Thesis

Capture and Reconstruction of the Topology of Undirected Graphs from Partial Coordinates: A Matrix Completion based Approach

Submitted by

Sridhar Ramasamy

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Spring 2017

Master's Committee:

Advisor: Anura Jayasumana

Randy Paffenroth
Indrajit Ray
Sudeep Pasricha

Abstract

Capture and Reconstruction of the Topology of Undirected Graphs from Partial Coordinates: A Matrix Completion based Approach

With the advancement in science and technology, new types of complex networks have become common place across varied domains such as computer networks, Internet, biotechnological studies, sociology, and condensed matter physics. The surge of interest in research towards graphs and topology can be attributed to important applications such as graph representation of words in computational linguistics, identification of terrorists for national security, studying complicated atomic structures, and modeling connectivity in condensed matter physics. Well-known social networks, Facebook, and twitter, have millions of users, while the science citation index is a repository of millions of records and citations. These examples indicate the importance of efficient techniques for measuring, characterizing and mining large and complex networks.

Often analysis of graph attributes to understand the graph topology and embedded properties on these complex graphs becomes difficult due to causes such need to process huge data volumes, lack of compressed representation forms and lack of complete information. Due to improper or inadequate acquiring processes, inaccessibility, etc., often we end up with partial graph representational data. Thus there is immense significance in being able to extract this missing information from the available data. Therefore obtaining the topology of a graph, such as a communication network or a social network from incomplete information is our research focus. Specifically, this research addresses the problem of capturing and reconstructing the topology of a network from a small set of path length measurements. An accurate solution for this problem also provides means of describing graphs with a compressed representation.

A technique to obtain the topology from only a partial set of information about network paths is presented. Specifically, we demonstrate the capture of the network topology from a small set of measurements corresponding to a) shortest hop distances of nodes with respect to small set of nodes called as anchors, or b) a set of pairwise hop distances between random node pairs. These two measurement sets can be related to the Distance matrix D, a common representation of the topology, where an entry contains the shortest hop distance between two nodes. In an anchor based method, the shortest hop distances of nodes to a set of M anchors constitute what is known as a Virtual Coordinate (VC) matrix. This is a submatrix of columns of D corresponding to the anchor nodes. Random pairwise measurements correspond to a random subset of elements of D. The proposed technique depends on a low rank matrix completion method based on extended Robust Principal Component Analysis to extract the unknown elements. The application of the principles of matrix completion relies on the conjecture that many natural data sets are inherently low dimensional and thus corresponding matrix is relatively low ranked. We demonstrate that this is applicable to D of many large-scale networks as well. Thus we are able to use results from the theory of matrix completion for capturing the topology.

Two important types of graphs have been used for evaluation of the proposed technique, namely, Wireless Sensor Network (WSN) graphs and social network graphs. For WSN examples, we use the Topology Preserving Map (TPM), which is a homeomorphic representation of the original layout, to evaluate the effectiveness of the technique from partial sets of entries of VC matrix. A double centering based approach is used to evaluate the TPMs from VCs, in comparison with the existing non-centered approach. Results are presented for both random anchors and nodes that are farthest apart on the boundaries. The idea of obtaining topology is extended towards social network link prediction. The significance of this result lies in the fact that with increasing privacy concerns, obtaining the data in the form of VC matrix or as hop distance matrix becomes difficult. This approach of predicting the unknown entries of a matrix provides a novel approach for social network link predictions, and is supported by the fact that the distance matrices of most real world networks are naturally low ranked.

The accuracy of the proposed techniques is evaluated using 4 different WSN and 3 different social networks. Two 2D and two 3D networks have been used for WSNs with the number of nodes ranging from 500 to 1600. We are able to obtain accurate TPMs for both random anchors and extreme anchors with only 20% to 40% of VC matrix entries. The mean error quantifies the error introduced in TPMs due to unknown entries. The results indicate that even with 80% of entries missing, the mean error is around 35% to 45%.

The Facebook, Collaboration and Enron Email sub networks, with 744, 4158, 3892 nodes respectively, have been used for social network capture. The results obtained are very promising. With 80% of information missing in the hop-distance matrix, a maximum error of only around 6% is incurred. The error in prediction of hop distance is less than 0.5 hops. This has also opened up the idea of compressed representation of networks by its VC matrix.

Acknowledgements

Firstly I would like to place my sincere gratitude to my advisor and mentor Prof. Anura Jayasumana. He has always been there as a pillar of support during tough times in motivating me to achieve this feat. I learnt to be an independent thinker and an efficient researcher. I also got to learn lessons to be a successful person in life and in career. Thank you Prof. Jayasumana, for believing in my abilities and moulding me into better student. I should say that, what I have learnt in this period of time will become the foundations for my future career.

I am also deeply thankful for Prof. Randy Paffenroth, who has been guiding me in understanding the important concepts through his video conferences and the praises that he has given time and again. Thank you Prof. Paffenroth for your strong confident words and belief in my abilities.

I would like to sincerely thank my committee members Prof. Indrajit Ray and Prof. Sudeep Pasricha. Thank you very much for your time and efforts.

I would like to thank Aly Fathi Boud and Gunjan Mahindre, my colleagues at CNRL lab for their valuable guidance. Thank you for being there as good friends and a critiques.

The most important people behind me are my parents C.Usha and R.Ramasamy. They are always there by my side giving me the best possible life by even sacrificing their own wishes. Thank you, mom and dad for teaching me to be independent and responsible. I fondly remember my grandfather Late. S.Chandrasekharan, friend, philosopher and guide. He would have been the proudest person to see me graduate. His memory will be with me always. Thanks to my ever-loving big brothers Chotu and Charan. Thank you to Shaiba,

for all her love and support. Thank you for the moral support that you have been giving me. I am also really grateful to my friend, Dhinesh, who helped through rough times.

I would like to mention some important people in my life who have been encouraging, appreciating and taking pride with my growth. I would like to thank my school mentor Rajkumar sir. My School teachers, Mrs. N Lakshmi, Mrs. Vijayalakshmi, Mrs. Krishna Prabha, Mrs. Jayalakshmi and Mrs. Devasena. My Undergraduate teachers, Prof A.L.Kumarappan, Prof. Gayathri Devi, Prof. Caleb, Prof. Alvin, Prof. Premanand.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1. Graphs

Graphs are used to model the relation between objects in a set. Graphs are used to represent relationships in wide variety of applications, including topology of communication and social networks and the adjacency relationships in biological information. A graph is a pair of sets (V, E), where V is the set of vertices and E is the set of edges, formed by pair of vertices. The set of edges could be ordered or unordered pair of vertices. A graph is said to be undirected if the set of edges are unordered. A simple graph is a type of graph that does not have multiple edges between adjacent nodes or self-loops. The Adjacency matrix representation of a finite simple graph is a square matrix with elements (0, 1), where 1 represents presence of a connection while 0 indicates absence of a connection. We consider two kinds of graphs here, graphs embedded in 2D and 3D physical spaces corresponding to Wireless Sensor Network (WSN) and multi-dimensional social network information graphs. This thesis addresses the problem of capturing and reconstructing the topology of simple undirected graphs, with only partial information about the connectivity.

1.1.1. Wireless Sensor Networks.

A Wireless sensor network is a mesh of wirelessly interconnected sensor nodes spanning a geographical area. Wireless sensor networks are deployed to sense the physical or environmental conditions such as temperature, humidity or for monitoring purpose in applications such as wildlife habitat monitoring, health monitoring [2] [3], etc. The sensor nodes work in unison to gather and route information amongst each other or to a base station. Recent advancements have led to novel applications of WSNs such as for disaster management (volcano studies, real-time flood control etc.,), deployed in smart grids to manage energy usage [4], precision agriculture [5] and industrial process monitoring [6]. Wireless sensor network may consist of hundreds or thousands of nodes spread across a wide area. The future of WSN could possibly involve even millions of nodes. Such a future demands inexpensive nodes with low hardware complexity. The major challenge is to select sensor nodes in such a way that it ensures a long, stable and operational WSN. A node in a WSN is limited by its transmission range capability, power and compute capability. Although, the nodes are distributed and networked together, increase in computing and communication capability will tend to make the sensor node more expensive [7].

Localization and positioning is a very significant problem in WSN's for algorithms related to routing, topology management and self-organization. While popular positioning technology, Global Positioning System (GPS) which uses Geographic Coordinate System is a candidate, owing to the drawbacks such as high energy consumption and high cost, is not a practical choice for many applications. Even when the geographic coordinates are available, routing is prone to be affected by physical voids and boundaries. This causes degradation in performance of geographic coordinate based routing protocols. An alternative to the Geographic Coordinate System is Virtual Coordinate System (VCS) [7]. Virtual Coordinate Systems provides a very efficient solution to localize a sensor without geographic coordinates. In a VCS, a subset of M sensor nodes is selected as landmarks or anchor nodes. VCS characterizes each node with a coordinate vector consisting of shortest path hop distances to those pre-selected anchors. Thus each node maintains a vector of size M, and thus the dimensionality of the coordinate system depends on the number of anchors. Thus VCS is an attractive alternative because it is easy to generate and also insensitive to physical voids.

The virtual coordinates in a multi-dimensional virtual domain are an attractive alternative to geographic domain coordinate. Thus, VCS is free of localization equipment such as

GPS and localization algorithms such as RSSI. Also, VCS is transparent to physical voids and VC generation based on hop-count makes it easy to extend the system to 2D as well as 3D sensor network. Due to these advantages, extensive research work is being done on VCS based sensor networks [8] [7]. However there are a few disadvantages namely,

1) A cube with hourglass void network.Number of anchors required and positioning them plays an important role in VC-based routing algorithms. With under-deployment of anchors WSNs suffer from identical node coordinates and improper placement leads to local minima problem.

2) The VCS loses directional information with respect to the nodes.

3) The Virtual coordinates are not orthogonal resulting in errors in distance estimation. [7]

Topology Preserving Map introduced in [7] is a novel technique that can overcome the disadvantages of VCs by generating topology maps of 2D and 3D networks that are homeomorphic to the corresponding physical maps. Singular Value Decomposition is used to obtain the topology coordinates of the sensor nodes. The TPM thus generated preserves the external and internal boundaries and basic shape of 2D and 3D WSN is obtained.

So far, the routing algorithms, TPMs have been requiring the complete set of VCs. Owing to node deaths and other unforeseen circumstances, the life of a node is at risk of being disconnected from the WSN. In such cases, there is lack of complete information about the WSN due to inaccessibility or improper routing.

1.1.2. Social Networks. A social network is a structure made up of social actors interacting and interconnected through relationship among them. The science of studying the social interpersonal relationship is called as Sociometry. Sociogram refers to the graphical representation of the social actors and their relationships. A Social network graph is a

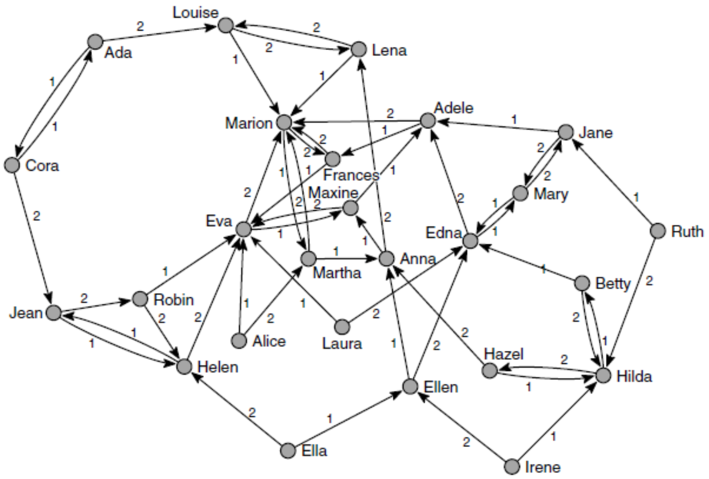


Figure 1.1. Sociogram of dining-table partners [1]

complex multidimensional graph. An example of a sociogram can be seen in Figure 1.1, which is from the book ‘Exploratory Social Network Analysis’ [1]

Figure 1.1 depicts the best choice of dining table partners. Each node of the graph points to a person/actor. The relationship between two actors need not be reciprocative and hence could be directed or undirected. A directed line is called an arc whereas an undirected line is an edge. Social network analysis of graph represented by the Figure 1.1 could lead to answers to questions such as who is the most/least popular dining partner? etc. This explains social network analysis in its simplest form. This is a key to understand the societal behavior.

A social network can be of many different types based on the relationship attribute. The different types of relationships possible are network on social networking sites, communication network (such as email), citation network, collaboration network, product co-purchasing network, road network, peer to peer network, online review network etc. All the above examples describe connection among people but each relationship is of different nature. Unlike Wireless Sensor Network, where connectivity between two sensor nodes depends on the communication range of a sensor node, social network formation purely depends on the relationship attribute.

Emergence of online social networking websites has revolutionized the study of human relationships. Social networking sites such as Facebook, Twitter, and Flickr have provided means to form social groups online. Earlier the social networking analysis was limited to information collected from individuals through difficult approaches. With the advent of online social networking websites the scale and accuracy of social network analysis has increased manifold. The term Monthly Active User (MAU) is widely used to report the active users of a social networking website. Facebook boasts the largest MAU of 1.57 billion users as of June 30 2016. Twitter another social networking website used for short message exchanges has around 313 million active users [9] [10]. The online social networks can be directed or undirected. Some examples of directed social network graph are citation network, twitter etc. On the other hand, there are undirected social networks such as collaboration network, Facebook friend network. Though availability of super-fast computers paves a way to study networks of huge size, the number of users of a social networking website is growing enormously and there is a need to study the social networks. The need to study online social networks comes out of its applications. It is widely used in issues pertaining to national security such as, for intelligence/counter-intelligence to combat terrorism, analysis of call detail records to study the relationship of suspects etc. [11]

A collaboration network is formed on the basis of scientific collaboration between authors of scientific journals submitted to a forum on a particular category. Each author is considered as a node and if an author i co-authored a paper with another author j then, the graph contains an undirected edge between two authors ‘i’ and ‘j’. E-mail communication covers all the email communication of a particular organization. Nodes of the network are the email addresses and an e-mail sent from one node to another denotes the edge. Also, in the field of medicine, it is applied for protein-protein interaction, spread of infectious diseases such as AIDS etc. [12]. The enormous growth of social networks and its significant applications has resulted in surge of interest in researching the structural and behavioral properties. Typically the information collected by using WEB crawler software leaves us with improper or inaccurate information. Thus there is immense significance in being able to extract the missing information from available data.

1.2. Contribution

A technique to capture the topology of undirected graphs from partial information from network path is presented. Specifically, we demonstrate the topology reconstruction for WSN graphs and social network graphs from a small set of measurements corresponding to shortest hop distances from each node to anchors, or shortest hop-distances between random pairwise nodes. The two measurements are related to Distance matrix D where an entry represents the shortest hop distance between two nodes.

The proposed technique depends on a low rank matrix completion method based on extended Robust Principal Component Analysis to extract the unknown elements. The application of matrix completion relies on the fact that many natural data sets are inherently low dimensional and the corresponding matrix is also low ranked.

To demonstrate the effectiveness of our technique, 2D and 3D WSN has been used for simulation. The anchor based VCS is a preferred choice for WSN graph representation owing to its compressing capability in the form of VC matrix and ease of obtaining Topology Preserving Map (TPM). The size of the tested network ranges from 500 to 1600 nodes. The number of anchors is much smaller when compared to the number of nodes. The Topology Preserving Map (TPM) is used to evaluate the effectiveness of the topology reconstruction from partial sets of entries of VC matrix. Results are presented for random anchors and extreme anchors. A double centering based approach is used to normalize the data and compare it with the TPMs from VCs from non-centered approach. Two metrics, mean error and neighbor error are introduced to quantify the error in the TPMs due to missing entries of VC matrix.

For social networks, three different real-world sub-networks have been chosen. They are Facebook, Collaboration and Email sub networks with nodes ranging from 750 to 4200. Since anchor based representation is also relevant for social networks, the social networks are represented as hop distance matrix D and also VC matrix (which is a subset of D). The topology is reconstructed with partial entries of VC matrix and with random entries of distance matrix. The technique is evaluated by two metrics, mean error and absolute hop distance error. The results indicate that, the topology of WSN and social network graphs can be reconstructed accurately. This research has given us methods to measure graphs and also provides means of describing graphs with a compressed representation.

1.3. Outline

Rest of the thesis is organized as follows. Chapter 3 explains the problem statement and motivation for this thesis. Chapter 2 reviews the background work in the area of WSNs and social network with respect to topology reconstruction. Chapter 4 describes the fundamentals of PCA, SVD and the theory of Matrix Completion. Chapter 5 discusses the results for 2D/3D WSN graphs. Chapter 6 discusses the results for social network graphs. Thesis is concluded with Chapter 7.

CHAPTER 2

Literature Review

2.1. Virtual Coordinate System

The two widely used coordinate systems for wireless sensor networks are 1) Geographic Coordinate System (GCS) and 2) Virtual Coordinate System (VCS). The GCS suffers from the drawbacks such as high cost, high energy consumption and routability issues around voids and in boundaries. We shall discuss the VCS in more depth here. Consider the Figure 2.1 which shows a very basic Wireless Sensor Network. The virtual coordinate system works by selecting a few landmark nodes as anchors. The anchors are colored.

Each sensor node maintains a set of coordinates, which are the shortest hop-distances to the chosen anchor nodes. In the above example, each node maintains a vector of length 4 (equal to number of anchors). The VCS is able to represent the connectivity information of the network much better than GCS. In a GCS, the actual distance between two nodes is known but it doesnt reveal the connectivity between them. The possibility of a void between the two nodes exists and thus there might not be an actual path between them. However in VCS, the actual hop distance between node is revealed and hence the shortest path between the node is also known [7].

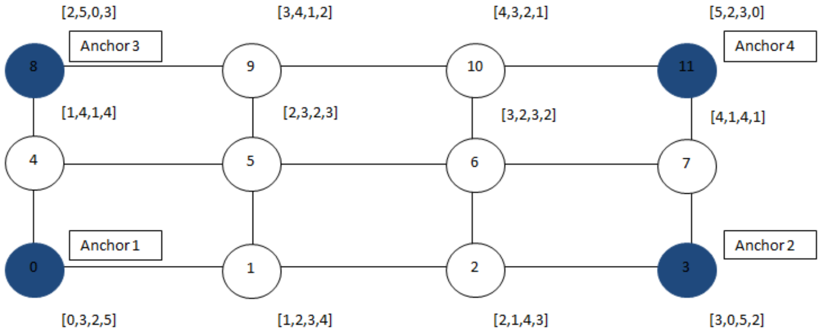


Figure 2.1. Example network to explain VCS

2.1.1. Anchor Selection and Novelty Filter Decomposition method: The selection of anchors has a significant impact on the performance of VCS. A key question therefore is number of anchors and its placement. If an adequate number of anchors are not deployed it may cause the network to suffer from identical coordinates and local minima [7]. To avoid this problem, most anchors placement techniques seek to place the nodes far away. The techniques mentioned in [7] discuss different approaches to select good set of anchors but these techniques requires flooding the network several times or it causes under deployment of anchors. Also, higher number of anchors increases overall energy consumption due to increased address and packet length. Finding the optimal number of anchors and proper placement of anchors are difficult problems to solve. The metric suggested in [7] called as Novelty Filter Decomposition gives a detailed explanation about novelty of an anchor. This method is useful while selecting the optimal number of anchors. For example if we have M anchors for a WSN, then on introduction of (M+1)th anchor, the novelty introduced by the new M+1th dimension is studied. This metric has shown that for networks of size approximately 500 nodes, there is not much new information available beyond 15 anchors. The uses of Novelty Filter Decomposition method are as follows:

(1) Identify a good subset of nodes as anchors

(2) Determine a terminating criterion to introduce new anchors

(3) Identify anchor locations.

2.1.2. Directional Virtual Coordinates and Extreme Node Search (ENS):. The VCS loses sense of direction because it has only the hop distance measurement between any two pair of nodes. To overcome this drawback Directional Virtual Coordinate (DVC) for WSNs was proposed in [13]. The concept behind introducing directionality in virtual coordinate space is based upon the sum and difference of hop-distance value. Each coordinate of DVCS is obtained using a pair of anchors A_1 and A_2. Now each node N_i is characterized by two coordinates which are the shortest hop-distances to the anchors. If we look at the shortest hop-distance of all nodes to anchor A_1, h_{N_i,A_1}, it loses directionality. Conversely, the function $f(h_{N_i,A_1}, h_{N_i,A_2})$ defined in [13], overcomes this drawback. The function defined in 2.1 is a linear mapping of virtual coordinates to real axis with positive and negative values. The center is at the midpoint of A_1 and A_2. This gives us the directional information in the VCS. The term $\frac{1}{2h_{A_2,A_1}}$ normalizes the distance.

$$f(h_{N_i,A_1}, h_{N_i,A_2}) = \frac{1}{2h_{A_2 A_1}}(h_{N_i,A_1} - h_{N_i,A_2})(h_{N_i,A_1} + h_{N_i,A_2}) \tag{2.1}$$

The ENS aims to assign extreme nodes such as furthest apart and corner nodes of the network as anchors. Extreme nodes are obtained as following, initially two random anchors are selected and the VC is flooded to all the nodes. Each node evaluate the corresponding DVC using the Equation 2.1 Finally, each node evaluates if it is local minima/maxima in DVCs within its h hop neighborhood. If the node is a local minima or maxima, it is chosen as an anchor.

2.1.3. Dimensionality reduction of VCS:. Identifying the set of anchors with best routability is a cumbersome procedure, hence dimensionality reduction of VCS paves way for this in such a way that routability remains fairly unaffected [7]. The Singular Value Decomposition of Virtual Coordinate matrix yields, unitary matrices, U, V of dimensions $(N \times N)$ and $(M \times M)$ respectively and diagonal matrix S of dimension $(N \times M)$. The diagonal elements of matrix S are non-negative real numbers called as singular values. The singular values decide which ordinate has a significant contribution. Therefore by ignoring lesser significant values the dimension of the VCS can be reduced to $(N \times R)$ where $R < M$.

2.1.4. Topology Preserving Maps: The virtual coordinates are invisible to physical voids and directional informational is lost. Thus a VCS system lacks information about geometric features and layout of original wireless sensor networks. The Topology Preservation Map (TPM) introduced in [14] preserves the physical features of a network such as geographical voids and boundaries. The TPMs are rotated and/or distorted versions of real physical node maps. The topological coordinates provided by the TPMs are a good substitute for geographical coordinates for applications that depends on connectivity and location. The topological coordinates preserves the relative Cartesian directional information when compared with original layout.

Consider a WSN with N nodes and M anchors ($M << N$). Each node is characterized be a VC vector of length M. Each dimension of the VC vector denotes the number of hops from the node to the anchors. Let P be the $N \times M$ matrix containing VCs of all sensor nodes in the network, The TPM is generated by principal component analysis of the matrix.

$$\begin{aligned} P &= USV^T \\ P_{SVD} &= P \times V \end{aligned} \tag{2.2}$$

U and V are unitary matrices of dimension $N \times N$ and $M \times M$ respectively. The matrix S contains non negative singular values. P_{SVD} is a $N \times M$ matrix containing the principal component values arranged in the descending order of information. P_{SVD} can be seen as a projection of networks VCs on matrix V. The 1st principal component captures the highest variance of the data set. The subsequent components contain the highest possible variance under the constraint that it is orthogonal to the previous components. Usually 1st PC is crucial in any SVD because it contains the most important information. But as shown in [7] for generating TPM, 1st PC is discarded. The 1st PC component contains the radial information about the nodes of 2D and 3D network and does not contribute to identify different nodes distinctly and results in a convex shape. Since SVD provides an orthonormal basis, the 2nd and 3rd components are orthogonal to 1st component. Hence 2nd and 3rd component can be selected as 2D topological coordinates.

$$[X_T, Y_T] = [P_{SVD}^{(2)}, P_{SVD}^{(3)}] \tag{2.3}$$

where $P_{SVD(i)}$ is the i^{th} column of P_{SVD} matrix. The X_T and Y_T are now $N \times 1$ vectors and its i^{th} row gives the X and Y topological coordinates. The topological coordinates for 3D WSNs can be obtained by considering the 2nd, 3rd and 4th principal components [7].

2.2. Related work on Social Networks

The social networks are complex in structure and with growth of internet they have also become massive in size. In recent times, research on social network analysis has witnessed a tremendous growth due to factors such as newer platforms of in the form of websites and commercial interests around it. Some of the challenging research work is going on in graph matching, community analysis, classification of user types and information propagation. The social networks are created based on the type of communication. Some of the attributes that contribute to different mode of communication seen in social networks are, network of friends, collaboration of scientific research, citation of papers and so on. It has been observed in [15] that the prime sociological grouping factors are gender, age, religion and education.

2.2.1. Properties of social networks: The properties of a graph help in understanding the topology better. Clustering coefficient, assortativity, degree of separation and avg path length are some of the commonly used metric to study the property of networks. There has been much research work about the social network topology and its properties.

Few research works point out that the properties of social networks are peculiar compared with other networks. A diverse set of metrics exist for measuring and characterizing the graphs, most typical ones are the statistics obtained from the degree, clustering coefficient, centrality, average path length etc. The [15] observes that, typical graph characteristics seen in social network are 1) power laws (of degree distributions, and other values), 2) small diameters and 3) community effects. The social networks are different than the other networks in two important properties. They have a non-trivial clustering or network transitivity, and they show positive correlations also called assortative mixing between the degrees of adjacent vertices [16].

The assortativity is a measure of probability for nodes to connect with nodes having similar degree. The research in [16] further observes that degrees of adjacent vertices are positively correlated in social network but negatively correlated in most other networks. The clustering is defined as a tendency for nodes to be connected if they have same neighbor nodes. Clustering is quantified as ratio of three times the number of triangles in the graph to the number of connected triples of vertices. The observation shows that correlation is far higher in social network than non-social network.

2.2.2. Classes of networks: Networks are often classified based on the connectivity distribution. The classifications are as follows, scale-free networks, broad-scale networks and single-scale networks [17]. Scale-free networks are characterized by a connectivity distribution with a tail that decays as a power law. We can see that, the new nodes emerging into the network tend to connect to those nodes having higher degree. These networks have high clustering coefficient. The broad-scale networks have a connectivity distribution that follows power law but has a sharp cut-off, while the single-scale network are characterized by a connectivity distribution with a fast decaying tail such as exponential or Gaussian. Most of the real-world social networks falls under the category of scale-free networks also called small-world networks. The diameter of the small world networks are very important. The average shortest distance between two nodes increases logarithmically with number of nodes [17]. This is the key property which makes the social networks to be called as small-world networks.

The research on measurement and analysis of social networks observes that though social-networks exhibit power-law degree distribution, they also differ from other power-law networks. One of the important observation made is that social networks have very similar indegree and outdegree distribution, when compared to other Web graphs. Further, they have significantly shorter average path length. This can be attributed to having high degree of reciprocity between nodes within a social network. The Joint Degree Distribution (JDD) gives us another important aspect of social network properties. The JDD measures the tendency of high-degree nodes connecting to other high-degree nodes. The social networks analyzed shows that it abides by this property. Social networks also differ in the assortativity coefficients. Social networks shows positive assortativity coefficients while other previously observed power-law networks have negative coefficients. Even the clustering coefficients is found to be one order more than other power-law graphs. As an addition, it is also seen that, nodes with lower outdegree have higher clustering coefficient suggesting significant clustering among low-degree nodes. The average group clustering coefficients are also higher indicating the presence of groups/communities inside the graph. Some of the small groups are also cliques. The low-degree nodes are part of few groups while, high degree nodes are part of multiple groups. These are some of the characteristics that classifies social networks into small-world networks[18] [19] [20].

2.2.3. Social network topology prediction : The link prediction problem is another research area in social network analysis. Researchers in this field collect information through WEB crawler software[21]. Many a times the results end up in partial information. Some of the reasons attributed to these are, the efforts of social network operators to block various subscribers, communication failure, non-cooperation of nodes, etc. Apart from recovering the complete information, they are very relevant for applications in different fields as well. For example, in the field of biotechnology, they are used for protein interactions, in online social networking sites for friend recommendation systems, in national security for predicting the links and identifying terrorists. In recent years, several algorithms have been proposed to solve this link prediction problem. The solutions are generally based on supervised machine learning, Bayesian probabilistic models or linear algebraic methods. The survey [22] gives in-depth explanation of different approaches for link prediction problem.

