Applying the Active Subspace Method in Network and Measurement Science

Tadas Krikstanavicius

Department of Mathematics and Statistics

University of Strathclyde

Glasgow

2021

This thesis is submitted to the University of Strathclyde for the degree of Doctor of Philosophy in the Faculty of Science.

Declaration of Authenticity and Author's Rights

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed: Tadas Krikstanavicius

Date: 15/03/2021

Acknowledgements

Throughout the writing of this dissertation I have received a great deal of support and assistance.

First and foremost, I would like to thank my supervisors Dr. Alison Ramage (University of Strathclyde), Prof. Des Higham (University of Edinburgh) and Prof. Alistair Forbes (National Physical Laboratory) for their dedicated support, encouragement and guidance. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level. Without their persistent help, the goal of this project would not have been possible.

I also want to pay my special regards to the National Physical Laboratory for the support and funding.

Finally, none of this could have happened without my family and friends. I would like to thank Regina, Tomas, Kestutis, Loreta, Juozas, Odeta, Petras, Karolina, Yongmei, Raymond for their patience, understanding and support throughout this research project.

Abstract

Mathematical models of real-world problems in uncertainty quantification, optimisation and sensitivity analysis use a large number of parameters which can be expensive and time consuming. Often simulations have to be run multiple times to effectively study inputs and outputs. However, each simulation can be extremely expensive. An active subspace approach allows us to identify an important linear combination of parameters instead of analysing all of them individually. This can significantly reduce the dimension of the parameter study. We develop these ideas in a network science context to find important edges in a given network. Then, we make a comparison with the Sobol method, which is an alternative approach to sensitivity analysis. Moreover, we apply the active subspace method to examples in measurement science.

Our analysis shows that the active subspace method on networks was able to identify important edge(s) or connections in a given graph. The active subspace method and Sobol method result in similar findings; however, the active subspace method is computationally less expensive. We are able to validate some of these results using extra information about the networks. In the measurement science setting, the results of the active subspace method match with the results obtained previously by NPL.

Contents

List of Tables viii

List of Figures ix

1 Introduction 1

1.1 Sensitivity Analysis 1

1.2 Examples of Sensitivity Analysis 3

1.2.1 The Active Subspace Method 3

1.2.2 Sobol Indices 3

1.2.3 Principal Component Analysis 3

1.3 Network and Measurement Science 7

1.3.1 Network Science Context 8

1.3.2 Measurement Science Context 8

1.3.2.1 Introduction to Laser Trackers 9

1.4 Summary of Thesis 12

2 The Method of Active Subspaces 14

2.1 Introduction 14

2.2 Background Theory 15

2.3 Practical Algorithm 18

2.4 Monte Carlo and Bootstrapping Methods 21

2.5 Numerical Examples 24

2.5.1 Comparison with a Global Linear Model 24

2.5.2 A Quadratic Model 26

2.6 Summary 29

3 Background Theory on Networks 30

3.1 Introduction to Network Theory 30

3.2 Centrality Measures 35

3.2.1 Degree Centrality 35
3.2.2 Katz Centrality 36
3.2.3 Eigenvector Centrality 37
3.2.4 Edge Betweenness Centrality 38
3.3 Gradient of Katz Centrality 38

4 Active Subspaces in Network Science 41
4.1 Overview 41
4.2 Finite Difference Method Comparison with Exact Derivative 43
4.3 Examples of Synthetic Networks 45
4.4 Zachary's Karate Club Network 59
4.5 Background on PPI Networks 66
4.5.1 PPI Network and the Active Subspace Method 68
4.6 Summary 71

5 Comparison with Sobol Indices 72
5.1 Background Theory on Sobol Indices 72
5.2 Introduction to Variance-Based Methods 73
5.3 Calculating Sobol Indices 74
5.4 Examples Using Sobol Indices 76
5.4.1 Example 1: A Simple Function f 77
5.4.2 Example 2: Sobol G-function 79
5.4.3 Gaussian Test Function 81
5.5 Comparison of the Sobol Method with Active Subspaces 82
5.5.1 Comparison of the Active Subspace Method with Sobol Method on the Sobol G-function 82
5.5.2 Comparison of Two Methods on a Synthetic Network 84
5.5.3 Sobol Method and Zachary's Karate Club Network 87
5.6 The Active Subspace Method vs The Sobol Method 90
5.7 Summary 91

6 Applying the Active Subspace Method to the Design of Experiments 93
6.1 Introduction 93
6.2 The Design of a Calibration Experiment 95
6.2.1 Least Squares Problem and Aggregate Measures of Uncertainty 95
6.3 NPL Approach to Designing a Calibration Experiment 98
6.4 Applying the Active Subspace Method 100

