in basic model is too limited, we consider adding message text information to improve our model, by text analysis and semantic network analysis, based on the above model, we create an improved model. The steps of the improved model are shown as below:

Step 1: Combine topic model and NMF model to construct a new weighted graph.

Step 2: Use dimension reduction (feature extraction) to collect main features.

Step 3: Use logistic regression to get the rank order.

The relationship between two models is shown in Figure 1.

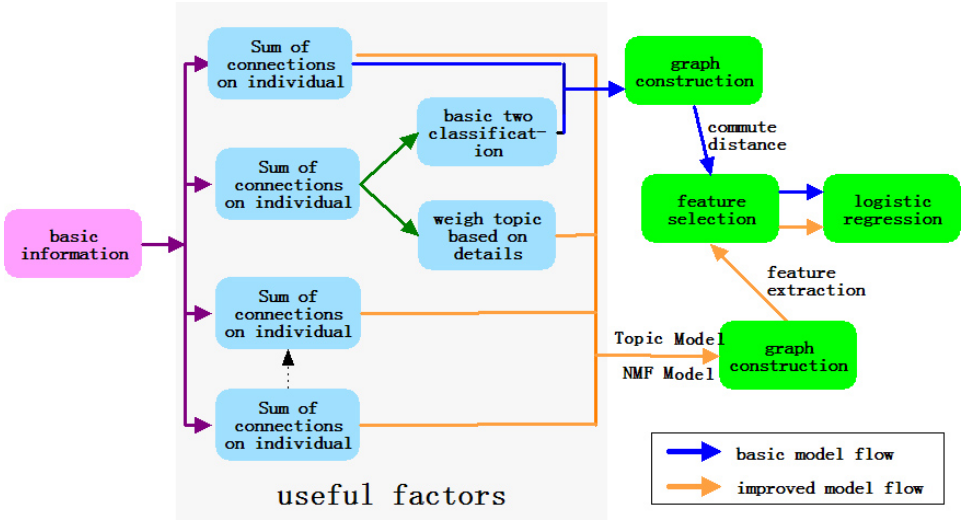

Figure 1: Framework of the Models

2 Notations

- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: graph used to model the message traffic network in the company;
- $\mathcal{V}$: vertices in the graph, corresponds to an individual in the company;
- $\mathcal{E}$: edges in the graph, corresponds to an connection between individuals;
- n: the number of individuals;
- m: the vocabulary size the whole text corpus;
- r: the number of latent topics, or the reduced dimension;
- W: graph affinity matrix, or transition matrix;
- D: degree matrix;
- L: graph Laplacian matrix;
- C: commute distance matrix;
- $L^{\dagger}$: pseudo inverse of the graph Laplacian L;
- A^T: transpose of matrix A;
- X: the individual-word matrix, or the data matrix;
- U: the word-topic matrix, or the basis matrix;

- V: the individual-topic matrix, or the low-dimensional representation matrix;
- Φ: the feature vector based on commute distance;

3 Assumptions

- The network we considered in one case is isolated from the outside world, which means that all members inside the network have no relationship with people who are not mentioned in the network;
- Apart from the information we collected in the case, other conditions on individuals are equal;
- It is reliable of using the data collected to predict conspirators, which means that regardless of some unrelated information and incompleteness of network message, most useful information has been accumulated;
- We change the network relationship into a weighted graph and the distance between two points represents some kind of relationship;
- In a certain distance measurement, the relationship between two points is closer when their distance is shorter.

4 Graph Construction in Basic Model

First of all, we need to construct a similarity graph to model the whole network. There's one important issue, how to construct the graph based on the message traffic, so that it properly encode the relationship between two individuals? Based on our assumptions, we constructed the similarity graph W in the following two steps:

- First, we assign weights to one correspondence between two individuals:

$$s_{ij}^{(t)} = \begin{cases} 1, & \text{if this message is about suspicious topics,} \\ \delta, & \text{otherwise.} \end{cases} \tag{1}$$

  Here we use $s_{ij}^{(t)}$ to denote the t-th correspondence between individual i and individual j, $1 > \delta > 0$ is a parameter denoting the relative weights between a suspicious message and a non-suspicious one. For the message that contains multiple topics, we simply split it into multiple single-topic messages. The directions are ignored.