1) FEATURE BASED LINK PREDICTION:

The link prediction problem can be modeled as supervised machine learning problem, where each data point denotes the link between pair of vertices in the social network graph. For any machine learning problem choosing the appropriate feature set is very significant. For link prediction problem, each data point represents some form of connectivity between two nodes. It is natural to choose the feature set to mirror the topology of graph. These features are called as graph topological features. Many works on link prediction as a machine learning problem focused on graph topological features[23][19]. This is straight forward approach as it is applicable for any type of graphs. The popular graph topological features are grouped into categories as follows,

1) Proximity features: The size of common neighbors is also a estimate for link prediction. For example if node x is connected to z and node y is also connected to node z then there

are chances of x and y being connected. This probability will increase with increase of size of common neighbors. Also, for a collaboration network, we can say that, sum of keyword match count (keywords of research papers) is a similar idea.

2) Topological features: Kleinberg [24][25] discovered that in social network most of the nodes are connected with a short hop-count. The idea that friends of a friend can become a friend suggests that, possible link between two nodes depends on the shorter hop count. On the other hand, small world effect brings to notice that most of the nodes are separated only by short hop distances. Thus using this feature of small hop count cannot be a top priority feature. A variant of this shortest path distance proposed by Leo Katz in [26]. This metric sum all the paths exists between a pair of vertices. A regularization parameter is applied to give more weightage for shorter paths than the longer ones. Clustering index is found [25] [19] to be an important feature in social network. It has been observed that, a node in a dense neighborhood tends to grow more edges than a node present in a more sparser neighborhood. The results in [27] shows that, for one of the dataset keyword match count was top ranked attribute followed by sum of neighbors and sum of papers. Shortest hop distance is ranked top among the topological features however it ranks less when compared with all the features. At the same time for another dataset, shortest distance was ranked first among the other features. There are much more classification algorithms available for link prediction in social networks. Some of the classification algorithms are Support Vector Machines, Decision Tree, K-Nearest Neighbors etc.

2) BAYESIAN PROBABILISTIC MODELS: A local probabilistic model for link prediction that uses Markov Random Field (MRF) was proposed by Wang et. al. [28]. This introduces the concept of *central neighborhood set*, which groups the local neighborhood of

either node x or y to predict the link between them. For example, one such *central neighborhood set* is x, y, w, z. This model computes the joint probability, which gives the probability of link between any two nodes. MRFs have been used by authors to solve this learning problem. Initially the *central neighborhood sets* are obtained. One way to find this set is to find the all possible shortest path between the two nodes and include all the nodes along this path. Further, the training data is obtained for the MRF model. The training data is obtained from the log-event of social network. The MRF model is then trained with the training dataset. Once the model is built, the joint probability can be estimated for the *central neighborhood set*. There are also other techniques such as, hierarchical probabilistic model and other probabilistic relational model for this link prediction problem.

3) LINEAR ALGEBRAIC METHODS: Linear algebraic method was proposed by Kunegis [29] which uses dimensionality reduction methods to solve the link prediction problem. This method involves learning of function F which is applied on the graph adjacency. Two adjacency matrices of training and test set are made available. The two matrices are called as source matrix and target matrix. The problem is modified as an optimization problem involving minimizing the Frobenius norm between the two matrices. A link prediction function is applied to source matrix. This problem is solved using eigenvalue decomposition. This general method can fit many possible spectral transformation functions. There are many graph kernels that can be used for this. The function that gives best possible solution is chosen.

One of the related work on application of matrix completion to the problem of social network graph reconstruction via low rank approximation [30]. The graph reconstruction problem has been addressed in the context of recovering the original graph from a randomized graph. The research employs eigen-decomposition for rank approximation of the adjacency matrix. Another related setting is, estimation of the sparse and low ranked matrices [31]. The paper addresses this problem in matrices that are block diagonal. The matrix decomposition addresses in Candes et al., [32] separates the matrix into low rank component (L) and sparse error component (S). On the other hand, this research addresses the problem of S being low rank and sparse at the same time. The evaluation has been done for Protein interactions and Social network (Facebook).

CHAPTER 3

Motivation, Problem statement and Contribution

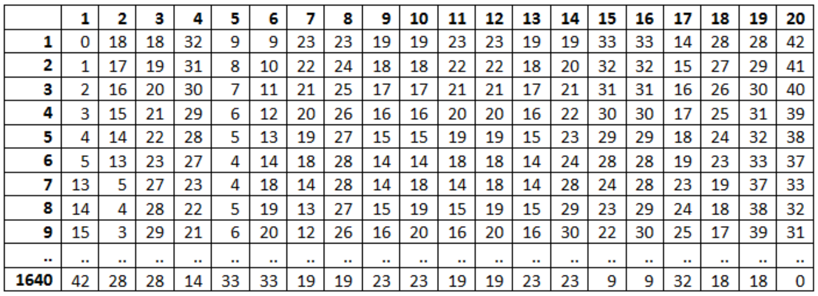
3.1. Motivation and Problem Statement

Graphs are commonly used to represent the relationships between nodes with many attributes. From the first work on graphs by Euler to till date the graph studies has evolved a lot. With advancement in computing capabilities and internet, it can be seen that, we are seeing totally different types of graphs that were not present historically.

The graph theoretical studies are not limited to just mathematical domain but it is extremely important various other domains such as physics, biology, computer science, sociology and is present virtually in almost every statistical analysis. The importance of applications of graph theory in these domains proves its significance. Well known examples are, network of communication, graph representation of words in computational linguistics, in identifying terrorist for national security, studying complicated atomic structures and connectivity in condensed matter physics. There are many graph theoretic analysis done on graphs of these kinds, but such studies faces hindrance in the wake of non-availability or inaccuracy of information. Besides, the dataset from Facebook, Twitter, journal citations, collaboration of authors, web graphs, online review networks, peer to peer networks is an indication to show the complexity of these graphs in terms of size, structural properties, number of attributes, different types of connectivity etc.

Often, the data we get is not complete or suffers from inaccuracy. Taking the case of WSNs, node deaths due to unforeseen circumstances leads issues such as improper routing, inaccessibility of nodes etc. Similarly, obtaining the connectivity information between different nodes in a social network may not always be possible. At the same time, studying the topology of these graphs is vital in tune with its applications. These are the triggers that lead to formulation of a solution to the problem of reconstructing the topology of graphs from incomplete information. The problem focused in this current research work is to reconstruct the topology from a small set of available measurements between the nodes. An accurate solution to this problem also provides means of describing graphs with compressed representation and a good method of measuring sample distances between nodes in order to construct the topology.

The measurement that is used throughout in this research work is shortest hop distance between the nodes. A graph can be expressed by its adjacency matrix. The hop distance matrix is another form of representation which contains the hop distances between pairwise nodes. An adjacency matrix can be obtained from hop-distance matrix and vice-versa. The hop distances of a node to others are considered as virtual coordinates because, it has the connectivity information within itself and it is found to be reliable for routing and obtaining topology. For localizing the nodes in WSNs, one of the most prominent virtual coordinate system used is anchor based method, where a node is addressed by a set of M coordinates. The m-dimensional virtual coordinate of a node is the shortest hop distances to the M chosen anchors. In this research, either a) shortest hop distances from each node to set of anchors or b) shortest hop distances between pairwise nodes are considered. It can be seen that a) is a subset of b). The shortest hop distance matrix is denoted by D as shown in Equation 3.1. The virtual coordinates are obtained by choosing $m = M$ number of columns. A sample VC matrix with 20 anchors can be seen in Figure 3.1. For WSNs, the existing routing protocols and topology mapping tools have been requiring complete set of VCs so far. The key question therefore is how can the topology of a wireless sensor network be reconstructed without the existence of complete virtual coordinates?



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	18	18	32	9	9	23	23	19	19	23	23	19	19	33	33	14	28	28	42
2	1	17	19	31	8	10	22	24	18	18	22	22	18	20	32	32	15	27	29	41
3	2	16	20	30	7	11	21	25	17	17	21	21	17	21	31	31	16	26	30	40
4	3	15	21	29	6	12	20	26	16	16	20	20	16	22	30	30	17	25	31	39
5	4	14	22	28	5	13	19	27	15	15	19	19	15	23	29	29	18	24	32	38
6	5	13	23	27	4	14	18	28	14	14	18	18	14	24	28	28	19	23	33	37
7	13	5	27	23	4	18	14	28	14	18	14	18	14	28	24	28	23	19	37	33
8	14	4	28	22	5	19	13	27	15	19	15	19	15	29	23	29	24	18	38	32
9	15	3	29	21	6	20	12	26	16	20	16	20	16	30	22	30	25	17	39	31
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
1640	42	28	28	14	33	33	19	19	23	23	19	19	23	23	9	9	32	18	18	0

FIGURE 3.1. VC matrix for network of 1640 nodes and 20 anchors

$$D = \begin{bmatrix} h_{n_1 n_1} & \dots & h_{n_1 n_M} & \dots & h_{n_1 n_N} \\ \vdots & \ddots & \vdots & \dots & \vdots \\ h_{n_M n_1} & \dots & h_{n_M n_M} & \dots & h_{n_M n_N} \\ \vdots & \ddots & \vdots & \dots & \vdots \\ h_{n_N n_1} & \dots & h_{n_N n_M} & \dots & h_{n_N n_N} \\ \\ \end{bmatrix} \tag{3.1}$$

Secondly, social networks studies have been gaining momentum from the end of last century. There is lots of research being done on social network topology such as link prediction. The data has usually been acquired using web crawler softwares. Often, web crawler software leaves us with partial information owing to improper acquirement process. Also, due to privacy reasons one may not get the complete data from social networking sites and it depends on the voluntary disclosure as well. The traditional research work on social network link prediction has depended on supervised machine learning techniques that use the features extracted from dataset.

The extension of our research to social networks leads us to the problem of predicting the links or topology of social networks. Unavailability of information due to reasons such as privacy concerns, secrecy, not cooperating to reveal connectivity etc. are the motivating factors for us to venture into the link prediction and topology reconstruction in social networks. The notion of anchor based virtual coordinate representation of nodes is relevant even in social networks. For a prominent node in a social network, the various hop distance denotes circles of nodes or friends around itself. It can be inferred that, with anchor as pivot, there will be huge set of nodes surrounding it creating a dense structure. Thus, anchor based perspective for social network will indeed pave way for obtaining partial measurements from graphs. So, the problem that is being addressed in this research work on social network graphs is to reconstruct the topology represented through anchor based virtual coordinate. Further, what if it not even feasible to obtain partial information in the form of VC matrix? This idea led us towards the research in predicting the topology of networks with just random pairs of hop distances between nodes.

3.2. Contribution

This thesis addresses the problem of reconstructing the topology of undirected graphs with limited information about the connectivity. The fractional information is in the form shortest hop distances between random pairs of nodes or shortest hop distances from each node to a set of anchors. As discussed above, WSN graphs and social network graphs have been used to evaluate the effectiveness of the technique. The WSN graphs contain number of nodes in the range of 500 to 1600 while social network graphs contain number of nodes in the range of 750 to 4200. A matrix completion approach based on extended Robust Principal Component Analysis is used in this research. The application of principles of matrix completion on these partially observed relies on the fact that many natural datasets are low dimensional and their corresponding matrices are low ranked.

To demonstrate the effectiveness of our technique, 2D and 3D WSNs has been used to validate the approach. The anchor based VCS is a preferred choice for WSN graph representation owing to its compressing capability in the form of VC matrix and ease of obtaining Topology Preserving Map (TPM). The TPMs give a good representation of the physical layout using topological coordinates. For an anchor based method the number and placement of anchors plays a vital role. In this research, random nodes and nodes that are farthest apart in boundaries are chosen as anchors. Further, various percentage of entries from the VC matrix have been removed and the TPM is reconstructed using the topological coordinates. The topology coordinates are the Cartesian equivalents in virtual domain.

In the case of social networks, it was observed earlier that obtaining all the connectivity information is very often not possible for reasons such as privacy concerns, secrecy etc. This became a motivating factor for us to apply this technique to reconstruct the topology of social networks. In this thesis, a method has been suggested to measure the connectivity information from a set of anchors to n number of nodes. This gives an anchor based virtual coordinate representation of social networks. The social network graphs are represented as Distance matrix and also as VC matrix. It is to be noted that, subset of VC matrix is in turn a subset of the hop-distance matrix. Since it may not be even possible to obtain social network information from an anchor perspective, the research has widened into scope of predicting the topology with only random entries of hop-distance matrix. Thus the topologies of social network graphs are reconstructed with partial entries in VC matrix and also with random entries from Distance matrix.

The results for WSNs indicate that with around 20% to 40% of entries in a VC matrix, TPM can be obtained with significant accuracy. Metrics have been introduced to quantify the error in the TPMs. The results show that for the case of random anchors, TPMs can be reconstructed with less than 40% error. The results for social networks indicate that, even with 80% of entries missing in the hop distance matrix the topology is obtained with an excellent accuracy. The error is very much less than 10% and can be predicted with an error of less than 0.5 hops.

CHAPTER 4

Theory of Matrix Completion

4.1. Introduction

The problem that we focus on is to successfully predict the topology of graphs. We have represented the graph as a matrix containing shortest hop distances. We have therefore attempted to complete the partial matrix using the principles of low rank matrix completion along with the basic assumptions and technique on which it is based. Before getting into the problem of matrix completion, other fundamental topics such as principal component analysis, singular-value decomposition, are discussed in the first few sections.

4.2. What is Principal Component Analysis?

The modern day applications involve usage of huge amounts of data. The data that we deal with is often not conceivable directly. The data has multiple dimensions or features and hence visualization and inferring results from it isnt an easy task. The point in 3D space can be identified with help of three coordinates known as the Cartesian coordinates. The basis of this representation is orthogonality of the three chosen axes. To demonstrate the need for a good basis, let us see an example where data captured is redundant and skewed. Let us consider a moving point in a 2D plane. The trajectory of the point is shown in Figure 4.1. Along with that, we are noting the position of the moving point in a mobile watch tower along a specified axis. There are two such towers A and B. These are chosen arbitrarily considering it as an experiment. The data is captured say for one minute and the angle between the two axes is not 90 degrees. Now if we look at the data that we obtain, there will be redundancy between them because both the towers are more or less trying to record the same information. To capture the path in terms of x and y values a orthogonal basis

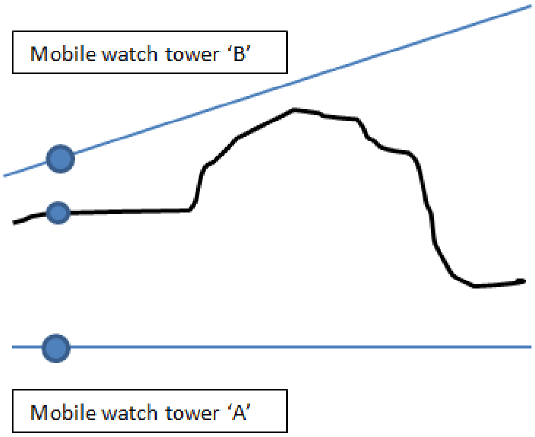


FIGURE 4.1. Example explaining PCA

is needed to record the data. Since the path of towers are not orthogonal, exact x and y values will not be obtained. Also, this example shows that data obtained from both towers will be correlated. The principal component analysis comes to the rescue for datasets like this. The principal component analysis is a transformation of correlated data to uncorrelated variables. This means, we want the variables to co-vary as small as possible with respect to other variables. Variance is a measure that gives the deviation of a random variable from its mean. Covariance is a measure of the change of two random variables together. Let us look at covariance matrix. Covariance matrix represents the covariance between two random variable vectors. Consider a matrix X whose columns are normalized to zero mean. Let C_x denote the covariance matrix of X.

$$C_x = \frac{1}{n} X X^T \tag{4.1}$$

The element C_x denotes the covariance matrix. The diagonal elements of this matrix denote the variance between the same vectors while off diagonal elements contains the covariance. The aim of PCA is to convert set of possibly correlated data into linearly uncorrelated variables called as principal components. The principal component analysis makes an assumption that the direction with largest variance is most important. By this assumption, the PCA selects first direction in which the variance of data encoded in matrix X is maximized. Secondly, the PCA finds the next highest variance possible but under the constraint of orthogonality. These basis vectors are orthogonal to each other and are considered as the principal components. This is transformation of matrix X into new matrix Y with new basis vector P related by $Y = PX$, such that covariance matrix C_y is diagonalized.

4.3. Singular Value Decomposition

The principal components are computed with singular value decomposition [33]. The SVD factorizes the given matrix X. The SVD can be expressed as

$$X = USV^T \tag{4.2}$$

where U and V are unitary matrices i.e. $U.U^T = I$, $V.V^T = I$ and S is a diagonal matrix with singular values along the diagonals. The singular values are non-negative real numbers sorted in descending order. An interesting relation between eigenvalue decomposition and singular value decomposition is to be noted. Eigenvalue decomposition, also known as spectrum decomposition is applicable only for certain square matrices which are symmetric and diagonalizable. An eigenvector v is non-zero vector which on linear transformation gives a scalar multiple of the same vector v. $T(v) = \lambda v$ gives the representation where T is the linear transformation and the scalar value λ is called Eigenvalue.

The eigenvalue decomposition of a symmetric matrix A is expressed as $A = QPQ^{-1}$ where matrix P consists of Eigenvalues arranged along the diagonal elements and Q is a matrix that contains eigenvectors of A. Also, for real values of A, Q is orthonormal matrix i.e. $Q^{-1} = Q^T$. Let us take the example matrix X. C_x is the covariance matrix of the given matrix X. C_x is symmetric and diagonalizable. Thus C_x can also be rewritten as $C_x = QPQ^T$. If we take SVD for the matrix X, we get 4.2. By constructing the covariance with 4.2 we get, $C_x = \frac{1}{n}US^2U^T$ which also gives the relation that square roots of eigenvalues are the singular values.

Thus, principal component analysis can be done with the data matrix or also with the eigenvalue decomposition of covariance matrix provided the values are real. U and V are unitary matrices (in case of complex numbers) or orthonormal matrices if the entries of X are real. The equation 4.2 can be re-written as $XV = US$. The resultant matrix of XV contains the principal components. The number of non-zero singular values is the rank of the matrix. We know that, the singular values are found as diagonal matrix S. The rank of diagonal matrix is equal to the number of non-zero entries. In SVD, U and V are orthogonal and so they are full rank. Therefore rank(X) = rank(S). The SVD is found to be more robust and numerically accurate than EVD of covariance matrix. A matrix is said to be low rank if there are only a few linearly independent rows, i.e. most of the rows can be expressed as a linear combination of those few independent rows. So this means it is in a way possible to write down the elements of other rows if we are given with the linearly independent rows. One of the main applications of PCA that we are interested in is low-rank approximation method for predicting the entries of matrix.

4.4. Low rank matrix and Matrix Completion

A matrix is said to be low rank if there are only a few linearly independent rows. In real world scenario, internet networks, social networks and many other networks are found to be relatively low-ranked. The matrix completion problem can be stated as, given a low ranked matrix, is it possible to predict the unknown matrix elements with only a set of sample entries? Let us consider the given matrix as M. Let us denote the set of location of observed entries m, as Ω i.e. (i, j) belongs to Ω. Let P_Ω be the set projection operator, so $P_\Omega(M)$ is the projection of M onto the set Ω. The optimization problem for matrix recovery can be stated as,

$$\begin{aligned} &min\ rank(L) \\ s.t.\ L_{(i,j)} &= M_{(i,j)}, (i, j)\epsilon\Omega \end{aligned} \tag{4.3}$$

This can be phrased as follows, The aim is to find a matrix L such that, the rank of L be minimized while sticking to the constraint that the elements of Ω found in L matches with M. At the same time L is free to take any values outside the set Ω to make sure the matrix is low rank. This problem is an NP hard problem and so this can be recast as a convex optimization problem under few assumptions.

$$\begin{aligned} &Min\ ||L||_* \\ s.t.\ L_{(i,j)} &= M_{(i,j)}, (i, j)\epsilon\Omega \end{aligned} \tag{4.4}$$

where $||L||_*$, the nuclear norm, is summation of largest k singular values [34]. This new problem is modified as a convex function to minimize. As we have seen earlier, the number of non-zero singular values is same the rank of matrix and so minimizing the nuclear norm of matrix is same as minimizing its rank. Further we will see the initial works on matrix completion and then proceed to the algorithm that we have used for our work. The very first work on matrix completion was done by Eckart and Young, popularly known as Eckart Young Theorem in 1936 [33]. The use of PCA for approximating one matrix by another of low rank was proposed. The theorem states that if the least-square criterion of approximation is adopted then the problem has a general solution though amount of computation will be excessive.

4.4.1. Robust PCA and Matrix Completion. There are many modern extensions of PCA [32] [35] on this problem and computationally fast solutions are available for the same. Robust PCA is a recent addition to deal with corrupted data matrices. In many of the recent researches in the field of image processing, biomedical informatics etc. the data obtained is grossly corrupted. The PCA of a data matrix is a widely preferred mathematical tool to obtain lower dimensional data. If error introduced in data is small then it does not cause wide changes to the number of non-zero singular values. On the other hand, for large errors the number of non-zero singular values might differ a lot. It is important to note that the number of non-zero singular values gives the rank of matrix. In such a case minimizing rank becomes a hard task. The RPCA gives a reliable solution to this problem of matrices corrupted with sparse errors. Under certain conditions, the RPCA decomposes the matrix M into $M = L+S$ where L is the low rank matrix component and S is the sparse component. The problem is solved by convex optimization as follows,

$$Min \; ||L||_* + \lambda ||S||_1 \; s.t. \; M = L + S \tag{4.5}$$

where λ is a constant and the value is set as $\lambda = \frac{1}{\sqrt{n}}$. Again, minimizing rank is achieved using nuclear norm which is convex relaxation of rank and separating the sparse component is using one-norm which is convex relaxation of sparse errors [34] [32]. To get a better understanding of convex relaxation of sparsity, let us consider a motivating example. Consider a vector x and the problem is to minimize the 0-norm of x such that, $Ax = b$. 0-norm is counting the number of non-zero entries of x. The linear system can have infinitely many solutions i.e. x can take any values to attain b. So could try a version of x that has a zero-norm of 1. If it doesn't matches the constraint then we would go for zero-norm of 2 and so on. This problem is NP hard. Now instead let us consider the 1-norm of x. The 1-norm of x is defined as

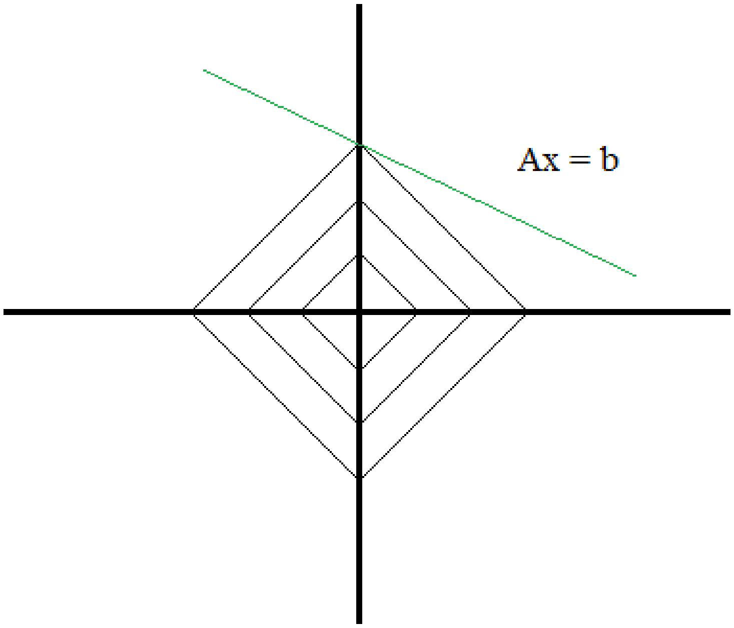


FIGURE 4.2. Figure showing the level set of the one-norm

the sum of absolute values of x. It can be seen that, the 1-norm will appear as concentric diamonds around origin for each absolute value. This denotes all the possible values that x can take to attain the particular absolute value. The constraint line $Ax = b$ will generally intersect at any of the vertex. These vertex points will provide a minimal 0-norm. However there are corner cases when it might not give a sparse solution but considering the fact that there are n dimensions, the occurrence of such a corner case is negligible. There are number of methods to solve RPCA but we are particularly interested in the algorithm based on Augmented Lagrange Multiplier (ALM) [36]. Lagrange multipliers are used for constrained optimization problems of following kind.

$$Min(f(x))\ subject\ to\ h(x) = 0 \tag{4.6}$$

where $f : \mathbb{R}^{\mathbb{N}} \mapsto \mathbb{R}$ and $h : \mathbb{R}^{\mathbb{N}} \mapsto \mathbb{R}^{\mathbb{M}}$. For an optimization problem, violation of equality constraints leads to imposition of penalty and it is borne by Lagrange multiplier. The augemented Lagrangian function for constrained optimization problems can be defined as,

$$L(X, Y, \mu) = f(X) + < Y, h(X) > + \frac{\mu}{2} ||h(X)||_F^2 \tag{4.7}$$

where μ is a positive scalar and Y Lagrange multiplier matrix. After replacing the function f(x) to be minimized and constraint h(X) according to our problem, this can be rewritten as,

$$L(L, S, Y, \mu) = ||L||_* + \lambda ||S||_1 + < Y, M - L - S > + \frac{\mu}{2} ||M - L - S||_F^2 \tag{4.8}$$

To solve, the lagrangian is minimized with respect to L with S fixed and then lagrangian is minimized with respect S with L fixed. Then the lagrange multiplier is updated based on $M - L - S$. This method is called as alternating direct method of multipliers [2447]. The main cost involved is computing SVD needed for updating L. The important implementation detail for the choice of μ and the stopping criterion can be seen in [34][36]. Lin et al, [36] proposed an algorithm for matrix completion based on RPCA. The prior work [34] on matrix completion showed that it is possible to recover a matrix of rank r, if there are p number of samples and it obeys the condition $p >= Cn(6/5)rlog(n)$ where C is a positive constant and n is the maximum of dimensions of considered matrix. Thus, the initial matrix completion equation 4.4 can be solved in a case with augmented Lagrange multiplier as follows,

$$L(L, S, Y, \mu) = ||L||_* + < Y, M - L - S > + \frac{\mu}{2} ||M - L - S||_F^2 \tag{4.9}$$

4.4.2. eRPCA Algorithm. The work done by Lin, et al[36] provides a robust method for matrix decomposition and a separate method for matrix completion. But the eRPCA algorithm is an extension that deals with globally noisy data and also can be used for incomplete case. The extended RPCA allows for point-wise error bounds and partial observation [35]. The problem can be formulated as,

$$min\ ||L||_* + \lambda||S||_1\ subject\ to\ |P_\Omega(M) - P_\Omega(L+S)| \leq \widetilde{\epsilon} \tag{4.10}$$

where, $\widetilde{\epsilon}$ is a matrix of point-wise error bounds. The eRPCA problem relaxes $|M-L-S| = 0$ to $|M-L-S| \leq \widetilde{\epsilon}$

The minimization problem can be re-written as,

$$\begin{aligned} L_1, S_1 = argmin_{L_0,S_0}\ ||L_0||_* + \lambda||S_{\widetilde{\epsilon}}(P_\Omega(S_0))||_1 \\ s.t. P_\Omega(M) - P_\Omega(L_0+S_0) = 0, \\ L, S = L_1, S_{\widetilde{\epsilon}}(P_\Omega(S_1)) \end{aligned} \tag{4.11}$$

The modified constrained optimization problem with augmented Lagrangian will be,

$$L(L,S,Y,\mu) = ||L||_* + \lambda||S_{\widetilde{\epsilon}}(P_\Omega(S))||_1 + \frac{\mu}{2}||P_\Omega(M-L-S) + \frac{1}{\mu}Y||_F^2 - \frac{1}{2\mu}||Y||_F^2 \tag{4.12}$$

The constant $\lambda = \sqrt{\frac{m}{|\Omega|}}$. The algorithm implementation details of eRPCA via ADMM can be seen in [37] [35]. Augmented Lagrangian Method is again used for solving this optimization problem owing to uncomplicated calculation method. The augmented Lagrangian term makes it easier to compute the differentials of Lagrangian with respect to L and S. Also it ensures that as every iteration proceeds, if the constraints aren't met, then appropriate penalty is levied to ensure correctness. The parameter μ is to be noted. The μ starts with a lower value and as every iteration proceeds, the value of μ grows. Let's see what happens to optimization with every iteration. In the initial iterations, the penalty is less and value of μ is also less. With every passing iteration, if the entries of L aren't filled according to constraints then penalty increases. So it can be inferred that initial iterations contributes to minimizing the objective and for later iterations the focus shifts to matching the constraints.