6.4.1 Example: Fitting a Straight Line to Given Data 102
6.4.2 Example: Fitting a Straight Line to Data with Extra Information 105
6.4.3 Laser Tracker Calibration Experiment 106
6.5 Conclusion 108

7 Conclusion 109
7.1 Thesis Achievements 109
7.2 Future Work 110

References

List of Tables

1.1 Data set of hours spent studying monthly by each student, mark obtained and their age. 4

4.1 Comparison of finite differences against exact derivative to 6 decimal places. 45

4.2 Katz centralities for the network in Figure 4.4 with all edge weights equal to 1. 47

4.3 Ordering of the edges in Figure 4.6. 49

4.4 Katz centralities for the network in Figure 4.8. 51

4.5 Ordering of the edges in Figure 4.9. 52

4.6 Katz centralities for the network in Figure 4.11. 54

4.7 Ordering of the edges in Figure 4.12. 55

4.8 Ten largest edge betweenness centralities for the network in Figure 4.13. 57

4.9 Katz centralities for the network in Figure 4.13. 57

4.10 Twenty largest edge betweenness centralities for the network in Figure 4.17a. 64

5.1 Approximate first and total order sensitivity indices for (5.14), as in Figure 5.1. 78

5.2 Approximate first and total order sensitivity indices for (5.15). . . . 80

5.3 The results of Sobol method on Sobol G-function in (5.15). 83

5.4 The results of the active subspace method on Sobol G-function in (5.15). 84

5.5 Five most important links or connections identified as important by either the active subspace method or the Sobol method. 90

List of Figures

1.1 A plot of data of hours spent studying monthly, mark obtained and age. 4

1.2 A plot of three principal components. 6

1.3 A plot of two principal components. 7

1.4 Examples of a laser tracker. From left to right: API's OT2 Core, FARO's Vantage, Leica Hexagon AT403 [38] 9

1.5 An example of a reflector [43] 10

1.6 The working principle of the laser tracker. 11

1.7 An example of an aircraft with target locations to be determined by the laser tracker. 12

2.1 The Monte Carlo simulation: basic process. 21

2.2 Monte Carlo approximation to $\mathbb{E}(e^{\mathbb{Z}})$, where $\mathbb{Z} \sim N(0,1)$. Crosses are the approximations, vertical line are 95% confidence intervals. Horizontal line is at height $\sqrt{e}$. 23

2.3 Left: Sufficient summary plot for the active subspace estimation with gradients for the function $f(x_1, x_2) = \sin(4\pi x_1) + 4x_2$. Right: Sufficient summary plot for the active subspace estimation with a global linear model. 25

2.4 Left: Sufficient summary plot for the active subspace estimation with gradients for the function $f(x_1, x_2) = \frac{1}{2}(0.7x_1 + 0.3x_2)^2$. Right: Sufficient summary plot for the active subspace estimation with a global linear model. 26

2.5 Three choices of eigenvalues for matrix $\mathbf{A}$. 27

2.6 True and estimated eigenvalues along with the bootstrap intervals. . 28

2.7 Estimated subspace error along with bootstrap intervals. 29

3.1 Facebook network in 2010. 31

3.2 A simple symmetric graph (left) and its adjacency matrix (right). . 32

3.3 A directed graph (left) and its adjacency matrix (right). 33
3.4 A weighted symmetric graph (left) and its adjacency matrix (right). 34
3.5 A weighted nonsymmetric graph (left) and its adjacency matrix (right). 34

4.1 Left: A weighted graph with 4 nodes and 10 edges. Right: A weighted adjacency matrix. 42
4.2 A simple network with 15 nodes and 14 edges. 44
4.3 Difference between exact derivative and finite difference approximation. 45
4.4 A synthetic network with 15 nodes and 28 edges. 46
4.5 The eigenvalues obtained by the active subspace method for the network in Figure 4.4. 48
4.6 The leading eigenvector for the network in Figure 4.4. 49
4.7 Sufficient summary plot of $f(\mathbf{x})$ (Katz centrality of node 1 for each sample) against the active variable ($\hat{\mathbf{w}}_1^T\mathbf{x}$). 50
4.8 A network with 20 nodes and 19 edges. 51
4.9 The leading eigenvector for the network in Figure 4.8. 52
4.10 Highlighted dominant edges. 53
4.11 A network with 13 nodes and 12 edges. 53
4.12 The leading eigenvector for the network in Figure 4.11. 54
4.13 A network with 20 nodes and 59 edges. 55
4.14 The leading eigenvector for the network in Figure 4.13. 56
4.15 Comparison of the results from the active subspace method with respect to node 19 and edge betweenness centrality for the network in Figure 4.13. 58
4.16 Comparison of the results from the active subspace method with respect to node 5 and edge betweenness centrality for the network in Figure 4.13. 59
4.17 Graphical representation of Zachary's Karate Club. 60
4.18 Two leading eigenvectors arising from quantities of interest based on the centrality of Node 34 (John A) and Node 1 (Mr. Hi). 62
4.19 The first five important links to either Node 34 (five green links) or Node 1 (five blue links) for Zachary's Karate Club network. 63
4.20 Zachary's Karate Club network: comparison of the results from the active subspace method with respect to node 34 and edge betweenness centrality for the network in Figure 4.17a. 65

