Reconstruction of Power Distribution Network

A Project Report

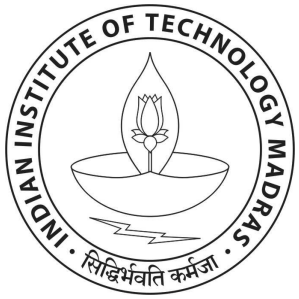
submitted by

AREJEET NAG

in partial fulfilment of the requirements

for the award of the degree of

MASTER OF TECHNOLOGY



DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2015

THESIS CERTIFICATE

This is to certify that the thesis titled **Reconstruction of Power Distribution Network**, submitted by **Arejeet Nag**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Ramkrishna Pasumarthy
Project Advisor
Associate Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 26th May 2015

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor Dr. Ramakrishna Pasumarthy and Dr. Nirav Bhatt for providing me with an opportunity to work as a research student under their guidance and giving me this enriching learning experience. Their invalueable guidance, constant encouragement and constructive criticism helped me over the course of my project. I would also like to thank Mr. Sai Krishna who is an epitome of hardwork and who has inspired me to push my limits.

I am also thankful to the entire faculty of Department of Electrical Engineering for constantly helping me develop interest and learn key concepts which I used in this project.

I would also like to thank my lab mates Mr. Manish Heda, Ms. Niharika Challapalli and Mr. Shubham Agarwal for their constant encouragement. I am also thankful to my friends at IIT Madras for making my stay as comfortable as possible.

Finally, I express my sincere thanks to my parents and sisters for their constant support and encouragement.

ABSTRACT

Energy distribution companies around the world are shifting from manual meters to smart meters which will give them continuous updates about the customer's electricity usage habits and the behavior of electrical energy in the network. The collected meter measurements from all the components may be used to identify the *Connectivity Model* of the distribution network up to a certain accuracy, which is an adjacency matrix showing the child-parent relationship between the various meters. When the connectivity model cannot be reconstructed from the meter data, we try to get a matrix which has the same row space as the connectivity model.

This thesis presents the various methods used to infer the connectivity model of power distribution network. Simulations have been performed on synthetic data as well as semi-synthetic data to understand the various methods and to compare the accuracy and efficiency of these methods with respect to scaling for large networks. Apart from traditional deterministic methods, heuristic approach like Randomized algorithms is explored. The results are also tested in a Hadoop based system to understand scalability.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

SVD	Singular Value Decomposition
PCA	Principal Component Analysis
SS	Substation
DT	Distribution Transformer
CM	Consumer Meter
LAPACK	Linear Algebra PACKage

CHAPTER 1

INTRODUCTION

This chapter describes the various components of a power distribution system which are considered in this thesis, how these are interconnected, and the connectivity model that we aim to infer from smart meter measurements. The need for connectivity model is also described.

1.1 Power Distribution Network

Majority of Electrical Energy is generated in substations which are miles away from the consumers. The transmission network is responsible for bringing the energy closer to the consumers. The distribution system is the final stage in delivery of electrical energy and is responsible for transferring power from the transmission system to individual customers.

The *distribution substation* is the first component in the distribution network. It transfers power from the transmission network to the distribution network. Its input voltage is typically of the order of 100kV. A substation typically serves thousands of customers. One or more 3-phase *feeders* are connected to each substation. The feeders typically have voltage between 4kV to 33kV depending on the size, density and power requirements in the region. *Distribution transformers* (DTs) are connected to the feeders and transfer power to 1-10 customers. Depending on the location, transformers can be single phase (Northern America) or 3 phase (Australia). This thesis deals with a north American type distribution network in which the DTs tap to one line of the 3

phase feeders and hence are single phase. Also, the 3 phases of a feeder are considered independent. That is, a particular feeder reading is the sum of all its three phases. In its operational state, the distribution network forms a tree structure with the customers as the leaf nodes. Fig. 1.1 shows the radial structure of the distribution network. SS is the distribution substation, F-1 and F-2 are the two 3-phase feeders and DT-1 and DT-2 are the single phase transformers. The consumer meters (C1- Cn) are the leaf nodes.

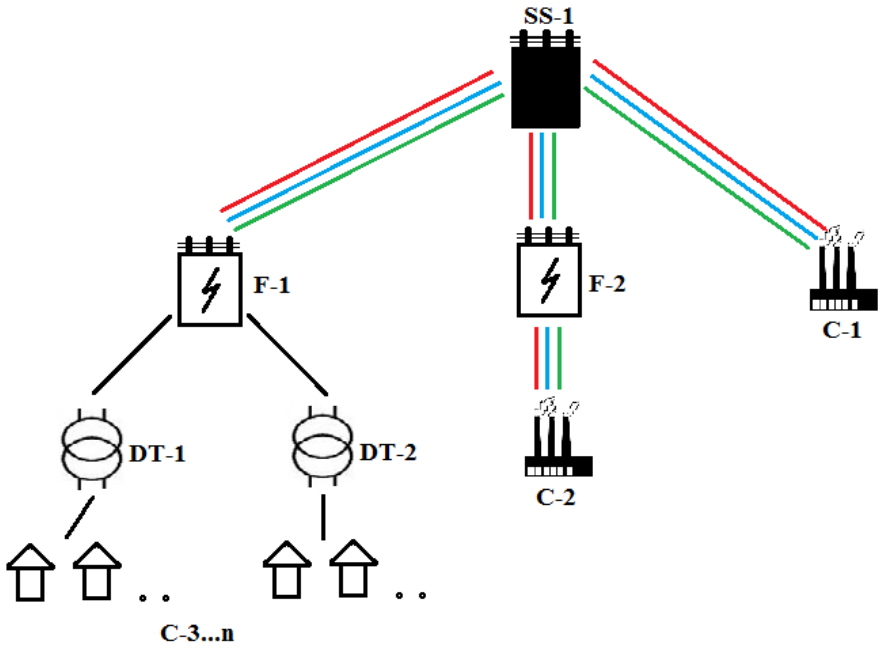


Figure 1.1: Radial distribution network.

1.2 The Connectivity Model

The interconnection between various elements in a network is given by its connectivity mode (Arya *et al.*, 2012)l. As already stated above, the three primary elements in the distribution system are the feeder, the DT and the customers. The connectivity model

should represent the parent-child relationship between the various components in the network. Consider the network shown in Fig. 1.2 (a). The network consists of a feeder (F-1), two distribution transformers (DT-1 and DT-2) and three consumers. Fig. 1.2 (b) is the tree structured connectivity model which we aim to achieve.

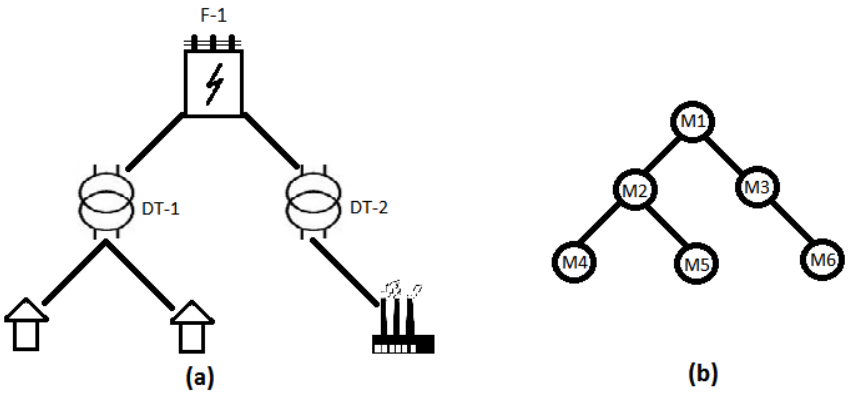


Figure 1.2: A sample distribution network (a) and its connectivity model (b)

The connectivity model is generally denoted in the form of an adjacency matrix or a *constraint matrix*. For the network shown in Fig. 1.2 (a), if Mi is the meter reading of the i^{th} meter, then by conservation of energy, the following relations hold incase of a lossless error-free network:

$$M1 = M2 + M3 \tag{1.1}$$

$$M2 = M4 + M5 \tag{1.2}$$

$$M3 = M6 \tag{1.3}$$

A below shows these relationships in the matrix form.

$$A = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

We seek to infer the connectivity model using the smart meter readings from all the elements in the distribution network.