It can be noted that, a new shrinkage operator $S_{\widetilde{\epsilon}} : \mathbb{R} \mapsto \mathbb{R}$ has been introduced. The shrinkage operator is defined as,

$$S_{\widetilde{\epsilon}}(x) = sign(x)\ max(|x| - \epsilon, 0) \tag{4.13}$$

where ϵ denotes the error. This can be extended to matrix shrinkage by applying this to each element of matrix $\widetilde{\epsilon}$. The proof for equivalence of the constraints after application of shrinkage operator can be seen in [35]. Also, by using the matrix shrinkage operator to the the S matrix, it eases the calculations further.

On performance side, if we had, all the entries of the matrix M then, the running time will be $\mathcal{O}(n^2)$. Let us take a look at the two differentials $\frac{\partial L(L,S)}{\partial S}$ and $\frac{\partial L(L,S)}{\partial L}$. The differential w.r.t S, depends on entries of S. So by possessing observed and partially observed entries, we can say that instead of n^2 entries, we will be calculating the differential only for the seen m entries. As far as the derivative w.r.t S, it depends on the desired rank (the number of singular values). By virtue of the chosen low rank $r << min(n1, n2)$, we can assure the fast convergence. We can also see in [37], that the Lagrangian in Equation 4.9 will run in $\mathcal{O}(m)$. The empirical result presented in [37] shows that, if there is a matrix M with $\mathcal{O}(m)$ entries that are observed or partially observed then each iteration in the optimization problem 4.11 costs $\mathcal{O}(m)$ in both time and memory. Each iteration in minimization is a function of number of observed entries rather than the total entries.

CHAPTER 5

Results for Wireless Sensor Network graphs

5.1. Introduction

The wireless sensor networks are widely used to sense the physical or environmental conditions or for monitoring purpose. The WSNs usually span a wide geographical area, and are built with nodes limited by memory and power. There are different algorithms for propagation of message in a wireless sensor network. The propagation of message involves having sense of location and directionality to reach the next nearest node. As seen in Section 2, the regular GPS based localization is expensive in terms of energy consumption and cost. This paved way for VCs for a wireless sensor network based on the relative hop distances. Out of the different VCS available, anchor based VCS have been found to be used widely for routing and also for topology studies. The anchor based VCs uses a set of anchors and minimum hop distances to all the nodes i.e. each node is characterized by a virtual coordinate which contains minimum hop distance to the anchors.

For example, geo-logical routing uses the virtual coordinates and the topological coordinates in three different modes to achieve successful routing. The routing algorithms so far needed the complete set of virtual coordinates, but the present work aims to capture the topology of the network with only partial information. This chapter discusses the efficiency of the proposed approach in reconstructing the topology of the wireless sensor networks.

To test the efficiency of this approach we have used four different WSNs. Two 2D network, and Two 3D networks have been used for evaluation. The 2D networks used are

(1) An odd shaped network.

(2) A circular network with three voids.

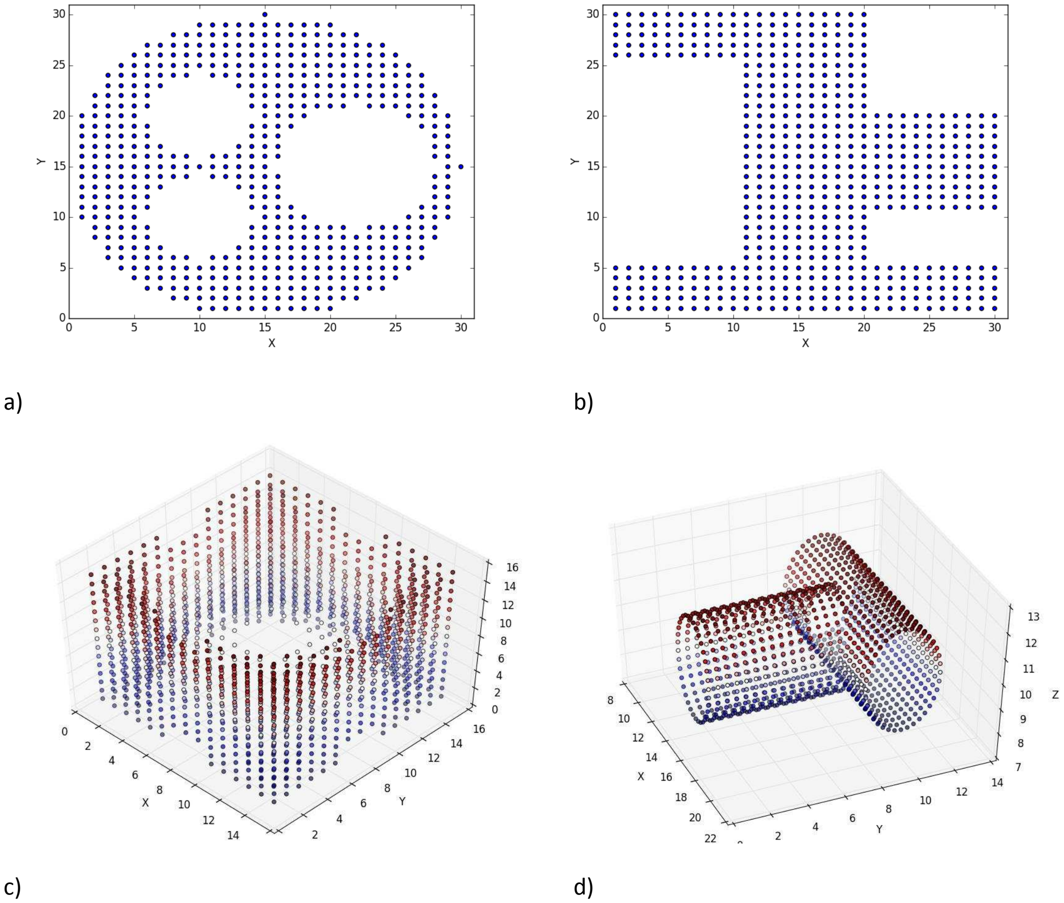


Figure 5.1. Physical layout of test WSNs: a) circular network with three voids, b) odd shaped network, c) cube network with hourglass shaped void, and d) hollow T shaped cylinder network

The 3D networks are

(1) A cube with hourglass void network.

(2) A hollow T shaped cylinder network.

The physical layout of the 4 networks are shown in Fig 5.1.

5.2. Anchor Selection

A good TPM representation depends on a good set of anchors. The question is the number and placement of anchors. The two problems are also interdependent and makes it a tough task. An optimum set of anchors are the minimum number of anchors that provide unique VCs (without redundant information) and 100% routing is achieved using the shortest path. If an adequate number of anchors are not deployed, it may cause the network to suffer from identical coordinates and local minima, resulting in logical/virtual voids. Having too many anchors, increases the cost of VCS generation as well as the address length. Apart from determining the right number of anchors, the optimal placement is also a challenge. We have obtained results for two different anchor selection methodology. The first one uses random positioning of anchors while the second approach uses Extreme Node Search to obtain nodes that are furthest apart and corner nodes of the network as anchors.

Algorithm 1 Extreme Node Search Algorithm

```
Neighbors set of $N_i$ is $K_h(N_i)$;
Two random nodes are chosen. Flooding is initiated to generate VCS
Each node locally generates its DVCS using Equation 2.1
Each node checks whether it is a local minimum/maximum in h-hop neighborhood
if $f(h_{N_iA_1}, h_{N_iA_2}) > f(h_{N_jA_1}, h_{N_jA_2}); \forall N_j \in K_h(N_i)$ then
    $N_i$ is an anchor
end if
OR
if $f(h_{N_iA_1}, h_{N_iA_2}) < f(h_{N_jA_1}, h_{N_jA_2}); \forall N_j \in K_h(N_i)$ then
    $N_i$ is an anchor
end if
Selected anchor nodes generate VCS
```

5.2.1. ENS anchor. The Extreme Node Search is an attempt to assign extreme nodes as anchors [7]. The extreme nodes are those, that are furthest apart and corner nodes of the network. The Extreme Node Search is explained in the algorithm 1.

5.3. Topology Preserving Map

As seen in Chapter 2 Topology Preserving Maps are maps that preserve the physical features of a WSN such as geographical voids, boundaries etc,. The TPMs are obtained by plotting the 2D/3D topological coordinates. Consider a WSN with N nodes and M anchors

($M << N$). Each node is characterized by a VC vector of length M. Each dimension of the VC vector denotes the number of hops from the node to the anchors. Let P be the $N \times M$ matrix containing VCs of all sensor nodes in the network. The topological coordinates are generated by principal component analysis of the VC matrix. The equation 4.2 gives the method to apply SVD to obtain the topological coordinates from the principal components. The 2nd, 3rd and 4th component can be selected as 3D topological coordinates while 2nd and 3rd alone gives the topological coordinates for 2D maps.

$$[X_T, Y_T, Z_T] = [P_{SVD}^{(2)}, P_{SVD}^{(3)}, P_{SVD}^{(4)}] \tag{5.1}$$

5.3.1. Normalization of Data. The virtual coordinate is of m dimensions made up of positive valued hop distances and this can be thought of as data present in 1st quadrant of 2D graph. When SVD is applied for such a data the first principal component is actually not picking up the most significant information owing to the reason that data is not normalized. This is overcome by applying double centering as given in [38]. Each observed entry is double centered using the following equation,

$$d_n(i,j) = \frac{-1}{2}(d(i,j) - \mu_i(P_\Omega(D)) - \mu_j(P_\Omega(D)) + \mu(P_\Omega(D))) \qquad (i,j)\epsilon\Omega \tag{5.2}$$

where D is a matrix formed by squaring individual elements of the considered matrix and here it is VC matrix , Ω is the set of location of observed entries so that P_Ω is a projection operator onto this set, $d(i,j)$ ϵ D, $\mu_{i,j}$ denotes the mean of all observed distances, μ_j gives the mean of i^{th} row and μ_i gives the mean of j^{th} column, m is the number of observed entries. The topological coordinates are generated by principal component analysis of the full set of double centered VC matrix. The Topology Preserving Map thus obtained, takes into account the 1st component also.

Table 5.1. Characteristics of Wireless Sensor Networks

Network	Number of Nodes	Number of random anchors	Number of ENS anchors
Circular network with three voids	496	20	8
Odd Network	550	20	7
Cube with hourglass shaped void network	1640	20	11
Hollow T shaped cylinder network	1245	20	10

5.4. Results

This section discusses the approach of reconstructing Topology Preserving Map using partial VCs unlike the existing TPM generation methods. The Table 5.1 summarizes the details of the number of nodes and anchors for the four representative WSNs whose physical layout can be seen in 5.1. We choose the number of anchors to be less than 5%. The results have been obtained for random anchors and ENS anchors. 1 has been applied to obtain the ENS anchors.

5.4.1. Approach.

First the rank of VC matrix is examined. The singular values of the VC matrix for random/ENS anchors as well as for double centered VC matrix can be seen in Figure 5.2. This shows that the VC matrix is relatively low ranked and hence supports the usage of matrix completion to recover unseen hop distances. Next 10%, 20%, 40%, 60% and 80% entries are randomly dropped from the VC matrix. Then the TPMs are recovered using eRPCA algorithm. To test the quality of TPMs obtained and quantify the error introduced due to missing VCs, two metrics have been proposed.

1) MEAN ERROR: To quantify the error introduced to the TPM due to missing VCs, mean error is introduced. The mean error evaluates the error in the distances between all pairs of nodes with respect to the distance for a TPM with full set of VCs. The distance here is euclidean measure between the nodes localized using its topological coordinates. we

define the mean error (E) as follows:

$$E = \left[\sum_{i=1}^{N} |d_{ij}(f) - d_{ij}(0)|\right] / \left[\sum_{i=1}^{N} d_{ij}(0)\right] \tag{5.3}$$

where, $d_{ij}(f)$ refers to the euclidean distance between nodes i and j when f fraction of random anchor coordinates are missing. The nodes are localized by its topological coordinates in TPMs.

2) NEIGHBORHOOD ERROR: The neighborhood error calculates the change in neighborhood around a node for a particular cardinality. Cardinality $N_{i,m}$of a node i refers to the m number of closest neighbors. Here again, the closest neighbors are considered based on the euclidean distance between the nodes localized using topological coordinates. Neighborhood error for a cardinality of m is defined as,

$$N_m = \left[\sum_{i=1}^{N} |N_{i,m}(0) \setminus N_{i,m}(f)|\right] / \left[\sum_{i=1}^{N} m\right] \tag{5.4}$$

where $N_{i,m}(f)$ refers to m number of closest neighbors of node i when f is fraction of random coordinates are missing.

5.4.2. Results for random anchors. First let us look at the TPMs obtained with random anchors. The TPMs obtained for the WSNs with random anchors can be seen in Figure 5.3 - Figure 5.10. It can be seen that much of the geometric shape has been preserved even when the mean error is high. The mean error for TPMs with un-normalized and normalized data can be seen in Figure 5.11 and Figure 5.12 respectively. The Mean error is obtained over 5 different iterations. Each iteration is carried out by removing different sets of random coordinates. The average value of Mean error with the standard deviation is plotted. It can be seen that, the Mean error is consistent for removal same fraction but different set of coordinates. The Mean error values for TPMs after normalizing the data is higher when compared to that of un-normalized one. Also, It can be seen that when upto 60% of entries are removed the mean error for TPMs obtained for normalized data is higher by only few percentage points but when 80% entries are removed the mean error increases by almost 10%. In the same context, the mean error for 3D networks differs more when compared to 2D networks. The mean error for each run with discarded entries such as 20%, 40%, 60%, 80% differs by a meager amount for all the four networks with un-normalized data, but the mean error differs by a larger amount in case of normalized dataset. The maximum mean error for TPMs obtained with un-normalized data is around 38% and for TPMs with normalized data is 48%. The neighborhood error plot for 2D networks and 3D networks can be seen in Figure 5.13 and Figure 5.14. The neighborhood error shows that for smaller cardinality the neighborhood change is very prominent and as cardinality increases the error decreases considerably. The Neighborhood error in general is also higher for TPMs obtained after normalizing the data compared to un-normalized one.

There is an exception to this trend for odd shaped network. It can be seen that the Neighborhood error with normalized data is less compared to un-normalized one. There are two main reasons that can be attributed for this. The topology map obtained with full set of VCs have more number of points overlapped on three of the sectors of its geometry. Since the points tend to overlap and shrink with removal of virtual coordinates upon matrix completion, it still overlaps in those set of points. Also, the map shrinks to the center of the geometry and the odd shaped network contains grid shaped points in the same central location. Thus clustering of points in the center in a way tries to retain the backbone of the network. This can happen only with a network like this and any other network which has a void around the center of its geometry will suffer more loss of geometry such as the network with three voids.

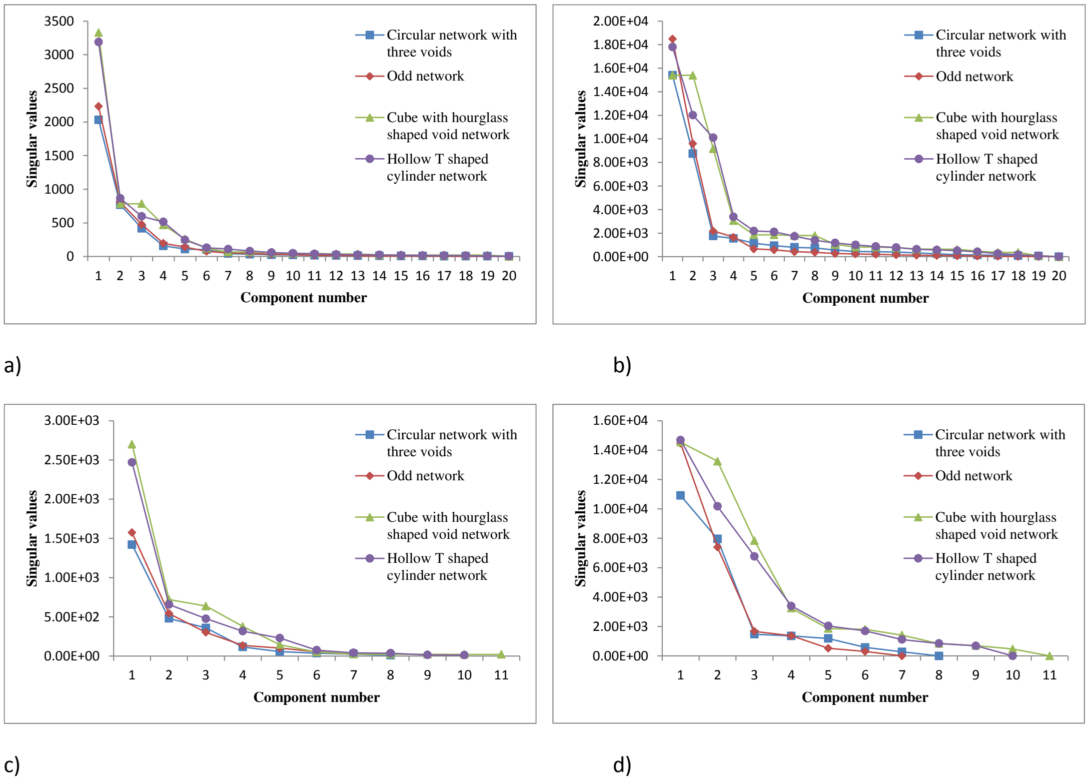
5.4.3. Results for ENS anchors. The trend observed in the error metric with random anchors chosen for WSN is also seen with ENS anchors but it can be seen that both the mean error and the neighborhood values are higher. The maximum mean error for 80% entries removed is almost 60% whilst, it is found to be around 73% for 80% entries removed with normalized dataset.

We can see that the rate of increase of neighborhood error, decreases with more entries being removed i.e. change in neighborhood error between 60% and 80% is less compared to 10% and 20%. The rate of fall of neighborhood error with cardinality is more when more random entries are removed.

Also, the TPM obtained for odd shaped network with normalized data gives lesser mean error and neighborhood error when compared to TPMs obtained with un-normalized data.

5.5. Summary

The application of matrix completion for WSNs satisfactorily proves that the TPMs can be obtained with incomplete information. The error introduced in the TPMs due to missing VCs is calculated with two error metrics namely mean error and neighborhood error. The results show that in spite of higher error values, much of the local neighborhood and geometric information is preserved. The results also indicate that this technique of obtaining TPMs with random anchors and un-normalized data is less prone to deformation even when much of the information is missing, whereas for TPMs obtained with lesser number of ENS anchors tend to lose its geometric information with loss of connectivity information. Also, we have proposed the method of generating topology map after normalizing the data. The next chapter discusses the results obtained for social networks. The future work is discussed in the Chapter 7.



a)

b)

c)

d)

Figure 5.2. a) Singular values of VC matrix with random anchors
b) Singular values of double centered VC matrix with random anchors
c) Singular values of VC matrix with ENS anchors
b) Singular values of double centered VC matrix with ENS anchors

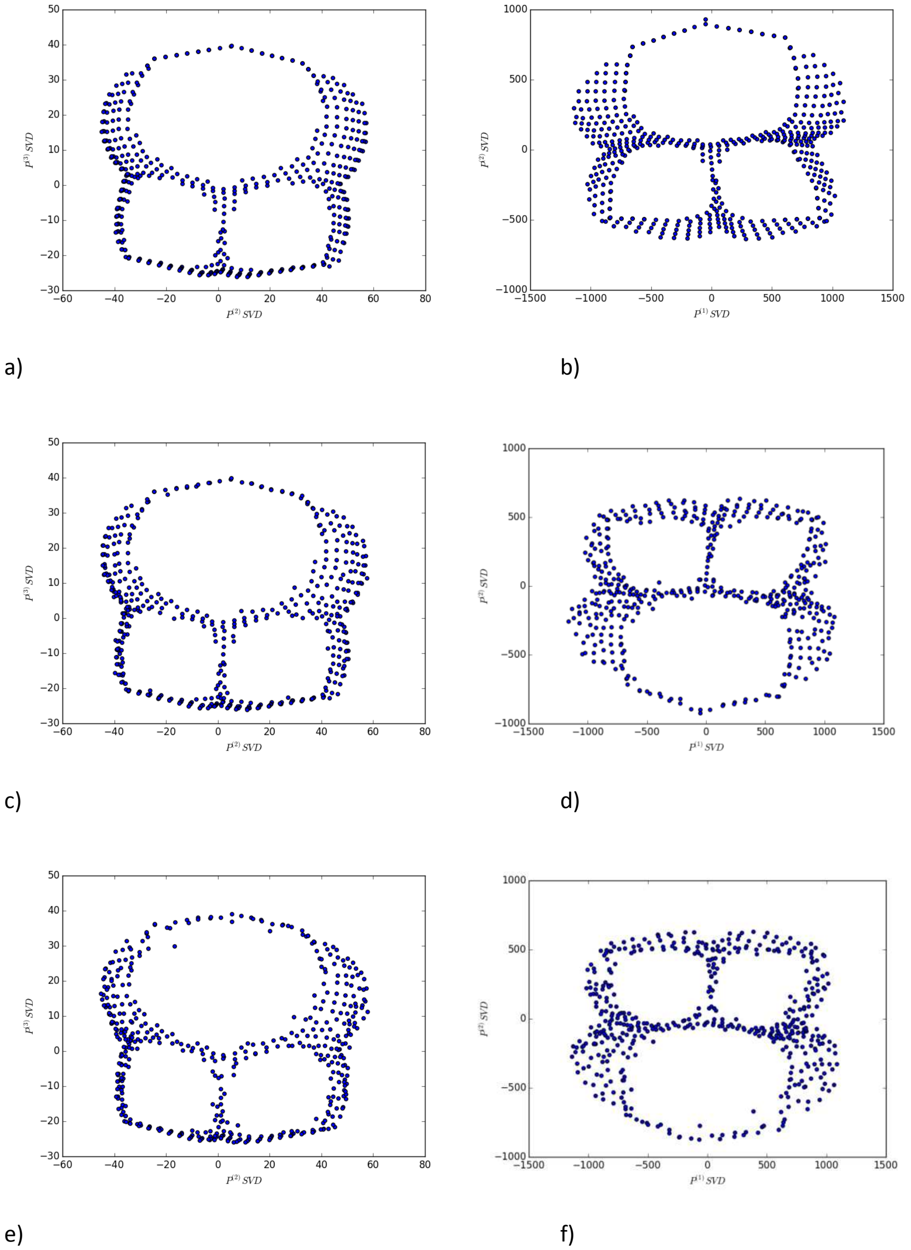


FIGURE 5.3. Topology Preserving Map for circular network with three voids - random anchors: TPM recovered from full set of VCs (a) non-centered approach, (b) centered approach; Recovered TPM with 10% random coordinates missing from VC matrix (c) non-centered approach, (d) centered approach; Recovered TPM with 20% random coordinates missing from VC matrix (e) non-centered approach, (f) centered approach

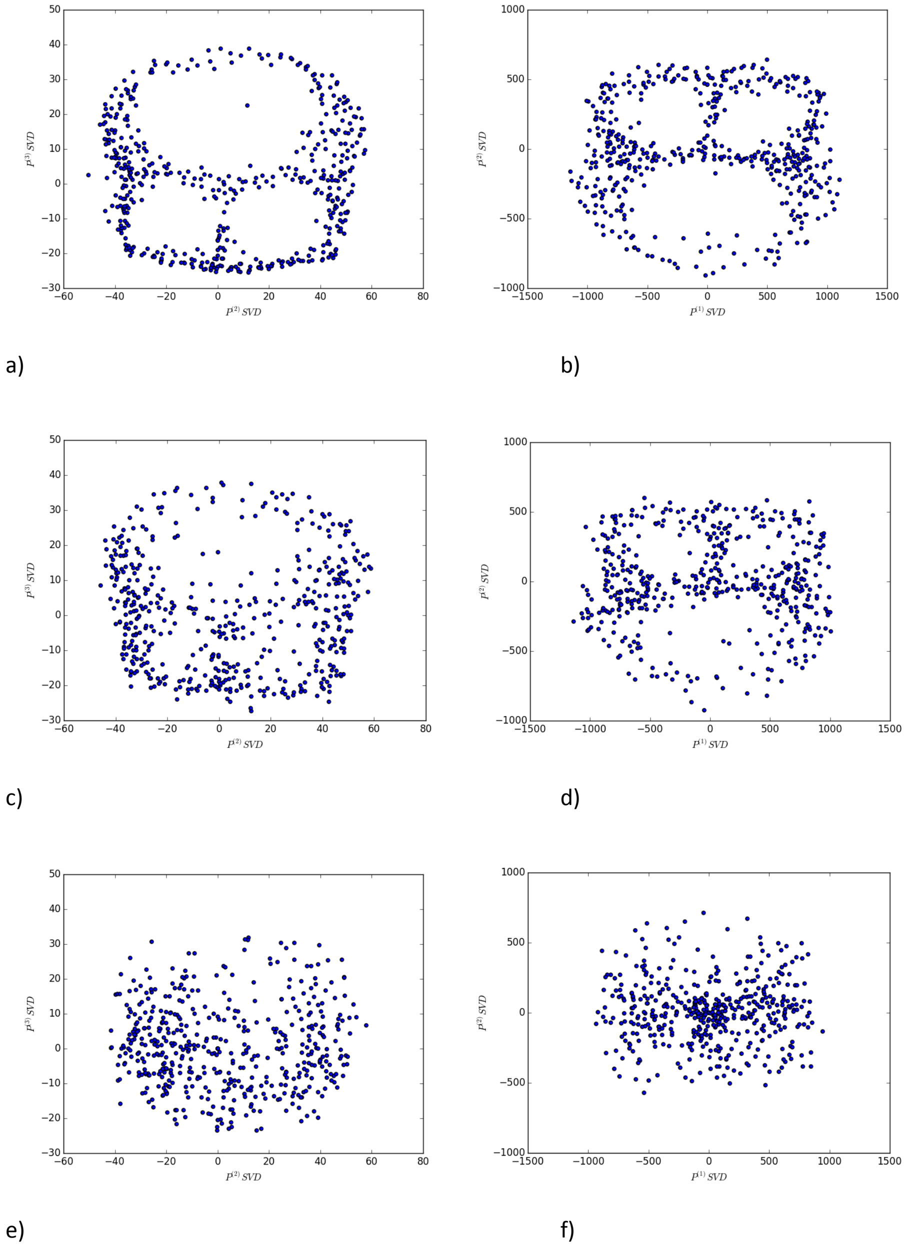


FIGURE 5.4. Topology Preserving Map for circular network with three voids - random anchors: Recovered TPM with 40% coordinates missing (a) non-centered approach, (b) centered approach; Recovered TPM with 60% coordinates missing (c) non-centered approach, (d) centered approach; Recovered TPM with 80% coordinates missing (e) non-centered approach, (f) centered approach

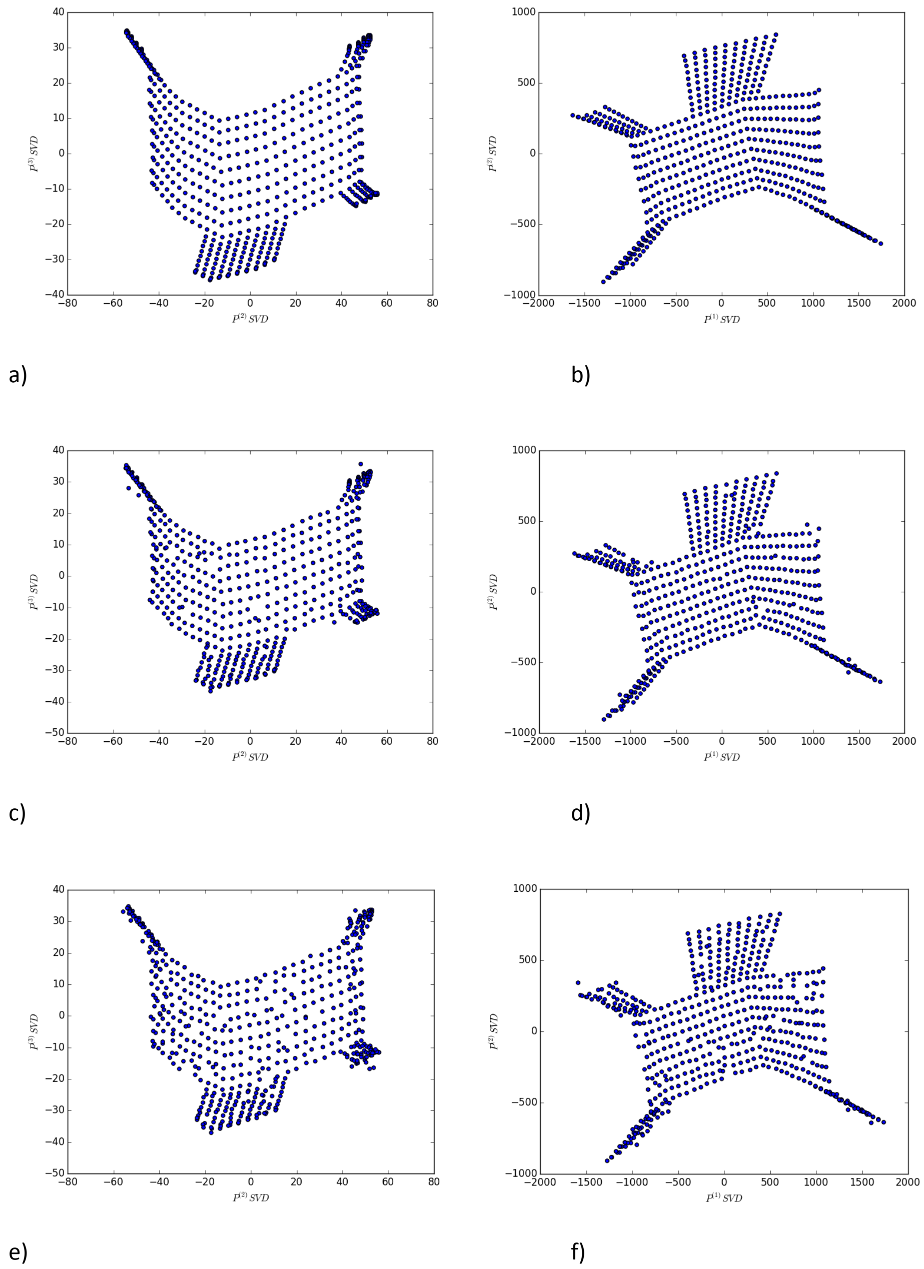


FIGURE 5.5. Topology Preserving Map for odd shaped network - random anchors: TPM recovered from full set of VCs (a) non-centered approach, (b) centered approach; Recovered TPM with 10% random coordinates missing from VC matrix (c) non-centered approach, (d) centered approach; Recovered TPM with 20% random coordinates missing from VC matrix (e) non-centered approach, (f) centered approach

