Factor Analysis of Social Aspects of Communication Networks

Diploma Thesis

By

Ioannis L. Sarris

Submitted to the Department of Electronic & Computer Engineering in partial fulfillment of the requirements for the ECE Diploma Degree.

Technical University of Crete

Advisor: Professor Sidiropoulos Nicholaos

Committee member: Professor Liavas Athanasios

Committee member: Professor Garofalakis Minos

September 2010

To my parents, sister and Alexandra for their patience and love.

Acknowledgement

I would like to express my gratitude to my advisor, Prof. Sidiropoulos Nicholas for his encouragement, understanding and expert guidance in the field of Communication Networks. His wide knowledge and logical way of thinking meant a great deal to me.

I would also like to thank my family for the support I received while doing my thesis. Last but not least, I would like to thank all of my friends for the encouragement given.

Abstract

Social network analysis has its origins in sociology. With the remarkable growth of online social networking sites, social network analysis has steered considerable interest in the fields of Computer Science and Telecommunications. The main objective of this thesis is to develop tools that enable extraction of social structures from communication network data (such as email, packet/flow or TCP data). The analysis aims to construct a map of the social network by analyzing communication patterns using traffic between nodes in the network. Such graphical interpretation of the data enables the data analyst to perform visual exploration and it is a first step in automated analysis using e.g., clustering tools. The main idea is that the higher the traffic between two nodes the lower their social distance. Hence, social distance can be defined as an appropriate monotone decreasing function of pairwise traffic. These pseudo-distances can be transformed to proper distances using a shortest path algorithm. We then invoke *Multidimensional Scaling* (*MDS*) to generate a social map from the processed distances. Exploiting the temporal dimension we further propose using *3-way MDS* to capture social dynamics. We illustrate our approach using the well-known Enron email corpus.

Contents

List of Figures

1

Introduction

Consider a network with N nodes that exchange messages. Let $\mathbf{M}$ be a matrix with elements m_{ij} equal to the number of messages that user i has sent to user j over a given period. We may define social distance as a monotically decreasing function of the number of messages $f(m_{ij})$. This models our intuition that social interaction implies a large number of exchanges. Possible choices for $f(m_{ij})$ could be:

- $f(m_{ij}) = \frac{1}{m_{ij}+c}$
- $f(m_{ij}) = Ae^{-bm_{ij}}$
- $f(m_{ij}) = A - bm_{ij}$

Since there are many choices that are consistent with our basic intuition, we need to investigate which ones are reasonable.

It is important to note that in order for $f(m_{ij})$ to be a distance metric, it must satisfy certain properties. Thus, for any two nodes (i, j) of the network:

- $f(m_{ij}) \geq 0$ (non-negativity).
- $f(m_{ij}) = 0$ only if $j = i$.
- $f(m_{ij}) = f(m_{ji})$. This can be enforced by construction, if we symmetrize (sum up) the message exchanges in both directions ($i \to j$ and $j \to i$).

- $f(m_{ij}) \leq f(m_{ik}) + f(m_{kj})$. This is not automatically ensured by "reasonable" choices of $f(.)$; however we propose using a shortest path algorithm for this purpose, as will be explained in the sequel.

Let $\mathbf{D}$ be a matrix holding the pairwise distances between the nodes of the network. In order to generate a social map of the users in the network from the given pairwise distance estimates, we can pose the following problem: given matrix $\mathbf{D}$ find points in 2-D or 3-D Euclidean space that generate these distances. This problem is known as *Multidimensional Scaling* (*MDS*) [3].

Noting that data may change over the time, we may aim to exploit the temporal dimension to better localize the individual nodes *and* capture social dynamics. Thus, instead of using classical matrix representation in order to store the data, we use a tensor array of order $N \times N \times K$ where N, K denote the number of network users and time-steps respectively. Consequently, we propose using *3-way MDS* and show superior results for the localization and tracking of dynamically changing datasets-in particular for the Enron data [12].

2

Multidimensional Scaling

MDS is a method that maps estimated distances between pairs of objects into a set of points, usually in a low-dimensional space, which approximately reproduce the given distances. MDS as a technique was discovered in Psychology [1, 2]. Since then, it has found numerous applications e.g., most recently for node localization in wireless sensor networks [4]. In this chapter we illustrate the basic idea behind *MDS* and its application to social networks..

2.1 Basic Idea

Denote the distance of object i and j as d_{ij}. The set of all distances between all objects yields the distance matrix $\mathbf{D}$. Let $\mathbf{X}_{\mathrm{N}\times\mathrm{m}}$ be the matrix of true coordinates of the nodes. Each row i of $\mathbf{X}$ indicates the coordinates of node i in m *dimensions*. The matrix of squared distances $\mathbf{P}$ can be expressed as [3]:

$$\mathbf{P} = \underline{\mathbf{c}}\,\underline{\mathbf{1}}^{\mathrm{T}} + \underline{\mathbf{1}}\,\underline{\mathbf{c}}^{\mathrm{T}} - 2\mathbf{X}\mathbf{X}^{\mathrm{T}} = \underline{\mathbf{c}}\,\underline{\mathbf{1}}^{\mathrm{T}} + \underline{\mathbf{1}}\,\underline{\mathbf{c}}^{\mathrm{T}} - 2\mathbf{B} \tag{2.1}$$

where $\underline{\mathbf{c}}$ is a $N \times 1$ vector of the diagonal elements of matrix $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ and $\underline{\mathbf{1}}$ is a $N \times 1$ vector of ones.

Multiplying from left and right by the centering operator $\mathbf{J} = \mathbf{I} - \underline{\mathbf{1}}\underline{\mathbf{1}}^{\mathrm{T}}/N$ and by the factor $-\frac{1}{2}$ gives:

$$\begin{aligned}
-\frac{1}{2}\mathbf{JPJ} &= -\frac{1}{2}\mathbf{J}(\underline{\mathbf{c}}\,\underline{\mathbf{1}}^{\mathrm{T}} + \underline{\mathbf{1}}\,\underline{\mathbf{c}}^{\mathrm{T}} - 2\mathbf{X}\mathbf{X}^{\mathrm{T}})\mathbf{J} \\
&= -\frac{1}{2}\mathbf{J}\underline{\mathbf{c}}\,\underline{\mathbf{1}}^{\mathrm{T}}\mathbf{J} - \frac{1}{2}\mathbf{J}\underline{\mathbf{1}}\,\underline{\mathbf{c}}^{\mathrm{T}}\mathbf{J} + \frac{1}{2}\mathbf{J}(2\mathbf{X}\mathbf{X}^{\mathrm{T}})\mathbf{J} \\
&= -\frac{1}{2}\mathbf{J}\underline{\mathbf{c}}\,\underline{\mathbf{0}}^{\mathrm{T}} - \frac{1}{2}\underline{\mathbf{0}}\,\underline{\mathbf{c}}^{\mathrm{T}}\mathbf{J} + \mathbf{J}(\mathbf{X}\mathbf{X}^{\mathrm{T}})\mathbf{J} \\
&= (\mathbf{I} - \frac{\underline{\mathbf{1}}\,\underline{\mathbf{1}}^{\mathrm{T}}}{N})\mathbf{X}\mathbf{X}^{\mathrm{T}}(\mathbf{I} - \frac{\underline{\mathbf{1}}\,\underline{\mathbf{1}}^{\mathrm{T}}}{N}) \\
&= (\mathbf{X}\mathbf{X}^{\mathrm{T}} - \mathbf{X}\mathbf{X}^{\mathrm{T}}\frac{\underline{\mathbf{1}}\,\underline{\mathbf{1}}^{\mathrm{T}}}{N} - \frac{\underline{\mathbf{1}}\,\underline{\mathbf{1}}^{\mathrm{T}}}{N}\mathbf{X}\mathbf{X}^{\mathrm{T}} + \frac{\underline{\mathbf{1}}\,\underline{\mathbf{1}}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\underline{\mathbf{1}}\,\underline{\mathbf{1}}^{\mathrm{T}}}{N^2}) \\
&= \mathbf{X}\mathbf{X}^{\mathrm{T}} = \mathbf{B}.
\end{aligned} \tag{2.2}$$

Given matrix $\mathbf{B}$ or a noisy estimate thereof we can then determine $\mathbf{X}$ by minimizing the function:

$$h(\mathbf{X}) = ||\mathbf{B} - \mathbf{X}\mathbf{X}^{\mathrm{T}}||_{\mathrm{F}}^2 \tag{2.3}$$

Thus, the node coordinates can be estimated by the n principal eigenvectors of $\mathbf{B}$.

$$\mathbf{B} \approx (\mathbf{U}_{\mathrm{n}}\mathbf{\Lambda}_{\mathrm{n}}^{1/2})(\mathbf{\Lambda}_{\mathrm{n}}^{1/2}\mathbf{U}_{\mathrm{n}}^{\mathrm{T}}) = \mathbf{Y}\mathbf{Y}^{\mathrm{T}} \tag{2.4}$$

Result: *The application of the classical MDS procedure with the correct number of dimensions and for the true distance matrix* $\mathbf{P}$*, returns estimates of the coordinates* $\mathbf{Y}$ *of all nodes of the network that are equal to the true coordinates* $\mathbf{X}$ *(up to rotation, reflection and translation)* [13].

2.2 A simple illustration of Classical MDS

A good way to understand the main idea of Classical MDS is by giving a simple example. Consider the distances between 5 cities measured on a map (see Table 2.1). Based only on the measurements of Table 2.1, our intention is to constuct a map of 5 points such that the distances between these points are equal to the distances

between the 5 cities in the original map. Note that the reconstructed map in Figure 2.1 correspond to the true distances between the 5 cities in Table 2.1.