1.3 Need for Connectivity Model

- Power companies use Automated Distribution Management systems to schedule operation and maintenance of distribution network. These systems need the connectivity model as one of the input.
- Connectivity models are required for performing accurate power-flow calculations which in turn are important for efficient delivery of electrical power.
- In case of catastrophes, connectivity model can be used to identify consumers affected by grid failure and this will lead to efficient allocation of manpower.
- Connectivity models can be used for detecting thefts in distribution network. These models are also used for energy auditing.

1.4 Contribution

As described above, the connectivity model is an important piece of information for efficient and theft proof functioning of the power distribution network. Moreover the connectivity model might change because of natural calamities or repairs. Existing methods depend on signal injection techniques which rely on power line communication require enhanced hardware (Caird, 2010). These solutions are impractical and expensive

since they require dedicated hardware. Alternately, companies send their representatives to manually map parts of the network, which is again wastage of manpower. This thesis discusses alternate solutions to the problem of inferring the connectivity model. Even if we aren't able to extract the exact model and instead get an estimate of the rows of the constraint matrix, we can use similar techniques to perform de-noising and fault analysis.

CHAPTER 2

PRE-REQUISITES

This chapter provides a short tutorial on the various mathematical preliminaries required to understand this work.

2.1 Singular Value Decomposition

The Singular Value Decomposition (SVD) is a matrix factorization technique ubiquitous in machine learning, signal processing, statistics etc. The SVD of an $m \times n$ matrix A is a factorization of the form A = U $\sum$ V$^\intercal$, where U is an $m \times m$ orthogonal matrix of left singular vectors, $\sum$ is an $m \times n$ diagonal matrix of singular values, and V$^\intercal$ is an $n \times n$ orthogonal matrix of right singular vectors. SVD can be understood as a method of transforming correlated variables into uncorrelated variables or a method to find out the direction in which most of the variance of the data is concentrated and hence reduce the dimensionality of the data by concentrating only on high variance directions (Baker, 2005). As an illustration, consider the matrix

$$A = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

On performing the SVD of A, we obtain three matrices, $U, S, V^\intercal$. The matrix S contains the singular values of A. Since we have only three independent variables, there are only 3 non zero singular values. We can verify that the product of U, S and $V^\intercal$ gives us A

again. In our experiments, we have used SVD algorithm provided by Linear Algebra PACKage (LAPACK).

$$U = \begin{bmatrix} -0.7370 & 0.3280 & -0.5910 \\ 0.5910 & 0.7370 & -0.3280 \\ 0.3280 & -0.5910 & -0.7370 \end{bmatrix}$$

$$S = \begin{bmatrix} 2.0608 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.5984 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0946 & 0 & 0 & 0 \end{bmatrix}$$

$$V^{\intercal} = \begin{bmatrix} 0.3576 & -0.2052 & 0.5400 & 0.4022 & 0.4022 & 0.4636 \\ -0.6444 & -0.2559 & -0.2403 & 0.4686 & 0.4686 & -0.1495 \\ -0.5168 & 0.5749 & 0.1334 & -0.0664 & -0.0664 & 0.6130 \\ 0.2868 & 0.4611 & -0.2997 & 0.7343 & -0.2657 & -0.0747 \\ 0.2868 & 0.4611 & -0.2997 & -0.2657 & 0.7343 & -0.0747 \\ 0.1592 & -0.3697 & -0.6733 & -0.0664 & -0.0664 & 0.6130 \end{bmatrix}$$

2.2 Principal Component Analysis

Principal Component Analysis is a technique which involves a linear transformation to reduce the dimensionality of dataset containing possibly correlated variables into set of values of linearly uncorrelated variables, while retaining as much variance as possible. Thus, PCA gives a basis which is a linear combination of the existing basis and which best re-expresses the data.

2.2.1 Solving PCA using eigenvector decomposition (Shlens, 2014)

Let X be our data set which is an $m \times n$ matrix, where m is the number of features and n is the number of samples. We aim to find out directions which leads to uncorrelated data and transform our data to those directions. Hence, our transformed data should have a diagonal covariance matrix. Our goal is to find an orthonormal matrix Q such that the covariance matrix of $Y = QX$ is a diagonal matrix. That is,

$$C_Y = \frac{1}{n} Y Y^\intercal \tag{2.1}$$

is a diagonal matrix. In this case, the rows of Q will be called the *principal components*.

$$C_Y = \frac{1}{n} Q X X^\intercal Q^\intercal \tag{2.2}$$

$$C_Y = Q C_X Q^\intercal \tag{2.3}$$

Any symmetric matrix A can be written as $A = EDE^\intercal$, where D is a diagonal matrix and E is a matrix of eigenvectors of A. If we select the matrix Q as an eigenvector matrix of S_X, then, $Q \equiv E^\intercal$ and $Q^{-1} = Q^\intercal$. Substituting the decomposed value of C_X as described above, we have,

$$C_Y = Q(Q^\intercal D Q) Q^\intercal \tag{2.4}$$

$$C_Y = (QQ^\intercal) D (QQ^\intercal) \tag{2.5}$$

$$C_Y = D \tag{2.6}$$

Since this choice of Q transforms the original data into uncorrelated data, Q is the linear transformation we were looking for originally. Summarizing the results of PCA on a matrix X as

1. The principal component of X are the eigenvectors of $C_X = \frac{1}{n} X X^\intercal$
2. The i^{th} diagonal value of C_Y is the variance of X along q_i, where q_i is the i^{th} row of Q.

2.3 Randomized Method to find SVD of a matrix

Randomized algorithms for finding out SVD of a matrix A proceed in two major steps (Halko *et al.*, 2009).

- Compute an approximation of the range of A using randomized techniques. That is, find a matrix P with $r(< m, n)$ orthonormal columns and A $\approx$ P P$^\intercal$ A.
- Construct a matrix B = P$^\intercal$ A having smaller number of columns than A. Now, compute the SVD of B by standard methods, and thus B = S $\sum$ V$^\intercal$.
  Since A $\approx$ P P$^\intercal$ A = P (S $\sum$ V$^\intercal$), we have computed an approximate SVD of A $\approx$ U $\sum$ V$^\intercal$ where U = P S.

Estimating the range of matrix A is the tricky step in this algorithm. We take a collection random vectors ω1,ω2 ... and examine the subspace formed by the action of A on each of these random vectors. If we form an $n \times l$ Gaussian random matrix Ω, compute Y = A Ω, and take the QR decomposition of Y, Q R = Y, then Q is an $m \times l$ matrix whose columns are an orthonormal basis for the range of Y.

2.3.1 Randomized SVD Algorithm

- Draw an $n \times k$ Gaussian random matrix Ω.
- Form the $m \times k$ sample matrix Y = A Ω.

- Form an $m \times k$ orthonormal matrix P such that Y = P R.
- Form the $k \times n$ matrix B = $\overline{Q}$ A
- Compute SVD of the smaller matrix B using existing packages like LAPACK's SVD: B = $\hat{U} \sum \overline{V}$
- Form the matrix U = P $\hat{U}$
  where, $\overline{X}$ represents the complex conjugate of a matrix X.

(Halko *et al.*, 2009) have proved strong bounds on the accuracy of randomized SVD when the singular values decays rapidly, which is the case in our application.

CHAPTER 3

PROBLEM FORMULATION AND SOLUTION

Our aim is to obtain the tree connectivity model of distribution network using the meter readings. Since the meter readings will have some measurement error because of sensor noise, we first consider the error free case and then apply our solution for different noise levels. This thesis presents two different approaches for doing the same. One is formulating a problem similar to subset sum problem and other is identifying the model using SVD or PCA. The second approach is studied in detail and experiments have been carried on synthetic as well as semi synthetic data using the same. In this study, we also try to identify the number of independent meter readings (consumer meter readings) using the meter data alone.

3.1 Modified Subset-sum problem

Consider ideal meters and an ideal network in which there are no losses. Applying conservation of energy at any DT, its meter reading will be the sum of the consumer meters. Similarly, the meter reading of any feeder will be the sum of all the independent components under it.

In the *subset-sum problem*, we are given a finite set S $\subset$ N and a target t $\in$ N. The problem asks whether there exists a subset S' $\in$ S such that sum of the S' adds up to t. The subset-sum problem is NP-complete (Cormen *et al.*, 2001). (Arya *et al.*, 2013) have defined a modified version of this problem which factors in the decimal values of the meter readings called the *Leaf connectivity(LC)* problem. Leaf connectivity problem

takes as input a time series of load measurements from a set of leaf meters *l* and a non-leaf meter *s* and determines the subset $l' \in l$ of leaf meters present in the subtree rooted at s. This is represented as $l' = LC(s, l)$.