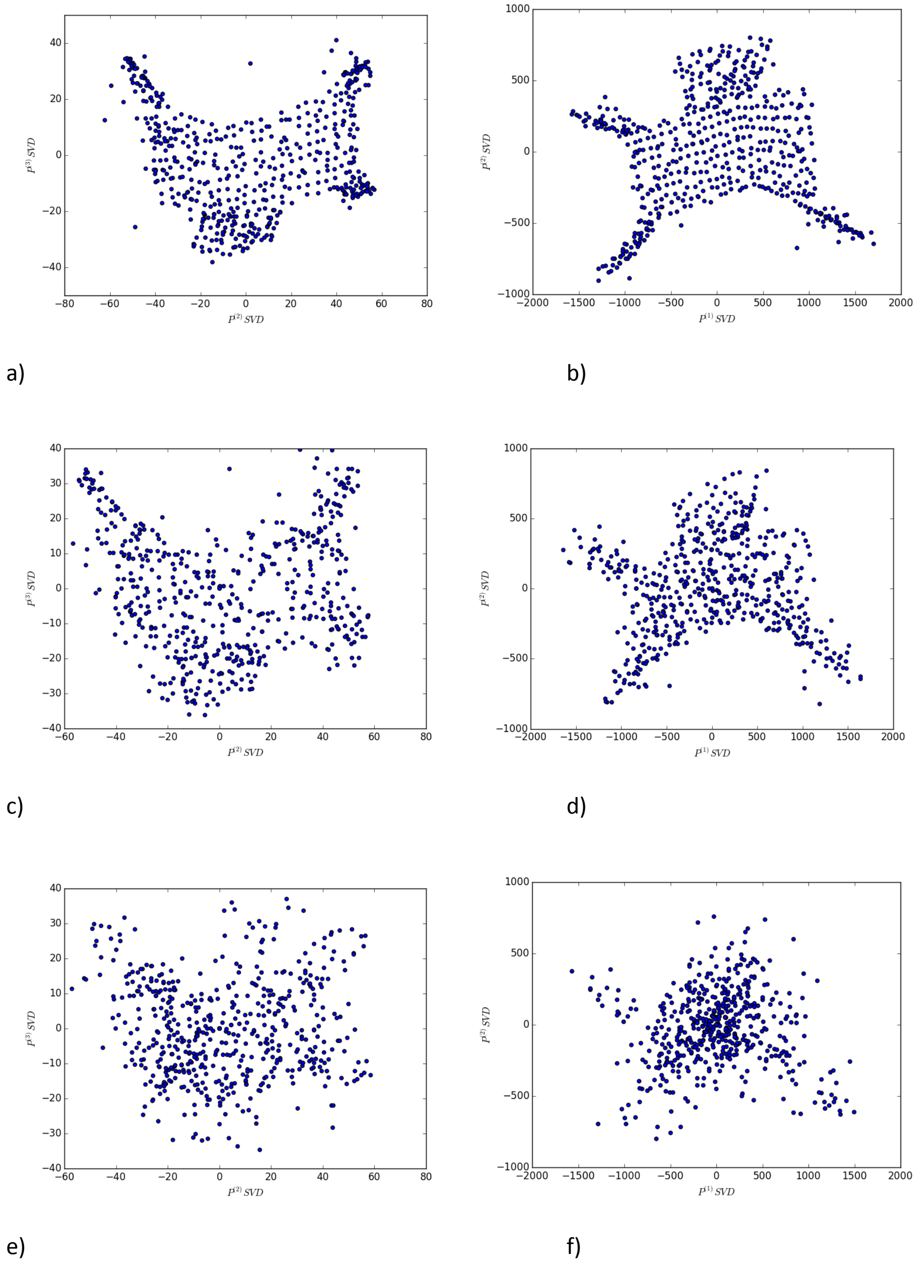


FIGURE 5.6. Topology Preserving Map for odd shaped network - random anchors: Recovered TPM with 40% coordinates missing (a) non-centered approach, (b) centered approach; Recovered TPM with 60% coordinates missing (c) non-centered approach, (d) centered approach; Recovered TPM with 80% coordinates missing (e) non-centered approach, (f) centered approach

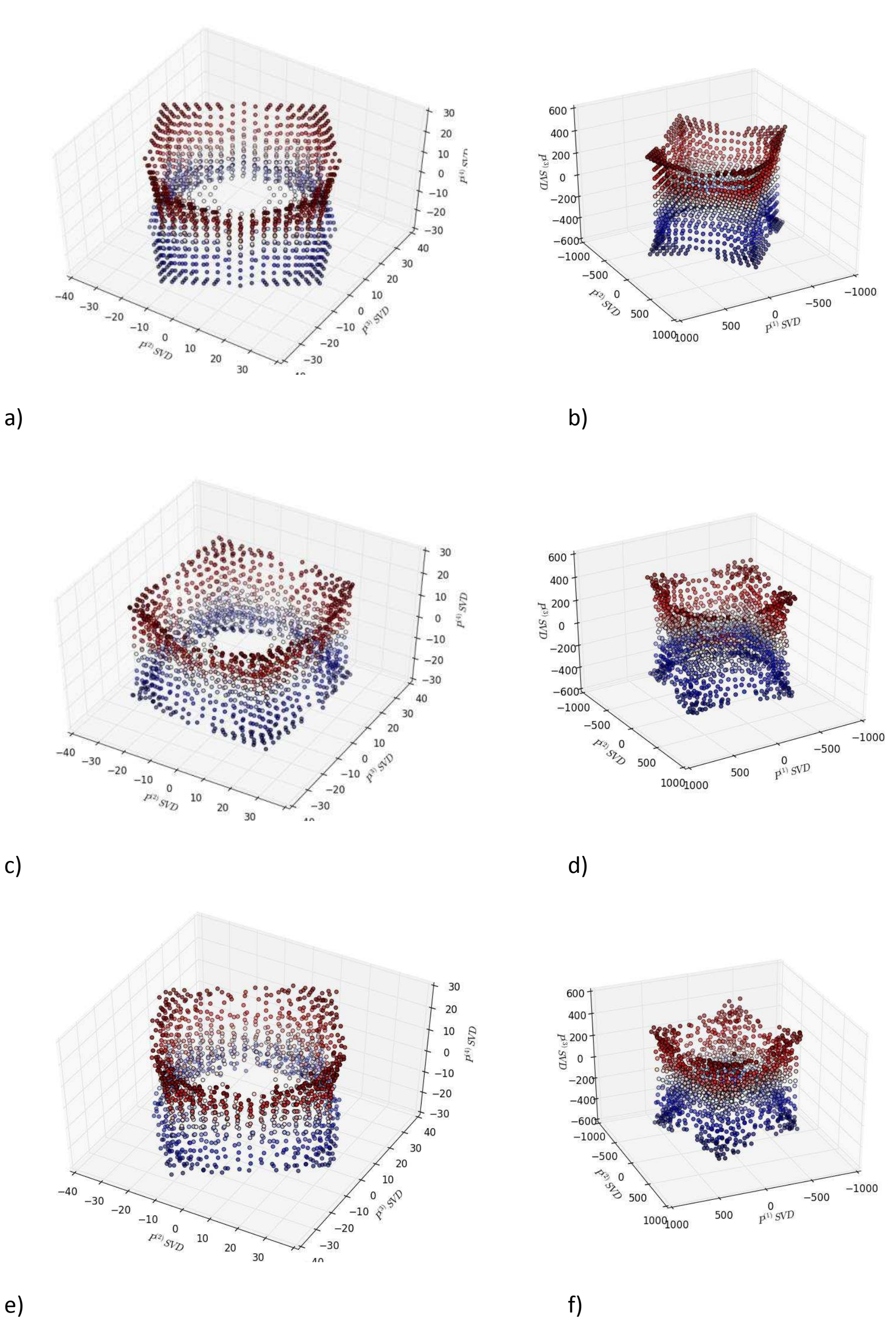


FIGURE 5.7. Topology Preserving Map for cube with hourglass shaped void - random anchors: TPM recovered from full set of VCs (a) non-centered approach, (b) centered approach; Recovered TPM with 10% random coordinates missing from VC matrix (c) non-centered approach, (d) centered approach; Recovered TPM with 20% random coordinates missing from VC matrix (e) non-centered approach, (f) centered approach

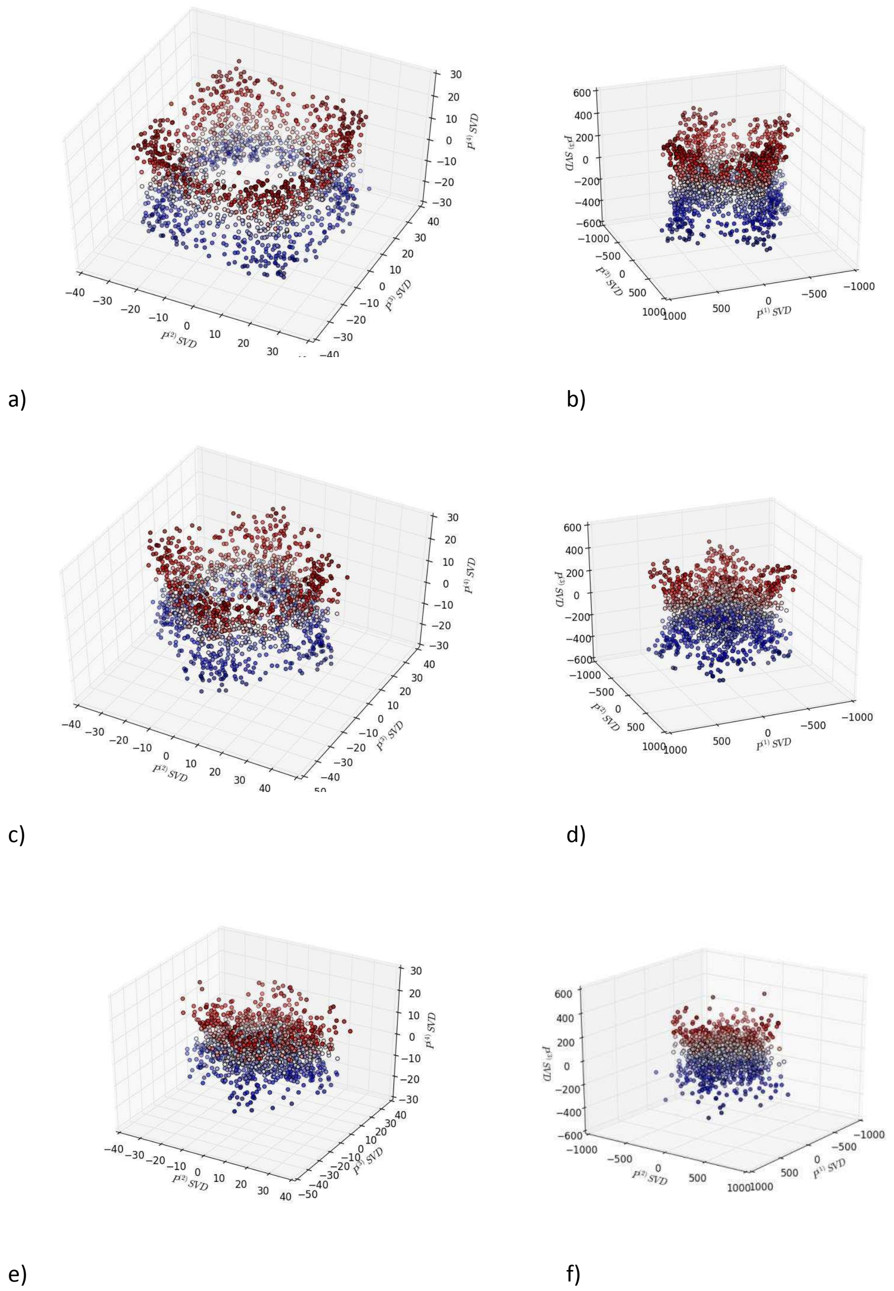


FIGURE 5.8. Topology Preserving Map for cube with hourglass shaped void - random anchors: Recovered TPM with 40% coordinates missing (a) non-centered approach, (b) centered approach; Recovered TPM with 60% coordinates missing (c) non-centered approach, (d) centered approach; Recovered TPM with 80% coordinates missing (e) non-centered approach, (f) centered approach

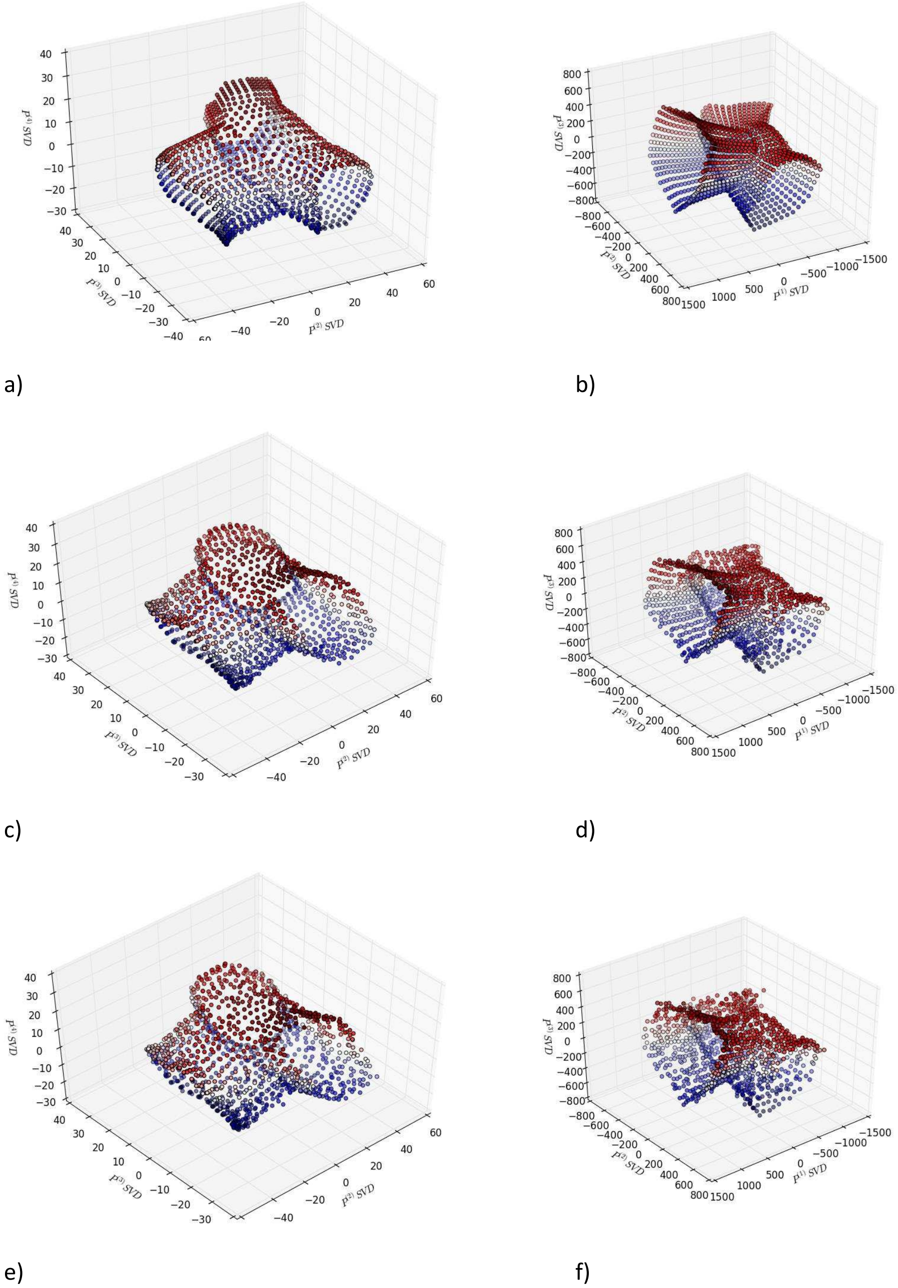


Figure 5.9. Topology Preserving Map for hollow T shaped cylinder network - random anchors: TPM recovered from full set of VCs (a) non-centered approach, (b) centered approach; Recovered TPM with 10% random coordinates missing from VC matrix (c) non-centered approach, (d) centered approach; Recovered TPM with 20% random coordinates missing from VC matrix (e) non-centered approach, (f) centered approach

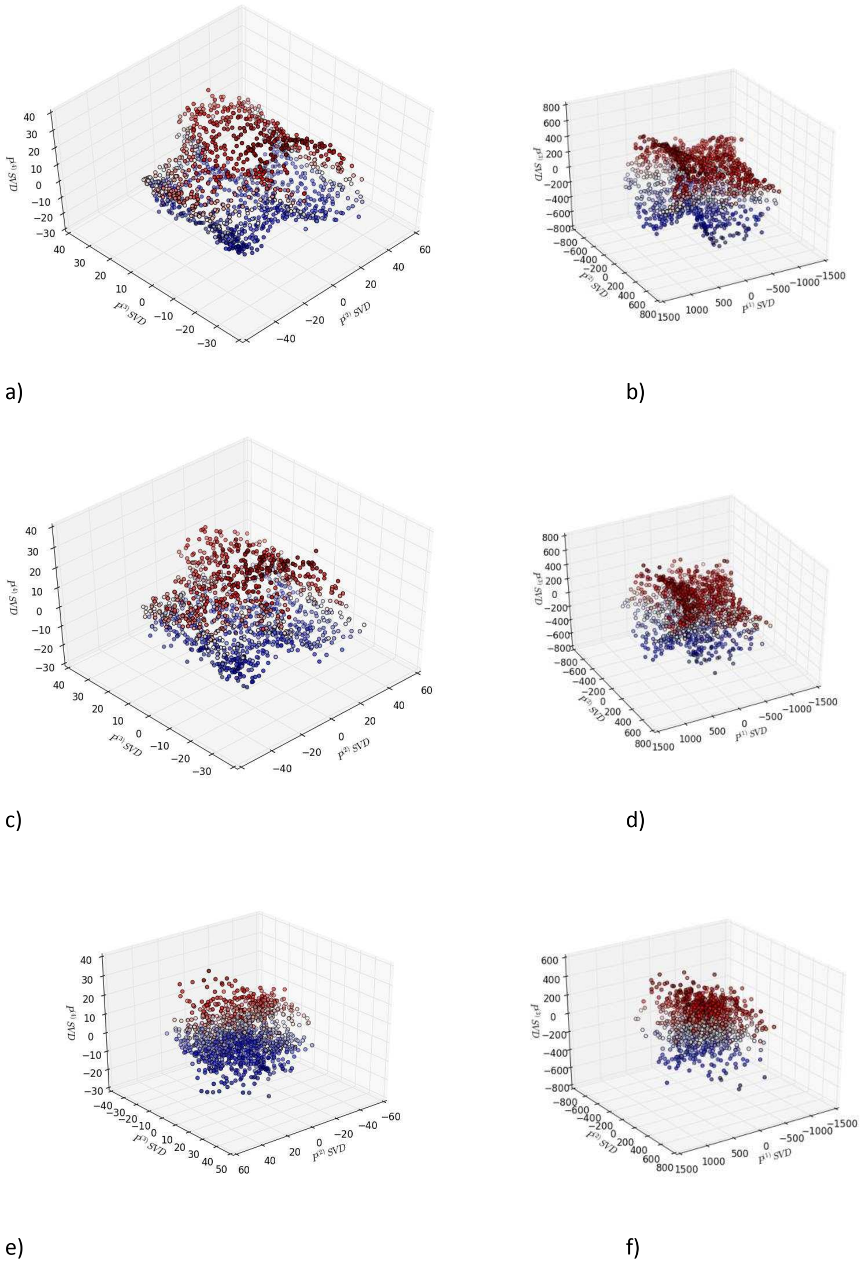


FIGURE 5.10. Topology Preserving Map for hollow T shaped cylinder network - random anchors: Recovered TPM with 40% coordinates missing (a) non-centered approach, (b) centered approach; Recovered TPM with 60% coordinates missing (c) non-centered approach, (d) centered approach; Recovered TPM with 80% coordinates missing (e) non-centered approach, (f) centered approach

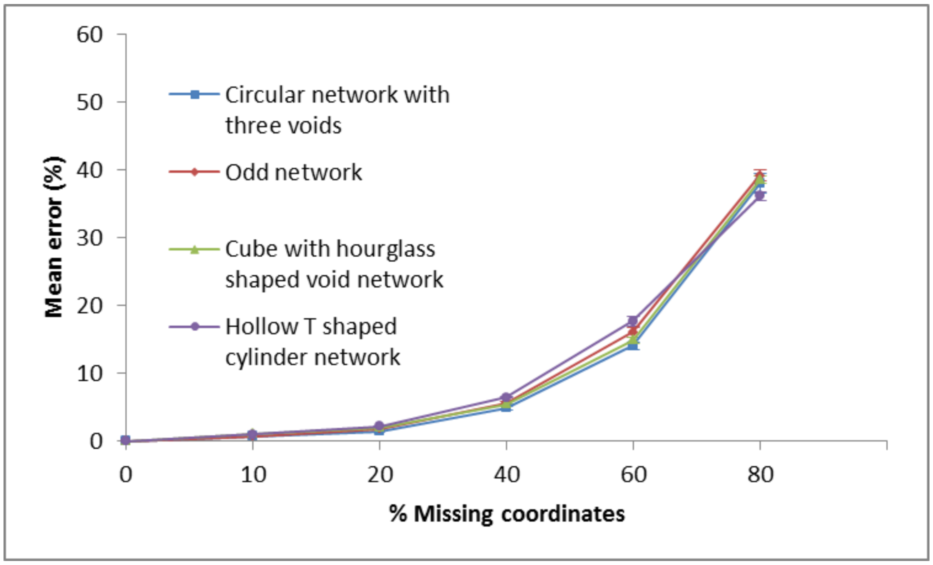


Figure 5.11. Mean error (with Standard Deviation) vs. the percentage of missing virtual coordinates for the circular network with three voids, odd shaped network, hollow T shaped cylinder network, and cube network with hourglass shaped void (non-centered approach)

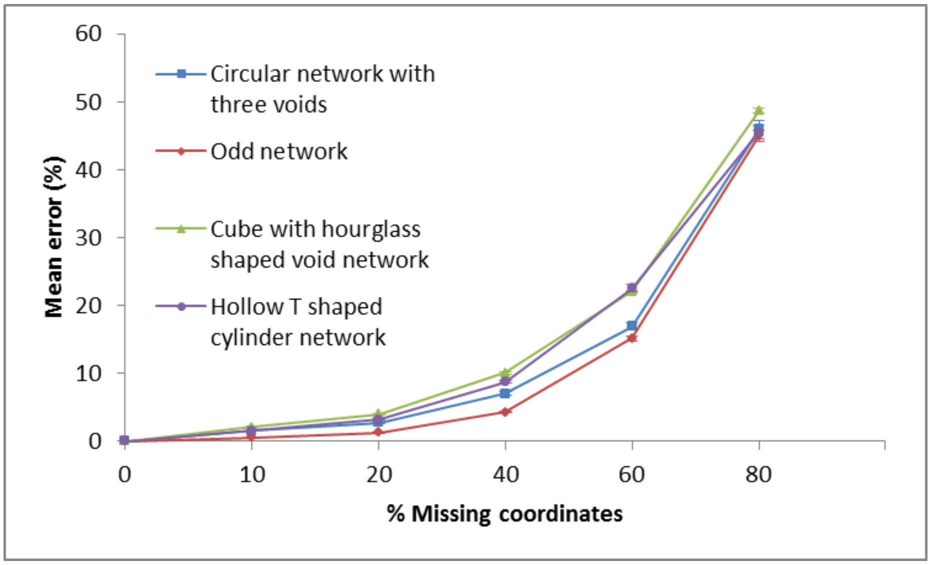
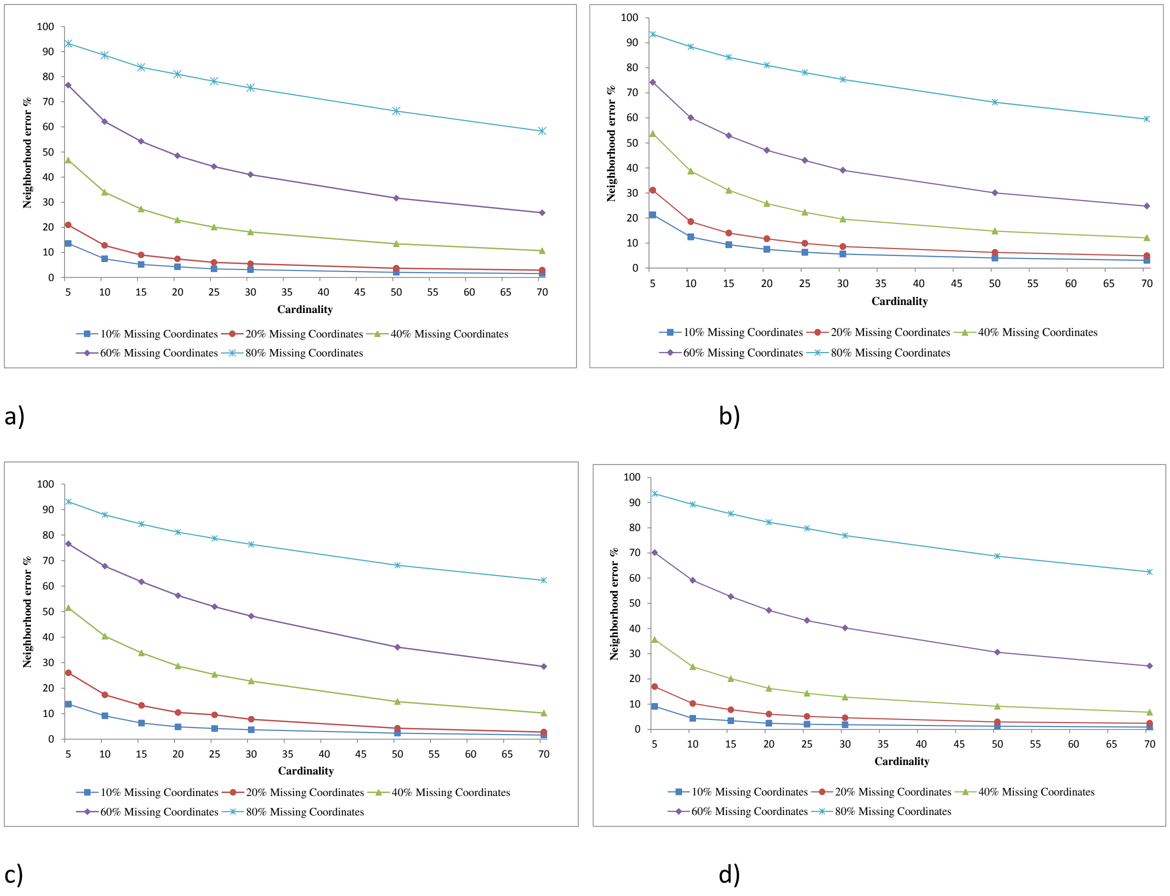