4.21 Zachary's Karate Club network: comparison of the results from the active subspace method with respect to node 1 and edge betweenness centrality for the network in Figure 4.17a. 66
4.22 The complete protein interaction network of yeast showing essential proteins in red and non-essential ones in blue, yellow circles correspond to proteins with unknown essentiality [78]. 67
4.23 Adjacency matrix of a PPI network. 67
4.24 Proportions of three types of edges after permuting the eigenvector. 69
4.25 Proportions of three types of edges with confidence intervals (or "envelopes"). 70

5.1 The importance of parameters to the output Y or contribution of each parameter to the variance of Y. 77
5.2 Second order sensitivity indices $\hat{S}_{1,2}$, $\hat{S}_{1,3}$, $\hat{S}_{2,3}$. 78
5.3 The importance of parameters: approximate first and total order sensitivity indices for Sobol G-function in (5.15). 79
5.4 Difference between analytical and estimated values with Sobol method for S_1 on (5.15). 80
5.5 Approximate first and total order sensitivity indices for (5.16). . . . 82
5.6 The active subspace method on Sobol G-function with finite difference method. 83
5.7 A simple network with 15 nodes and 28 edges. 85
5.8 Simple synthetic network with 15 nodes and 28 edges using Sobol method. 85
5.9 The leading eigenvector obtained by using the active subspace method with $M = 10^4$ samples. 86
5.10 A combination of Figure 5.9 and the first order sensitivity indices in Figure 5.8. 87
5.11 The Sobol method for Zachary's Karate Club network. 88
5.12 Results from the Sobol method and the active subspace method on Zachary's Karate Club network. 88

6.1 The dominant eigenvector with respect to the model in (6.15) and (6.16). 103
6.2 The dominant eigenvector with respect to the observation matrix in (6.17). 104
6.3 The dominant eigenvector with respect to the matrix in (6.18). . . . 106

6.4 The leading eigenvector of a black-box problem obtained by the active subspace method. 107

Chapter 1

Introduction

1.1 Sensitivity Analysis

Many mathematical models of real-world problems involve a large number of parameters. For instance, consider an example in uncertainty analysis in the automotive industry where the Noise Vibration Harshness (NVH) is investigated [1]. Specifically, the analysis of 46 different sheet thicknesses of a car body against the sound pressure level at certain frequencies is carried out. This is a real-world example of a problem with many parameters [1, 2]. We want to know which sheet thickness or sheet thicknesses are the most important when analysing the sound pressure level. This can be done by using sensitivity analysis.

Simply put, the input is the set of numbers you feed into the model, the output is what you obtain after the calculations are done in the model. The basic idea of sensitivity analysis is to analyse how inputs affect the output or how sensitive the quantity of interest is to different input parameters. The quantity of interest is a quantity that is measured in order to answer a specific question about the model. For instance, the example we just encountered considers calculating the quantity of interest (sound pressure level at certain frequencies) given input parameters (46 different sheet thicknesses of a car body) [1].

Sensitivity analysis is a fundamental approach used to identify influencial parameters and understand complex models, especially computationally expensive models with a large number of parameters. It is known that sensitivity analysis can improve parameter calibration efficiency, reduce model uncertainty and reveal model structure [3, 4, 5, 6]. The more complex a given problem is, the harder it is to correctly identify important parameters. We want to correctly identify parameters that have the most leverage or influence on the quantity of interest [7]. Statisticians and engineers often need simulations that have to be run multiple times to effectively study inputs and outputs. Each simulation can be computationally expensive [8].