Table 2.1: Distances between 5 cities in km

	Athens	Salonica	Herakleion	Istanbul	Rome
Athens	0	344	359	637	1230
Salonica	344	0	698	568	1040
Herakleion	359	698	0	806	1530
Istanbul	637	568	806	0	1590
Rome	1230	1040	1530	1590	0

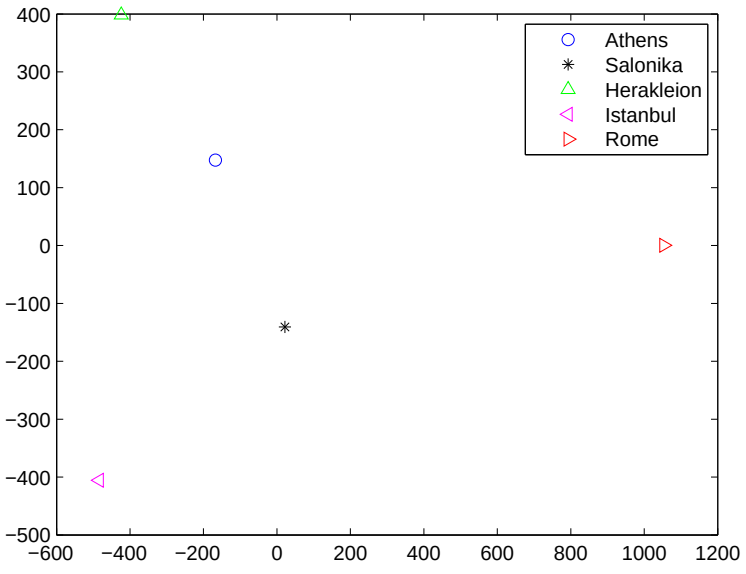


Figure 2.1: Reconstruction of the map of 5 cities using MDS

Although the reconstructed map has an unconventional orientation, this can be easily adjusted, if we look at the map upside-down. This is the main problem of

bilinear decomposition that is discussed below: **rotational freedom**

2.3 Classical MDS and social network analysis

In the following section, we present an example applied to social network clique analysis. Firstly, we create 25 points in the Euclidean space in such a way in order to create 5 groups of 5 points each (c.f. Figure 2.2).

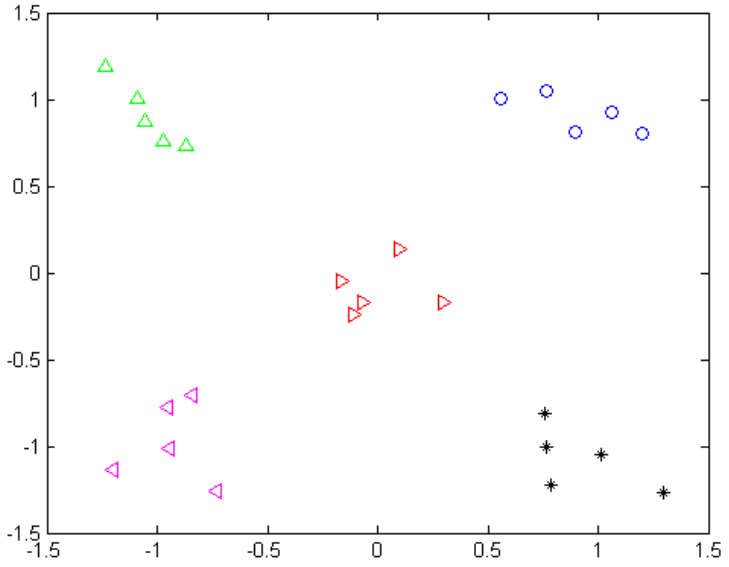


Figure 2.2: Illustration of 5 groups of 5 nodes.

Afterwards, we generate packets between the nodes at a rate of packet (probability generation and transmission) that is inversely proportional to their distance. Hence, $m_{ij} = \frac{1}{d_{ij}}$, where m_{ij} is the number of packets that user i sends to user j and d_{ij} their distance for $i, j = 1 \dots 25$. Thus, we model 5 groups of 5 individuals sending packets to each other. We define the social distance of the nodes as $\hat{d}_{ij} = f(m_{ij}) = \frac{1}{m_{ij}}$. Then, we use Dijkstra algorithm for each node in order to transform the pseudo - distances to proper distances. Finally, *MDS* is applied using *Singular Value Decomposition (SVD)* in order to cluster them in groups (c.f. Figure 2.3). Notice the difference between Figure 2.2 and Figure 2.3 because of **rotational freedom**.

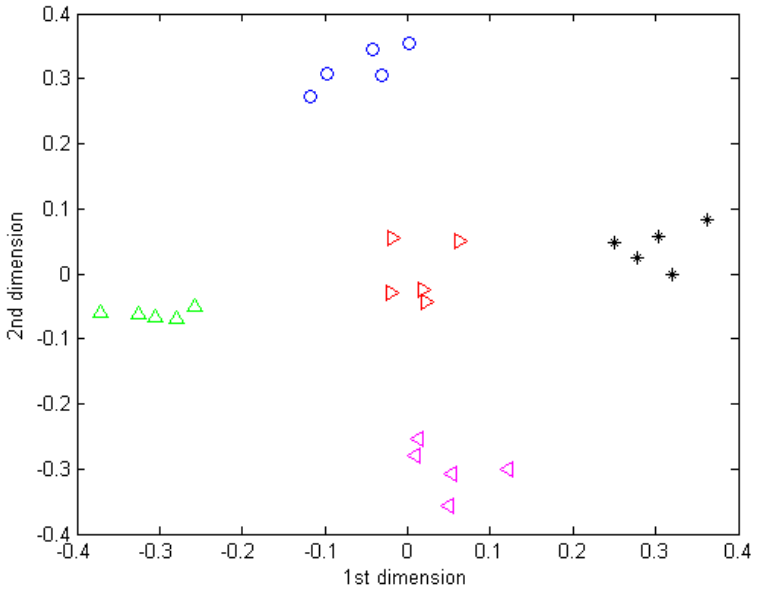


Figure 2.3: Illustration of 5 cliques using Classical MDS.

3

All pairs shortest path algorithm

Consider a weighted, directed graph $G = (V, E)$ with weight function $w : E \to \mathbb{R}$ that maps edges to real-valued numbers. The weight of path $p = \langle u_0, u_1, \ldots, u_n \rangle$ can be defined as the sum of the weights of its constituent edges:

$$w(p) = \sum_{j=1}^{n} w(u_{j-1}, u_j)$$

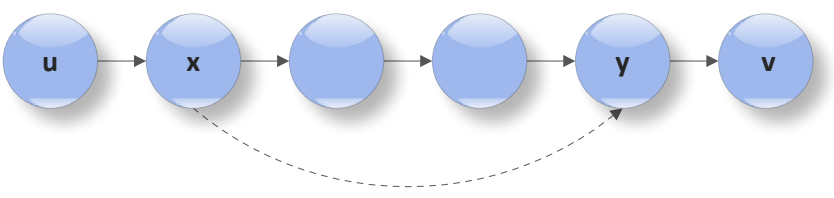
Then the *shortest path* from vertex u to vertex v is:

$$\delta(u, v) = \begin{cases} \min \left\{ w(p) : u \xrightarrow{p} v \right\} & \text{if there is a path from u to v,} \\ \infty & \text{otherwise.} \end{cases}$$

3.1 Properties of shortest paths

Theorem 1: (Optimal substructure) *A subpath of a shortest path is a shortest path.*

Proof



Let's consider a subpath $p(x, y)$ of the shortest path $\delta(u, v)$. Assume that $p(x, y)$ isn't a shortest path. Then, there exists a shortest path $\delta(x, y)$ that goes from x to y. If we have a shorter path from x to y than $p(x, y)$ then we can replace the older path $p(x, y)$ with the shorter one $\delta(x, y)$. Then there must exist a path from u to v which is shorter than the shortest one $\delta(u, v)$ which is a contradiction.

The following is a key property of shortest paths:

Theorem 2: (Triangle inequality) *$\forall u, v, x \in V$, we have $\delta(u, v) \leq \delta(u, x) + \delta(x, v)$* where $\delta(x, y)$ denotes the length of the shortest path from x to y.

Proof *By contradiction and the definition of shortest path.*

3.2 All pairs shorthest paths problem

All pairs shortest paths is an optimization problem that finds all shortest paths for every pair of u and v in the network graph $G(V, E)$. This problem can be solved:

- by executing a *single-source shortest paths* algorithm for all the vertices V of the graph.
- by executing an *all pairs shortest paths* algorithm

Single-source shortest paths algorithms

Consider a weighted, directed graph $G(V, E)$ with source s and weighted function $w : E \rightarrow \mathbb{R}$. The *Bellman-Ford* algorithm solves the single source shortest paths problem in which edge weights may be negative. If there is a negative-weight cycle that is reachable from the source, the algorithm indicates that there is no solution. Otherwise, the algorithm finds the shortest paths and their weights from the source. The running time of *Bellman-Ford* algorithm is $O(|V||E|)$ and by running *Bellman-Ford* process $|V|$ times the complexity is $O(|V|^2|E|)$.