Figure 5.12. Mean error (with Standard Deviation) vs. the percentage of missing virtual coordinates for the circular network with three voids, odd shaped network, hollow T shaped cylinder network, and cube network with hourglass shaped void (centered-approach)



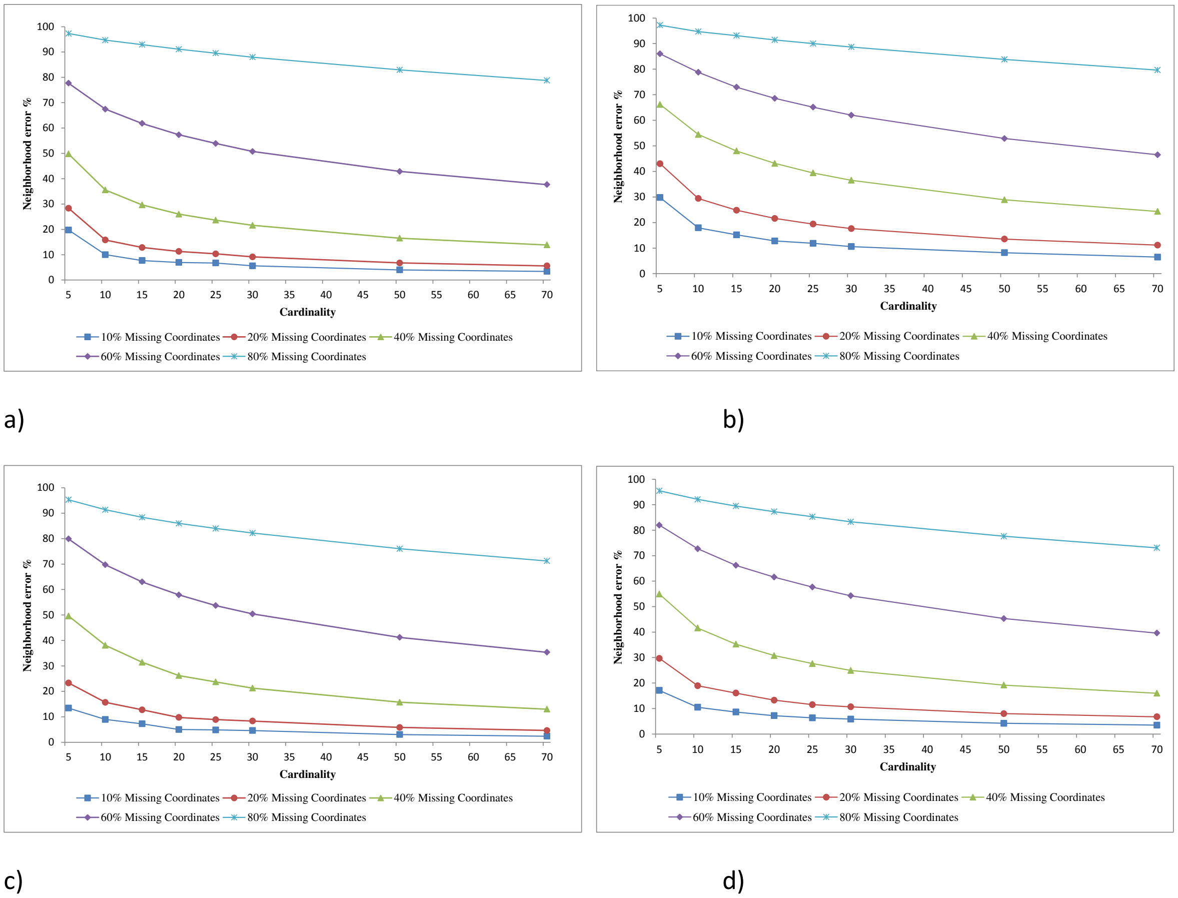
a)

b)

c)

d)

Figure 5.13. Neighborhood error:
circular network with three voids - random anchors (a) non-centered approach, (b) centered approach; odd shaped network - random anchors (c) non-centered approach, (d) centered approach



a)

b)

c)

d)

Figure 5.14. Neighborhood error:
cube with hourglass shaped void network - random anchors (a) non-centered approach, (b) centered approach; hollow T shaped cylinder network - random anchors (c) non-centered approach, (d) centered approach

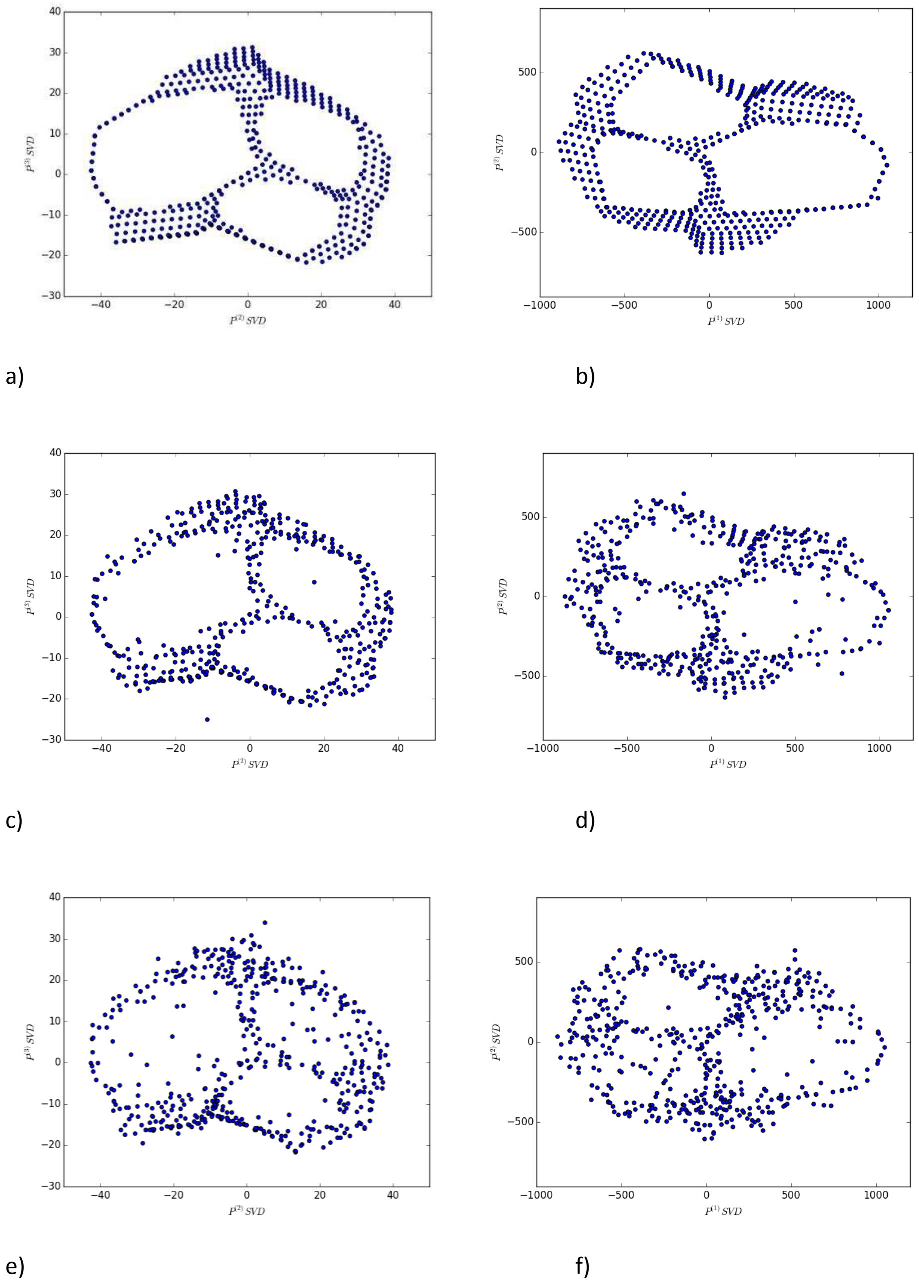


FIGURE 5.15. Topology Preserving Map for circular network with three voids - ENS anchors: TPM recovered from full set of VCs (a) non-centered approach, (b) centered approach; Recovered TPM with 10% random coordinates missing from VC matrix (c) non-centered approach, (d) centered approach; Recovered TPM with 20% random coordinates missing from VC matrix (e) non-centered approach, (f) centered approach

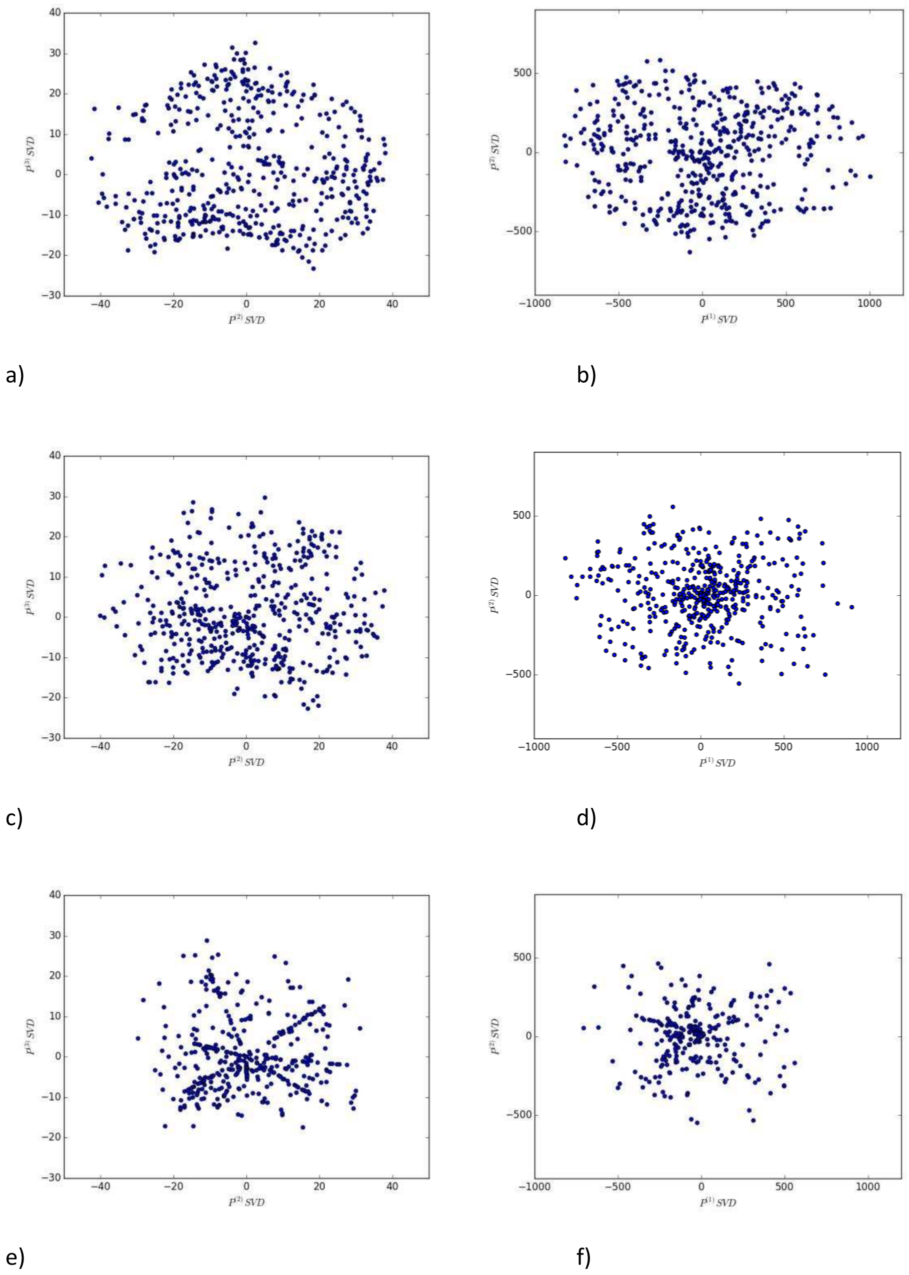


FIGURE 5.16. Topology Preserving Map for circular network with three voids - ENS anchors: Recovered TPM with 40% coordinates missing (a) non-centered approach, (b) centered approach; Recovered TPM with 60% coordinates missing (c) non-centered approach, (d) centered approach; Recovered TPM with 80% coordinates missing (e) non-centered approach, (f) centered approach

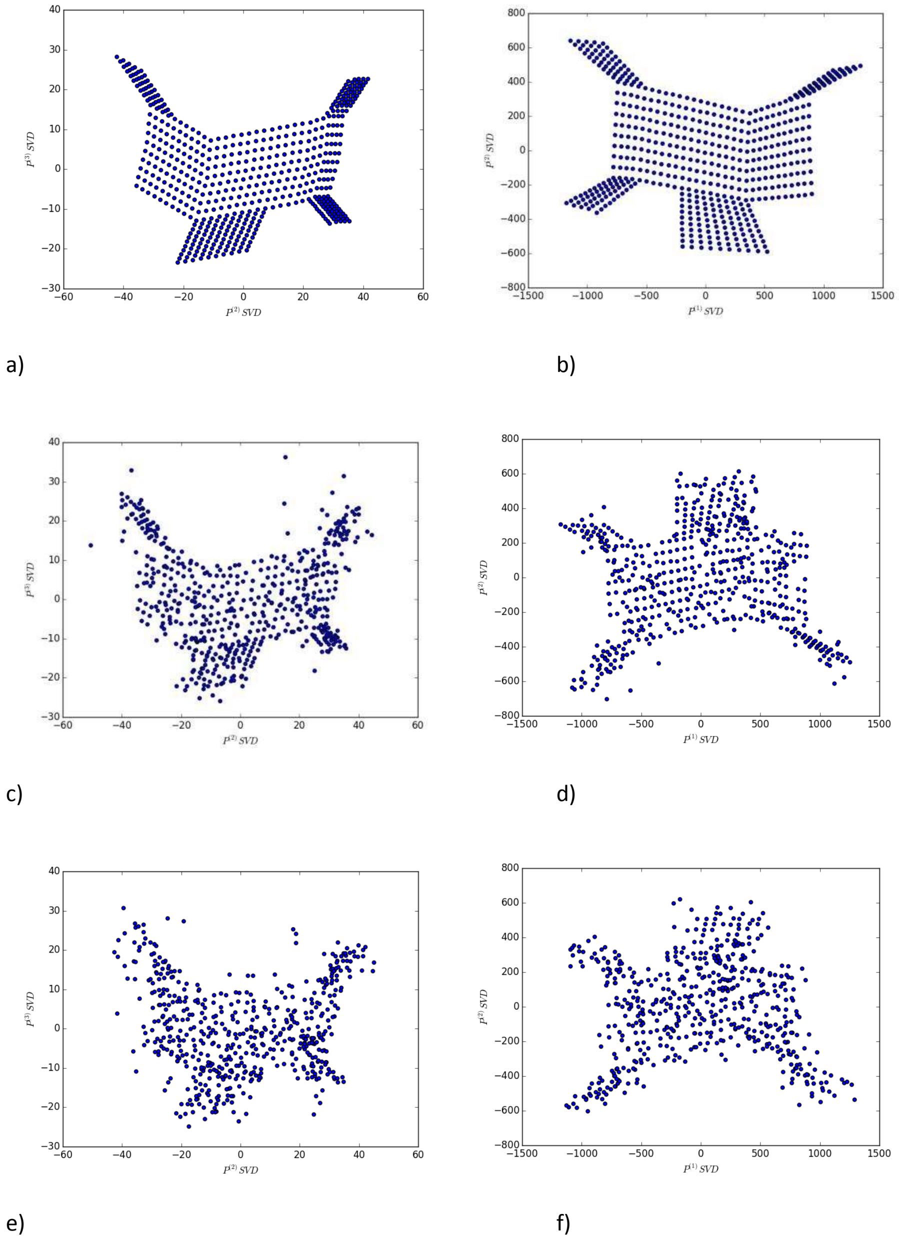


FIGURE 5.17. Topology Preserving Map for odd shaped network - ENS anchors: TPM recovered from full set of VCs (a) non-centered approach, (b) centered approach; Recovered TPM with 10% random coordinates missing from VC matrix (c) non-centered approach, (d) centered approach; Recovered TPM with 20% random coordinates missing from VC matrix (e) non-centered approach, (f) centered approach

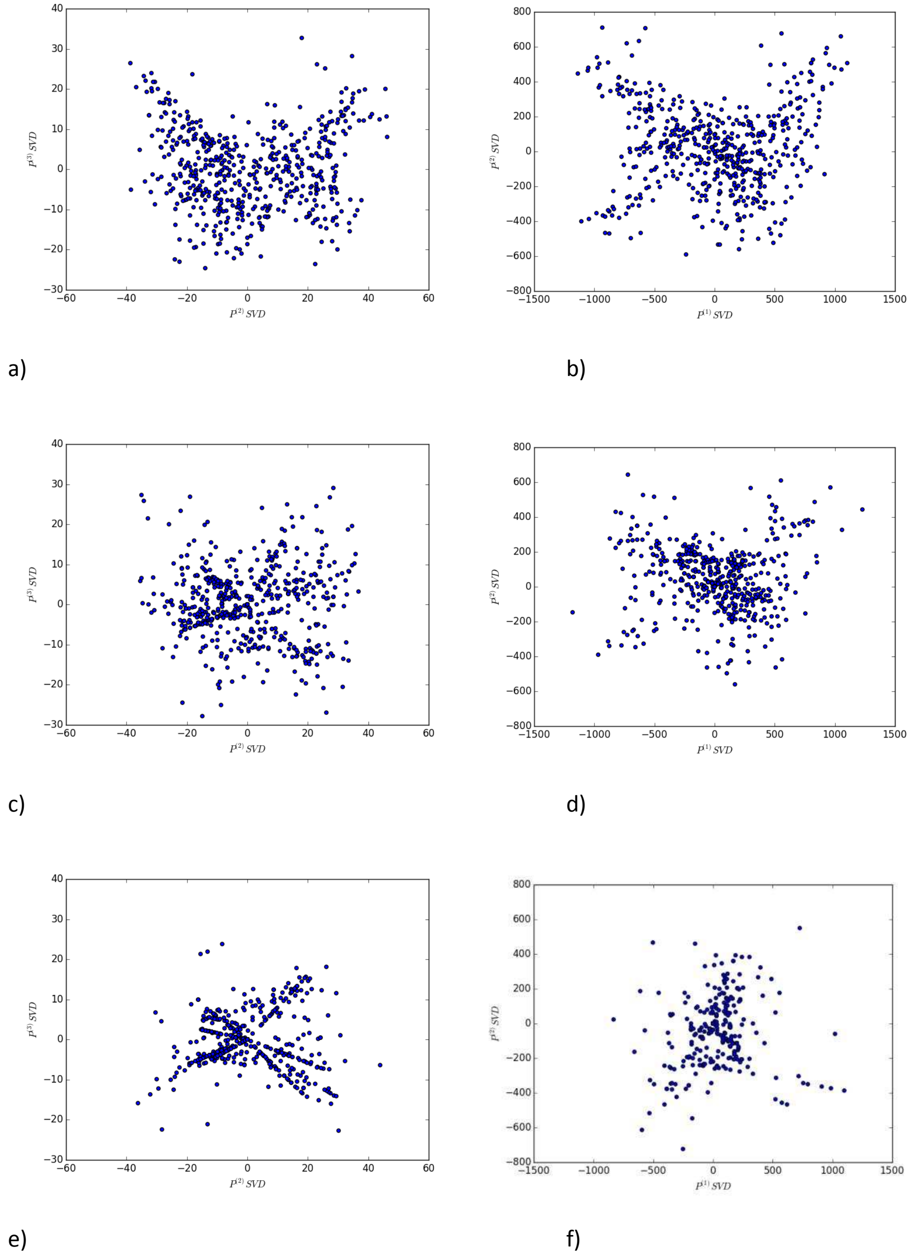


FIGURE 5.18. Topology Preserving Map for odd shaped network - ENS anchors: Recovered TPM with 40% coordinates missing (a) non-centered approach, (b) centered approach; Recovered TPM with 60% coordinates missing (c) non-centered approach, (d) centered approach; Recovered TPM with 80% coordinates missing (e) non-centered approach, (f) centered approach

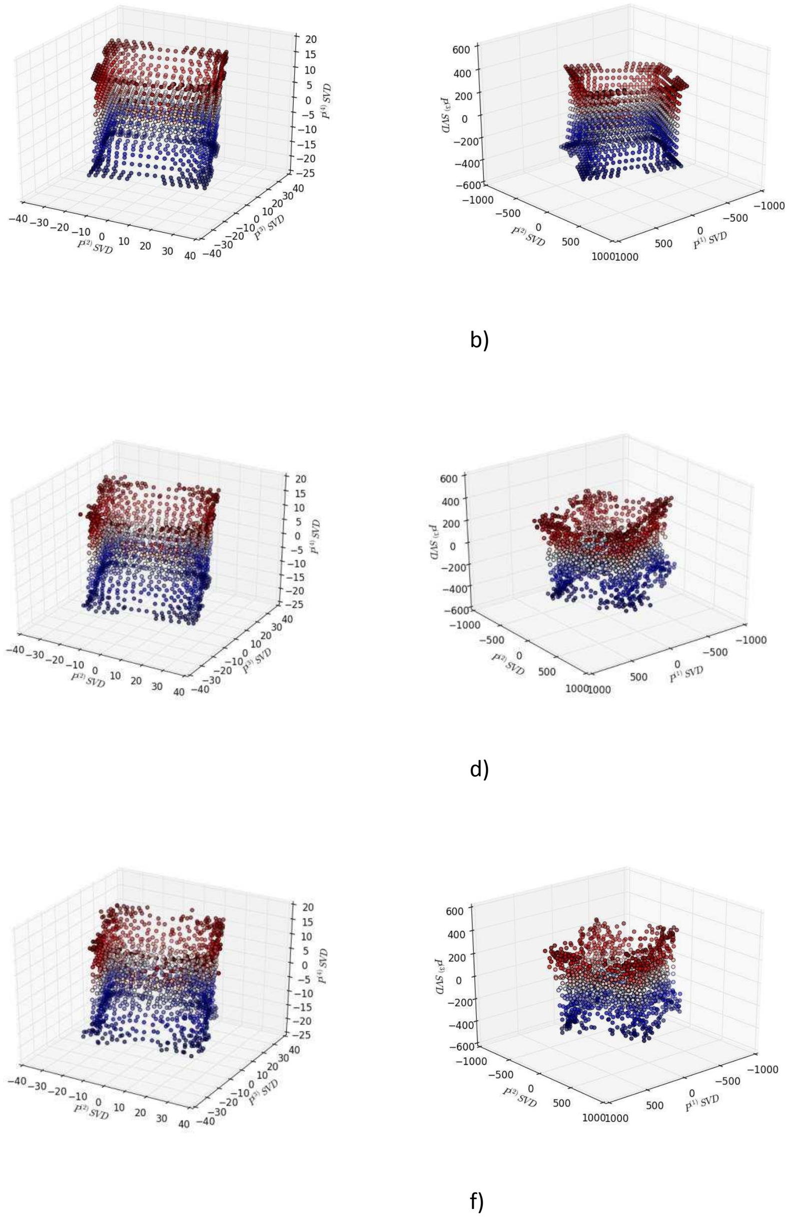


FIGURE 5.19. Topology Preserving Map for cube with hourglass shaped void - ENS anchors: TPM recovered from full set of VCs (a) non-centered approach, (b) centered approach; Recovered TPM with 10% random coordinates missing from VC matrix (c) non-centered approach, (d) centered approach; Recovered TPM with 20% random coordinates missing from VC matrix (e) non-centered approach, (f) centered approach

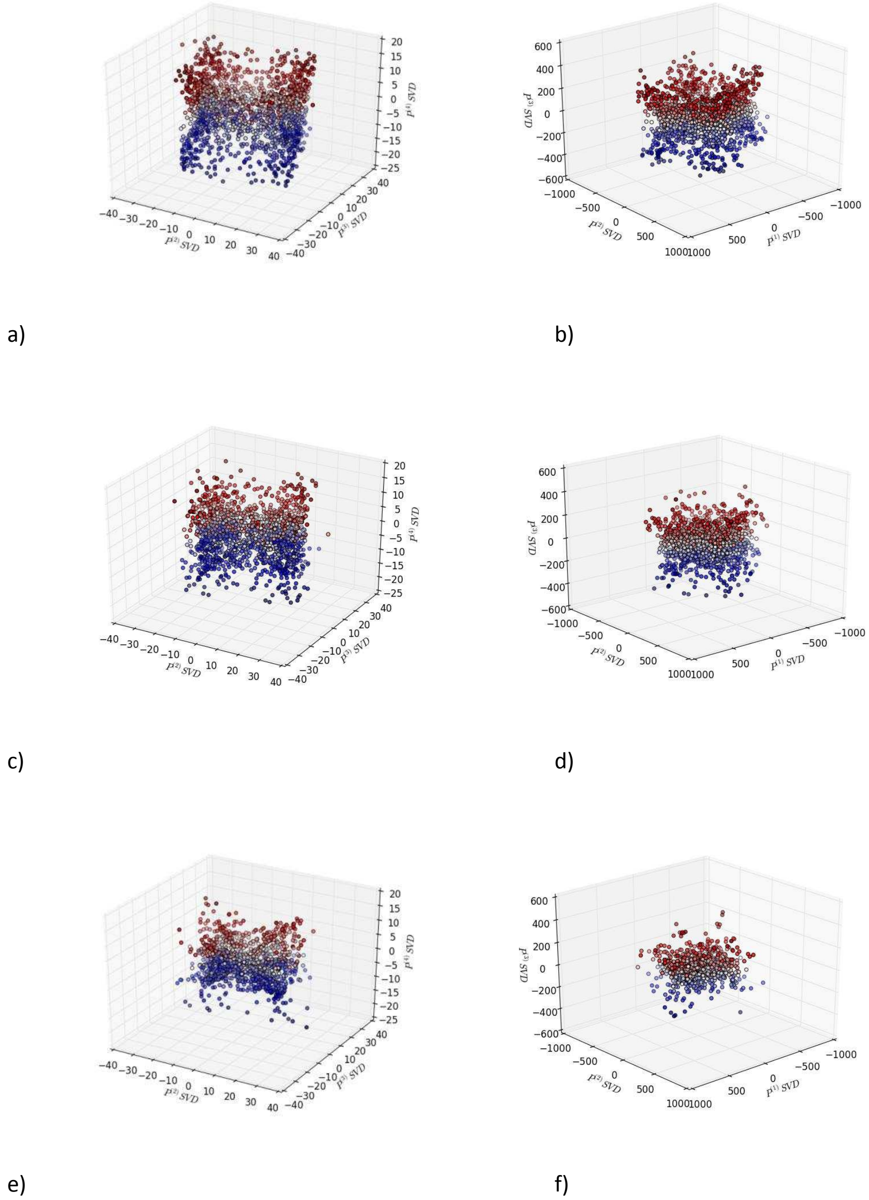


Figure 5.20. Topology Preserving Map for cube with hourglass shaped void - ENS anchors: Recovered TPM with 40% coordinates missing (a) non-centered approach, (b) centered approach; Recovered TPM with 60% coordinates missing (c) non-centered approach, (d) centered approach; Recovered TPM with 80% coordinates missing (e) non-centered approach, (f) centered approach