- Then, the graph affinity matrix is defined as:

$$W_{ij} = \begin{cases} \sum_t s_{ij}^{(t)} / \sum_t 1, & \text{if they have correspondence records,} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Note that we used the total number of correspondence $\sum_t 1$ between two individuals to normalize the weights. The rationality of doing this is to avoid the case where two individuals have a lengthy conversation on a non-suspicious topic, resulting in an even larger affinity than other couples who converse on suspicious topics.

4.1 More on the Weights

The weight settings in our graph construction is just for a simple illustration. We ignored the directions how the messages are sent, the detailed topics it talks about, and the relative importance between suspicious topics. As is suggested by requirement 3, if we have more information about the content of the topics, we can enhance our model by optimizing the weights in the affinity graph.

Just for another illustration, since topic 13 is considered key in the conspiracy plan, we can add more weights on the edge talking about this topic.

5 Modeling Similarity: Commute Distance

5.1 Motivation

As is stated by the problem, the key task for us is to determine who is one of the conspirators, while who is not. Since there are already several individuals whose identity has been determined, we can therefore solving the problem by asking: who is more likely to be the conspirator? Who is less likely? This inspires us to find a "distance" to properly measure the similarity and dissimilarity between individuals based on the network provided. Once the measure is determined, one's identity could be easily determined relying on the distances between him/her and the individuals whose identity already known to us.

5.2 Definition

If we use a connected, undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to model the whole network, then we can define a Markov random walk on the graph where for each vertex i and j using the graph affinity matrix, or transition matrix $W = [w_{ij}]$. As a result, there is an associated probability w_{ij} that the walker transit from i to j in one step. Then a natural distance measure between a vertex pair (i, j) would be the average number of steps needed by a random walker from vertex i to j for the first time. Note here that the number of average steps does not necessarily be the same for (i, j) and (j, i). Therefore, we use *commute distance* [Fouss et al. (2007)] c_{ij}, which is defined as the expected time it takes for the random walk to travel from vertex i to j and back between two vertices i and j. All the pairwise distance forms a commute distance matrix C.

The two nice properties for commute distance are:

- The commute distance between two vertices decreases if there are many different short ways to get from vertex i to j, which is different from the shortest distance, and complies to the fact that the more intimate contacts one has with a conspirator, the more likely he/she is involved.
- Vertices which are connected by a short path in the graph and lie in the same high-density region of the graph are considered closer to each other than points which are connected by a short path but lie in different high-density regions of the graph. It complies to the fact that conspirators are likely to be tightly and compactly connected.

5.3 Calculations

The commute distance on a graph can be computed using the generalized inverse $L^\dagger$ of the graph Laplacian $L = D - W = [L_{ij}]$ (where D is the diagonal degree matrix, whose entries are column or row sums of W, *i.e.*, $(D_{ii} = \sum_j W_{ij})$, $L = D - W$ is the Laplacian matrix). Denote $\mathbf{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0)^T$ as the i-th unit vector.To define $L^\dagger$, the matrix L can be decomposed as $L = U\Lambda U^T$ where U is the matrix containing all eigenvectors as columns and Λ the diagonal matrix with the eigenvalues $\lambda_1, \ldots, \lambda_n$ on the diagonal. Since at least one of the eigenvalues is 0 (a property of graph Laplacian), the matrix L is not invertible. Instead, we define its generalized inverse as $L^\dagger = Q\Lambda^\dagger Q^T$ where the matrix $\Lambda^\dagger$ is the diagonal matrix with diagonal entries $1/\lambda_i$ if $\lambda_i \neq 0$ and 0 if $\lambda_i = 0$. The entries of $L^\dagger$ can be computed as $L^\dagger_{ij} = \sum_{k=2}^{n} \left[1/\lambda_k q_{ik} q_{jk}\right]$. The matrix $L^\dagger$ is positive semi-definite and symmetric.