Algorithm 1 BELLMAN-FORD ALGORITHM

Ensure: The shortest paths from s to all $u \in V - s$.

```
1: INITIALIZE-SINGLE-SOURCE(G, s)
2: for i ← 1 to |V[G]| − 1 do
3:     for all (u, v) ∈ E[G] do
4:         RELAX(u, v, w)
5:     end for
6: end for
7: for all (v, u) ∈ E[G] do
8:     if d[v] ≥ d[u] + w(u, v) then
9:         return FALSE
10:    end if
11: end for
12: return TRUE
```

Algorithm 2 INITIALIZE-SINGLE-SOURCE(G, s)

```
1: for all v ∈ V[G] do
2:     d[v] ← ∞
3:     pr[v] ← ∅
4: end for
5: d[s] ← 0
```

Algorithm 3 RELAX(u, v, w)

```
1: if d[v] ≥ d[u] + w(u, v) then
2:     d[v] ← d[u] + w(u, v)
3:     pr[v] ← u
4: end if
```

On the other hand, *Dijkstra* algorithm solves the single source shortest paths problem in which all edges weights are nonnegative. Given a set S of vertices whose shortest paths are known from the source s, the algoritm selects the vertex $u \in V - S$

whose estimated distance from s is the minimum shortest path, and updates all estimated distances of vertices that are adjacent to u. In our implementation we use a min-priority queue Q of vertices. The running time of $Dijkstra$ algorithm depends on how the min-priority queue is implemented. If we use an array the complexity of the algorithm is $O(|V|^2)$ and by running $|V|$ times *Dijkstra* process the complexity is $O(|V|^3)$. For the single sourtest paths case, we can achieve a running time of $O(|V| \log |V| + |E|)$ by implementing the min-priority queue with a Fibonacci heap [10].

Algorithm 4 DIJKSTRA ALGORITHM

Require: The source vertex s and the min-priotity queue Q of $G(V, E)$.

Ensure: The shortest paths from s to all $u \in V - s$.

```
1: INITIALIZE-SINGLE-SOURCE(G, s)
2: S ← ∅
3: Q ← V[G]
4: while Q ≠ ∅ do
5:     u ← EXTRACT-MIN(Q).
6:     S ← S ∪ u
7:     for all v ∈ Adj[u] do
8:         RELAX(u, v, s)
9:     end for
10: end while
```

All pairs shortest paths algorithms

Floyd-Warshall algorithm solves the all-pairs shortest-paths problem on a directed graph $G(V, E)$. Negative-weight edges may be present but no negative-weight cycles. We define a recursive to the all-pairs shortest-paths problem. Let $d_{ij}^{(k)}$ be the weight of a shortest path from vertex i to vertex j for which all intermediate vertices are in $\{1, 2, \ldots, k\}$. When $k = 0$, then a path from vertex i to vertex j with no intermediate vertex numbered higher than 0, has no intermediate vertices. Thus, $d_{ij}^{(0)} = w_{ij}$. A recursive solution is given by

$$d_{ij}^{(k)} \begin{cases} w_{ij} & \text{if } k = 0, \\ \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}) & \text{if } k \geq 1. \end{cases}$$

$Floyd - Warshall$ algorithm runs in $\Theta(|V|^3)$.

Algorithm 5 FLOYD-WARSHALL ALGORITHM

Require: The weight matrix W $(n \times n)$.

Ensure: All shortest paths for every pair $(u, v) \in G(V, E)$.

```
1: n ← rows[W]
2: D^(0) ← W
3: for k ← 1 to n do
4:     for i ← 1 to n do
5:         for j ← 1 to n do
6:             d_ij^(k) ← min(d_ij^(k-1), d_ik^(k-1) + d_kj^(k-1))
7:         end for
8:     end for
9: end for
10: return D^(n)
```

In our case we have to solve the all-pairs shortest-paths problem on a undirected, dense graph $G(V, E)$ with weight function $w : E \rightarrow \mathbb{R}^+$. As we presented above, the running time of *Bellman-Ford* algorithm is the most costly; the *Bellman-Ford* complexity depends on the number of edges $|E|$ of the graph $G(V, E)$ and when the graph is dense $|E| >> |V|$. The above optimization problem can be solved by using either *Dijkstra* or Floyd-Warshall algorithm as described above. The pseudocodes of all the above algorithms were collected from [10].

4

Three way multidimensional scaling

The information in large or complex datasets is often difficult to describe. This is common in real world applications where the data are not static and change over time. Exploiting the temporal dimension, we use *3-way MDS* to localize better the individual nodes and capture social dynamics. The distance - estimates are stored in a *tensor array* of order $N \times N \times K$ where N, K denote the number of network users and time-steps respectively.

4.1 Individual Scaling

Given the double centered distance matrices $\mathbf{S}_{\mathrm{i}}$ for $\mathrm{i} = 1, \ldots, K$, our goal is to minimize the function

$$h(\mathbf{X}, \mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_{\mathrm{K}}) = \sum_{\mathrm{i}=1}^{\mathrm{K}} ||\mathbf{S}_{\mathrm{i}} - \mathbf{X}\mathbf{W}_{\mathrm{i}}\mathbf{X}^{\mathrm{T}}||_{\mathrm{F}}^2 \tag{4.1}$$

where $\mathbf{X}$ is a $I \times r$ matrix, $\mathbf{W}_{\mathrm{i}}$ is a diagonal weight matrix of size $r \times r$. The optimization problem described above is called *INDSCAL*. There is no analytical solution for the equation (c.f. Equation 4.1) for any $\mathbf{X}$ and $\mathbf{W}_{\mathrm{i}}$. Caroll and Chang [5] proposed a method called *CANDECOMP/PARAFAC* that solves the above equation iteratively.

More specifically, given the equation:

$$h(\mathbf{X}, \mathbf{Y}, \mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_{\mathrm{K}}) = \sum_{\mathrm{i}=1}^{\mathrm{K}} ||\mathbf{S}_{\mathrm{i}} - \mathbf{X}\mathbf{W}_{\mathrm{i}}\mathbf{Y}^{\mathrm{T}}||_{\mathrm{F}}^2 \tag{4.2}$$

where $\mathbf{X}$ is a $I \times r$ matrix, $\mathbf{Y}$ is a $J \times r$ matrix, $\mathbf{W}_{\mathrm{i}}$ is a diagonal matrix $r \times r$ and r is the rank approximation that we want to achieve, Caroll and Chang claimed that, when *CANDECOMP* converges $\mathbf{X}$ and $\mathbf{Y}$ are column-wise proportional. Note that each row i of $\mathbf{X}$ indicates the coordinates of node i in r dimensions. The dimension weights w_{rri} for every dimension r and time-slice i are non-negative and the time differences are possible only in the weights on the dimensions of $\mathbf{X}$.

4.2 Parallel Factor Analysis

4.2.1 Linear Algebra Properties

Definition 1: *The rank of a matrix* $\mathbf{X}$ *can be defined as the minimum number of outer products (rank-one factors) that generate* $\mathbf{X}$ *as their sum.*

$$\mathbf{X} = \underline{\mathbf{a}}_1\underline{\mathbf{b}}_1^{\mathrm{T}} + \underline{\mathbf{a}}_2\underline{\mathbf{b}}_2^{\mathrm{T}} + \ldots + \underline{\mathbf{a}}_{\mathrm{r}}\underline{\mathbf{b}}_{\mathrm{r}}^{\mathrm{T}}$$

or alternatively,

$$\mathbf{X} = \mathbf{A}_{\mathrm{r}}\mathbf{B}_{\mathrm{r}}^{\mathrm{T}}$$

where $\mathbf{X} \in \mathbb{C}^{I \times J}$ matrix, $\underline{\mathbf{a}}_{\mathrm{i}}$, $\underline{\mathbf{b}}_{\mathrm{i}}$ for $\mathrm{i} = 1, \ldots, r$ are so - called "loading"/ "score" column vectors, $\underline{\mathbf{a}}_{\mathrm{i}}\underline{\mathbf{b}}_{\mathrm{i}}^{\mathrm{T}}$ are the rank - one factors, and r is the rank of matrix $\mathbf{X}$.

Property 1: *Given* $\mathbf{X} = \mathbf{A}_{\mathrm{r}}\mathbf{B}_{\mathrm{r}}^{\mathrm{T}}$ *there are infinitely many equivalent decompositions of* $\mathbf{X}$*. This is the basic principle behind bilinear decomposition, called rotational freedom.*

Proof

Let $\mathbf{X}$ be a matrix of order $(I \times J)$. Suppose that we have found $\mathbf{A}$, $\mathbf{B}$ that decompose $\mathbf{X}$. So,

$$\mathbf{X} = \mathbf{A}\mathbf{B}^{\mathrm{T}} = \mathbf{A}\mathbf{M}\mathbf{M}^{-1}\mathbf{B}^{\mathrm{T}} = \mathbf{A}\mathbf{M}(\mathbf{B}\mathbf{M}^{-\mathrm{T}})^{\mathrm{T}} = \mathbf{A}_1\mathbf{B}_1^{\mathrm{T}}$$

where $\mathbf{M}$ is a nonsingular matrix, $\mathbf{A}_1 = \mathbf{AM}$ and $\mathbf{B}_1 = \mathbf{BM}^{-\mathrm{T}}$.

Let's now consider a three-way array $\underline{\mathbf{X}} \in \mathbb{C}^{I \times J \times K}$. The *trilinear decomposition*, also known as *PARAFAC*, represents $\underline{\mathbf{X}}$ as the sum of outer products of three vectors. Thus, for each slice i:

$$\mathbf{X}_{\mathrm{i}} = \sum_{\mathrm{r}=1}^{\mathrm{R}} \underline{\mathbf{a}}_{\mathrm{r}} \underline{\mathbf{b}}_{\mathrm{r}}{}^{\mathrm{T}} c_{\mathrm{ir}}$$

or alternatively,

$$\mathbf{X}_{\mathrm{i}} = \mathbf{A}\mathbf{D}_{\mathrm{i}}(\mathbf{C})\mathbf{B}^{\mathrm{T}}.$$

where $\underline{\mathbf{a}}_{\mathrm{r}}$, $\underline{\mathbf{b}}_{\mathrm{r}}$ are the rth columns of the loading matrices $\mathbf{A} \in \mathbb{C}^{I \times R}$, $\mathbf{B} \in \mathbb{C}^{J \times R}$, c_{ir} is the element of $\mathbf{C} \in \mathbb{C}^{K \times R}$ and $\mathbf{D}_{\mathrm{i}}(\mathbf{C})$ is a diagonal matrix with main diagonal constructed by the *ith* row of $\mathbf{C}$.