Let *n* be the total number of leaf meters and *m* be the time for which the measurements are taken. Let s_k and L_{kj} denote the loads measured by the non-leaf meter s and leaf meter j in the interval k respectively, $1 \leq$ k $\leq$ m, $1 \leq$ j $\leq$ n. Now let $A = [L_{kj}]_{m \times n}$denote the matrix of all leaf meter measurements and b = $[s_j]_{m \times 1}$ denote the vector of source meter measurements. Each row of A corresponds to one time series measurement and each column of A corresponds to load measurements from one meter over multiple time intervals. Our aim is to identify the subset of leaf meters $l' \in l$ present in the subtree rooted at s. We define an indicator variable $x(j)$ such that $x(j) = 1$ if leaf is in the subtree rooted at s, otherwise $x(j) = 0$.

Let $X = [x_j]_{n \times 1}$. Using conservation of energy

$$b = AX + e \tag{3.1}$$

where $e = [\epsilon_k]_{m \times 1}$ is the error in the meter readings. The leaf connectivity problem is to determine the unknown binary vector $X \in \{0, 1\}^n$ given A, b, and unknown e. The $LC(s, l)$ problem is called with each DT and Feeder meter reading as *s* and all the meter readings *l*. This will give us the set of leaf meters under the given DT or feeder. The matrix A, also known as the data matrix, is a collection of time series of meter readings arranged in different rows. Each column represents the reading for a particular meter. We can pose an 0-1 Integer linear programming (ILP) with zero objective function.

$$\begin{aligned} & \min \quad 0^\intercal X \\ \text{s.t.} \quad & AX = \quad b \\ & x_j \in \{0, 1\}, \quad 1 \leq j \leq n \end{aligned} \tag{3.2}$$

As an illustration, consider the meter readings shown in Table 3.1 corresponding to the network in Fig. 1.2.

$time$	$M1$	$M2$	$M3$	$M4$	$M5$	$M6$
1	5.4	3	2.4	1.6	1.4	2.4
2	4.8	3.2	1.6	1.7	1.5	1.6
3	5.1	3.1	2	2.1	1.0	2.0

Table 3.1: Meter readings for three time instants for the network shown in Fig. 1.2

Here, M1 is the feeder meter, M2 and M3 are the DT meters and M4,M5 and M6 are the customer meters. For the above data, if we call different instances of LC(s,l) we can get full connectivity information of the network. Table 3.2 shows the different instances of LC problem which will give us full connectivity information. ILP solver from MATLAB was used to solve the above equation. If the number of readings is much less than the number of leaf meter readings, then the solution to the above optimization equation isn't unique.

s	l (leaf nodes)	Connectivity Information Obtained
M1	M4,M5,M6	Customer Feeder
M2	M4,M5,M6	Customer DT
M3	M4,M5,M6	Customer DT
M1	M2,M3	DT Feeder

Table 3.2: Instances of leaf connectivity problem

Incase of noisy measurements (Arya *et al.*, 2013) proposes two approaches, with and without sparsity based regularization. Moreover the regularization can be L1 or L2.

$$\begin{aligned} &\min_X \quad \| \, b - AX \, \|_1 \\ &x_j \in \{0,1\}^n \; / \; x_j \in [0,1]^n \end{aligned} \tag{3.3}$$

$$\begin{aligned} &\min_X \quad \| \, b - AX \, \|_2^2 \\ &x_j \in \{0,1\}^n \; / \; x_j \in [0,1]^n \end{aligned} \tag{3.4}$$

$$\begin{aligned} &\min_X \quad \lambda \textstyle\sum_j x_j + \| \, b - AX \, \|_1 \\ &x_j \in \{0,1\}^n \; / \; x_j \in [0,1]^n \end{aligned} \tag{3.5}$$

$$\begin{aligned} &\min_X \quad \lambda \textstyle\sum_j x_j + \| \, b - AX \, \|_2^2 \\ &x_j \in \{0,1\}^n \; / \; x_j \in [0,1]^n \end{aligned} \tag{3.6}$$

Equation 3.3 and 3.4 are without sparsity based regularization, while 3.5 and 3.6 are with sparsity based regularization.

3.2 Model identification using SVD/PCA

In the previous formulation, apart from the meter data, we need more information like whether a given reading is from the consumer or the transformer. Also, we need the number of consumers in the network. This information is not always available. Hence we formulate another problem to infer the network.

Let $x(t) \; : \; n \times 1$ be a vector of true meter measurements(error free) for a time instant. Let A be the connectivity model for the tree structured distribution network. As described in the subset sum problem formulation, only $n-m$ of these meter readings are independent, so the connectivity model is an $n \times m$ matrix. The term *constraint model* is also used interchangeably with connectivity model in data reconciliation literature. So, the following relation holds:

$$Ax(t) = 0 \tag{3.7}$$

Since, the connectivity model is constant with time unless the structure of the network changes, if X : $n \times N$ is a collection of meter reading for N time samples, then extending 3.1, we have

$$AX = 0 \tag{3.8}$$

If we compare this formulation with the $LC(s, l)$ formulation, we observe that the roles of A and X matrix are switched. In this formulation, the A is the constraint matrix and X is the data matrix. Since the vectors $x(t)$ span an $n - m$ dimensional subspace of $\mathbb{R}^n$, the row space of A is an m dimensional subspace of $\mathbb{R}^n$ using Rank-nullity theorem. Moreover, these two subspaces are orthogonal to each other. Let us represent the vector space spanned by $x(t)$ as V_x and the orthogonal subspace spanned by Row space of A as $V_x^{\perp}$. So, our problem reduces to finding m linearly independent vectors in the row space of A. The covariance matrix of the meter data is given by

$$S_X = \frac{1}{N} X X^T \tag{3.9}$$

We can write the SVD of the scaled matrix as

$$svd(\frac{X}{\sqrt{N}}) = U_1 S_1 V_1^{\intercal} + U_2 S_2 V_2^{\intercal} \tag{3.10}$$

where, U_1 are the orthogonal vectors corresponding to the $n - m$ largest eigenvalues of S_X while U_2 are the orthogonal eigenvectors corresponding to the remaining m eigenvalues of S_X. As stated earlier, the eigenvectors corresponding to U_2 are orthogonal to X and hence gives an estimate of rows of A matrix. That is, an estimate of A matrix is

$$\hat{A} = U_2^{\intercal} \tag{3.11}$$

The estimate of constraint matrix obtained is optimal in the least square sense. The estimate of the constraint matrix derived using PCA can differ from the true constraint

matrix form (that we desire) by a rotation matrix. As the sample size goes to infinity, the estimated and true constraint matrices are related as

$$\hat{A} = RA \tag{3.12}$$

where R is some non-singular matrix.

3.2.1 Comparing Estimated model with original model

Since the estimated constraint matrix and the true constraint matrix differ by a constraint matrix, only the row spaces of the true and estimated constraint matrices can be compared. The criterion used to measure similarity between the two row spaces as given in (Narasimhan and Shah, 2008) and (Narasimhan and Bhatt, 2015)are used. These are:

- the subspace angle between the row subspaces of the estimated and true constraint matrices.
- the sum of orthogonal distances of the row vectors of the estimated constraint matrix from the subspace defined by the rows of the true constraint matrix denoted by α, where,

  $$\alpha = \sum_i \alpha_i \tag{3.13}$$

  where

  $$\alpha_i = \parallel \hat{A}_i - \hat{A}_i A^\intercal (AA^\intercal)^{-1} A \parallel \tag{3.14}$$

  where $\hat{A}_i$ are the rows of the estimated constraint matrix.
- Element by element comparison between the regression matrix relating the dependent and independent variables of the original and estimated matrix.
  The meter readings are divided into dependent and independent variables. The leaf measurements are the independent readings. The number of dependent variables should be chosen equal to the number of constraints. The constraint given in equation 3.7 is rewritten as

  $$A_D x_D + A_I x_I = 0 \tag{3.15}$$

where, A_D and A_I are the sub-matrices of A corresponding to dependent and independent variables, respectively. the regression matrix relating the dependent variables to the independent variables can be obtained as

$$x_D = -A_D^{-1} A_I x_I = R x_I \tag{3.16}$$

Similarly, the estimated regression matrix can be obtained from the estimated constraint matrix as

$$\hat{R} = -\hat{A}_D^{-1} \hat{A}_I \tag{3.17}$$

An element by element comparison can be made between R and $\hat{R}$.