Sensitivity analysis methods are generally grouped into two types: local methods and global methods [9]. Local methods only make one parameter change at one time, while the other parameters are kept as constant [10, 11]. It is known that local methods are computationally cheap. However, local methods are not capable of determining the interactions between parameters. Local sensitivity analysis is used less than global sensitivity analysis (which will be described shortly) due to the fact that sensitivity parameter estimation can be affected by non-linearities and/or interactions in a given model. However, for a given problem where it is extremely expensive to obtain reasonable results using global methods, local sensitivity analysis method can still be useful. This is due to the fact that fewer simulations are required in order to obtain estimated sensitivity indices [10]. The main examples of local methods are the one-at-a-time (OAT) method and the Morris method [12].

In contrast to local methods, global sensitivity methods enable us to obtain more sophisticated or detailed results. Specifically, they overcome the disadvantages of local methods by changing more than one of the model parameters at the same time [4, 9, 11]. In general terms, global sensitivity analysis focuses on the variance of model output and how the input variability influences the output variance. There are a number of global methods that have been proposed and used, such as Analysis of Variance (ANOVA) [13], the Fourier amplitude sensitivity test (FAST) [9], the Sobol method [10], and the active subspace method [8] (which is the focus of this thesis).

This project is sponsored by the National Physical Laboratory (NPL), which is the national measurement laboratory of the UK. Given their interest in precision in measurement, it is obvious why NPL is invested and interested in the study and applicability of sensitivity analysis. This is the reason why this thesis is mainly focused on sensitivity analysis.

1.2 Examples of Sensitivity Analysis

In this section we highlight three specific methods of sensitivity analysis which will be important in the context of this thesis. The first one we discuss is the active subspace method, which is the main topic of this thesis.

1.2.1 The Active Subspace Method

The key idea of the active subspace method is that it allows us to choose a linear combination of important parameters instead of analysing all of them individually. This way we are reducing the dimension of the parameter study [8, 14]. The active subspace method will be discussed throughout this thesis (the background theory, algorithm and examples are presented in Chapter 2). The main reference of this thesis is the book “Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies” by Paul G. Constantine [8].

1.2.2 Sobol Indices

The Sobol method is one of the most powerful global sensitivity analysis methods [3, 4, 15]. It relates the contribution of each input variable and the interactions between them to the overall variance in the output of the model. This is achieved by computing first-order, second-order, higher-order and overall sensitivity indices. In other words, given a deterministic function, the variance of the output is explained by the variance in the inputs. This then allows us to determine which individual input parameters are important or which interactions (if any) are important [16, 17, 18].

Since this is a well-known and powerful method, it is no surprise that NPL is currently utilising the technique. The Sobol method will be discussed in more detail in Chapter 5 for comparison with the active subspace method.

1.2.3 Principal Component Analysis

Pricipal Component Analysis (PCA) is a method that uses a vector space transformation to reduce the dimensionality of a given data set. We discuss PCA in more detail here than the first two methods due to the fact that we will not consider this method in later chapters. If the original data set involves many variables, it is possible that the data can be interpreted in terms of just a few variables (i.e., the principal components) [19]. PCA seeks a low-dimensional linear parameterisation of a vector. In other words, the goal of PCA is to simplify the description of the data set and extract the most important information. The main references used here for details of the PCA are [19, 20, 21].

To illustrate the method, we consider a synthetic PCA example here for simplicity. Imagine that we have collected some 3-dimensional data, say, we have asked students how many hours monthly they spent studying MM104 (Statistics and Data Presentation) at the University of Strathclyde, the mark that they received at the end of the academic year, and their age. Let H be the first dimension (monthly hours spent studying), M be the mark received by a student, and A be their age. The data set is detailed in Table 1.1 and plotted in Figure 1.1.

Hours H	6	15	6	5	25	20	16	10	14	0	5	19	18	16	18	7	16	7	6	26	22	18	12	16	3	8	21	23	18	25
Mark M	42	59	52	44	97	89	77	57	65	25	43	70	71	81	85	46	63	58	50	97	89	81	62	69	40	49	75	75	86	90
Age A	18	19	35	25	28	19	37	27	25	20	21	22	21	21	25	19	20	25	22	23	19	26	25	22	21	20	19	21	21	20

Table 1.1. Data set of hours spent studying monthly by each student, mark obtained and their age.