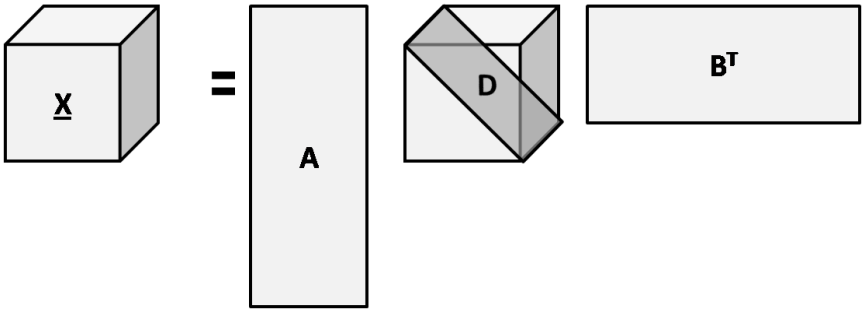


Figure 4.1: A representation of PARAFAC decomposition, where $\mathbf{A} \in \mathbb{C}^{I \times r}$, $\mathbf{B} \in \mathbb{C}^{J \times r}$ and $\mathbf{D} \in \mathbb{C}^{r \times r \times K}$.

Property 2: *Trilinear decomposition is unique under mild assumptions.*
The necessary and sufficient conditions for the uniqueness of this analysis is described in the cited literature [8, 6, 9].

4.2.2 Khatri-Rao product

The *Khatri-Rao product (KR product)* can be defined as the columnwise Kronecker product of two matrices with the same number of columns F.

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1 | \mathbf{a}_2 \otimes \mathbf{b}_2 | \ldots | \mathbf{a}_{\mathrm{F}} \otimes \mathbf{b}_{\mathrm{F}}]$$

or alternatively,

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \mathbf{BD}_1(\mathbf{A}) \\ \mathbf{BD}_2(\mathbf{A}) \\ \vdots \\ \mathbf{BD}_\mathrm{I}(\mathbf{A}) \end{bmatrix} \tag{4.3}$$

where $\mathbf{A} \in \mathbb{C}^{I \times R}$, $\mathbf{B} \in \mathbb{C}^{J \times R}$ and $\odot$, $\otimes$ are the *Khatri Rao* and *Kronecker* operator respectively.

The *PARAFAC* model can be expressed in different ways as described above. Thus, let's consider again the three way array $\underline{\mathbf{X}} \in \mathbb{C}^{I \times J \times K}$. A useful formulation of $\underline{\mathbf{X}}$ in matrix representation is by the use of the *KR product*:

$$\mathbf{X}^{(JI \times K)} = (\mathbf{A} \odot \mathbf{B})\mathbf{C}^\mathrm{T}. \tag{4.4}$$

$$\mathbf{X}^{(IK \times J)} = (\mathbf{C} \odot \mathbf{A})\mathbf{B}^\mathrm{T}. \tag{4.5}$$

$$\mathbf{X}^{(KJ \times I)} = (\mathbf{B} \odot \mathbf{C})\mathbf{A}^\mathrm{T}. \tag{4.6}$$

where $\mathbf{X}^{(JI \times K)}$, $\mathbf{X}^{(IK \times J)}$ and $\mathbf{X}^{(KJ \times I)}$ are obtained by unfolding $\underline{\mathbf{X}}$ along the third, second and first dimension respectively.

4.2.3 Trilinear Alternative Least Squares

A widely used method that fit the *PARAFAC* model is the *TALS-Trilinear Alternative Least Squares* algorithm. Consider,

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \|\mathbf{X}^{(JI \times K)} - (\mathbf{A} \odot \mathbf{B})\mathbf{C}^\mathrm{T}\|_\mathrm{F}^2.$$

1. Initialization of **A, B, C**.
2. Least Squares updates:
$\mathbf{A}^\mathrm{T} = (\mathbf{B} \odot \mathbf{C})^\dagger \mathbf{X}^{(KJ \times I)}$. $\quad \mathbf{B}^\mathrm{T} = (\mathbf{C} \odot \mathbf{A})^\dagger \mathbf{X}^{(IK \times J)}$.
$\mathbf{C}^\mathrm{T} = (\mathbf{A} \odot \mathbf{B})^\dagger \mathbf{X}^{(JI \times K)}$.
3. Go to step 2 until convergence criterion holds.

Table 4.1: TALS

where † denotes matrix pseudo-inverse.

4.3 Application of 3-way MDS to social networks

In this section we present an example of *3-way MDS* applied to social networks that change over the time, using *trilinear* and *singular value decomposotion - SVD*.

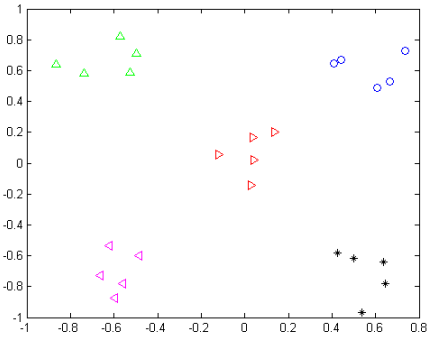
We create 25 points in the 2-D space in such a way in order to create 5 groups of 5 nodes each. The coordinates of the nodes are stored in a matrix $\mathbf{X} \in \mathbb{R}^{25\times 2}$ (c.f. Equation 4.1). In order to model a network that dynamically change, the node coordinates are multiplied by a diagonal matrix $\mathbf{W}_k \geq 0$ for $k = 1 \ldots 5$. Afterwards, we generate packets between the nodes at a rate of packet (probability generation and transmission) that is inversely proportional to their distance as in section (2.3). Thus, $m_{ijk} = \frac{1}{d_{ijk}}$ where m_{ijk} is the number of packets that user i sends to user j at k-th time slice and d_{ijk} their distance for i, $j = 1 \ldots 25$ and $k = 1 \ldots 5$. Thus, we define the social distance of the nodes as $\hat{d}_{ijk} = \frac{1}{m_{ijk}}$. Instead of using classical matrix representation in order to store the data, we use a tensor array $\underline{\mathbf{S}} \in \mathbb{R}^{25\times 25\times 5}$.

4.3.1 Noiseless case

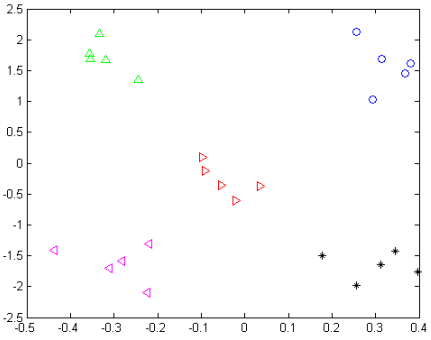
Firstly, we sum the *3-way array* $\underline{\mathbf{S}}$ along the time dimension. Then, by using *Dijkstra algorithm* we transform the pseudo-distances to proper distances. Classical MDS is applied, by the use of SVD, in order to localize the individual nodes in the Euclidean space.

Secondly, we use the 3-way MDS model in order to decompose the tensor $\underline{\mathbf{S}} \in \mathbb{R}^{25\times 25\times 5}$. After the application of *Dijkstra algorithm* for each time slice, 3-way MDS is applied by the use of *Parafac*[1] decomposition. The above procedures have allowed to classify the 5 cliques of the network.

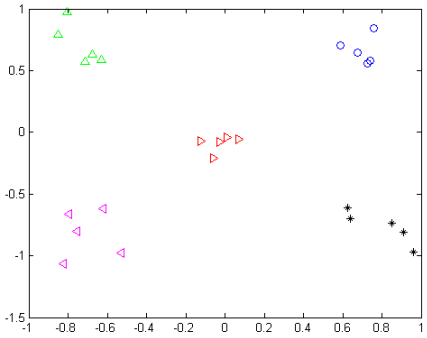
[1]We use parafac procedure of *nway Toolbox for Matlab* (ver. 7.3) under the constraint: C-mode non-negativity constraint. We use non-negativity of the weights according to the weighted Euclidean model.



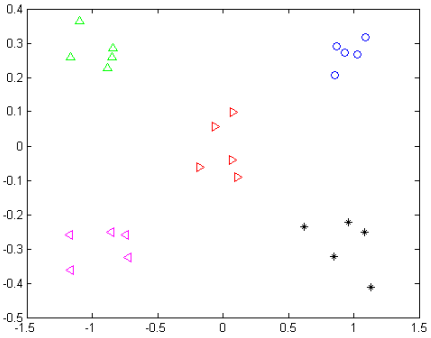
(a) k = 1



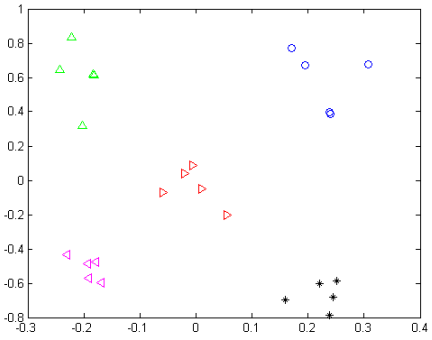
(b) k = 2



(c) k = 3



(d) k = 4



(e) k = 5

Figure 4.2: Illustration of 5 groups of 5 nodes that dynamically change.