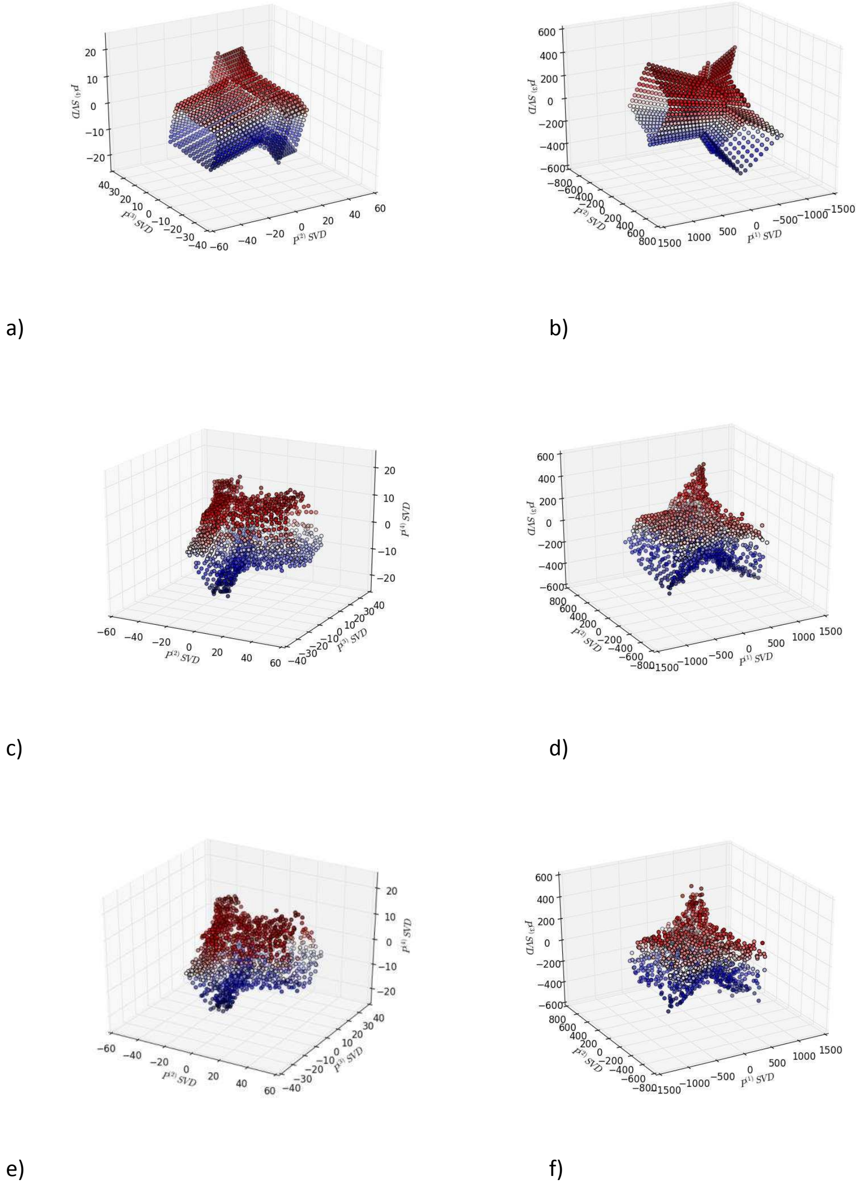


FIGURE 5.21. Topology Preserving Map for hollow T shaped cylinder - ENS anchors: TPM recovered from full set of VCs (a) non-centered approach, (b) centered approach; Recovered TPM with 10% random coordinates missing from VC matrix (c) non-centered approach, (d) centered approach; Recovered TPM with 20% random coordinates missing from VC matrix (e) non-centered approach, (f) centered approach

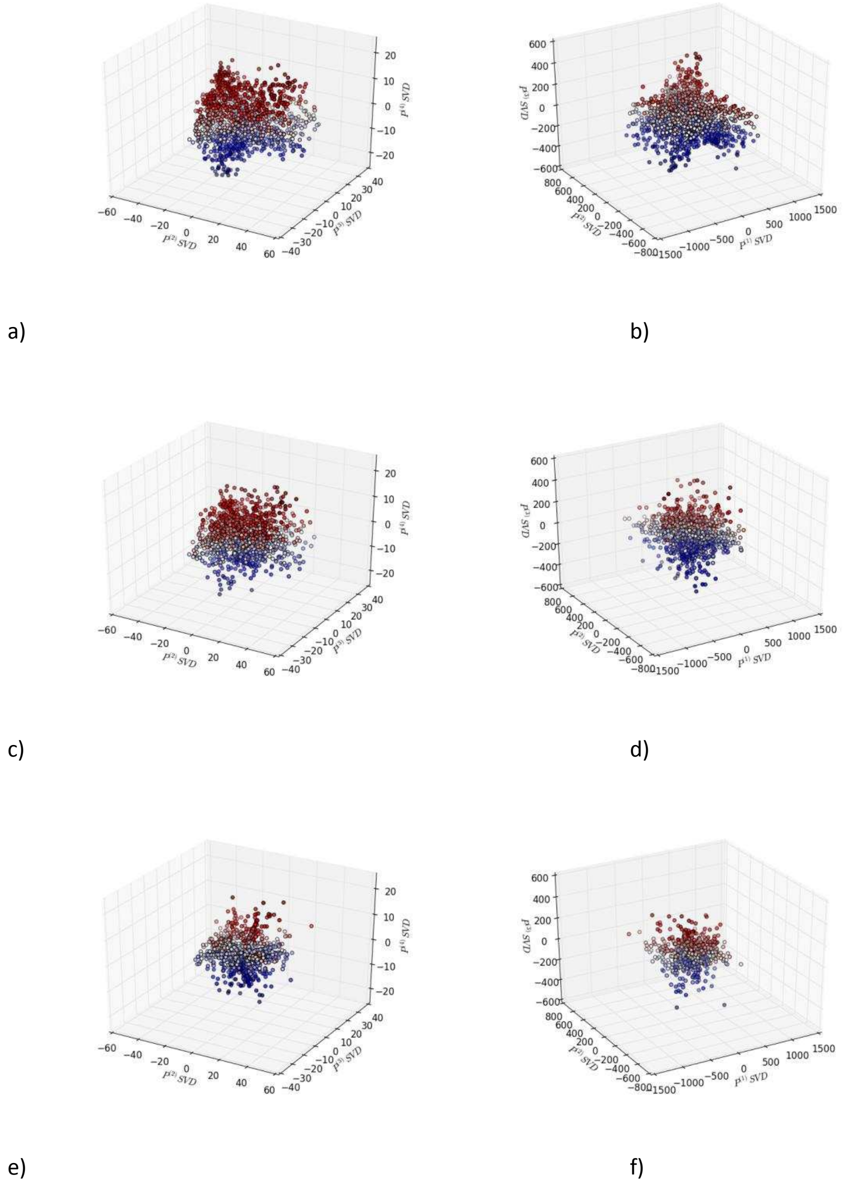


FIGURE 5.22. Topology Preserving Map for hollow T shaped cylinder - ENS anchors: Recovered TPM with 40% coordinates missing (a) non-centered approach, (b) centered approach; Recovered TPM with 60% coordinates missing (c) non-centered approach, (d) centered approach; Recovered TPM with 80% coordinates missing (e) non-centered approach, (f) centered approach

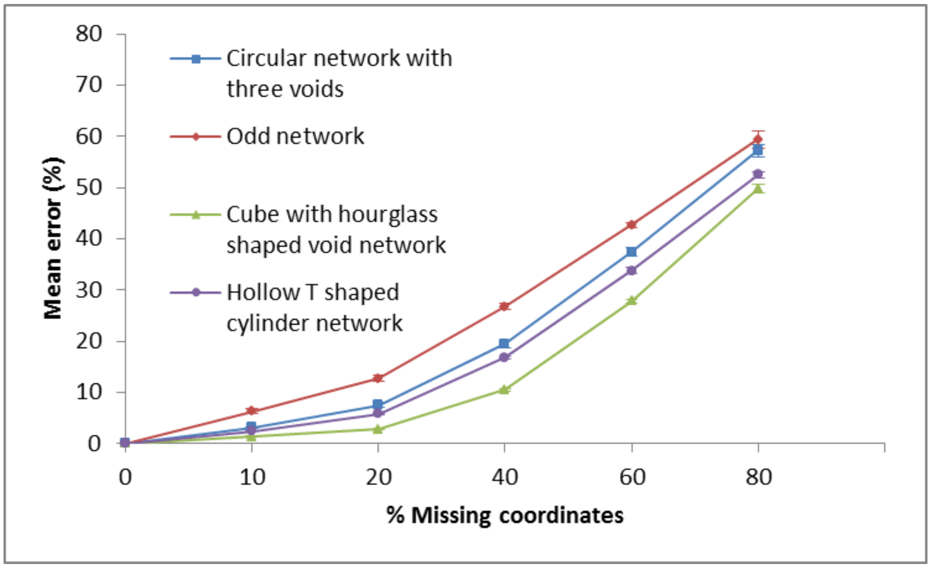


FIGURE 5.23. Mean error (with Standard Deviation) vs. the percentage of missing virtual coordinates for the circular network with three voids, odd shaped network, hollow T shaped cylinder network, and network with hourglass shaped void (non-centered approach)

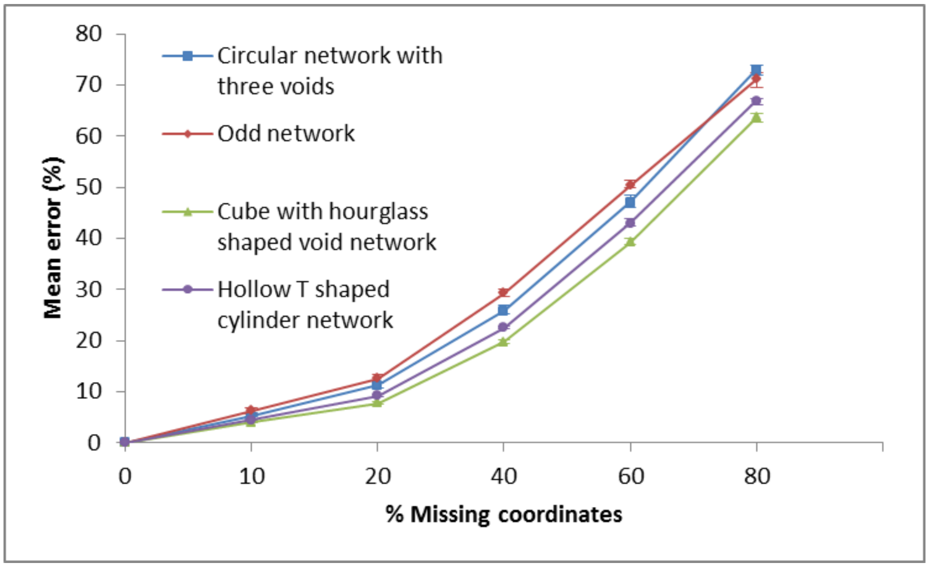


FIGURE 5.24. Mean error (with Standard Deviation) vs. the percentage of missing virtual coordinates for the circular network with three voids, odd shaped network, hollow T shaped cylinder network, and network with hourglass shaped void (centered approach)

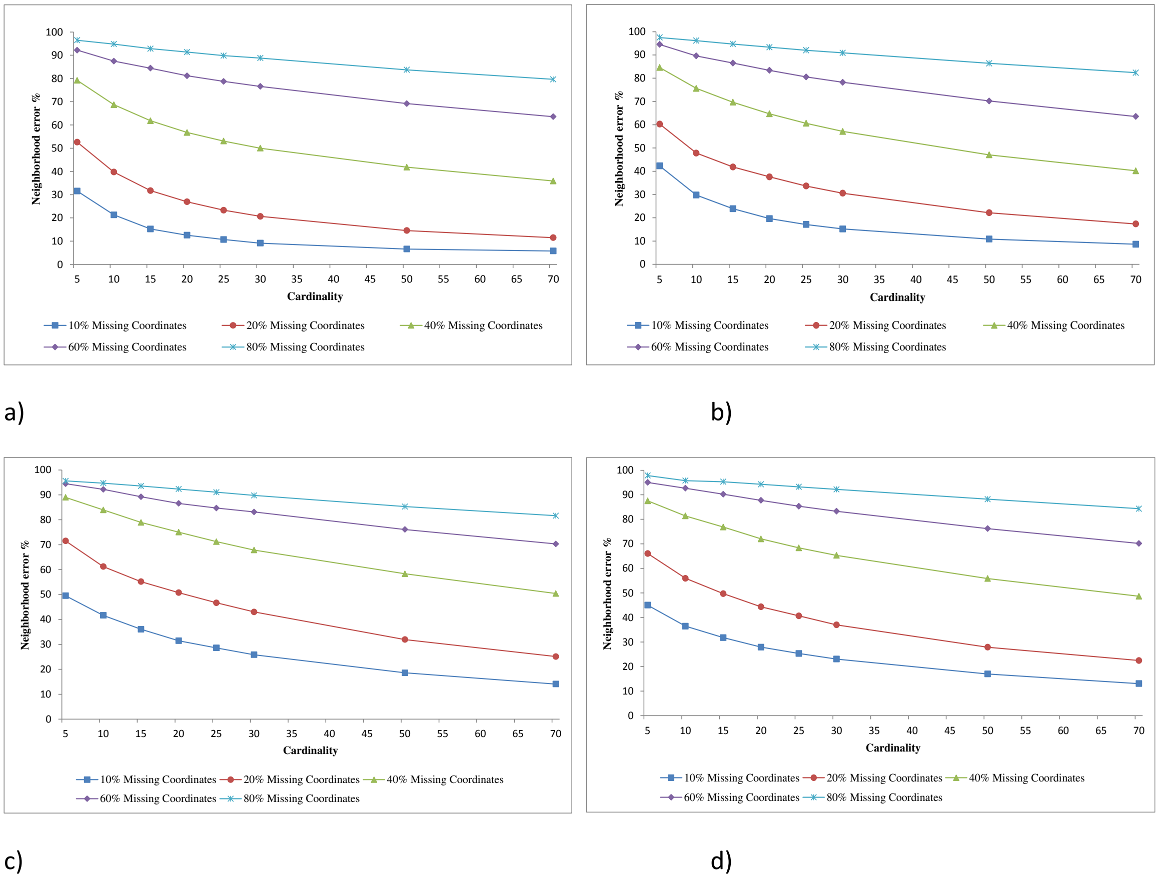
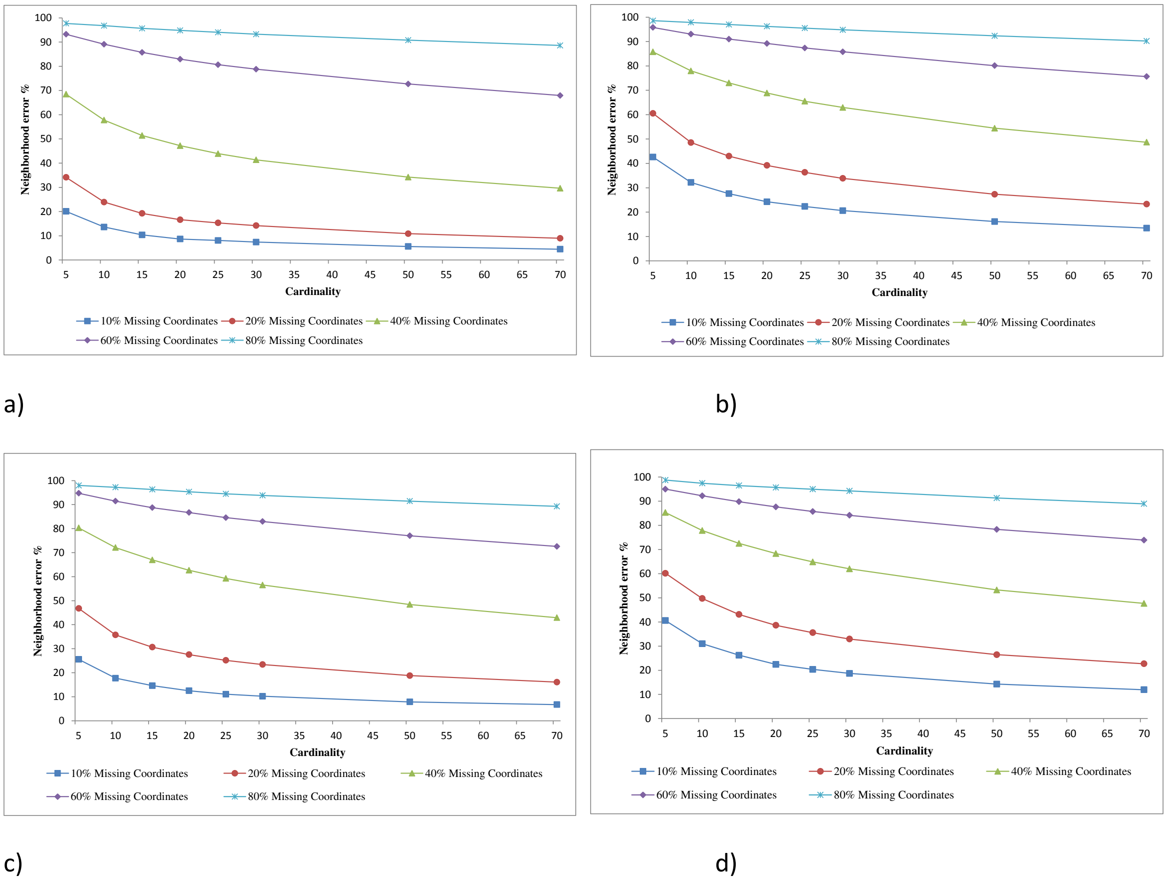


FIGURE 5.25. Neighborhood error:
circular network with three voids - ENS anchors (a) non-centered approach, (b) centered approach; odd shaped network - ENS anchors (c) non-centered approach, (d) centered approach



a)

b)

c)

d)

FIGURE 5.26. Neighborhood error:
cube with hourglass shaped void network - ENS anchors (a) non-centered approach, (b) centered approach; hollow T shaped cylinder network - ENS anchors (c) non-centered approach, (d) centered approach

CHAPTER 6

Results for Social Network graphs

The social networks are networks that are formed by the interaction between social actors. As mentioned in 2, social networks exhibit different properties compared to other types of networks. This gives us an inquisitive reason to study these graphs. Traditionally machine learning algorithms have been used predominantly for link prediction between two nodes in a social network. This chapter discusses the application of eRPCA algorithm for link prediction of partially observed networks. The social networks used in this research have been taken from Stanford Network Analysis Project [39].

6.1. Description of social network

Three different networks have been used to demonstrate the effectiveness of the approach. A facebook sub network, a scientific collaboration sub network (Arxiv GR-QC (General Relativity and Quantum Cosmology)) and Enron Email sub network consisting of 744, 4158 and 3892 nodes respectively. Unlike the WSNs social networks are dense locally and sparse globally. It can also be seen that the maximum shortest path length is less compared to WSNs. The Table 6.1 gives details of the social networks.

6.2. Approach

The social network link prediction is applied for the three sample networks with two different measurements. As an extension to the anchor based virtual coordinate representation, random anchors have been chosen for social networks to express it as VC matrix . There is rich literature that suggests ways to capture information from social networks with web crawlers [18] [40] [41] [42].

Table 6.1. Characteristics of social networks

Network	Size	Diameter	Avg. path length	Number of random anchors
Facebook network	744	7	2.55	20/100/150
Collaboration network	4158	17	6.049	100/150/200
Enron Email network	3892	5	3.139	100/150/200

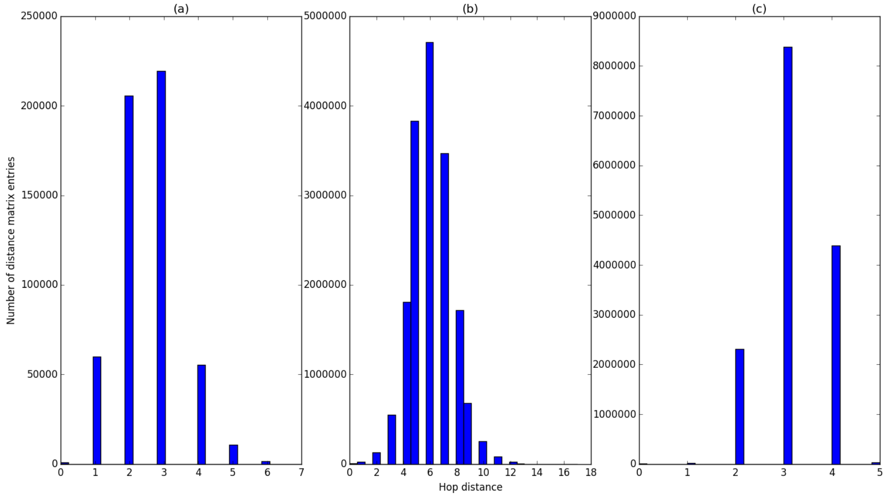


Figure 6.1. Histogram of hop-distances of a) Facebook network, b) Collaboration network, c) Enron Email network

Some of the common graph sampling techniques on social network employs either graph traversal method or random walk methods. The common graph traversal methods are Breadth First Search (BFS) or Depth First Search (DFS). However, it may lead to bias towards higher degree nodes. The latest work on random walk based graph sampling addresses this issue of bias [42]. Owing to massive size of such graphs, the power of parallel and distributed computing can be utilized to capture data at a rapid rate. Attention is to be paid while choosing the origin node that initiates the graph sampling to make sure that the right population is sampled. The anchors may be self-selected random nodes, e.g., each node becomes an anchor with some probability. This is the strategy used for results in this research. Alternatively, some type of strategy, e.g. based on connectivity or ease of measurement, may also be used. An anchor based method is relevant for social network, because many prominent nodes are being closely followed on social networks. This will help to study the complex structure around these prominent anchors. A sampling technique employing a limited broadcast originated by each prominent anchor and nodes responding with the hop count will give us a good sub sample of Virtual Coordinate matrix. The Algorithm 2 gives a generic method to capture the connectivity from a social network. The current research work is on undirected graphs and hence, the reciprocity of nodes are assumed.

Algorithm 2 Measuring Connectivity from Anchors

```
procedure ANCHOR SELECTION
    $M$ anchors are chosen randomly.
end procedure

procedure AS AN ANCHOR NODE
    for all anchor nodes $M$ do
        Send broadcast message to its neighbors
        with a TTL of $t$
    end for
    Response will constitute the
    subset of VC matrix.
end procedure

procedure AS AN INDIVIDUAL NODE/ANCHOR NODE
    if $t > 0$ then
        Reply back to source anchor with hop-count
        Forward the message to neighbor with
        $t$ decreased by one if it is an individual node.
    else
        Stop forwarding.
    end if
end procedure
```

This approach can be implemented on social networks by broadcasting message to friends and in turn encouraging them to forward the message. Not all of them would be interested to cooperate and resulting measure will be a clear subset of VC matrix. A time to live t

should be set to limit the messages in the network. As mentioned earlier, the social networks usually abide by the “six degrees of separation” rule [43] and so it can be set to 6. It’s assumed that random walks from anchors will lead to overlap of nodes. For a collaboration network, we can use this idea by looking at the co-authors of a paper and further branching out to look for other collaborators. For an E-mail network, the same method can be applied by sending e-mails from one node to another with a limit in broadcasting.

The matrix completion algorithm is applied on sub sample of virtual coordinate matrix to reconstruct the topology. With increased privacy of nodes, it is fair enough to say that, it gets quite tough to obtain the hop distance information from an anchor perspective. The nodes may not be interested to reveal the connectivity information. This in turn becomes a bottleneck in predicting the links between nodes. Thus we further extended this approach for any random subset of information available not just limited to anchor based methods. A sub sample of virtual coordinate matrix is also a sub sample of the total hop distance matrix. These random samples of graphs can be obtained by using web crawlers or the data captured from the anchor based method can be reorganized to get it as a sub sample of Distance matrix D.

Two metrics have been introduced to validate the accuracy of the proposed approach.

1) MEAN ERROR The mean error gives the accuracy of prediction of the matrix. The mean error is a measure of percentage error in prediction with respect to the original matrix. The mean error is defined as follows,

$$E = \left[\sum_{i,j=1}^{N,M} |P_{ij}(f) - P_{ij}(0)| \right] / \left[\sum_{i,j=1}^{N,M} P_{ij}(0) \right] \tag{6.1}$$

where, $P_{ij}(f)$ refers to the VC matrix element denoted by row i and column j when f fraction of random anchor coordinates are missing.

2) ABSOLUTE HOP DISTANCE ERROR The absolute hop distance error is the average error in hop distance found in the prediction. The absolute hop-distance error is defined as:

$$H = \left[\sum_{i,j=1}^{N,M} |P_{ij}(f) - P_{ij}(0)| \right] / [N.M] \tag{6.2}$$

where $P_{ij}(f)$ refers to VC matrix element in i^{th} row, j^{th} column and f denotes the percentage of missing coordinates . $N.M$ denotes the total number of elements in the VC matrix.

6.3. Results

This research shows that, with partial information either as shortest hop distances between each node to a set of anchors or as shortest hop distances between random pairwise nodes the social network link prediction and hence the topology can be obtained with good accuracy. Also, this provides us with means of describing graphs in compressed representation. The two subsections below discuss the results obtained with the two different measurements. The Figure 6.2 is a logarithmic plot of singular values of the distance matrix of the three networks. This shows that, the principles of matrix completion can be applied for these matrices.

6.3.1. Matrix completion from partially observed entries of Virtual Coordinate matrix.

The results have been obtained for different number of anchors for each of the networks. We have chosen 20,100 and 150 anchors for Facebook network and 100,150 and 200 anchors for Collaboration network and Enron Email network respectively. To demonstrate the effectiveness of the approach let us also study the singular values of VC matrices for all the three networks. The Figure 6.3 - 6.5 shows the plot of singular values of each component for different anchor numbers. It can be seen that as we choose more number of anchors, the resultant matrix becomes low rank relatively compared to the VC matrix with lesser number of anchors. The problems with singular values such as these can be solved using matrix completion techniques. Next we randomly discard 10%, 20% up to 90% from the VC matrix. The matrix completion is applied and predicted matrix is obtained.

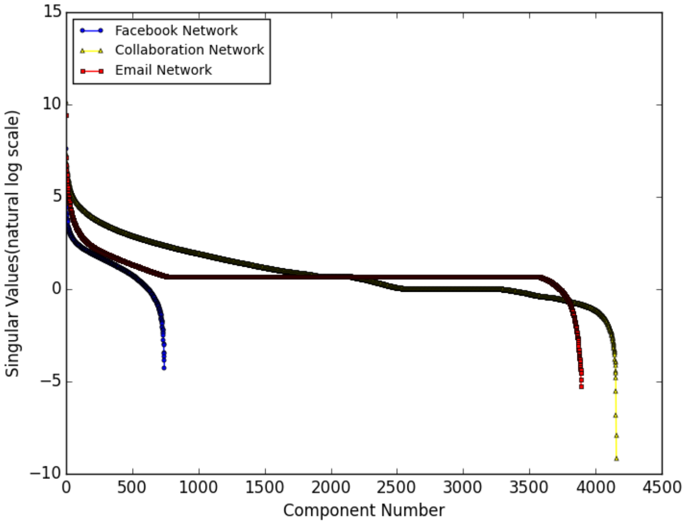


FIGURE 6.2. Singular values(Log) of distance matrix of all the three networks indicating that they are naturally close to low-rank

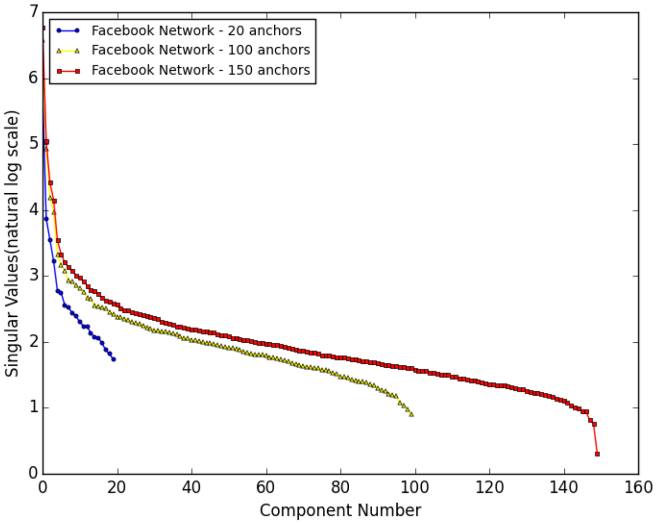


FIGURE 6.3. Singular values(Log) of VC matrix of Facebook network indicating that they are naturally close to low-rank

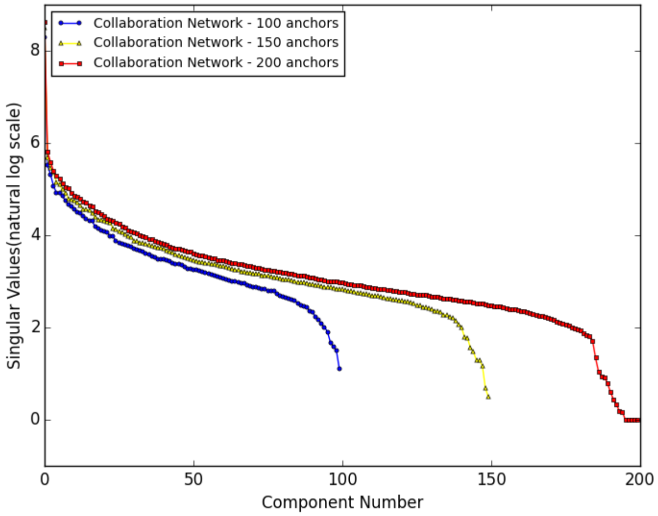


FIGURE 6.4. Singular values(Log) of VC matrix of Collaboration network indicating that they are naturally close to low-rank

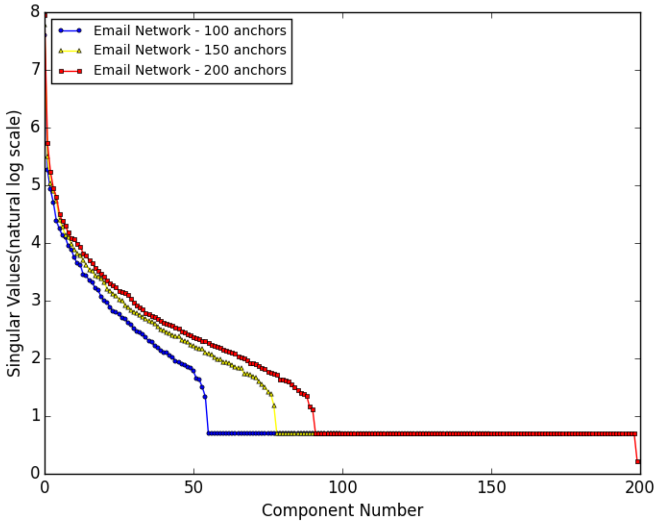


FIGURE 6.5. Singular values(Log) of VC matrix of Email network indicating that they are naturally close to low-rank

The accuracy of predicted matrix is evaluated with the above mentioned two metrics, mean error and absolute hop distance error. The predicted matrix is rounded off before evaluating the metrics. The Figure 6.6 and Figure 6.7 shows the mean error and absolute error for Facebook network. Though the mean error is around 15% the absolute error in predicting the entries is almost less than a hop. The accuracy for Collaboration network can be seen in Figure 6.8 and Figure 6.9. The maximum mean error is around 15% while the entries are predicted within a error of 0.5 hop. Similarly, for Email network, the Figure 6.10 and Figure 6.11 shows the accuracy of the approach. The mean error in prediction is around 8% while the absolute hop distance error is found to be very less. The entries are predicted within 0.2 hop error. The Figures 6.12 - 6.14 shows the histogram plot of predicted matrix with different percentage of virtual coordinate matrix entries missing for Facebook network (20 anchors), Collaboration network (150 anchors) and Enron Email network (150 anchors) respectively. This is obtained for one of the simulation run. This indicates that with 90% elements missing, most of the entries of predicted matrix can be seen to be well within 2 hop distance error for Facebook and Collaboration network and within 1 hop for Enron Email network. There are entries for other higher hop distance error values. The range is limited in the figure. The maximum observed absolute hop distance error is found to be 6 hops for Facebook network when 90% entries are removed. Similarly the maximum absolute hop distance error is found to be 12 hops and 5 hops respectively for Collaboration network and Enron Email network.