The close relationship between $L^\dagger$ and C is illustrated in the following formula:

$$c_{ij} = n(L^\dagger_{ii} - 2L^\dagger_{ij} + L^\dagger_{jj}) = n(\mathbf{e}_i - \mathbf{e}_j)^T L^\dagger (\mathbf{e}_i - \mathbf{e}_j). \tag{3}$$

Therefore, we could use graph Laplacian to conveniently derive the pairwise commute distance.

6 Prioritization & Classification: Logistic Regression

Based on the pairwise commute distance, we get a 14-dimensional feature vector for each individual (We cannot determine which Elsie is the conspirator, so we just use the rest), which measures the distances between one individual and the identified people. Then in the next stage, the task becomes how to use the feature vectors to predict the each individual's likelihood of being part of the conspiracy (requirement 1 and 2).

Here we use the *logistic regression* to solve the problem. Logistic regression, also called a logit model is used in statistics and machine learning for prediction of the probability of occurrence of an event by fitting data to a logistic function:

$$\sigma(z) = \frac{e^z}{e^z + 1} = \frac{1}{1 + e^{-z}} \tag{4}$$

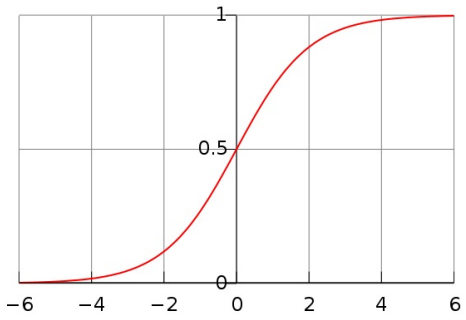

Figure 2: The logistic function: $z - f(z)$

6.1 Motivation: Relationship with Posterior Probability

It is important to note here that to determine if an individual is a conspirators is intrinsically a problem of two-class classification. And the main advantage of logistic regression is its close relationship to the posterior probabilities in the two-class classification problem. Thus we can easily utilize the posterior probability to infer and rank an individual. Consider two classes ζ_1 and ζ_2 and a feature vector ϕ, then according to the Bayes' theorem, the posterior probability for class ζ_1 is:

$$p(\zeta_1|\phi) = \frac{p(\phi|\zeta_1)p(\zeta_1)}{p(\phi|\zeta_1)p(\zeta_1) + p(\phi|\zeta_2)p(\zeta_2)} = \frac{1}{1+\exp(-a)} = \sigma(a) \tag{5}$$

where we have defined $a = \ln p(\phi|\zeta_1)p(\zeta_1)/p(\phi|\zeta_2)p(\zeta_2)$, $\sigma(\cdot)$ is the logistic function, $p(\phi|\zeta_k)$ is the class-conditional densities, $p(\zeta_k)$ is prior probability.

Similarly, class ζ_2(the no-conspirator class) can be represented as:

$$p(\zeta_2|\phi) = 1 - p(\zeta_1|\phi) \tag{6}$$

The inverse of the logistic sigmoid is given by

$$a = \ln(\frac{\sigma}{1-\sigma}) \tag{7}$$

and is known as the logit function.

6.2 Maximum Likelihood for Regression

As we have discussed above, the posterior probability of class ζ_1 can be written as a logistic function acting on a linear function of the feature vector ϕ so that:

$$p(\zeta_1|\phi) = y(\phi) = \sigma(\omega^T\phi), \tag{8}$$

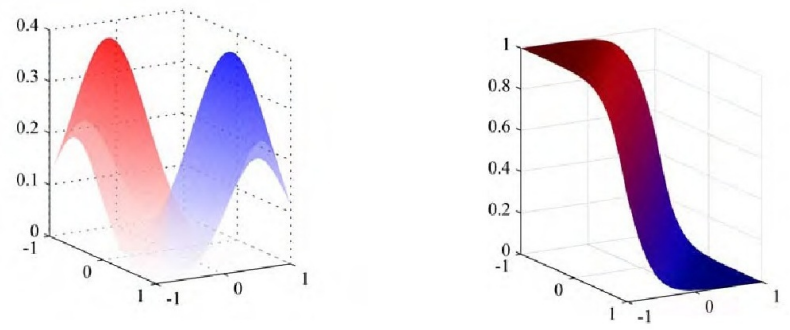

Figure 3: Relationship between Posterior Probability and Logistic Regression

The left-hand plot shows the class-conditional densities for two classes, denoted red and blue. On the right is the corresponding posterior probability $p(\zeta_1|x)$, which is given by a logistic sigmoid of a linear function of x. The surface in the right-hand plot is coloured using a proportion of red ink given by $p(\zeta_1|x)$ and a proportion of blue ink given by $p(\zeta_2|x) = 1 - p(\zeta_1|x)$.

where ω is the 14-dimensional coefficients, or parameters of the model.