It has been observed that the subspace angle is not a very reliable quantity to compare the two matrices. We observe that when the number of samples is more than the number of independent variables, the element wise comparison of R and $\hat{R}$ defined above is very accurate. For other cases, sum of orthogonal distances provides good insight about the distance between true and estimated constraint matrix.

3.2.2 Identifying model order

Till now we have assumed that we know the model order or the number of consumers, which is generally the case. In the leaf connectivity formulation, this information is absolutely necessary along with labels whether a given meter reading is from a leaf meter (consumer) or non leaf meter. But in case we don't have labels which tell us the number of consumer meters, we can still estimate the model order accurately provided our measurements aren't too noisy. Generally a SCREE plot is used to estimate the model order but in case of Smart meter data, due to the range of singular value, using a SCREE plot becomes challenging. In such a scenario, we have used a modified SCREE plot, plotting logarithm of the singular values with their index. As shown later in the simulations, these perform accurately in the low noise case and when the number of elements aren't too high. We look for steep descent in the value of logarithm of the singular

value. There might be multiple steep descents in the plot. Generally the first descent is because of the fact the first principal component contains the maximum variance and hence its singular value is high compared to the other singular values. Hence we neglect the first steep descent in case of multiple descents. In case there aren't steep descent, we can do curve fitting to identify three regions, namely the sharp descent which contains variables having the maximum variance, then a straight line having low gradient and then another curve having a high gradient. The last curve will tell us the number of dependent components.

CHAPTER 4

SIMULATION ON SYNTHETIC DATA

This chapter describes the experimental setup and the assumptions in generating the data to infer the network using techniques described in the previous chapter.

4.1 Generating test data

There are a number of smart meter manufacturers and distribution companies. The noise levels in these smart meters, albeit following some standards vary depending on the meter condition. Similarly, the size of distribution network depends on the population density and power requirements of the area. Once the size is fixed, multiple instances are generated with different noise levels.

We have generated data for network of different sizes. The small network consists of around 20 components, the medium sized network consists of around 300 components and the large network consists of around 3000 components. Random graphs of the above mentioned sizes are used as the distribution network.

Technical standards define requirements for accuracy for measuring of power in different operation modes. In Europe and other non-European countries these are usually the standards *IEC 62053-24* and, for the American market, the standards of the *ANSI C12* series. The accepted accuracy is $\pm 1.5\%$ for most smart meters (VACUUMSCHMELZE, 2012). So in the data generating process, we have assumed a gaussian noise model with 6 σ = 3% of the error free value. However, since the meter accuracy

may decrease over the course of usage, we have considered higher noise levels too for testing our solution.

4.2 Modified Subset-sum method

We use the problem formulation described in section 3.1. We use this technique only for small sized network to verify its results. We simulate a distribution network with 15 consumer meters, 4 DTs and 2 feeders. We first consider noise less meters and then consider meters with upto $\pm 1.5\%$ accuracy. We use ILP solver provided by MATLAB^{TM} to solve the noise less LC problem given by equation 3.2. We call 4 instances of LC, one for each DT and feeder with all the consumer meters as the leaf meters and one with the DTs as the leaf meter. This gives us the complete network. We observe that the number of meter readings should be atleast close to the total number of meters to obtain any useful solution.

Next we try to solve the noisy LC problem. We use the formulation given in equation 3.3 and use CPLEX (put source) to solve this equation. This formulation also gives a matrix which is close to the original constraint matrix. However we don't perform any further computations using this method because these formulations are NP hard and we need to know the labels of meters to be able to get the connectivity model.

4.3 Model identification using SVD

In this section, we use the meter measurements without labels; that is, we don't have the information whether a particular meter reading is from a leaf node or a non leaf node. First we infer the model order and then we obtain the connectivity model.

4.3.1 Identifying model order

In the previous set of experiments, we have assumed that we know the number of leaf meters(customer meters) in the tree network. These leaf meters are independent meter readings. When calculating the SVD of a matrix, only the independent components will have a non zero singular value. Hence, we can identify the number of customers by plotting the singular values in decreasing order. Such a plot is called a SCREE plot. However in our case, since the number of independent variables are high particularly in the large network and also the difference between the lower singular values being very close due to noise, we plot the logarithm of the singular value and identify the model order. The model order can be approximated by looking at the sharp drop in the values of log(singular value). The model order can be predicted with high accuracy in case of error free network as the singular values will be very close to 0 for all the dependent variables. In our experiments, for the noiseless case, Fig. 4.1 shows the change in value of log(singular values). The number of leaf readings (300 and 3000 respectively for medium and large network) can be obtained accurately.

Figure 4.2 and 4.3 shows the modified SCREE plots for medium and large networks respectively with varying noise levels. We observe that as the noise levels increase, the accuracy of obtaining the model order decreases as expected. For noise level with $\sigma = 0.5$ (the operational accuracy set by metering standards), we can obtain an accurate estimate of the model order.

4.3.2 Inferring connectivity model

As described in section 3.2, we aim to infer the connectivity model. We have tried two approaches, one is calculating the exact SVD using standard linear algebra library LAPACK and another using Randomized SVD as described in [Enter reference randsvd].