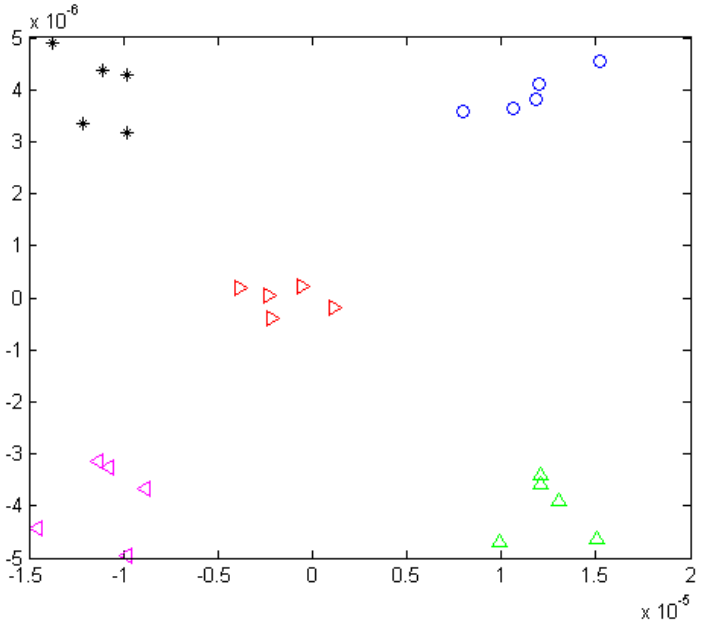


Figure 4.3: Illustration of 5 cliques using parafac procedure (noiseless case).

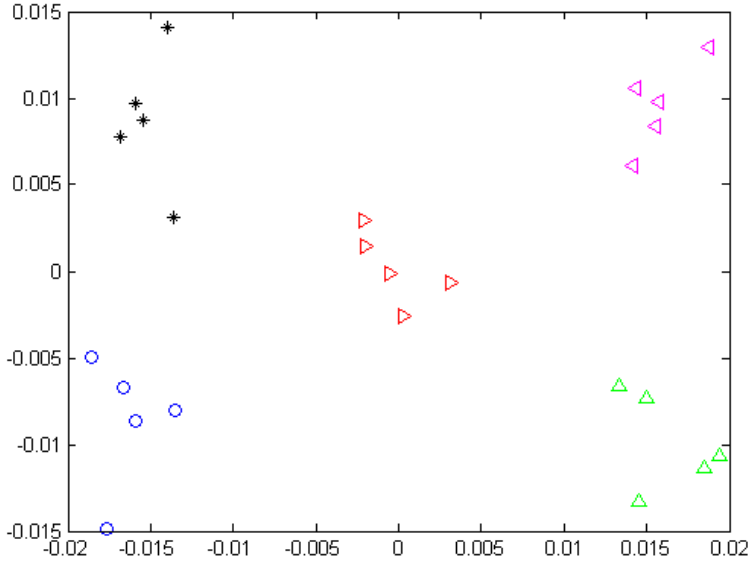


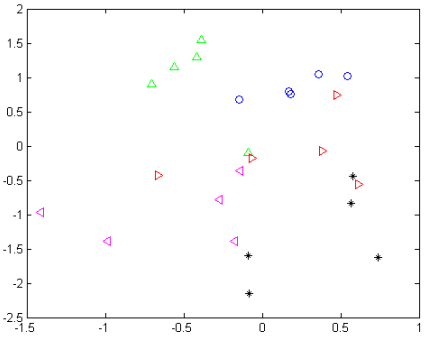
Figure 4.4: Illustration of 5 cliques using SVD (noiseless case).

4.3.2 Noisy case

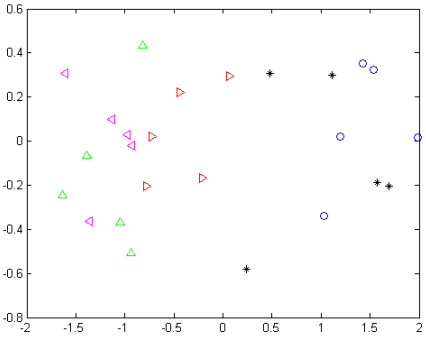
In the presence of noise the matrix $\mathbf{X} \in \mathbb{R}^{25\times 2}$ that holds the node coordinates becomes:

$$\mathbf{X} = \mathbf{X} + \mathbf{W}$$

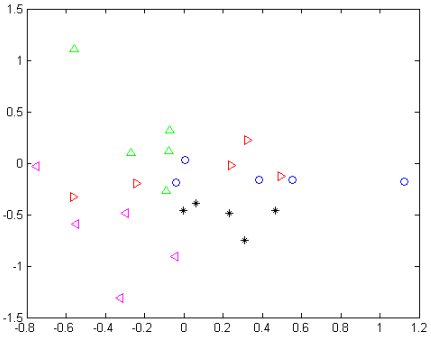
Note that noise can model data inaccuration or even the loss of data. We use again SVD as described above to represent the nodes of the graph. Unlike *SVD*, *3-way MDS* using *Parafac* has allowed to classify the cliques of the network. The reason for this is that *3-way MDS* assumes and exploits more structure in the data, which structure in this case is correct (by construction). The same structure often holds because different individuals weight latent dimensions differently over time e.g., as their needs/ life evolve.



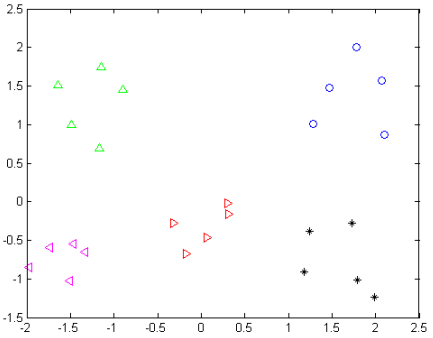
(a) k = 1



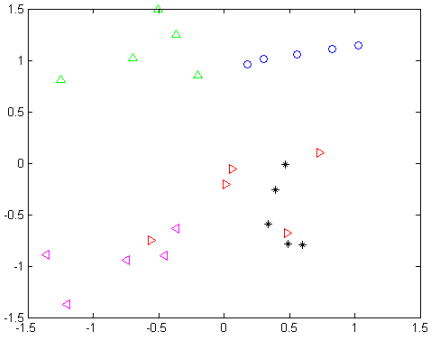
(b) k = 2



(c) k = 3



(d) k = 4



(e) k = 5

Figure 4.5: Illustration of 5 groups of 5 nodes that dynamically change (noisy case).

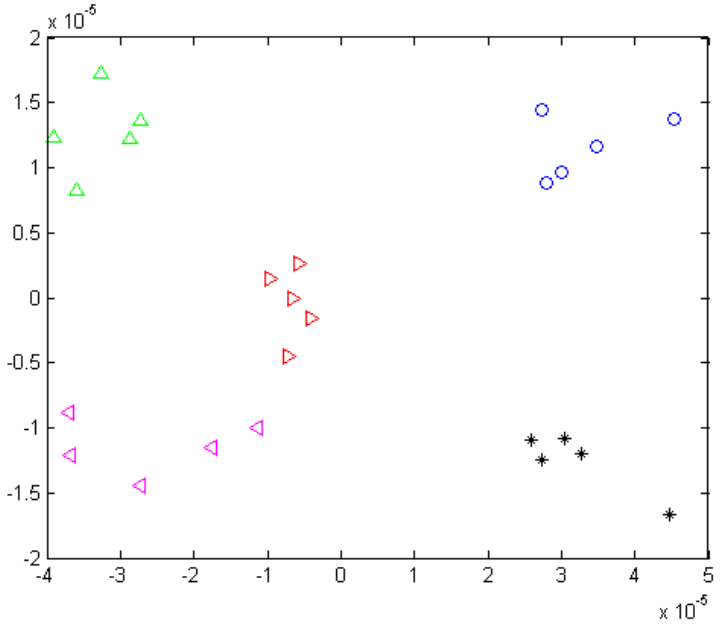


Figure 4.6: Illustration of 5 cliques using parafac procedure (noisy case).

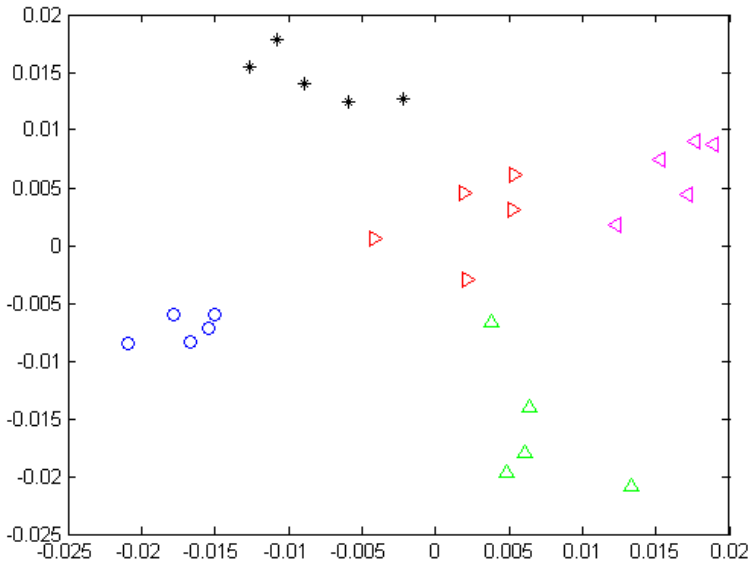


Figure 4.7: Illustration of 5 cliques using SVD (noisy case).