In order to determine the parameters ω, we use the maximum likelihood method. With the data set (ϕ_i, t_i) ($t_i \in \{0, 1\}$ is the class-indicator, with 1 denoting conspirator), according to the derivative of the sigmoid function $d\sigma/da = \sigma(1-\sigma)$, the likelihood of the whole data set can be written as:

$$p(\mathbf{t}|\omega) = \prod_{i=1}^{n} y_i^{t_i}\{1 - y_i\}^{1-t_i}, \tag{9}$$

where $\mathbf{t} = (t_1, \ldots, t_n)^T$ and $y_i = p(\zeta_1|\phi_i) = \sigma(\omega^T\phi_i)$.

And the error function (log-likelihood) is in the form:

$$\mathcal{O}_E = -\ln p(t|\omega) = -\sum_{i=1}^{n} [t_i \ln y_i + (1 - t_i)\ln(1 - y_i)]. \tag{10}$$

And finally the derivative is:

$$\nabla\mathcal{O}_E = \sum_{i=1}^{n}(y_i - t_i)\phi_i. \tag{11}$$

The maximum likelihood problem could be solved by setting the derivative of Eq.(11) to be zero. We can also use the Laplace approximation to handle a more complicated Bayasian logistic regression [Bishop (2006)], where the parameter ω itself has a multivariate Gaussian prior.

7 Results

7.1 Prediction: Correctness and Robustness

To prove the correctness and robustness of the model, we use part of the known identities to predict the others and judge whether the prediction is accurate. Good models should have stable predictions.

- For requirement 1, we randomly pick 3 known conspirators from 7 (Jean, Alex, Paul) and 3 known non-conspirators from 8 (Chris, Paige, Darlene) as the predict-known information. Using the model we build (parameter $\delta = 0.01$) , 12 people are identified as conspirators (The workers with likelihood larger than 50% are conspirators) and the following is the rank list (For the workers have same name, we use the order in file "Names.xls" to distinguish, all the followings take the same strategy):

  Jean, **Paul**, **Alex**, **Yao**, **Elsie(7)**, **Ulf**, Neal(17), Beth(38), Elsie(37), Jerome(16), **Harvey**, Marion (The known conspirators in bold.)

  All the 7 known conspirators are in the list and all the 8 known non-conspriators are not in the list.

- For requirement 2, we also randomly pick 3 known conspirators from 8 (Chris, Jean, Alex) and 3 known non-conspirators from 7 (Paige, Darlene, Tran) as the predict-known information. Using the model we build (parameter $\delta = 0.01$), 8 people are identified as conspirators (The workers with likelihood larger than 50% are conspirators) and the following is the rank list:

  Chris, **Jean**, **Alex**, **Paul**, **Yao**, **Elsie(7)**, **Ulf**, Neal(17) (The known conspirators in bold.)

  7 known conspirators from 8 are in the list and all the 8 known non-conspriators are not in the list.

The prediction results flunction little changing the predict-known identifies. The high predict accuracy proves our model is both accurate and robust.

7.2 Our Results

In Requirement 1, we use the above model with parameter $\delta = 0.01$ and threshold likelihood 50%. The results shows that there are 15 conspirators and the top 20 likelihood rank list is as Table 1, the Table is read column by column and the right name, left likelihood to be conspirators, the calculated conspirators in bold:

To make the results more concise and directly, the likelihood of all workers is shown in Figure 4 by grayscale, the higher gray level, the higher probability (workers in the order of the file "Names.xls") .