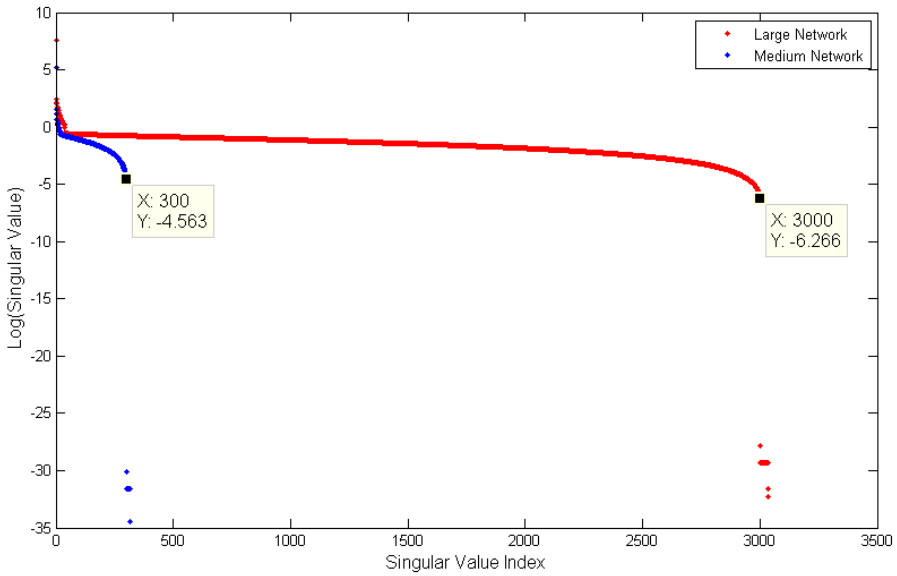


Figure 4.1: Identifying model order in noiseless case using modified SCREE plot

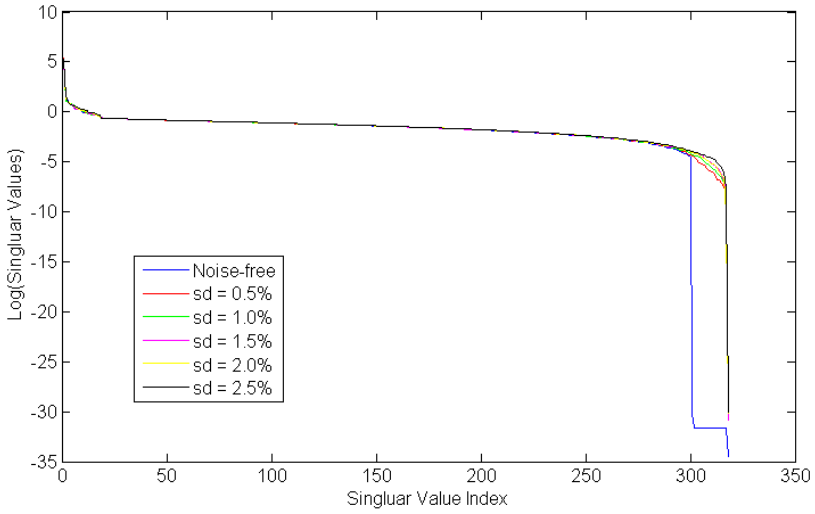


Figure 4.2: Identifying model order in medium sized noisy network

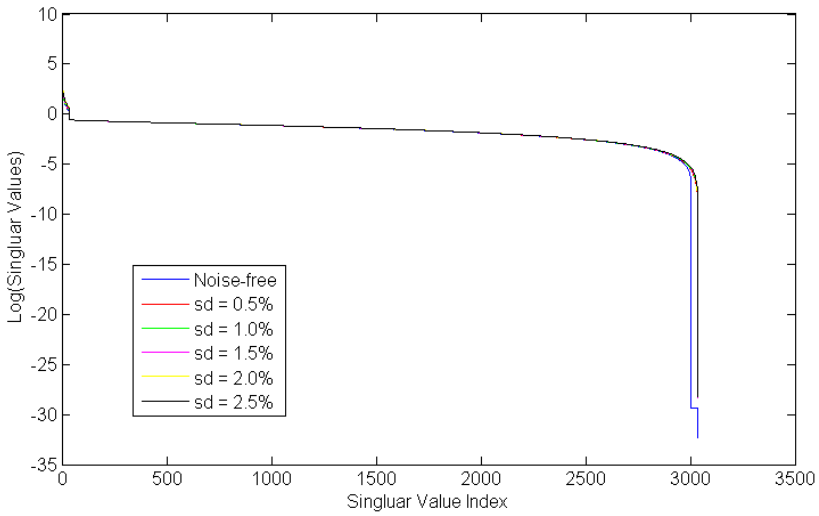


Figure 4.3: Identifying model order in large noisy network

Fig 4.4 shows the plot of α as a function of the number of meter measurements in case of no loss condition for medium and large network. We observe that once the number of measurements equals the number of independent variables in the network (number of leaf nodes), we have obtained the network completely.

Fig 4.5 and 4.6 shows the similar results in case of noisy data. All these results use SVD provided by LAPACK library.

4.3.3 Inferring connectivity model using Randomized SVD

In section 2.3, we described a randomized algorithm for calculating the SVD of a matrix. In case of randomized SVD, we find the SVD of a much smaller matrix and hence it is expected to give results faster. In the first step of the algorithm, we try to estimate the row space of the matrix. In case of smaller matrices, the computation time saved by computing the SVD of a smaller matrix is negated by the computations in estimating the row space. Hence we use this algorithm for large datasets only. In this section, we

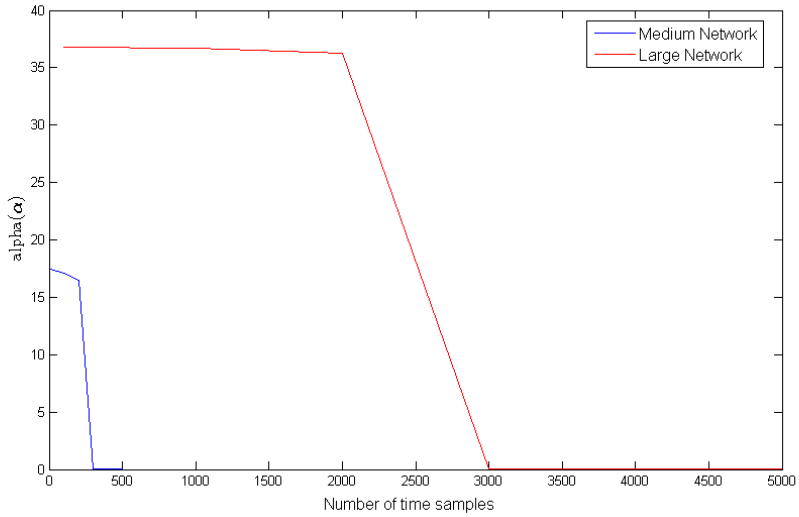


Figure 4.4: Inferring connectivity model for no noise case

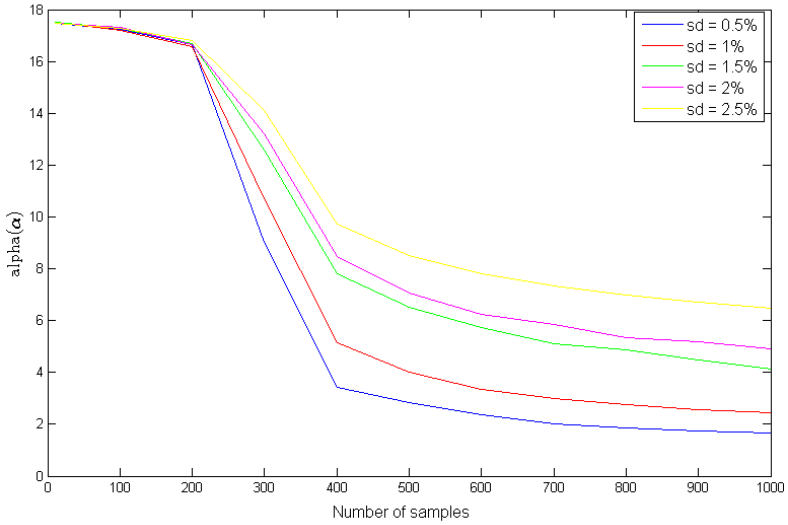


Figure 4.5: Inferring meter measurements for Medium network in case of noisy network for different noise levels

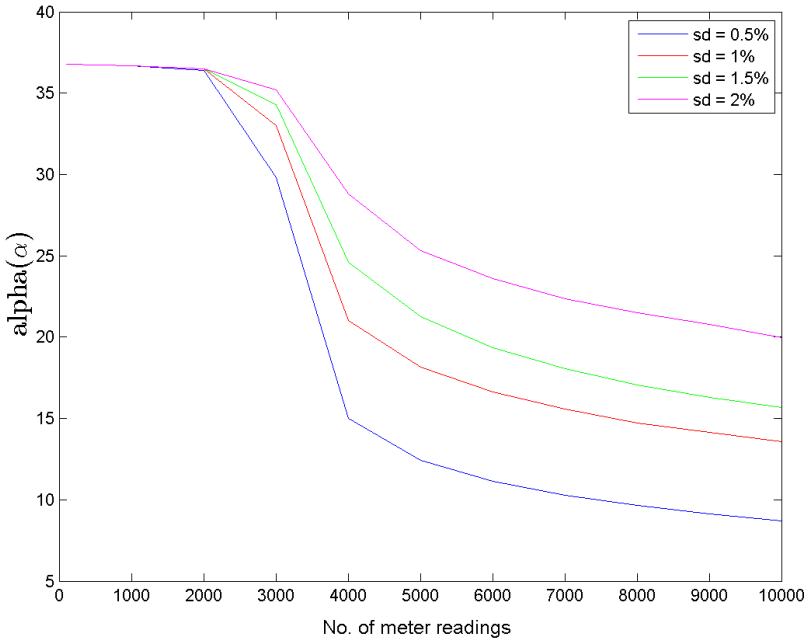


Figure 4.6: Inferring meter measurements for Large network in case of noisy network for varying noise levels

compare the performance of randomized SVD against LAPACK SVD based on computation time and accuracy. Fig 4.7 shows variation of α in case of LAPACK SVD and randomized SVD. We have used a reduced dimension of 1000 to compute randomized SVD. We observe that randomized SVD matches the performance of deterministic algorithms very closely. Fig 4.8 shows the plot of computation time. We observe that asymptotically, randomized algorithm performs better than normal SVD in computing the constraint matrix.