6.3.2. Matrix completion from partially observed entries of the distance matrix. This section analyzes the performance of the second method based on distance matrix. The same three networks have been used for this. The distance matrix is symmetric i.e. $D_{ij} = D_{ji}$. To remove one connectivity, both D_{ij} and D_{ji} should be dropped. Thus, entries from both lower and upper triangle of distance matrix should be dropped as a pair. The results have been obtained for 4 cases, 20%, 40%, 60%, 80% random drop of the distance matrix. The singular values of the distance matrix were observed to be low rank. The accuracy of this approach is evaluated the same way as evaluated for virtual coordinate

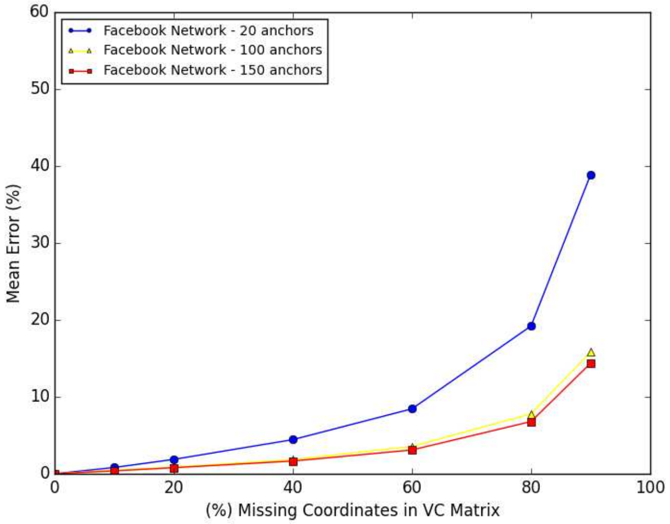


Figure 6.6. Mean error vs percentage of missing coordinates of VC matrix for different anchors - Facebook Network

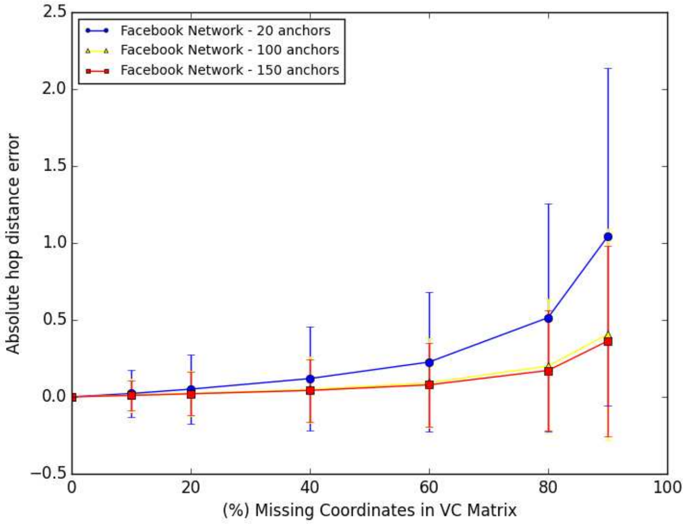


Figure 6.7. Absolute error in hop-distance with std. deviation vs percentage of missing coordinates of VC matrix for different anchors - Facebook Network

matrix. In addition, the known entries of the original matrix are replaced in the predicted matrix. Also, the predicted matrix is rounded off to the closest integer value and finally the error is calculated. Mean error defined in 6.1 is computed for the predicted distance matrix.

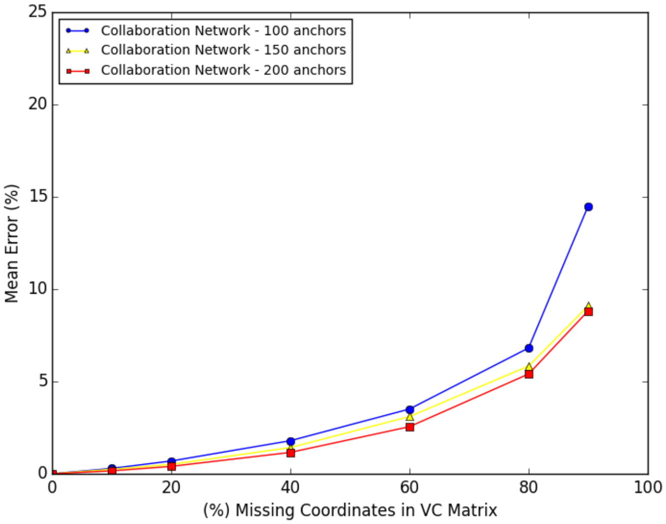


FIGURE 6.8. Mean error vs percentage of missing coordinates of VC matrix for different anchors - Collaboration Network

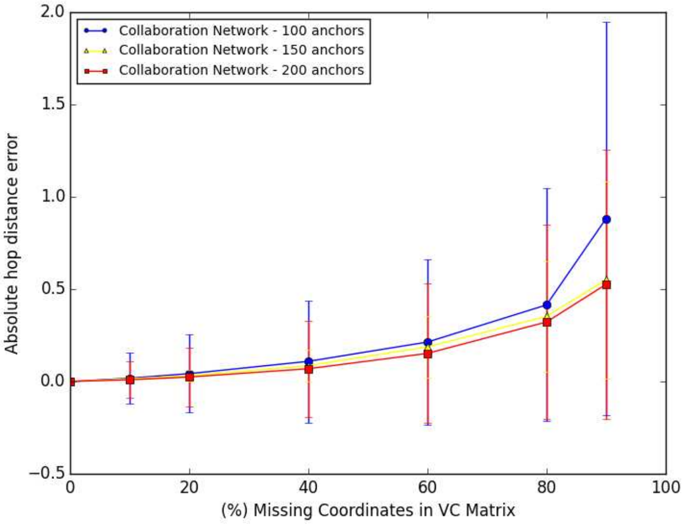


FIGURE 6.9. Absolute error in hop-distance with std. deviation vs percentage of missing coordinates of VC matrix for different anchors - Collaboration Network

The Figure 6.15 shows the mean error with percentage of missing distance matrix entries. The maximum mean error(%) for the three networks is around 6%. Also, the absolute hop-distance error is calculated from 6.2 and the Figure 6.16 shows the absolute hop-distance

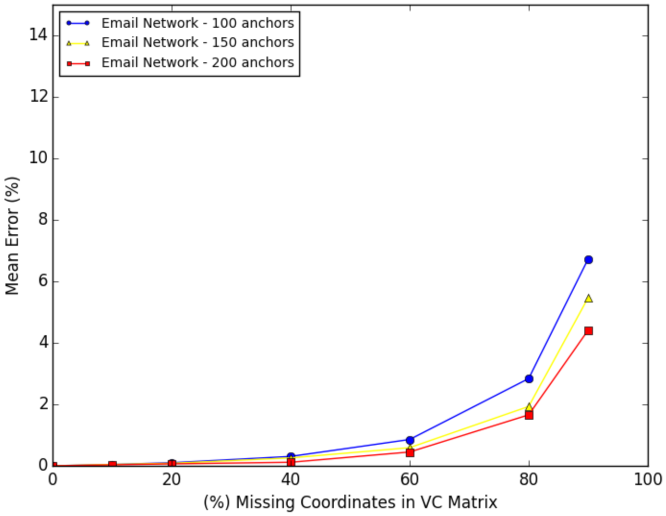


FIGURE 6.10. Mean error vs percentage of missing coordinates of VC matrix for different anchors - Email Network

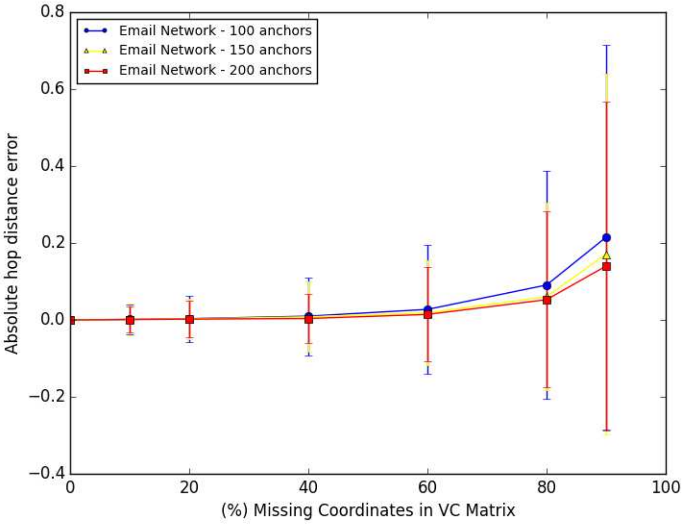


FIGURE 6.11. Absolute error in hop-distance with std. deviation vs percentage of missing coordinates of VC matrix for different anchors - Email Network

error with standard deviation vs percentage of missing coordinates in distance matrix. The entries are predicted within an average hop distance error of less than 0.5.

The Figures 6.17 - 6.19 shows the histogram plot of predicted matrix with different percentage of distance matrix entries missing for Facebook network, Collaboration network

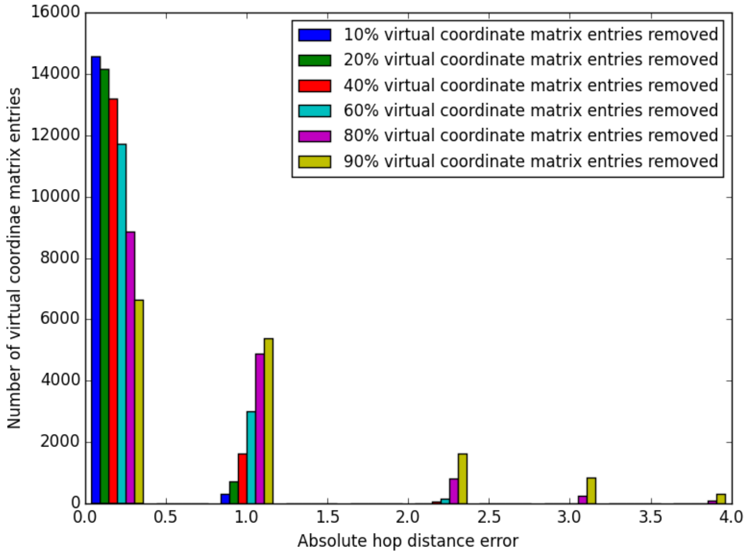


FIGURE 6.12. Histogram of the absolute hop distance error for different missing percentage of virtual coordinate matrix for Facebook network with 20 anchors

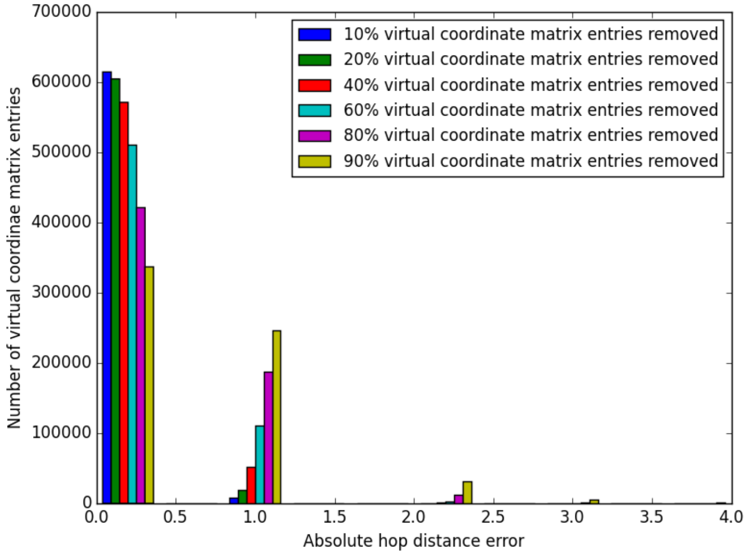


FIGURE 6.13. Histogram of the absolute hop distance error for different missing percentage of virtual coordinate matrix for Collaboration network with 150 anchors

and Enron Email network respectively. It can be seen that with 80% elements missing, most of the entries of predicted distance matrix can be seen to be well within 1 hop distance error for Facebook and Enron Email network and 2 hop distance error for Collaboration network.

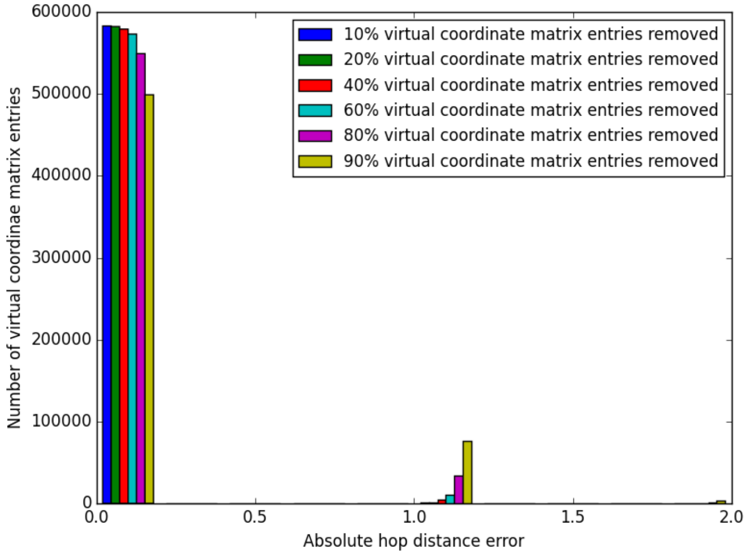


FIGURE 6.14. Histogram of the absolute hop distance error for different missing percentage of virtual coordinate matrix for E-mail network with 150 anchors

The histogram also has entries for higher absolute hop distance error values. The range is limited in the figure. The maximum observed absolute hop distance error is found to be 4 hops for Facebook network when 80% entries are removed. Similarly the maximum absolute hop distance error is found to be 14 hops and 3 hops respectively for Collaboration network and Enron Email network.

There is an important point to note here. Looking at the Table 6.1, it can be inferred that the average path length for the networks are different. The average path length of collaboration network is 6 and almost all the entries are predicted within an error of 2 for 80% elements missing. Similarly for Facebook and Enron E-mail network with average path length of 2.55 and 3.139 respectively, the hop distances are predicted within an error of 1 hop. So we can infer that the error obtained is reasonable considering the average path length and the diameter.

The graph used for evaluation is made of one complete connected component, however the reconstructed graph obtained is not connected. It contains many sub-components and

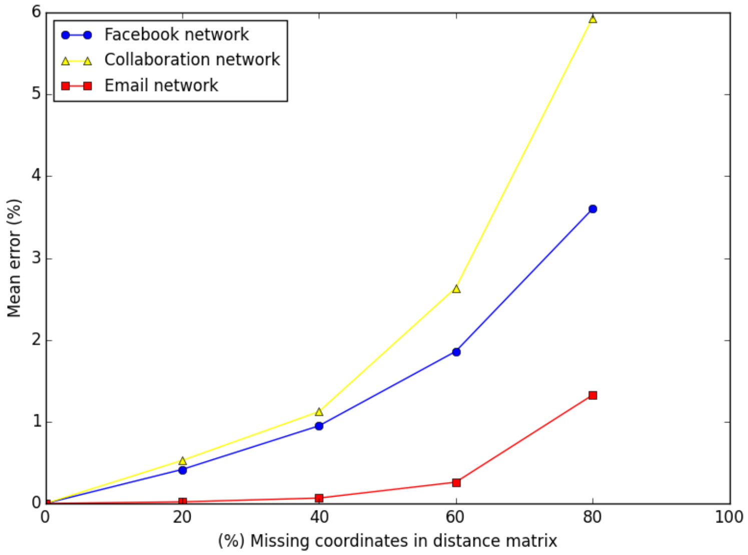


FIGURE 6.15. Mean error vs percentage of missing coordinates of distance matrix

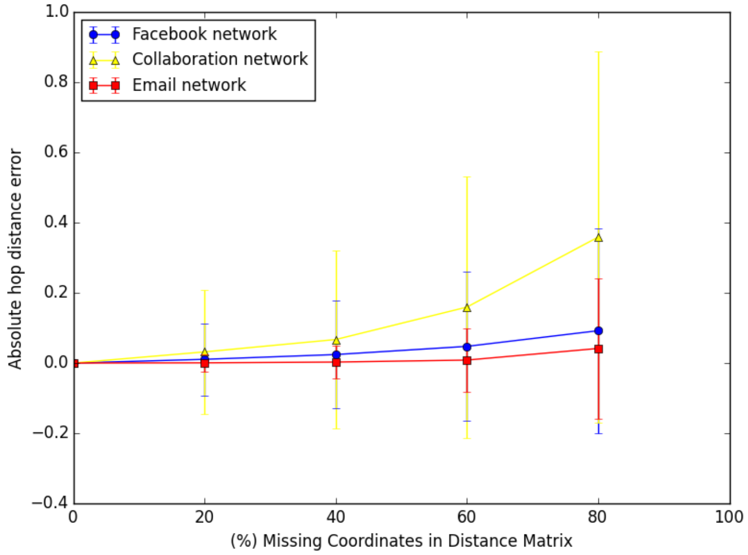


FIGURE 6.16. Absolute error in hop-distance with std. deviation vs percentage of missing coordinates of distance matrix

the sub components tend to increase with increase in the missing entries. The Table 6.2 shows the number of sub-components seen with increasing percentage of missing entries.

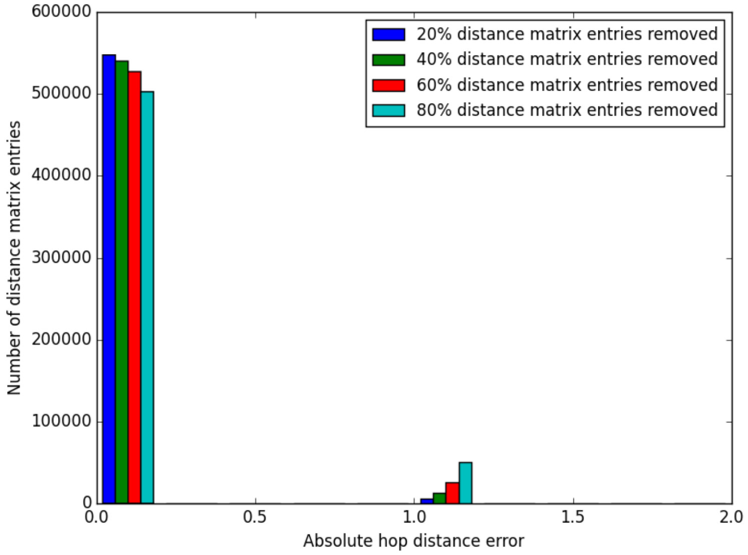


Figure 6.17. Histogram of the absolute hop distance error for different missing percentage of distance matrix for Facebook network

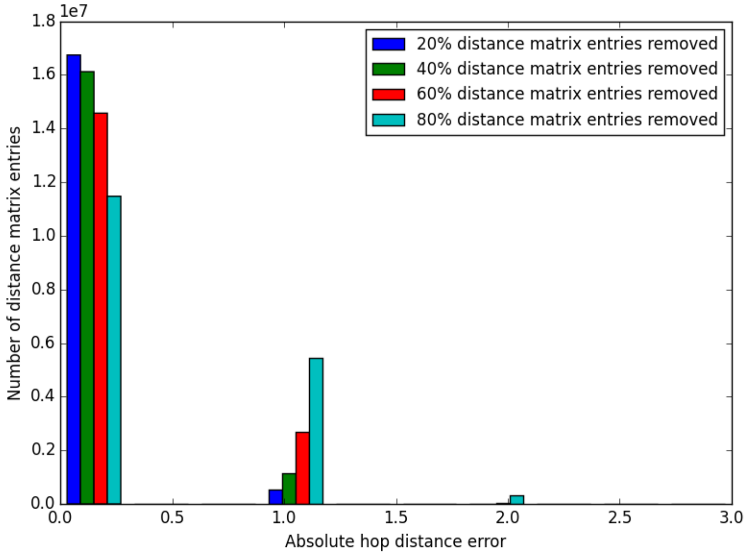


Figure 6.18. Histogram of the absolute hop distance error for different missing percentage of distance matrix for Collaboration network

On comparing of the diameter of the reconstructed matrix, it can be observed that the diameter more or less remains the same with a very negligible increase. The Table 6.3 shows the trend in change of diameter.

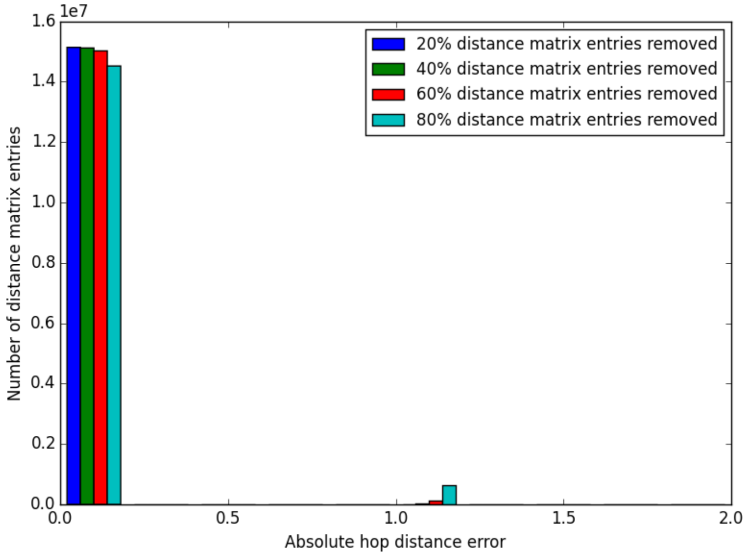


FIGURE 6.19. Histogram of the absolute hop distance error for different missing percentage of distance matrix for E-mail network

TABLE 6.2. Number of sub-component graphs

% Missing entries	Facebook network	Collaboration network	Email network
20	1	110	3
40	4	300	7
60	6	694	9
80	24	1790	36

TABLE 6.3. Diameter of reconstructed graph

% Missing entries	Facebook network	Collaboration network	Email network
20	7	17	5
40	7	19	6
60	7	18	6
80	8	20	7

Also it can be observed that, the avg. path length of reconstructed matrix is almost the same. The Table 6.4 shows the average path length for various percentage of missing entries.

The predicted distance matrix is desired to be symmetric as it is with the full set of distance matrix. The error in symmetricity of the predicted matrix is shown by Figure 6.20. The ratio of sum of absolute error between the lower and upper triangular halves of predicted

TABLE 6.4. Average path length of reconstructed graph

% Missing entries	Facebook network	Collaboration network	Email network
20	2.558	6.048	3.14
40	2.556	6.047	3.1402
60	2.553	6.046	3.1406
80	2.543	6.019	3.13

matrix with respect to the sum of entries of lower triangle of original distance matrix. It can be seen that Facebook network is found to perfectly symmetric, while the other two networks aren't. The reason is when matrix completion is applied, to have a faster convergence, max rank is specified. For Facebook network, the matrix was predicted with full rank while for other two networks, the rank was limited to 150. This is the reason behind not obtaining a perfectly symmetric matrix.

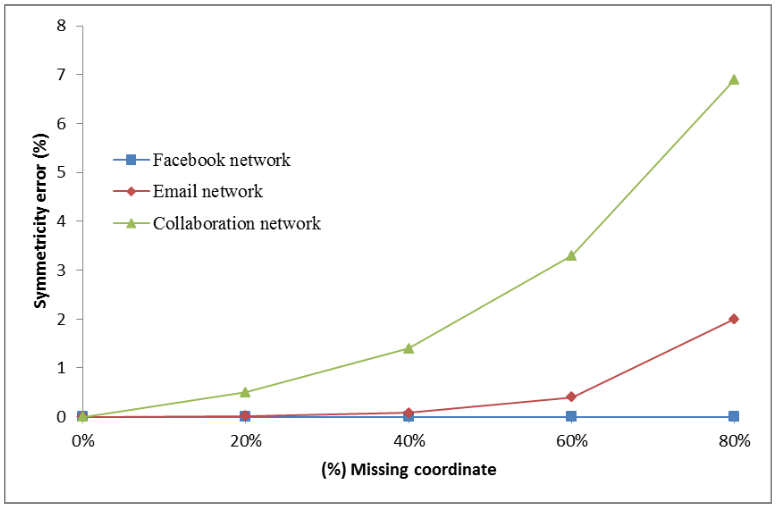


FIGURE 6.20. Error in symmetricity of predicted distance matrix

Also, the bar plot of symmetricity ratio in Figure 6.21 - 6.22 for Collaboration network and Email network shows that 99% of the entries in any half is within 1 hop error. Symmetricity ratio is defined as the number of entries in upper/lower triangles observed within 'k' hop distance to the total number of entries in upper/lower triangle of the matrix.