In Requirement 2, we use the same parameter δ and threshold likelihood. The results shows that there are 12 conspirators and the likelihood rank list is as Table 2:

Table 1: Likelihood Rank of Requirement 1 (Top 20, Column by Column)

Yao	0.9999	**Alex**	0.9998	**Jerome(16)**	0.8462	Stephanie	0.2507
Paul	0.9999	**Elsie(7)**	0.9998	**Beth(38)**	0.8434	Priscilla	0.2498
Ulf	0.9999	**Neal(17)**	0.9908	**Marion**	0.7955	Christina	0.2458
Jean	0.9999	**Dolores**	0.9894	**William**	0.7211	Sherri	0.0487
Harvey	0.9999	**Elsie(37)**	0.9082	**Patrick**	0.5768	Gretchen(4)	0.0390

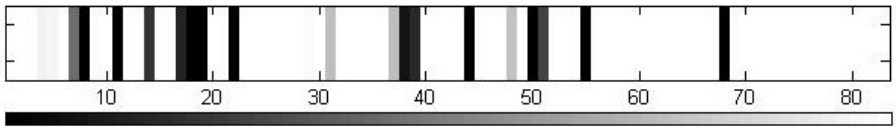

Figure 4: Likelihood Rank of Requirement 1
The horizontal axis represents the workers in order; the gray level represents the probability of being conspirators(the higher gray level, the higher probability).

Table 2: Likelihood Rank of Requirement 2 (Top 20, Column by Column)

Yao	0.9999	**Harvey**	0.9999	**Marion**	0.5926	Beth(38)	0.3031
Alex	0.9999	**Chris**	0.9999	**Christina**	0.5573	Sherri	0.2984
Paul	0.9999	**Elsie(7)**	0.9996	Julia	0.4367	Crystal	0.2615
Ulf	0.9999	**Neal**	0.9871	Jerome(16)	0.3762	William	0.1866
Jean	0.9999	**Dolores**	0.9825	Elsie(37)	0.3323	Neal(31)	0.1035

The likelihood of all workers is shown in Figure 5.

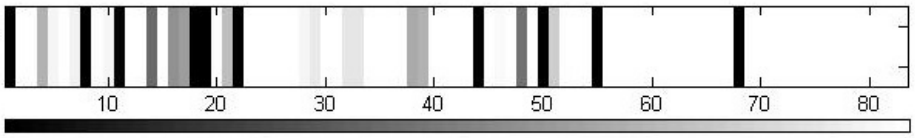

Figure 5: Likelihood Rank of Requirement 2
The horizontal axis represents the workers in order; the gray level represents the probability of being conspirators(the higher gray level, the higher probability).

The results can also help us judge whether the three senior managers Jerome, Delores, and Gretchen are involved. By the information from requirement 1, the model indicates that Delores and Jerome(16) are conspirators. If the information

from requirements 2 is used, Dolores is conspirator. Actually, the top 15 conspirators from both requirements are relatively the same, Jerome(16) is not the conspirator in the later situation just because the likelihood falls bellow 50%. So if Jerome(16) is the manager, the two managers Jerome and Delores both should be suspected.

7.3 Parameter Sensitiveness

In the model, there is just one parameter δ, then whether the results are sensitive to δ? We adjusted δ and the results are in Figure 6 (conspirator likelihood threshold is still 50%):

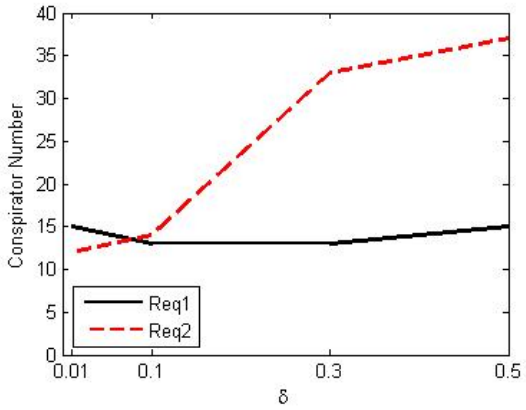

Figure 6: Influence of conspirator Number by δ

For every two likelihood rank lists, we measure the distance between them by calculating the *Spearman's rank correlation coefficient*, which is defined as $\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$, d_i is the ith-order difference, n is the total number. Table 3 is ρ between every two rank for different δ:

Table 3: ρ Between Ranks of Different δ

$\delta_1 - \delta_2$	$0.01 - 0.1$	$0.01 - 0.3$	$0.01 - 0.5$	$0.1 - 0.3$	$0.1 - 0.5$	$0.3 - 0.5$
ρ of Req 1	0.9571	0.9634	0.9194	0.8787	0.8231	0.9852
ρ of Req 2	0.9696	0.9070	0.8605	0.9643	0.9258	0.9877

As we can see, the correlation coefficient is relatively big, so the order of the rank list changes little. Actually, in most situations, the conspirator number changes little with δ, however in others, the absolute likelihood changes but not

the relative rank. So the threshold may be better considered. In sum, the likelihood rank list is insensitive to δ.

8 Model Analysis: a Revisit

Next we would like to take a step further, consider if we have access to the content and context of the message traffic, then what can we do with the information using tools from semantic network and text analysis? To the best of our knowledge, no existing model in this research field has been specially designed for handling the problem we are faced with. So we first revisit our original model for inspiration.

8.1 Explicit Representation for Commute Distance

In our model, we have used the so called commute distance to measure the similarity between an individual and a conspirator/non-conspirator. However, if we define $V = Q\Lambda^{-1/2} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n)^T$, it is easy to check the commute distance matrix C and V have the nice relationship that [Von Luxburg (2007)]:

$$\frac{1}{n}c_{ij} = (\mathbf{v}_i - \mathbf{v}_j)^T(\mathbf{v}_i - \mathbf{v}_j) = \|\mathbf{v}_i - \mathbf{v}_j\|_2^2. \tag{12}$$

It indicates that if we use $\mathbf{v}_i$ to denote each individual, then the commute distance, or the key information we would like to utilize later would be inherited in the form of Euclidean distance. This discovery has led to much research work in extracting features from a graph, or the so-called *graph embedding* problem.

8.2 Feature Extraction From a Graph

Given a graph W reflecting the intrinsic geometric structure of the network, the task of graph embedding is to extract features that preserve this structure. A classical graph embedding method is *Laplacian eigenmap* [Belkin and Niyogi (2001)], which achieves the result by preserving the local structures of the graph. It tries to minimize the following objective function under appropriate constraints:

$$\mathcal{Q} = \sum_{i=1}^{n}\sum_{i=1}^{n'}\sum_{z=1}^{r}(v_{iz} - v_{i'z})^2 W_{ii'}, \tag{13}$$

where $\mathbf{v}_i$ is the representation for i-th individual. The objective function with our choice of weight W_{ij} bears a heavy penalty if closely connected individuals i and j are mapped far apart. Therefore, minimizing it is an attempt to ensure that if i and j are "close" in the network, then v_i and v_j should be close as well.

Take the derivative of Eq.(13) and set it to be zero, we can find that an optimal embedding, or representation for individuals, is the (generalized) eigenvectors of the graph Laplacian matrix. Please see [Belkin and Niyogi (2001)] for the detailed derivation.

9 An Improved Model

From the above analysis, we can see that our model is essentially composed of three stages. The first is to find an explicit representation for each individual, or each node in the network that preserves certain properties. Then the pairwise distances between one individual and the identified individuals are calculated and then used to transform the representation. Finally, a logistic regression model is used to prioritize their probabilities of being part of the conspiracy.

It is important to note that the key stage or our model is the first one, *i.e.*, how to find an appropriate representation for later stages. Intuitively, merely with the messages one sent to others, we can already make a good inference on whether or not s/he is a conspirator. On the other hand, poor representation could lead to poor decisions. In fact, how to learn a proper representation is also a central problem in text analysis and even the whole area of artificial intelligence. Over the years, many models have been proposed and studied. Let us first give a brief description of the two models, topic models and Nonnegative Matrix Factorization (NMF) [Lee et al. (1999)]. They will both serve as basic models for our improved model.