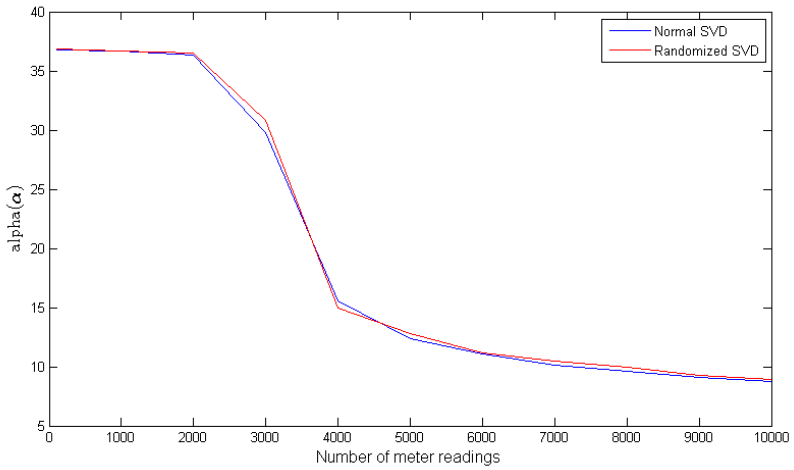


Figure 4.7: Comparing performance of Randomized SVD and LAPACK SVD to infer connectivity model

4.3.4 SVD using Apache Spark™

Apache Spark™ is a cluster computing framework. Its machine learning library (MLlib) has the SVD algorithm which we use for this experiment. Spark is run in pseudo-distributed mode in these set of experiments. For installing Spark, refer (Prabeesh, 2014) and (Geusebroek, 2014). We run Spark in pseudo distributed mode since we don't want to deal with network ping etc. We use an Intel *i7* processor for performing

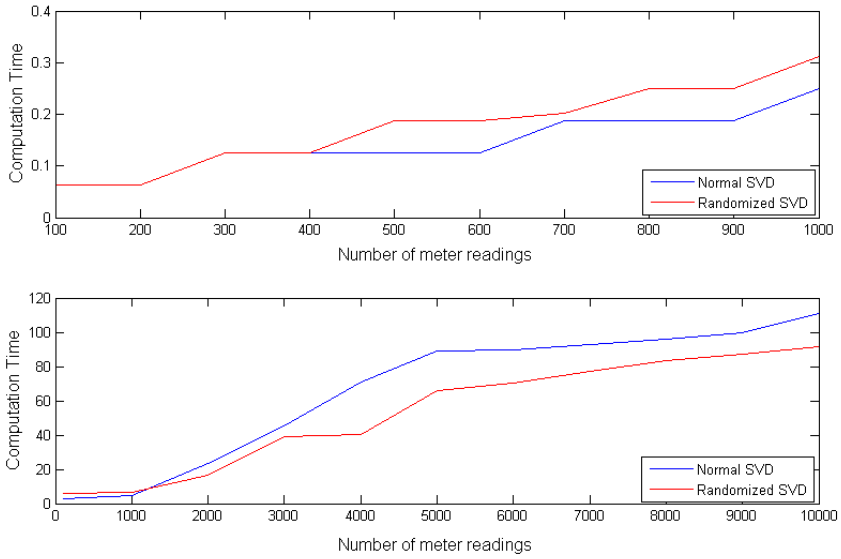


Figure 4.8: Comparison of computation time between Randomized SVD and LAPACK SVD

our experiments. We convert our data into an instance of Resilient Distributed Dataset (RDD) and use Spark to perform the necessary computation. Fig 4.9 shows a comparison in accuracy of inferring connectivity model by using Spark distributed algorithm and LAPACK library. Since SPARK uses standard deterministic SVD algorithm, there is no loss of accuracy while inferring the constraint matrix. Moreover the resilient distributed database can handle network failure and store larger dataset which can't be loaded in the RAM of traditional systems. We have used Spark only on the large network meter readings since the initial setup time and overhead neutralizes the time saved by Spark. Fig 4.10 compares the computation time required by Spark with randomized SVD and LAPACK SVD. We observe that Spark performs slightly better than traditional SVD on large data. This effect will be more dominant if the data size is even larger and when the data cannot be loaded into RAM.

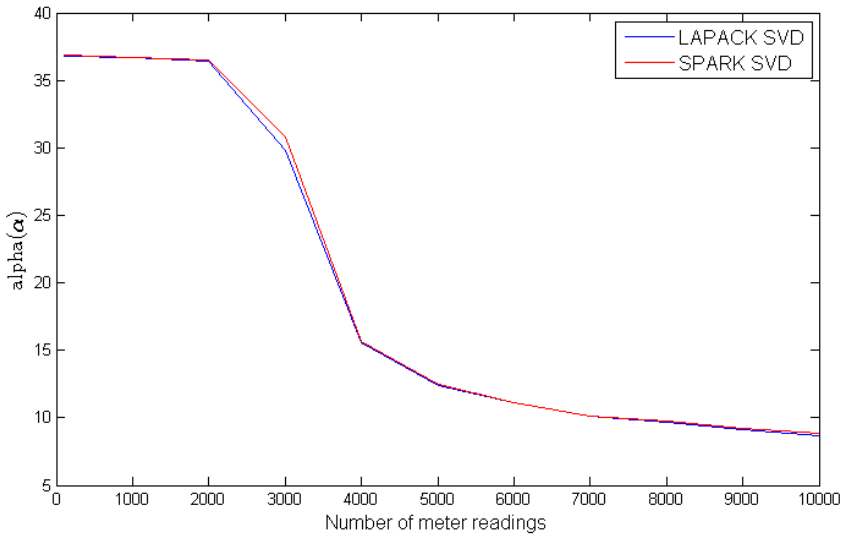


Figure 4.9: Comparing performance of Spark SVD and LAPACK SVD to infer connectivity model

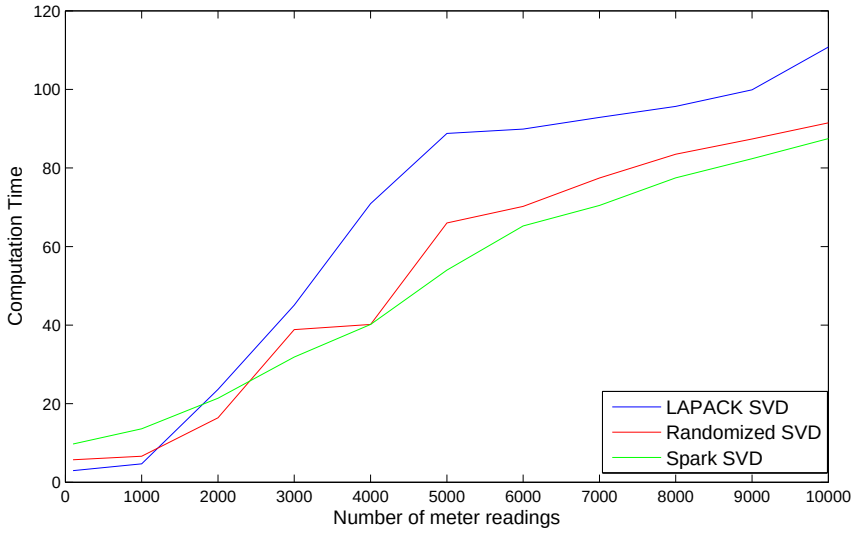


Figure 4.10: Comparison of computation time between Spark SVD, Randomized SVD and LAPACK SVD

CHAPTER 5

SIMULATION ON SEMI-SYNTHETIC DATA

5.1 Description of the dataset

In the previous chapters, the meter readings we used were generated using pseudo-random number generators. In this chapter, we test our solution on semi-synthetic data. (Bristol City Council, 2014) has published anonymous household meter readings from 5 buildings. The summary of the data can be seen in Table 5.1. As seen in the previous section, if the noise levels are within the specifications, the amount of information added by increasing the number of meter readings once the number of readings has reached the number of independent components is very less. Since we have assumed that the meter readings are spatially indpendent, we have broken down the large number of meter readings from a single meter into readings from multiple meters for shorter duration. The rearranged data is described in Table 5.2.

Building No.	No. of meteres	No. of days of meter readings
1	3	362 , 331, 365
2	1	337
3	1	360
4	1	348
5	1	354

Table 5.1: Bristol City Council Smart Meter Readings

As in the previous chapter, we consider two cases, lossy-erroneous meter and non lossy error free meters. We assume each building to be under one distribution tranformer. In the case of erroneous meters, we have considered the standard deviation of

Building No.	No. of meters	No. of days of meter readings
1	35	30
2	11	30
3	12	30
4	11	30
5	11	30

Table 5.2: Rearranged Meter readings from Bristol City Council

the meters to be 0.5% of its mean value. The noise is assumed to be gaussian white noise.