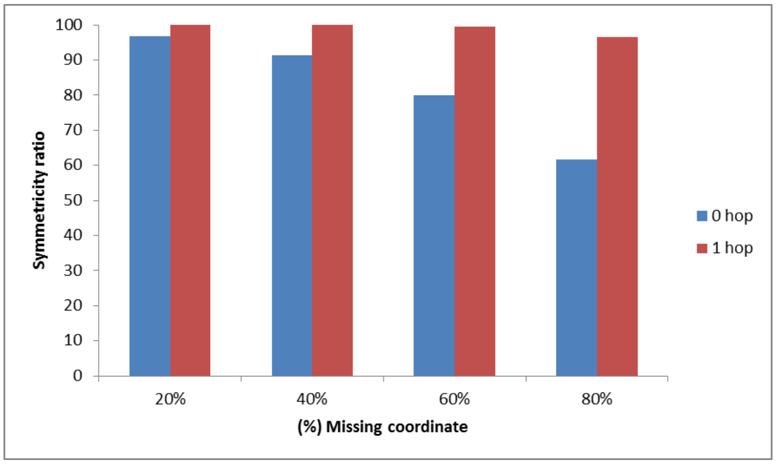


FIGURE 6.21. Symmetricity ratio for Collaboration

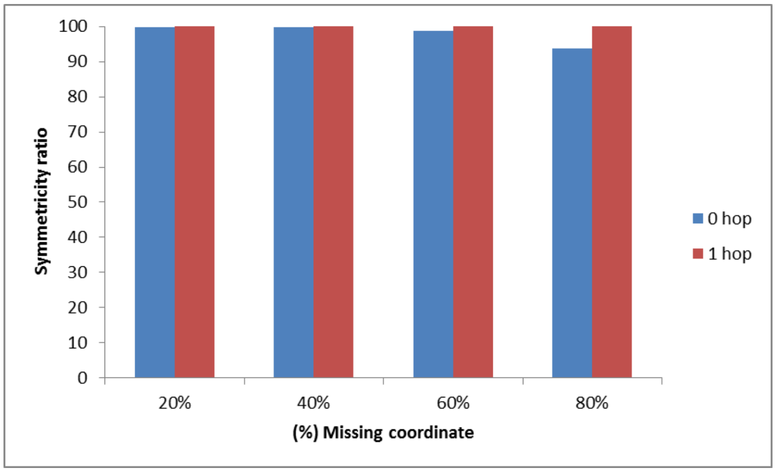


FIGURE 6.22. Symmetricity ratio for Email network

6.4. SUMMARY

A technique to capture social network topology was proposed in this section. The proposed technique is evaluated with two approaches. The first being, capturing the topology with fraction of information available in virtual coordinate matrix. The other uses shortest hop-distances between random pairs of nodes. The results indicate that, for networks of such kind, the social network links can be predicted with very good accuracy. The error seen in matrix prediction is found to be around 6% and all the entries are predicted within 0.5 hop error for a network of size 4200. In summary, with partial information obtained through any of the two measurements, can be used to predict the social network links. Considering the relevance of anchor based representation, it can also be used in social networks for its ability of compression of the data.

CHAPTER 7

Contribution, Conclusion and Future Work

7.1. Contribution

This thesis is an attempt to provide a solution to the problem of capturing and predicting the topology of simple undirected graphs with partial information. Two very important types of graph have been used to demonstrate the technique. They are WSN graphs and social network graphs. The graphs are represented by its shortest hop distance matrix. Two different measurements are considered as fractional information to reconstruct the topology.

The matrix completion algorithm based on extended Robust Principal Component Analysis is used in this research. The principles of matrix completion can be applied to those matrices that are inherently low dimensional and relatively low in rank.

7.1.1. Wireless Sensor Network graphs.

The contribution of this thesis is in two areas. Firstly, The Topology Preserving Map (TPM) for WSN graphs is reconstructed with fractional information about the Virtual Coordinate matrix. Topology Preserving Map is a rotated/distorted version of the original physical map in virtual domain. These TPMs are obtained by PCA of the VC matrix. To prove the efficiency of the technique, four different types of WSN have been used for simulation in our research. Two networks each in 2D and 3D. The 2D WSNs are a circular network with three voids and an odd shaped network. For 3D WSN, a cube with hourglass shaped void network and a hollow T shaped cylinder network has been used. The number of nodes ranges from 500 to 1600. The number of anchors and its placement is important. In this research the number of anchors is limited to less than 5% of nodes. On the anchor placement side, two different anchor placements approach has been followed. The anchors are randomly chosen or anchors are chosen at the boundaries/shape defining locations. To apply the MC for any matrix, the matrix should be relatively low ranked. Using SVD, it has been shown that the VC matrices used for simulation in this research are all low ranked. Various percentages of entries such as 10%, 20%, till 80% are removed from VC matrix. For each of the case, matrix completion is used for predicting the unknown entries. From the predicted matrix, Topology Preserving Map is reconstructed. The error in TPMs introduced by the unknown entries is evaluated with different metrics. The TPMs are plotted using topological coordinates. The topological coordinates are Cartesian equivalents in virtual domain.

The first metric used is mean error. The mean error evaluates the error in the distances between all pairs of nodes with respect to the distance for a TPM with full set of VCs. The distance here is Euclidean measure between the nodes localized using its topological coordinates.

The second metric is neighborhood error. This error has been introduced to find out the change in k neighborhood cardinality for each node. This error is calculated as follows. For each node, k closest neighbors of the TPM obtained with full set of VCs is compared with k closest neighbors of the same node for TPMs obtained with reconstructed VCs. The number of node replacements are summed up and the ratio with respect to the number of k neighbors for TPM with full set of VCs is computed. This ratio gives a measure of node replacements in k neighborhood for each node. Here again, the closest neighbors are considered based on the Euclidean distance between the nodes localized using topological coordinates. Further, the TPMs obtained with double centered VC matrix are compared with TPMs obtained with non-centered approach.

To summarize the results for WSNs, the accuracy in obtaining the TPMs with random anchors and extreme anchors for various percentage of missing entries has been evaluated.

The results are impressive due to the fact that, with only 20% to 40% of entries in VC matrix i.e., 0.8% to 2% of distance matrix, the Topology Preserving Map can be obtained with 35% to 40% mean error for random anchors. The non-centered approach seems to yield better TPMs compared to the centered approach. The TPMs obtained with random anchors is better than the extreme anchors. Since the subset of VC matrix is a very small subset of distance matrix, the technique is very efficient in reconstructing the TPMs with very less information. Thus we conclude that, by this method of measuring the hop-distances from nodes to anchors, the topology can be efficiently reconstructed using the MC algorithm with very small connectivity information. This method further reiterates the compression of the data in the form of a VC matrix.

7.1.2. SOCIAL NETWORK GRAPHS. The second contribution is extending this idea for real world social networks. Three different types of social networks have been used to test the effectiveness of the approach. A Facebook network (744 nodes), An Email communication network of erstwhile Enron Company (3892 nodes), A collaboration network of Arxiv (General relativity and Quantum cosmology category) with 4158 nodes have been used for this. The datasets used for this have been taken from Stanford Network Analysis Project [39]. Many of the real world networks are naturally low ranked, and this is the case for the simulation networks that has been chosen in this research. A method to measure the hop distances in social network from anchor perspective has been suggested. Firstly, the social networks have been represented using VC matrix. The representation of a social network in the form of VC matrix supports the idea of graph compression. Various percentages of entries such as 10%, 20% till 90% are removed from VC matrix. The MC algorithm is used to predict the unknown entries in the matrix. For social networks, obtaining the information in the form of a VC matrix may not be feasible all times, so research widened into the scope of topology prediction with random pairwise distances between the nodes. This gives lot more flexibility in acquiring the measurement from social networks. To demonstrate this approach 20%, 40%, 60% and 80% of entries from complete distance matrix is removed and MC algorithm is applied on it to predict the missing entries.

For both the approaches using a) shortest hop distance from each node to set of anchors and b) shortest hop distances between random pairs of nodes, the accuracy is evaluated using two metrics. The first metric is mean error in matrix prediction. The sum of error between each entries of complete matrix and predicted matrix to the ratio of sum of all the entries of complete matrix gives the mean error in matrix prediction. The second metric gives the absolute hop distance error in matrix prediction. This error is the average of error between the complete matrix and the matrix predicted back with missing entries. The standard deviation is also computed. From the metrics, stunning results can be seen. The results prove the fact that, even with 80% of entries missing from hop distance matrix, the topology can be obtained with significant accuracy. The mean error in topology prediction for network missing 80% entries of distance matrix is only 6%. Also, each of the entries has been predicted with an average error of less than 0.5 hops.

7.2. Summary and Conclusion

The research has come up with a low complexity technique for topology reconstruction with insufficient data. The lack of information about a graph, need for a compressed representation and a good measurements for graphs are the prime motivating factor that drove us towards this research. The problems solved in this research are as follows. The topology of WSN graphs and social network graphs were reconstructed with fractional information about either shortest hop-distances between each node to a set of anchors or shortest hop-distances between random pairs of nodes. Since, shortest hop-distances between each node to anchors is a subset of distance matrix, it can be observed that the measurements made in this form provides means of describing graphs with compressed representation. Also, the research has proved that, availability of measurements in two different forms can be used for topology reconstruction. Thus the research concludes that, with two different forms of measurements, the topology of WSN graphs and social network graphs of this kind can be efficiently reconstructed with high degree of accuracy.

The work presented here not only paves the way for topology reconstruction and link prediction but also has great potential in a wide variety of networks and applications. This technique of using virtual coordinates for graph opens a new method in graph compression. Also the possibility of obtaining the topology of a network with very small amount information is impressive owing to the reason that it takes much lesser computational power and time. Many a times, obtaining information in the form of virtual coordinate is not easy, and this research shows that in such cases, even pairwise distance measurements between random nodes are sufficient to capture the network topology.

7.3. Future Work

The matrix completion can be applied for those matrices that are low ranked. For a dataset represented by a matrix, if there is a lot of redundant information in each of the dimensions, then there is high likelihood that the matrix is relatively low rank. This indicates two important things, there should be equal significance given for both number of anchors and the placement of anchors. So for matrix completion to be more successful on WSNs, the anchors should be chosen in such a way that it contains all the vital information regarding the geometry and local neighborhood of the network. Thus one of the future works involves choosing optimal anchors for WSNs which can reproduce more accurate topology maps. Further, with the topological coordinates obtained with minimal information, the routing ability of the network should be evaluated. A Heuristic solution can be provided to auto correct the topological coordinate to minimize the error in TPMs.

The future application of matrix completion on social networks has immense significance. Instead of traditional machine learning approaches, linear algebraic methods such as matrix completion can also be used for link prediction based problems. One advantage with this method is that, with random information from hop distance matrix, it is now possible to recover the complete connectivity of the network. The current method uses shortest hop distance in all the implementations. The chances that the random samples of hop distance need not be shortest. Thus, this opens up a new area of research. One possible future work is setting up a lower/upper bound on the hop distance for every pair of nodes while applying the matrix completion algorithm and this will lead to even more accurate link prediction results.

Bibliography

[1] A. M. Vladimir Batagelj, Wouter de Nooy, *Exploratory Social Network Analysis with Pajek*. Cambridge University Press, 2011.

[2] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, WSNA '02, (New York, NY, USA), pp. 88–97, ACM, 2002.

[3] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, pp. 254–263, April 2007.

[4] Y. Liu, "Wireless sensor network applications in smart grid: Recent trends and challenges," *International Journal of Distributed Sensor Networks*, vol. 8, Sept. 2012.

[5] J. G.-H. Antonio-Javier Garcia-Sanchez, Felipe Garcia-Sanchez, "Wireless sensor network deployment for integrating video-surveillance and data-monitoring in precision agriculture over distributed crops," *Computers and Electronics in Agriculture*, vol. 75, pp. 288–303, Feb. 2011.

[6] G. Zhao, "Wireless sensor networks for industrial process monitoring and control: A survey," *Network Protocols and Algorithms*, vol. 3, Apr. 2010.

[7] D. C. Dhanapala, *Anchor Centric Virtual Coordinate Systems in Wireless Sensor Networks: from Self-Organization to Network Awareness.* PhD thesis, Colorado State University, Fort Collins, Dec 2012.

[8] A. Jayasumana, R. Paffenroth, and S. Ramasamy, “Topology maps and distance-free localization from partial virtual coordinates for iot networks,” in *Proceedings of the 2016 IEEE International Conference on Communications*, IEEE, 2016.

[9] “Facebook company statistics,.” Available: `http://newsroom.fb.com/company-info/#statistics`.

[10] “Twitter company statistics,.” Available: `https://about.twitter.com/company`.

[11] S. Ressler, “Social network analysis as an approach to combat terrorism: Past, present, and future research,” *Homeland Security Affairs*, vol. 2, July 2006.

[12] A. S. Klovdahl, “Social networks and the spread of infectious diseases: The aids example,” *Social Science and Medicine*, vol. 21, no. 11, pp. 1203–1216, 1985.

[13] D. Dhanapala and A. Jayasumana, “Anchor selection and topology preserving maps in WSNs – 2014; a directional virtual coordinate based approach,” in *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, pp. 571–579, Oct 2011.

[14] D. Dhanapala and A. Jayasumana, “Topology preserving maps from virtual coordinates for wireless sensor networks,” in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pp. 136–143, Oct 2010.

[15] D. F. Nettleton, “Data mining of social networks represented as graphs,” *Computer Science Review*, vol. 7, pp. 1–34, 02/2013 2013.

[16] M. E. J. Newman and J. Park, “Why social networks are different from other types of networks,” *Phys. Rev. E*, vol. 68, p. 036122, Sep 2003.

[17] L. A. N. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley, “Classes of small-world networks,” vol. 97, PNAS, 2000.

[18] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, (New York, NY, USA), pp. 29–42, ACM, 2007.

[19] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *J. Am. Soc. Inf. Sci. Technol.*, vol. 58, pp. 1019–1031, May 2007.

[20] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong, "Analysis of topological characteristics of huge online social networking services," in *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, (New York, NY, USA), pp. 835–844, ACM, 2007.

[21] M. Fire, L. Tenenboim-Chekina, R. Puzis, O. Lesser, L. Rokach, and Y. Elovici, "Computationally efficient link prediction in a variety of social networks," *ACM Trans. Intell. Syst. Technol.*, vol. 5, pp. 10:1–10:25, Jan. 2014.

[22] M. A. Hasan and M. J. Zaki, "A survey of link prediction in social networks," in *Social Network Data Analytics*, pp. 243–275, Springer, 2011.

[23] H. Kashima and N. Abe, "A parameterized probabilistic model of network evolution for supervised link prediction," in *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, (Washington, DC, USA), pp. 340–349, IEEE Computer Society, 2006.

[24] J. M. Kleinberg, "Navigation in a small world," *Nature*, vol. 406, pp. 845–845, Aug. 2000.

[25] M. E. J. Newman, "The structure of scientific collaboration networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, pp. 404–409, January 2001.

[26] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[27] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *SDM06: workshop on link analysis, counter-terrorism and security*, 2006.

[28] C. Wang, V. Satuluri, and S. Parthasarathy, "Local probabilistic models for link prediction," in *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, (Washington, DC, USA), pp. 322–331, IEEE Computer Society, 2007.

[29] J. Kunegis and A. Lommatzsch, "Learning spectral graph transformations for link prediction," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, (New York, NY, USA), pp. 561–568, ACM, 2009.

[30] L. Wu, X. Ying, and X. Wu, "Reconstruction from randomized graph via low rank approximation.," in *SDM*, pp. 60–71, SIAM, 2010.

[31] P.-A. Savalle, E. Richard, and N. Vayatis, "Estimation of simultaneously sparse and low rank matrices.," in *ICML*, icml.cc / Omnipress, 2012.

[32] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, pp. 11:1–11:37, June 2011.

[33] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[34] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, p. 717, 2009.

[35] R. C. Paffenroth, P. D. Toit, R. Nong, L. L. Scharf, A. P. Jayasumana, V. Bandara, P. C. Du Toit, and V. Banadara, "Space-time signal processing for distributed pattern

detection in sensor networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 38–49, 2013.

[36] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," Tech. Rep. ENG-09-2215, UIUC, Nov. 2009.

[37] R. Paffenroth, R. Nong, and P. Du Toit, "On covariance structure in noisy, big data," *SPIE Optical Engineering+ Applications*, pp. 88570E–88570E, 2013.

[38] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction.* Springer Publishing Company, Incorporated, 1st ed., 2007.

[39] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection." `http://snap.stanford.edu/data`, June 2014.

[40] S. Catanese, P. D. Meo, E. Ferrara, G. Fiumara, and A. Provetti, "Crawling facebook for social network analysis purposes," *CoRR*, vol. abs/1105.6307, 2011.

[41] C.-I. Wong, K.-Y. Wong, K.-W. Ng, W. Fan, and K.-H. Yeung, "Design of a crawler for online social networks analysis," in *WSEAS TRANSACTIONS on COMMUNICATIONS*, 2014.

[42] L. Backstrom and J. Leskovec, "Supervised random walks: Predicting and recommending links in social networks," in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, (New York, NY, USA), pp. 635–644, ACM, 2011.

[43] J. Travers, S. Milgram, J. Travers, and S. Milgram, "An experimental study of the small world problem," *Sociometry*, vol. 32, pp. 425–443, 1969.

APPENDIX A

Appendices

Simulations for this thesis are done in Python 2.7. Packages used are *networkx Scipy Numpy Matplotlib*. First the constant variables that are used in the code are presented:

A.1. T coordinate network generation

```
import pandas as pa
import numpy as np
import scipy.spatial as sp
import scipy.misc as spm
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
def cylinder():
    centre1 = np.array((10,10,10))
    a = np.reshape(centre1,(1,3))
    rad = 2;
    angle = np.zeros(36)
    for i in range(0,36):
            angle[i] = 10*i

    #print angle
    angle = np.deg2rad(angle)
    #angle = np.array
       ((0,30,60,90,120,150,180,210,240,270,300,330))

    #different angle vector that cuts open a piece in the
       centre of cylinder
    va = 23
    new_angle = np.zeros(27)

    for i in range(0,27):
        if i<= 13:
            new_angle[i] = 10*i
        if i >= 14:
            new_angle[i] = 10*va
            va = va +1
```

```
#print new_angle
new_angle = np.deg2rad(new_angle)
new_x = np.empty(27)
new_x.fill(10)
new_x=np.reshape(new_x,(27,1))

new_cos = np.multiply(rad,np.cos(new_angle))
new_sin = np.multiply(rad,np.sin(new_angle))
new_y = np.reshape(np.add(10,new_cos),(27,1))
new_z = np.reshape(np.add(10,new_sin),(27,1))
#The code to create varying angle to cut open a piece of
   cylinder is done
#the usage is done inside the loop that extends a cylinder

#The piece of code that is used to add extra points while
   cutting the cylinder1 open
va = 21
new_angle1 = np.zeros(31)

for i in range(0,31):
    if i<= 15:
        new_angle1[i] = 10*i
    if i >= 16:
        new_angle1[i] = 10*va
        va = va +1

new_angle1 = np.deg2rad(new_angle1)
new_x1 = np.empty(31)
new_x1.fill(10)
new_x1=np.reshape(new_x1,(31,1))

new_cos1 = np.multiply(rad,np.cos(new_angle1))
new_sin1 = np.multiply(rad,np.sin(new_angle1))

new_y1 = np.reshape(np.add(10,new_cos1),(31,1))
new_z1 = np.reshape(np.add(10,new_sin1),(31,1))

#the code ends here

fig = plt.figure()
ax = Axes3D(fig)
```

```
# This piece is to create the first circle with 10,10,10 as
    the centre

cos = np.multiply(rad,np.cos(angle))
sin = np.multiply(rad,np.sin(angle))
y1 = np.reshape(np.add(10,cos),(36,1))
z1 = np.reshape(np.add(10,sin),(36,1))
x1 = np.empty(36)
x1.fill(10)
x1 = np.reshape(x1,(36,1))
res1 = np.concatenate((x1,y1,z1),axis=1)
#print res1.shape
ax.scatter(x1,y1,z1)

# this piece extends the circle to a cylinder
i = 0.5
var = 1
while var == 1:
    #the below conditional part is to cut open a piece in
       the cylinder
    if i==3 or i==7:
        change_x = np.add(i,new_x1)
        ax.scatter(change_x,new_y1,new_z1)
        temp = np.concatenate((change_x,new_y1,new_z1),axis
           =1)
        res1 = np.concatenate((res1,temp))
        i = i + 0.5
        continue
    if i>3 and i<7:
        change_x = np.add(i,new_x)
        ax.scatter(change_x,new_y,new_z)
        temp = np.concatenate((change_x,new_y,new_z),axis
           =1)
        res1 = np.concatenate((res1,temp))
        i = i + 0.5
        continue
    change_x = np.add(i,x1)
    ax.scatter(change_x,y1,z1)
    temp = np.concatenate((change_x,y1,z1),axis=1)
    res1 = np.concatenate((res1,temp))
    if(i == 10):
        break
    i = i + 0.5
```

```
print res1.shape

#The below code is to add small piece add the end to finish the T joint
#adding one extra circle to the cylinder 2 with varying degrees

va = 4
vb = 22
new_angle = np.zeros(22)

for i in range(0,22):
    if i<= 10:
        new_angle[i] = 10*va
        va = va+1
    if i >= 11:
        new_angle[i] = 10*vb
        vb = vb +1


print new_angle
new_angle = np.deg2rad(new_angle)
new_x = np.empty(22)
new_x.fill(8.5)
new_x=np.reshape(new_x,(22,1))

new_cos = np.multiply(rad,np.cos(new_angle))
new_sin = np.multiply(rad,np.sin(new_angle))
new_y = np.reshape(np.add(15,new_cos),(22,1))
new_z = np.reshape(np.add(10,new_sin),(22,1))

ax.scatter(new_y,new_x,new_z)
temp = np.concatenate((new_y,new_x,new_z),axis=1)
res1 = np.concatenate((res1,temp))

#the one circles code ends here

x2 = np.reshape(np.add(15,cos),(36,1))
z2 = np.reshape(np.add(10,sin),(36,1))
y2 = np.empty(36)
y2.fill(8)
y2 = np.reshape(y2,(36,1))
```

```
    ax.scatter(x2,y2,z2)
    temp = np.concatenate((x2,y2,z2),axis=1)
    res1 = np.concatenate((res1,temp))

    change_y = y2
    print change_y.shape
    i = 7.5
    var = 1
    while var == 1:
        change_y.fill(i)
        ax.scatter(x2,change_y,z2)
        temp = np.concatenate((x2,change_y,z2),axis=1)
        res1 = np.concatenate((res1,temp))
        if(i == 1):
            break
        i = i - 0.5

    print res1.shape
    np.save(’T.npy’,res1)
    df = pa.DataFrame(res1)
    plt.show()
    #circle = np.zeros((12,3))
```

A.2. Removing entries from virtual coordinate matrix

```
def rand_individ(VC,C):
    global row_size
    global col_size
    R=np.arange(row_size)
    C_dict = {}
    C_dict_key = {}
    C_dict_nor = {}
    R_dict = {}
    R_dict_key = {}
    VC_dict = {}
    total_tuples = {}
    global n
    #VC_dict_rem={}
    for i in range(0,col_size):
        C_dict[str(C[i])]=i
        C_dict_key[i]=C[i]
        C_dict_nor[C[i]]=i
    for i in range(0,n):
        R_dict[str(R[i])]=i
        R_dict_key[i]=R[i]
    for i in range(0,n):
        for j in range(0,col_size):
            t =str(i)+’,’+str(j)
            total_tuples[(i,j)] = 1
            VC_dict[str(t)]=VC[i][j].astype(int)
            #VC_dict_rem[(i,j)]=VC[i][j].astype(int)
    VC_dict_rem={}
    VC_dict_rem = VC_dict
    print len(VC_dict_rem)
    global fraction
    n_rem=np.rint(fraction*np.prod(VC.shape)).astype(int)
    rem_tuples = random.sample(total_tuples,n_rem)
    new_rem_tuples = list(rem_tuples)
    for c in range(len(rem_tuples)):
        i = rem_tuples[c]
        if i[0] in C_dict_nor.keys():
            #t1 = str(C_dict_key.get(i[1]))+’,’+str(C_dict_nor.
               get(i[0]))
            count1+=1
            t2 = C_dict_key.get(i[1]),C_dict_nor.get(i[0])
            new_rem_tuples.append(t2)

    new_rem_tuples = list(set(new_rem_tuples))
    for c in new_rem_tuples:
        t = str(c[0])+’,’+str(c[1])
        del VC_dict_rem[str(t)]
    print_element(R_dict,C_dict,VC_dict_rem)
```

A.3. Removing entries from distance matrix

```
def remove_distance(A):
    print "hello"
    global row_size
    global col_size
    R=np.arange(row_size)
    C=np.arange(col_size)
    #print R.shape
    C_dict = {}
```

```
C_dict_key = {}
C_dict_nor = {}
R_dict = {}
R_dict_key = {}
mat_dict = {}
total_tuples = {}
global n

for i in range(0,col_size):
    C_dict[str(C[i])]=i
    C_dict_key[i]=C[i]
    C_dict_nor[C[i]]=i
for i in range(0,n):
    R_dict[str(R[i])]=i
    R_dict_key[i]=R[i]
for i in range(0,n):
    for j in range(0,col_size):
        #print (R[i],C[j])
        t =str(i)+’,’+str(j)
        total_tuples[(i,j)] = 1
        mat_dict[str(t)]=A[i][j].astype(int)
        #VC_dict_rem[(i,j)]=VC[i][j].astype(int)
mat_dict_rem={}
mat_dict_rem = mat_dict
print len(mat_dict_rem)
global fraction
n_rem=np.rint(fraction*np.prod(A.shape)).astype(int)
rem_tuples = random.sample(total_tuples,n_rem)
new_rem_tuples = set(rem_tuples)
print len(new_rem_tuples)
for c in range(len(rem_tuples)):
    i = rem_tuples[c]
    t = (i[1],i[0])
    new_rem_tuples.add(t)
print len(new_rem_tuples)
count = 0
tot_set = set(list(total_tuples.keys()))
remain_set = tot_set - new_rem_tuples
remain_set = list(remain_set)
new_rem_tuples = list(new_rem_tuples)
f = open(’remain.json’,’w’)
json.dump(remain_set,f)
f.close()
```

```
    f = open('deleted.json','w')
    json.dump(new_rem_tuples,f)
    f.close()
    print float(len(new_rem_tuples))/float(len(total_tuples))
    for c in new_rem_tuples:
        t = str(c[0])+','+str(c[1])
        count +=1
        del mat_dict_rem[str(t)]
    print "count deleted is",count
    print_element(R_dict,C_dict,mat_dict_rem)
```

A.4. Neighborhood Error

```
count = np.zeros((6),dtype=np.float)
denom = np.zeros((6),dtype=np.float)
def __2d(v1,v2):
    global count
    global denom
    #coors = np.load(v3)
    var1 = np.array(pa.read_excel(v1,sheetname='Sheet1',sep='\t
       ',header=None))
    var2 = np.array(pa.read_excel(v1,sheetname='Sheet2',sep='\t
       ',header=None))
    var3 = np.array(pa.read_excel(v1,sheetname='Sheet3',sep='\t
       ',header=None))
    var4 = np.array(pa.read_excel(v1,sheetname='Sheet4',sep='\t
       ',header=None))
    var5 = np.array(pa.read_excel(v1,sheetname='Sheet5',sep='\t
       ',header=None))
    var6 = np.array(pa.read_excel(v1,sheetname='Sheet6',sep='\t
       ',header=None))
    li = []
    li.append(var1)
    li.append(var2)
    li.append(var3)
    li.append(var4)
    li.append(var5)
    li.append(var6)
    la = []
    print len(var1)
    print len(li)
    for i in range(len(li)):
        l = spa.cdist(li[i],li[i],'euclidean')
        la.append(l)

    for i in range(len(var1)):
        __dist(la,i,v2)
    a = np.divide(count,denom)
    print np.multiply(a,100)

def __dist(la,no,v2):
    global count
    global denom
    for i in range(0,len(la)):
        obj = la[i]
        obj0 = la[0]
        one = np.argsort(obj0[no,:])
        two = np.argsort(obj[no,:])
        a = one[1:1+int(v2)]
        b = two[1:1+int(v2)]
        s0 = set(a)
        s1 = set(b)
        count[i] += float(len(s0-s1))
        denom[i] += float(len(s0))
```

A.5. Extreme Node Search Algorithm

```
def ens_anchor():
    var = np.load('coordinates.npy')

    D = spa.pdist(var,'euclidean')
    sqr=spa.squareform(D)
    r=1

    adj=spa.squareform(D<=r)
    row_size = len(var)
    sps = sp.sparse.coo_matrix(adj)
    #print sps
    A = np.array(dijkstra(sps))

    C = np.array(random.sample(range(0,row_size),2))
    VC = A[:,C]
    DVCS = VC
    dv = np.square(DVCS)
```

```
dis = A[C[0],C[1]]
dis = 2*dis

dv = np.subtract(dv[:,0],dv[:,1])

dv = np.divide(dv,dis)
count = 0
anchor = []
anchor_range = int(1)

for i in range(0,len(dv)):
    ind = np.where(A[i,:]==anchor_range)
    arr = np.take(dv,ind[0])
    if dv[i] < np.amin(arr) or dv[i] > np.amax(arr):
        anchor.append(i)
print anchor
print C
n_C = list(np.append(anchor,C))
print n_C
```

A.6. List of Abbreviations

ENS: Extreme Node Search

GC: Geographical Coordinates

GCS: Geographical Coordinate System

GPS: Global Positioning System

TC: Topological Coordinates

TPM: Topology Preserving Maps

VC: Virtual Coordinates

VCS: Virtual Coordinate System

WSN: Wireless Sensor Networks

MC: Matrix Completion

SVD: Singular Value Decomposition

PCA: Principal Component Analysis

RPCA: Robust Principal Component Analysis

eRPCA: extended Robust Principal Component Analysis

EVD: Eigenvalue decomposition