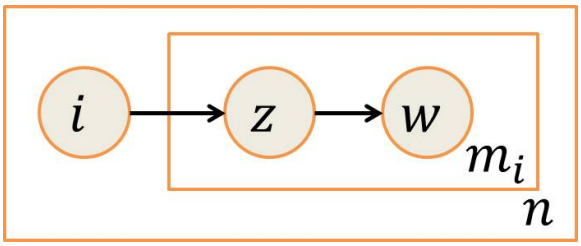

Figure 7: Plate notation representing the PLSI model.
i is the variable indexing individuals, z is a topic drawn from the topic distribution for this individual, $P(z|i)$, and w is a word drawn from the word distribution for this topic, $P(w|z)$. The i and w are observable variables, the topic z is a latent variable.

9.1 Topic Models

A topic model is a type of statistical model for discovering the abstract "topics" that occur in a collection of text data. One of the most classical topic models is *Probabilistic Latent Semantic Indexing* (PLSI) [Hofmann (1999)]. Specifically for the current case, we could view each individual in the company as a document, which is represented as a distribution over the topics. This is reasonable since different people have different topic focuses during the message traffic. For example, conspirators should be more likely to talk about embezzling funds from the company, while non-conspirators are more likely to talk about decent affairs. Moreover, different people would have different term preferences. From the case of Investiga-

tion EZ, Bob and Dave (both turn out to be criminals) used the slang "darn" more likely than others. Detecting such features is very helpful in identifying the guilty parties wisely.

Then how are the topics represented? In the PLSI model, each topic is a distribution over the words. This is also reasonable since when expressing different topics, people generally use different terms. For example, a topic on computer science would use words like algorithms, compilers, memories, *etc*. more frequently.

For such reasons, PLSI has been widely used to organize, summarize, and retrieve text data. Let's go more formally about this model in the following. Suppose the i-th person want to express the word w, it can be generated in the following way:

- First, pick a latent topic z with probability $P(z|i)$;
- Then, generate a word w with probability $P(w|z)$,

where the Multinomial distribution [Bishop (2006)] is often used to model the probabilities.

As a result, one obtains an observed pair (i, w), which means the i-th person expresses the word w, yet the latent topic is z is discarded. Then the joint probability $P(i, w)$ is:

$$P(i, w) = \sum_{z=1}^{r} P(z|i)P(w|z), \tag{14}$$

where r is the (predefined) number of topics.

Next, we could model the whole network using a probabilistic framework. Suppose we use X_{wi} to denote the number of occurrences of a word w in the text of individual i, the log-likelihood is:

$$\mathcal{J} = \sum_{i=1}^{n} \sum_{w=1}^{m} X_{wi} \log P(i, w), \tag{15}$$

where n is the number of individuals, and m is the vocabulary size. Many algorithms like Expectation-Maximization method [Bishop (2006)] have been developed to solve the maximum likelihood problem. Please see [Hofmann (1999)] for the details.

9.2 Nonnegative Matrix Factorization

Another model is Nonnegative Matrix Factorization, which has also been intensively applied to text processing and information retrieval. NMF focuses on matrices whose elements are all non-negative, which is often encountered in real world data such as intensity value of image pixels, document-term matrix, rating matrix, *etc*. Given such a matrix, NMF decomposes it into two non-negative matrices for low-rank approximation. In NMF, each data point can be explained as an additive-only linear combination of the nonnegative basis vectors, leading to a *parts-based representation* of very straightforward interpretability.

Suppose we have n data points of size m. Let $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) \in \mathbb{R}_+^{p \times n}$ denote the corresponding data matrix. Here $\mathbb{R}_+$ is the set of nonnegative real numbers. The standard NMF objective function is based on the Frobenius norm:

$$\min_{U,V} \mathcal{O}_F = \|X - UV^T\|_F^2, \tag{16}$$

or K-L Divergence:

$$\min_{U,V} \mathcal{O}_D = \sum_{i=1}^{n} \sum_{w=1}^{m} \frac{X_{wi}}{(UV^T)_{wi}} - X_{wi} + (UV^T)_{wi}, \tag{17}$$

where $U = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r) = [u_{wz}] \in \mathbb{R}_+^{m \times r}$ consists of r basis vectors, $V = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n)^T = [v_{iz}] \in \mathbb{R}_+^{n \times r}$ is the r-dimensional representation of the original inputs. A classical algorithm to solve the optimization problem is called the *multiplicative update* algorithm, which takes turns to optimize U and V.