5.2 Model Identification using SVD

5.2.1 Inferring Model order

As described in Section 3.2.2, we first identify the model order from the meter readings. Figure 5.1 shows the modified SCREE plots for the dataset with and without noise. Gaussian noise is generated with $\sigma = 0.5\%$ of the mean meter reading, which translates to an accuracy of 1.5%. The blue colored plot shows the descent of logarithm of singular values in case of noise free network, while the red colored plot shows shows descent for noisy network with $\sigma = 0.5\%$ of the mean meter reading. From the plot, we infer that there are around 80 consumer meters in the network. In case of noisy network, obtaining accurate results is difficult. However the modified SCREE plot gives a close estimate. For the next step, we assume that the number of consumer meters are 80.

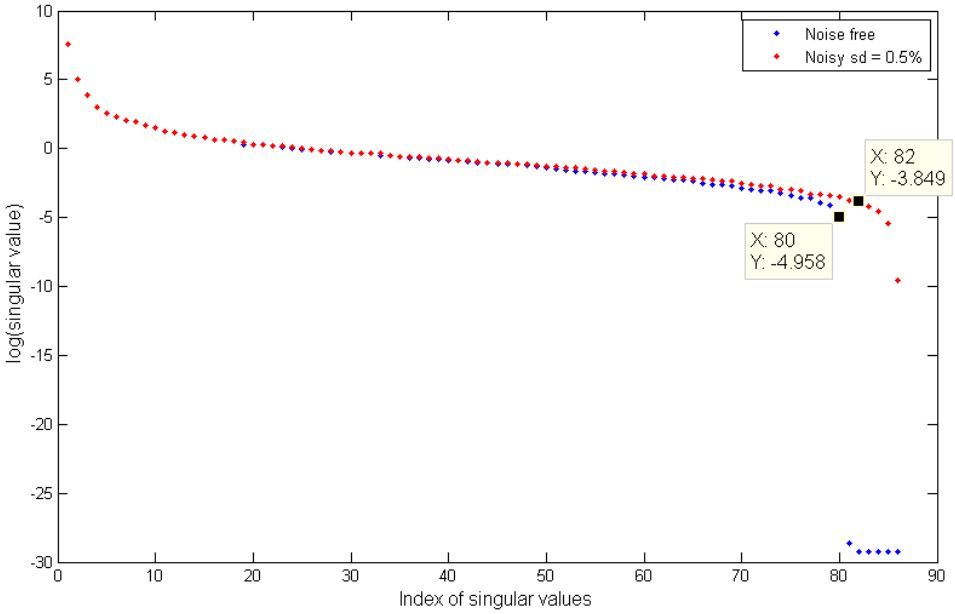


Figure 5.1: Modified SCREE plot to infer model order

5.2.2 Building connectivity model

In Section 3.2, we described how to obtain the constraint matrix from meter data. We apply the same procedure on this dataset. First we consider the no noise case and then we consider the case of noisy network. We perform SVD on the dataset while varying the number of meter readings. This will give us an estimate of the number of readings required to obtain a solution which can be used to perform further calculations. We have used to metrics to guess the accuracy of the estimate. These metrics have been defined section 3.2.1. Fig 5.2 shows these two metrics in no noise condition. The top plot shows the variation of α as a function of number of meter readings. We observe that as the number of readings increase, α decreases until the number of readings equals the number of components. We identify the complete network with maximum accuracy and any subsequent meter reading doesn't add any information. The same pattern is seen

in the bottom plot showing distance between the estimated regression matrix and the actual regression matrix. These two matrices are supposed to be same element wise. So we compare their norm distance. Once again, as the number of meter readings approach the number of independent components, maximum reconstruction is done.

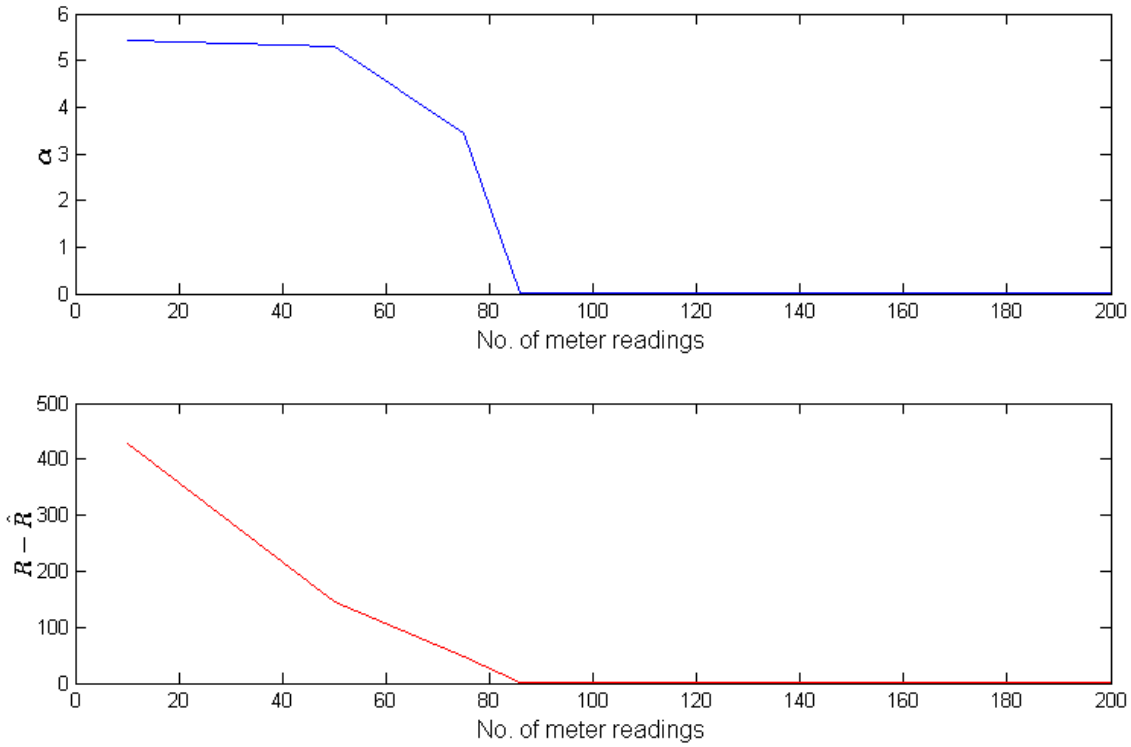


Figure 5.2: Variation of α and $\| R - \hat{R} \|$ (estimated regression matrix from original matrix) with no of meter readings for noise-free network

Fig 5.3 shows these two metrics in noisy condition. The top plot shows the variation of α as a function of number of meter readings. Similar to the earlier plot, as the number of observations increases, α decreases. But since its noisy data, we cannot infer the exact model, but with more data, we reach a good estimate of the network. The same trend is followed by $\| R - \hat{R} \|$. We also observe that given the size of dataset, we aren't able to

infer a very accurate connectivity model.