5

Enron email data processing

Enron was a U.S. energy company that was formed in 1985, under the direction of Keneth Lay, through the merger of a utility and a gas pipeline company. Enron quickly became the nation's seventh-largest company in revenue by buying electricity from generators and selling it to consumers. From 1999, Enron created offshore "special purpose entities" (SPE), to hide losses from equity and look more profitable than it actually was. In August 2000, Enron's stock price hit its highest value of 90$ billions. In December 2000, Jeffrey Skilling took over the position of chief executive from Keneth Lay and helped make Enron the biggest wholesaler of gas and electricity, trading over 27$ billions per quarter.

At this point Enron executives, who possessed the inside information on the hidden losses, began to sell their stock, while at the same time, the Enron's investors were told to buy the stock. As the executives sold their shares, the price began to drop. In August 2001 Skilling surprisingly resigned, stating personal reasons for quitting. As October closed, the Enron's stock price had fallen to 15$ billions. At the end of 2001 Enron filed for bankruptcy and the "Enron scandal" quickly followed. The Securities and Exchange Commission (SEC) and the Federal Energy Regulatory Commission (FERC) started inquiry into Enron. In May 2002, FERC released a corpus of the emails from about 150 employees during its investigation.

Enron email dataset is the only substantial collection of real email data that is public. For this study, the email data were collected from [7]. There are 184 email accounts with emails logs during a period of 44 months (1998-2002). However, there were email accounts which belong to the same person, thus we firstly identify the emails that belong to the same person and add them to one account. So, the 184 email accounts belong to 151 users. Afterwards, we store the email data in a tensor array $\underline{\mathbf{S}}$ of order $151 \times 151 \times 44$.

5.1 Preprocessing steps

In order to perform network analysis we must specify the kind of the relations between the individuals in the network graph. There are two kind of relations: a) the number of emails exchanged between the employees and b) the content of the emails. In this thesis we use the first kind of relation (communication networks). All the steps described below are performed in MATLAB version 7.6.0.324 (R2008a) released in February 10, 2008.

Goal

According to [11] Enron data are multi-mode (work relationship, friendship), multi-link and multi-time period. Our goal is to look at the profiles of the network graph and extract social structures.

Definition of social distance

The social distance of the nodes of the graph can be represented by means of the exponential function $e^{-m_{ij}}$ where m_{ij} is the number of emails that user i sends to user j. This is based on the assumption that social distance between nodes is inversely proportional to the number of emails.

Undirected graph

Each slice of the tensor $\underline{\mathbf{S}}$ represents a directed graph. We construct an undirected graph in order to perform a *3-way MDS* analysis. Thus, we symmetrize each slice:

$\mathbf{S}_\mathrm{i} = \frac{(\mathbf{S}_\mathrm{i}+\mathbf{S}_\mathrm{i}^\mathrm{T})}{2}$ for $\mathrm{i} = 1 \dots K$.

Triangle inequality

Symmetry of the data was imposed by the procedure described above. Non-negativity holds for $\mathbf{S}_\mathrm{i} \geq 0$. In order to achieve a graphical interpretation of the data and ensure that social distances are real distances we must ensure that the triangle inequality also holds. In this case we used an all pairs shortest path algorithm for each slice.

Double centering

After the all pairs shortest paths step, we square the produced distances and we apply the centering operator as in Equation (2.2). Then, each slice can be written as:

$$\mathbf{S}_\mathrm{i} = \mathbf{Y}_\mathrm{i}\mathbf{Y}_\mathrm{i}^\mathrm{T}$$

where $\mathrm{i} = 1 \dots 44$, $\mathbf{Y}_\mathrm{i} \in \mathbb{R}^{151 \times r}$ and r is the rank of matrix $\mathbf{Y}_\mathrm{i}$. Assuming that each slice $\mathbf{S}_\mathrm{i}$ differs from each other, by a weight on each of the dimensions r (according to the weighted Euclidean model) we have:

$$\mathbf{S}_\mathrm{i} = \mathbf{A}\mathbf{W}_\mathrm{i}\mathbf{A}^\mathrm{T} = (\mathbf{A}\mathbf{W}_\mathrm{i}^{1/2})(\mathbf{W}_\mathrm{i}^{1/2}\mathbf{A}^T) = \mathbf{Y}_\mathrm{i}\mathbf{Y}_\mathrm{i}^\mathrm{T}$$

where $\mathbf{W}_\mathrm{i} \in \mathbb{R}^{r \times r}$ is a diagonal matrix with the weights for each dimension r and $\mathbf{A} \in \mathbb{R}^{151 \times r}$. Note that each $\mathbf{W}_\mathrm{i} \geq 0$. The point at this stage is to specify the rank of $\mathbf{Y}_\mathrm{i}$ or alternatively, to specify the dimensions of the common stimulus space $\mathbf{A}$. For this reason, spectral analysis of Enron data is done in the following section.

5.2 Spectral analysis of Enron data

Let $\mathbf{B}_\mathrm{i} = \mathbf{Y}_\mathrm{i}\mathbf{Y}_\mathrm{i}^\mathrm{T}$ for $\mathrm{i} = 1 \dots 44$, thus we have tensor $\underline{\mathbf{B}}$. We sum tensor $\underline{\mathbf{B}}$ along third (time) dimension. Let,

$$\mathbf{C} = \sum_{\mathrm{i}=1}^{\mathbf{K}} \mathbf{B}_\mathrm{i}$$

In the following we illustrate that matrix $\mathbf{C}$ has a low rank approximation by applying SVD to matrix $\mathbf{C}$ and plotting the singular values. Compact SVD of matrix $\mathbf{C} \in \mathbb{R}^{151\times151}$ can be defined as:

$$\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$$

where $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular vectors respectively. $\mathbf{U} \in \mathbb{R}^{151\times r}$ and $\mathbf{V} \in \mathbb{R}^{151\times r}$ are orthogonal matrices ($\mathbf{U^T U} = \mathbf{I}$ and $\mathbf{V^T V} = \mathbf{I}$). $\mathbf{\Sigma} \in \mathbb{R}^{r\times r}$ is a diagonal matrix that contains the singular values of $\mathbf{C}$. Note that singular values are non-negative and the diagonal entries in $\mathbf{\Sigma}$ are placed in descending order.

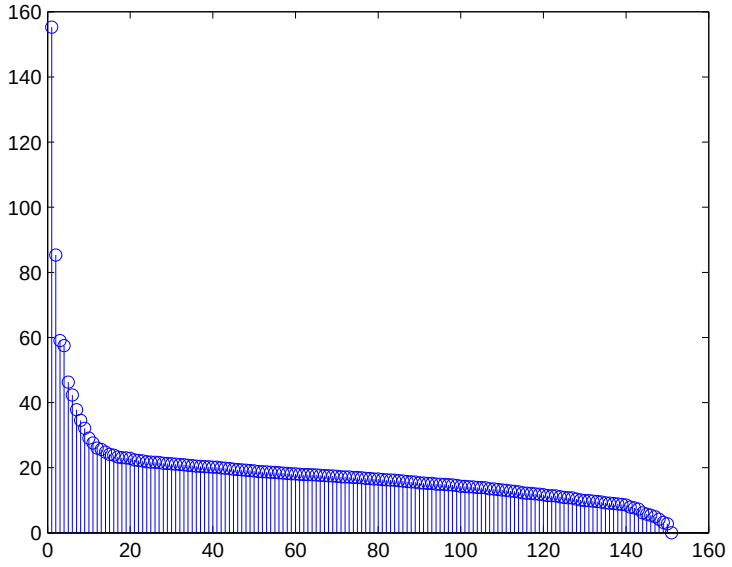


Figure 5.1: Singular values of matrix $\mathbf{C}$ shows that the two largest singular values are clearly above the rest - although a "significant dozen" more singular values appear, as typical for real data.

The largest two singular values of $\mathbf{C}$ are 155 and 85 respectively and the rest singular values are below 60.

5.3 Algorithm

After the application of double centering, trilinear decomposition with 2 components is applied (note that the distance coordinates have a 2-rank approximation) in order to find the profile of each user and plot it in 2-D space.

Algorithm 6 FACTOR ANALYSIS OF ENRON DATA

Require: Tensor array $\underline{\mathbf{S}} \in \mathbb{R}^{151\times151\times44}$ which contains the number of emails over 44 months.

for all $i = 1 \dots 44$ **do**
  for all $j = 1 \dots 151$ **do**
    for all $k = 1 \dots 151$ **do**
      $s_{ijk} = e^{-s_{ijk}}$
    end for
  end for
  $\mathbf{S}_i = \frac{(\mathbf{S}_i + \mathbf{S}_i^T)}{2}$
  $\mathbf{S}_i = \mathbf{S}_i - diag(diag(\mathbf{S}_i))$
  $\mathbf{D}_i = dijkstra(\mathbf{S}_i)$
  $\mathbf{D}_i = -\frac{1}{2}\mathbf{J}\mathbf{D}_i^2\mathbf{J}$
end for
$[\mathbf{A}, \mathbf{B}, \mathbf{C}] = parafac(\underline{\mathbf{D}}, 2)$

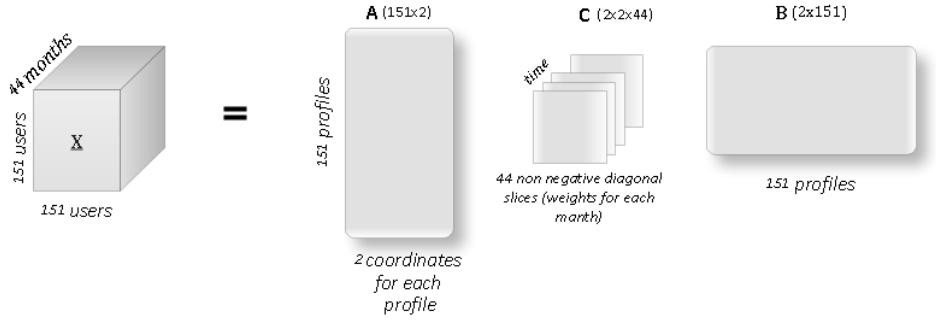


Figure 5.2: Profiles of Enron users over the 44 month period.