It is important to note that the models PLSI and NMF have very close connections, *i.e.*, it is equivalent to maximize the log-likelihood Eq.(15) of PLSI and to minimize the objective function Eq.(17) of NMF. Please see [Ding et al. (2008)] for a detailed justification. Therefore, we will just focus on one of them, namely NMF, for further investigation.

9.3 Representation Enhancement

As is stated above, we would like to utilize the power of text analysis tools to enhance the effectiveness of our representation in the first stage, while it can still preserve the nice property and structure of distances. So a natural intuition is to incorporate the existing model into our framework. Take the NMF model as an example, we introduce the technique of *regularization*, and design a penalty term as:

$$\mathcal{P} = \sum_{i=1}^{n} \sum_{i'=1}^{n} \sum_{z=1}^{r} \left(v_{iz} \log \frac{v_{iz}}{v_{i'z}} + v_{i'z} \log \frac{v_{i'z}}{v_{iz}} \right) W_{ii'}, \tag{18}$$

which could be then be combined into the objective function Eq.(17) of NMF as:

$$\min_{U,V} \mathcal{O}_A = \sum_{i=1}^{n} \sum_{w=1}^{m} \frac{X_{wi}}{(UV^T)_{wi}} - X_{wi} + (UV^T)_{wi} + \lambda \mathcal{P}, \tag{19}$$

where $\lambda > 0$ is the regularization parameter, which can in turn be determined by model selection. This problem can also be solved by a multiplicative algorithm. Please see [Cai et al. (2011)] for the details.

The rationality of designing such a penalty term is that, as with Eq.(13), it tries to preserve the information about the structure of the network that the graph W transfers, so that the learned representations have the nice properties that will benefit later stages of the model.

Once we have learned the features, we can calculate the pairwise distances and then use logistic regression for prediction and prioritization.

9.4 Generalization

Similarly, for other applications where we have image or chemical data for the nodes, our goal is very much the same as the crime conspiracies and message data. On the one hand, we want to enhance the representation by using information from the data. On the other hand, we still want to keep the nice property of distance measure between an arbitrary individual and the identified ones so that the network structure is inherited in the representation. We would also turn to the dimension reduction techniques for help.

The motivation for us to choose dimension reduction to find the proper representation is that, many vectorized data (such as in computer vision, the image is linearized to a very high dimensional) have redundant and noisy features. Moreover, high dimensionality leads to high computational complexity. We can add our prior knowledge to any of the state-of-the-art dimension reduction methods, assuming that in the representation, points close to each other indicated by the graph should be close as well. This can be easily achieved by adding $\mathcal{P}$ or $\mathcal{Q}$ to the original objective function.

Finally when applied to solving the problems with very large databases of message traffic (thousands of people with tens of thousands of messages and possibly millions of words), we have a few notes about our model's scalability:

- For computing the commute distance, the most time-consuming part is to compute the pseudo inverse, or alternatively the eigenvalue decomposition. Fortunately the graph constructed in our model is quite sparse (with lots of zero entries), therefore many acceleration algorithms like the Nyström's method [Fowlkes et al. (2004)] can be applied.
- For the advanced model and its generalized versions, typically the optimization scheme has an iterative nature, thus we can stop it somewhere earlier, and make a compromise between the algorithm's effectiveness and efficiency.

References

Belkin, M. and Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*, 14:585–591.

Bishop, C. (2006). *Pattern recognition and machine learning*, volume 4. springer New York.

Cai, D., He, X., Han, J., and Huang, T. (2011). Graph regularized nonnegative matrix factorization for data representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1548–1560.

Ding, C., Li, T., and Peng, W. (2008). On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis*, 52(8):3913–3927.

Fouss, F., Pirotte, A., Renders, J., and Saerens, M. (2007). Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *Knowledge and Data Engineering, IEEE Transactions on*, 19(3):355–369.

Fowlkes, C., Belongie, S., Chung, F., and Malik, J. (2004). Spectral grouping using the nystrom method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):214–225.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM.

Lee, D., Seung, H., et al. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.