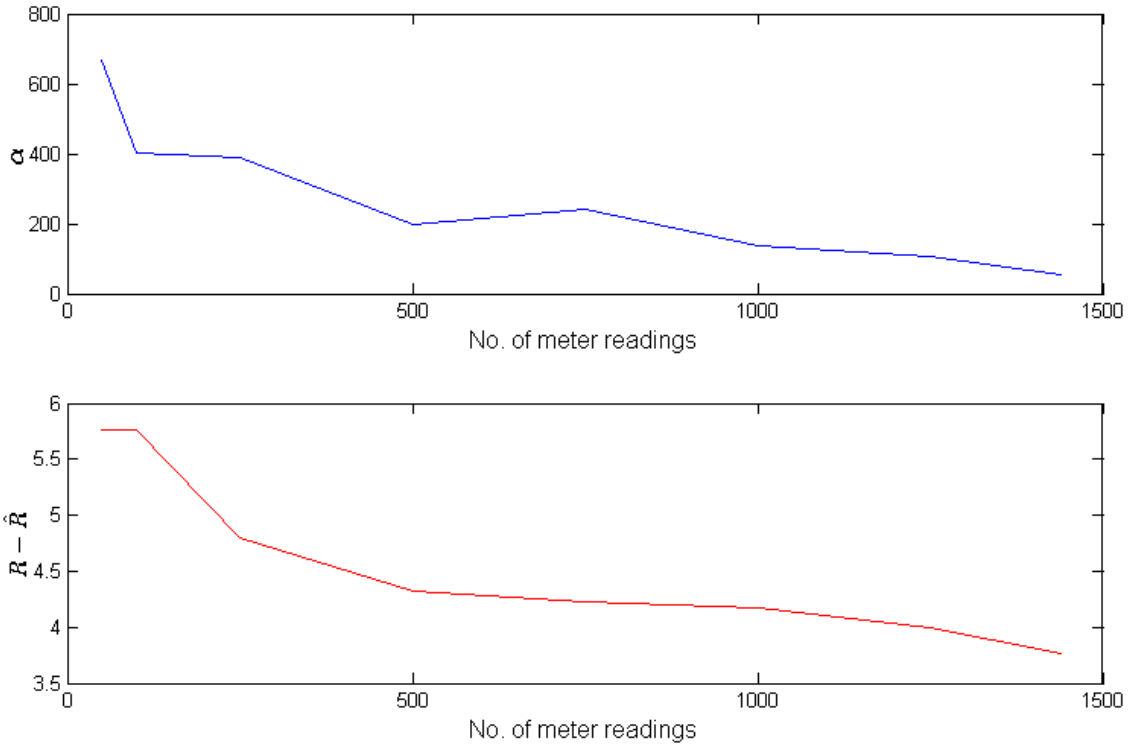


Figure 5.3: Variation of α and $\parallel R - \hat{R} \parallel$ (estimated regression matrix from original matrix) with no of meter readings for noisy network

So, we get back to our original dataset described in Table 5.1 and modify it such that we have more instances of meter readings as opposed to the last time(1440 readings). Our new modified dataset is described in Table 5.3. In this dataset, we have 100 days of meter reading for each meter giving us 4800 meter readings for each meter. Also notice that the number of independent meters have also decreased which should give us a better estimate. Once again we perform SVD on this dataset and plot our observations. These observations are plotted in Fig 5.4. We observe that as we increase the number of meter readings, our estimate of the constraint matrix gets more accurate. We are able to extract the constraint model from the data.

Building No.	No. of meteres	No. of days of meter readings
1	9	100
2	3	100
3	3	100
4	3	100
5	3	100

Table 5.3: Rearranged Meter readings from Bristol City Council with large number of time samples

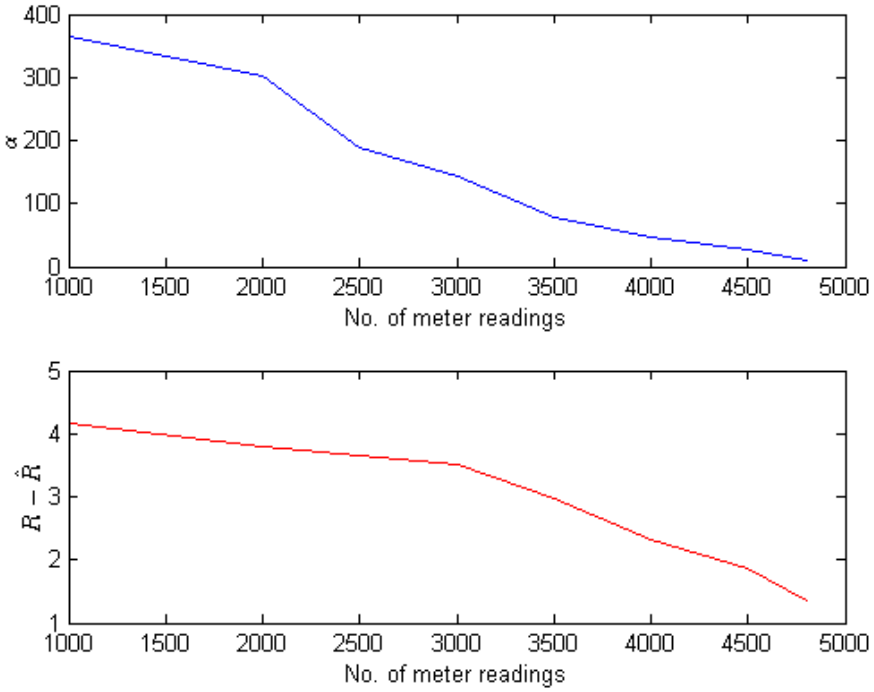


Figure 5.4: Variation of α and $\parallel R - \hat{R} \parallel$ (estimated regression matrix from original matrix) with no of meter readings for noisy network with more meter readings.

CHAPTER 6

Conclusion and Future Scope

In chapter 4 and chapter 5, we saw SVD being used to identify the connectivity model of power distribution system. In most of the cases, we could only get a matrix containing the same row space as the original matrix. This information can be used to perform Fault diagnosis in the distribution network (Yoon and MacGregor, 2000). We compared three different implementations of SVD to calculate the connectivity model with respect to computation time and accuracy of results. We saw that the deterministic SVD algorithm provided by LAPACK is very accurate but has limitations in the form of computation speed and dependence on RAM. We also observed that randomized SVD can perform as good as deterministic SVD with high probability. It has a lower computation time then the first SVD. Finally we used an instance of Spark to compute the SVD in distributed setting. We only used pseudo distributed mode in order to ignore network delays. It outperformed LAPACK SVD in computation time while giving similar performance. For larger datasets, it will perform even better given its asymptotically improving computation time and ability to handle extremely large datasets.

The above methodologies are also tested on real dataset provided by (Bristol City Council, 2014). The network provided by the above methodologies are very close to the original network thus validating these techniques.

Given the increasing size of power networks and the continuous improvement in storage technology, we expect smart meter data to play a much bigger role. An ideal network identification method should be able to make use of this increasing availability of data. A distributed computing based algorithm, which uses randomness to reduce the

size of computation would be an ideal choice. Such an algorithm can be developed and tested in future. Also, fault diagnosis systems for linear process can be tested on Spark which will reduce the computation time and help in making real time decisions.

REFERENCES

1. **Arya, V.**, **T. S. Jayram**, **S. Pal**, and **S. Kalyanaraman** (2012). Determining a connectivity model in smart grids.

2. **Arya, V.**, **T. S. Jayram**, **S. Pal**, and **S. Kalyanaraman**, Inferring connectivity model from meter measurements in distribution networks. *In Proceedings of the Fourth International Conference on Future Energy Systems*, e-Energy '13. ACM, New York, NY, USA, 2013. ISBN 978-1-4503-2052-8. URL `http://doi.acm.org/10.1145/2487166.2487186`.

3. **Baker, K.** (2005). Singular value decomposition tutorial. *The Ohio State University*, **2005**, 1–24.

4. **Bristol City Council** (2014). Energy Consumption for selected Bristol buildings from smart meters by half hour. URL `http://data.gov.uk/dataset/`.

5. **Caird, K.** (2010). Meter phase identification.

6. **Cormen, T. H.**, **C. Stein**, **R. L. Rivest**, and **C. E. Leiserson**, *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001, 2nd edition. ISBN 0070131511.

7. **Geusebroek, K.** (2014). Local and Pseudo-distributed CDH5 Hadoop. URL `http://blog.godatadriven.com/local-and-pseudo-distributed-cdh5-hadoop-on-your-laptop.html`.

8. **Halko, N.**, **P.-G. Martinsson**, and **J. A. Tropp** (2009). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *ArXiv e-prints*.

9. **Narasimhan, S.** and **N. Bhatt** (2015). Deconstructing principal component analysis using a data reconciliation perspective. *Computers & Chemical Engineering*, **77**(0), 74 – 84. URL `http://www.sciencedirect.com/science/article/pii/S0098135415000873`.

10. **Narasimhan, S.** and **S. L. Shah** (2008). Model identification and error covariance matrix estimation from noisy data using {PCA}. *Control Engineering Practice*, **16**(1), 146 – 155.

11. **Prabeesh, K.** (2014). Install Apache Spark on Ubuntu-14.04. URL http://blog.prabeeshk.com/blog/2014/10/31/install-apache-spark-on-ubuntu-14-dot-04/.

12. **Shlens, J.** (2014). A Tutorial on Principal Component Analysis. *ArXiv e-prints*.

13. **VACUUMSCHMELZE**, *Current transformers and Power Line transformers for smart metering*. VACUUMSCHMELZE GmbH & Co., 2012.

14. **Yoon, S.** and **J. F. MacGregor** (2000). Statistical and causal model-based approaches to fault detection and isolation. *AIChE Journal*, **46**(9), 1813–1824. ISSN 1547-5905. URL http://dx.doi.org/10.1002/aic.690460910.