5.4 Results

The relations that are developed between the working staff of the company according to the emails that they exchanged are examined as for:

- their position in the company
- the Department of the company they belong

In the first case, among the 151 users we have adequate information concerning their position in the company only for 126 of them. Among those 126, we choose to place in the following diagram those who belonged to the largest categories so that the diagram be clear as much as possible. Consequently, our sample amounts up to 121 individuals (cf. Table 5.1).

In the second case, as shown in the second table below, we made a list of 7 categories of the working staff of the company, which account for up to 78 individuals totally. For those who are not included in the table, either their Department was unknown or they formed small groups which again would complicate the diagram (cf. Table 5.2).

Table 5.1: Number of Employees per position

President	4
Vice - President	23
Director	15
Managing - Director	2
Manager	16
Employees	42
CEO	4
Traders	17
COO	1
In House Lawyer	1
Asst. General Counsel	1
n/a	11
xxx	14
Summary	**151**

Table 5.2: Number of Employees per Department

Gas - Trading	37
Gas - Pipeline (ETS)	10
Finance	8
Government & Reg. Affairs	5
Legal Department	12
Marketing Department	3
Risk Management	3
Summary	**78**

The first diagram (cf. Figure 5.5) shows the working staff's relations according to their position as follows:

- *The Presidents* of Enron Corporation communicate more frequently with the Chief Executive Officers and the Managers.. Furthermore, there is an intercommunication network between the Presidents of Enron Company as underscored by the small distance of their nodes in the diagram (cf. Figure 5.3).

- Notice the dense node distribution of the *Vice - Presidents* in the diagram and a more frequent communication with the high - profiled members of the company (cf. Figure 5.4).

- The *Directors*, due to their position in the company, communicate with the high - profiled members as well as with the *Employees* of the company.

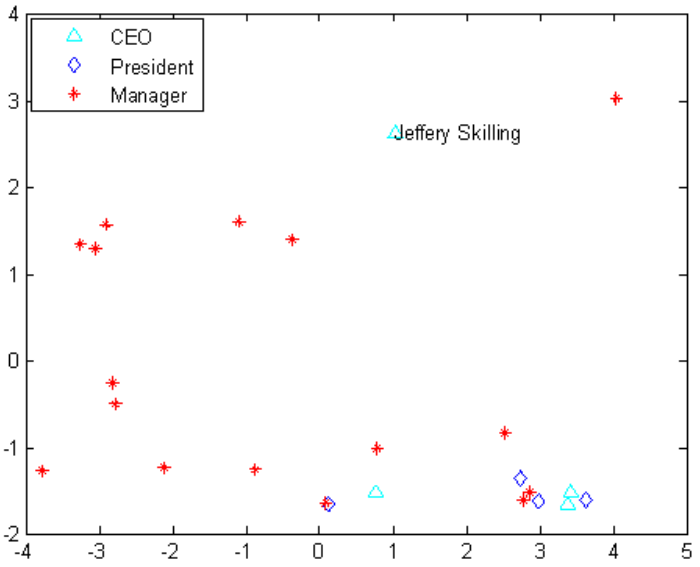


Figure 5.3: Profiles of CEO, Presidents, Managers & VP

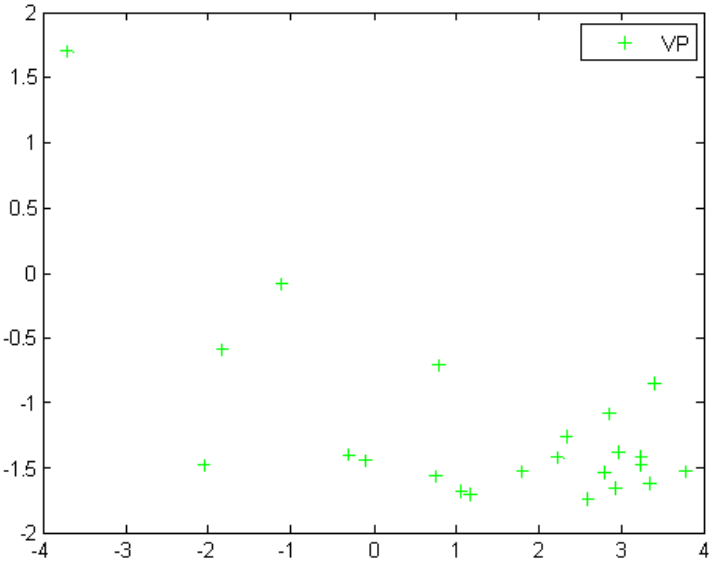


Figure 5.4: Profiles of Vice President

- The *Chief Executive Officers* are strongly connected with the Presidents and Vice - Presidents of the company. Only, Jeffery Skilling communicates occasionally with a large number of individuals.

- There is a large distribution of *Managers* in the diagram. This demonstrates that this group communicate with all the working staff in the company.

- Both *Traders* and *Employees* communicate with all the categories cited above as well as with the individuals of their own group.

The second diagram (cf. Figure 5.6) shows the relation between the Departments of the comany as follows:

- The employees of *Legal Department* are strongly connected. In that sense, we can speak again of an intercommunication network between the members of this group.

- The nodes that represent the *Pipeline Company (ETS-Enron Transportation Services)*, *Gas Trading Department* and *Marketing Department* which promote and circulate the gas, are closer to each other. Therefore, there is a frequent communication between these Departments during the period of 44 months.

- The *Finance Department* of the company communicate mostly with the *Managers* and *Marketing Department*. However, there are some employees that communicate with the employees of *Legal Department* and *ETS* company too.

- The *Risk Management Department* communicate with *Gas Trading*, *ETS*, *Government & Regulatory Affairs* and *Legal Department* employees. It is obvious that there is no communication between this department and the *Marketing Department*.

- It is expected the nodes of *Government & Regulatory Affairs* to be closer to the nodes of *Legal Department*. This shows that there is a communication of this community of employees with the lawyers of the company. Additionally, we notice the communication between this department with *ETS* and *Gas*

Trading Departments, something which is normal because the role of this sector is to ensure that their company comply with all of the regulations and laws pertaining to their business.

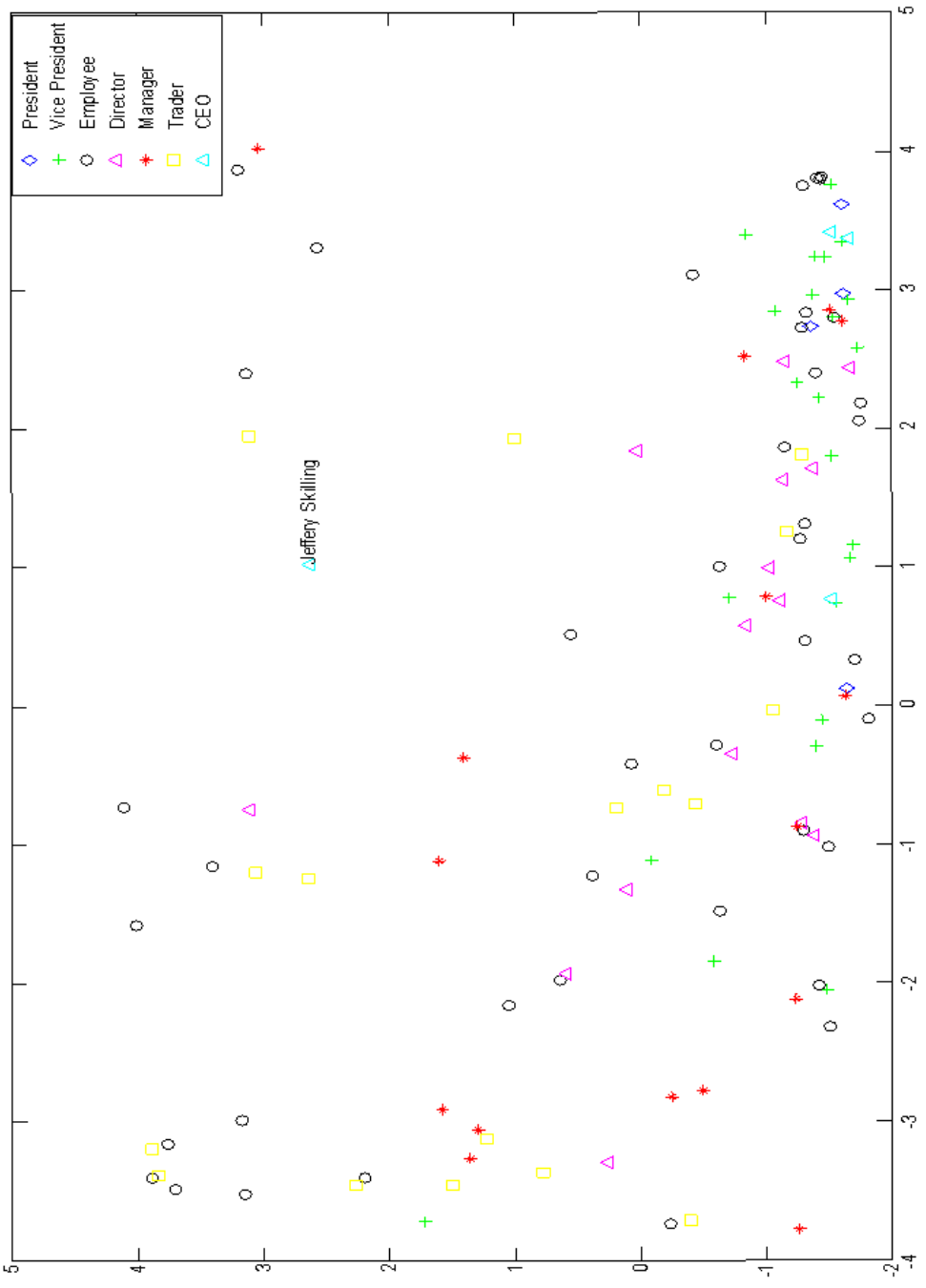


Figure 5.5: Visualization clustering of Enron data, color-coded per position.

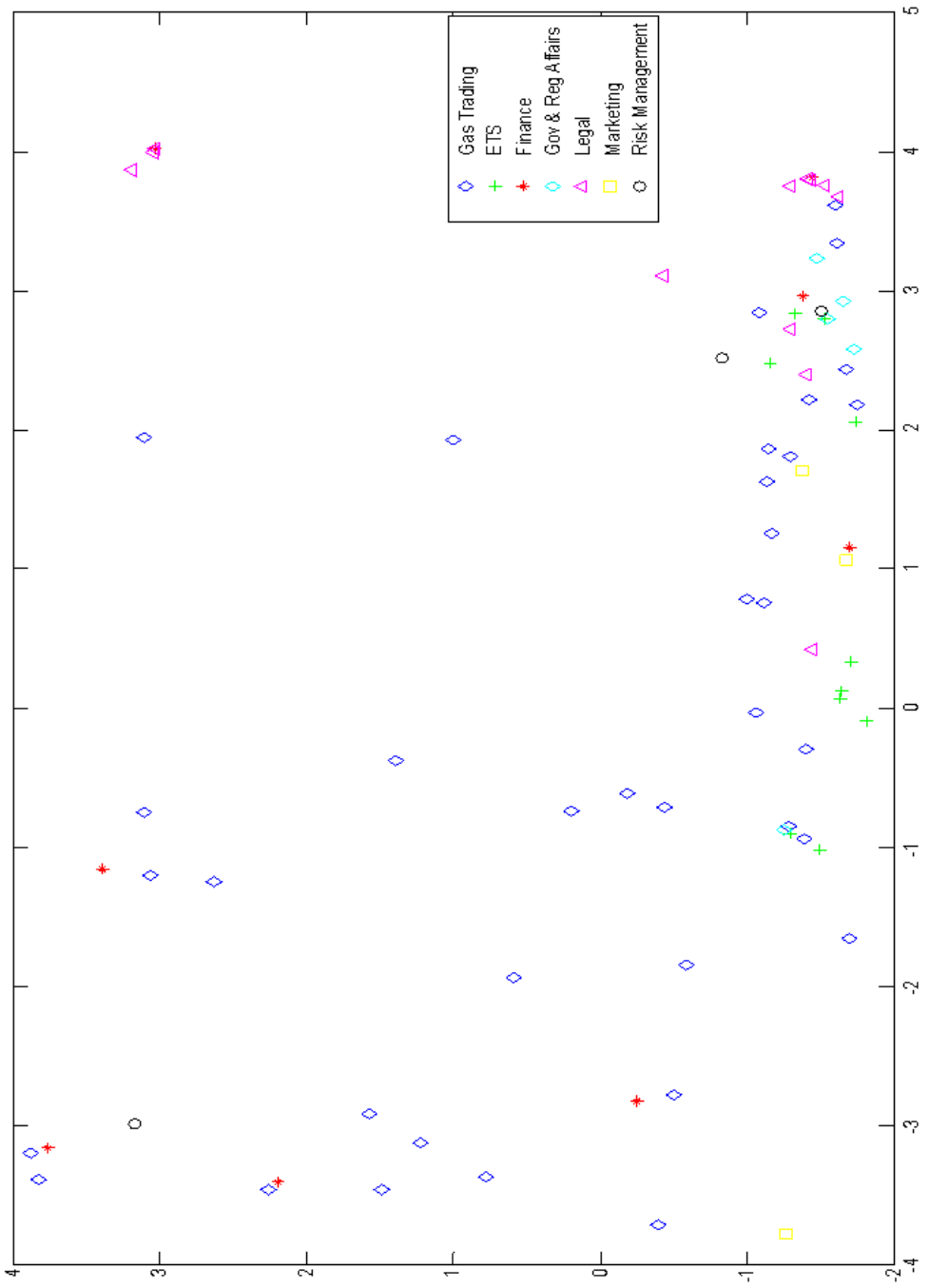


Figure 5.6: Visualization clustering of Enron data, color-coded per department.

6

Conclusion and Future Work

Our analysis concetrated on constructing a map of a social network by analyzing communication patterns using only the traffic between the nodes in the network. Our basic intuition was that the higher the traffic between two nodes the lower their social distance. Thus, we define social distance as a positive, monotically decreasing function of pairwise traffic.

Since there were many possible choices that were consistent with our basic intuition, we examined which ones are reasonable (c.f. Appendix A). However, we don't have a good analytical insight on how to choose this function. We only know the properties that this function should have in order to be as close as possible to the "true" function. Hence, only reasonable choices can be made. Ideally, we would like to adopt a function that is motivated /corroborated by research in the social sciences.

Finally, we illustrated our approach using the Enron email corpus. We propose to use *3-way MDS* in order to capture social dynamics and examine the relations among the employees of the company according to their Department and their position they belong (c.f. Ch. 5).

Appendix A

Function that generates social distances

As mentioned above this function must be non negative and monotically decreasing in order to represent the social distance between the nodes of the communication network. The choice of the proper function depends on the kind of data one has.

Consider a peer to peer network as described in section (4.3). In such a network the range of variation of the number of packets exchaged among different peers is large. Thus, the function that generates the pseudo-distances of the nodes should not necessarily decrease fast. In this case, a reasonable choice of the function that maps messages to social distances could be $\frac{1}{x}$ where x denotes the number of packet-exchanges.

In contrast, email data are by nature sparse and the range of variation of the number of emails that users exchange is not large (typically in Enron data the biggest number of emails that have been exchanged between two users over a period of a month is 147 emails). Thus, in this case in order to visually explore the social structures of the email graph we need a function that quickly decreases. Thus, we define as social distance the function e^{-x}.

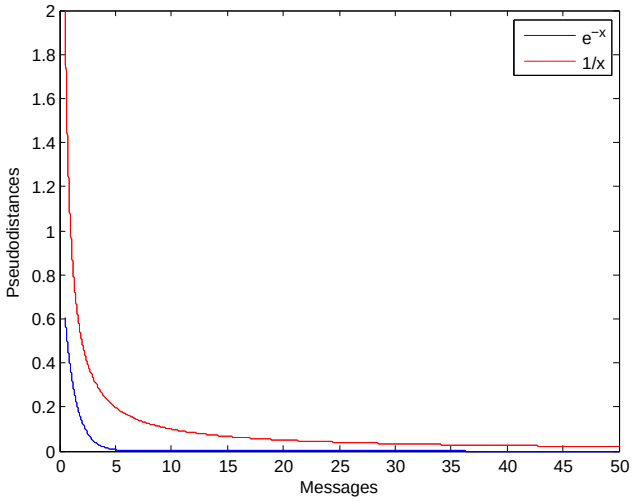


Figure A.1: Functions that generate pseudo - distances from given pairwise packet estimates or messages.

Bibliography

[1] W.S Torgenson, "Multidimensional Scaling of Similarity", *Psychometrica*, vol. 30, pp. 379-393, 1965.

[2] W.S Torgenson, "Multidimensional Scaling: I. Theory and method", *Psychometrica*, vol. 17, pp. 401-419, 1952.

[3] B. Ingwer, G. Pattrick, "Modern Multidimensional Scaling: Theory and Applications", *Springer-Verlag*, New York.

[4] G. Latsoudas, N.D Sidiropoulos, "A Fast Effective Multidimensional Scaling Approach for Node Localization in Wireless Sensor Networks", *IEEE Trans. on Signal Processing*, 55(10):5121-5127, Oct. 2007.

[5] J.D Caroll, J.J Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition", *Psychometrica*, 35, pp. 283-319, 1970.

[6] N.D Sidiropoulos and R. Bro, "On the uniqueness of multilinear decomposition of N-way arrays", *J. Chemometrics*, vol. 14, no.3, pp. 229-239, May 2000.

[7] http://cis.jhu.edu/ parky/Enron/enron.html.

[8] J.M.F ten Berge, N.D Sidiropoulos, "On uniqueness in CANDECOMP/PARAFAC", *Psychometrica*, 67, Feb. 2003.

[9] N.D Sidiropoulos, R. Bro and G.B Giannakis, "Parallel factor analysis in sensor array processing", *IEEE Trans. Signal Processing*, vol. 48, no. 8, pp. 2377-2388, Aug. 2000.

[10] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", *MIT Press*, 2^{nd} Edition.

[11] Jana Diesner, Kathleen M. Carley, "Exploration of Communication Networks from the Enron Email Corpus", *Carnegie Mellon University.*

[12] Jana Diesner, Terrill L. Frantz and Kathleen M. Carley, "Communication Networks from the Enron Email Corpus "It's Always About the People. Enron is no Different "", *Computational & Mathematical Organization Theory*, vol. 11, no. 3, pp. 201-228, 2005

[13] Petros Drineas, Malik Magdon-Ismail, Gopal Pandurangan, Reino Virrankoski, and Andreas Savvides, "Distance matrix reconstruction from incomplete distance information for sensor network localization",*Proc. of the 3rd Annual IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pp. 536-544, 2006