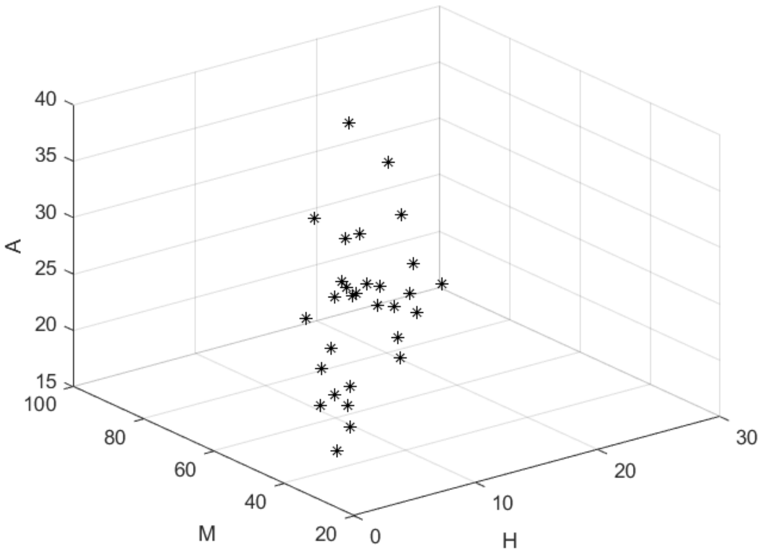


Figure 1.1. A plot of data of hours spent studying monthly, mark obtained and age.

Having defined the data, the standard procedure of the PCA is now to calculate the covariance matrix, which here is a 3 × 3 matrix. For our 3-dimensional case, the covariance matrix is expressed as

$$C = \begin{bmatrix} Cov(H,H) & Cov(H,M) & Cov(H,A) \\ Cov(M,H) & Cov(M,M) & Cov(M,A) \\ Cov(A,H) & Cov(A,M) & Cov(A,A) \end{bmatrix},$$

where

$$Cov(H,M) = \frac{\sum(H_i - \bar{H})(M_i - \bar{M})}{n-1}, Cov(H,A) = \frac{\sum(H_i - \bar{H})(A_i - \bar{A})}{n-1},$$

$$Cov(M,A) = \frac{\sum(M_i - \bar{M})(A_i - \bar{A})}{n-1}, i = 1, \dots, n.$$

Here, $\bar{H}$ is the average of hours spent studying of all students, $\bar{M}$ is the average mark, $\bar{A}$ is the average of student's age and n is the number of parameters (students), i.e., $n = 30$ in this case. The covariance matrix for this example is therefore

$$C = \begin{bmatrix} 53.34 & 130.44 & -1.41 \\ 130.44 & 358.81 & 8.52 \\ -1.41 & 8.52 & 19.91 \end{bmatrix}.$$

Having defined the covariance matrix, the next step of PCA is to obtain eigenvalues and eigenvectors of the covariance matrix. For this example, the eigenvalues are 407.07, 20.93 and 4.06, with corresponding eigenvectors $[-0.3459, -0.9381, -0.0194]^T$, $[0.2526, -0.0732, -0.9648]^T$ and $[0.9037, -0.3386, 0.2622]^T$, respectively. The eigenvalues are ordered from largest to smallest in order to determine the importance of the components. The eigenvector associated with the largest eigenvalue is called the principal component of the data set.

The idea of PCA is that we can choose to ignore the components of lesser significance. We obviously lose some information in the data by doing that, however, if the eigenvalues are small, we don't lose a great deal of information. If we ignore some components in a given problem, then the final data will have fewer dimensions [19]. The purpose is to retain meaningful properties of the original data while we reduce the dimensionality and computational time. This is the main idea of the

PCA method [22].

The last step of the PCA method is to derive a new data set with respect to the principal components or eigenvectors. In this particular example, we have a 3-dimensional problem and so we choose to derive a new data set (i.e., transform the existing data set) with respect to the two eigenvectors that correspond to the largest eigenvalues. Then, we standardise our original data in such a way that $\hat{H}_i = \dfrac{H_i - \bar{H}}{\sigma_H}$, where σ_H is the standard deviation of H, $\hat{M}_i = \dfrac{M_i - \bar{M}}{\sigma_M}$, where σ_M is the standard deviation of M and $\hat{A}_i = \dfrac{A_i - \bar{A}}{\sigma_A}$, where σ_A is the standard deviation of A. This is done so that $\hat{H}_i, \hat{M}_i, \hat{A}_i \sim N(0,1)$. We then combine all standardised vectors of data to form a 30×3 matrix called the Feature Matrix. The idea is then to obtain transformed data with respect to the Feature Matrix and the two leading eigenvectors (positioned colum-wise, in order of importance, to form a 3×2 matrix). By multiplying the Feature Matrix and the leading eigenvectors, we obtain a 30×2 matrix of transformed data. Notice that our original data was 3-dimensional, but the transformed data is now 2-dimensional [19, 22, 23].

The principal components for our example are plotted together with the standardised data in Figure 1.2. We can clearly see that there is not much contribution from the green direction (the third principal component).

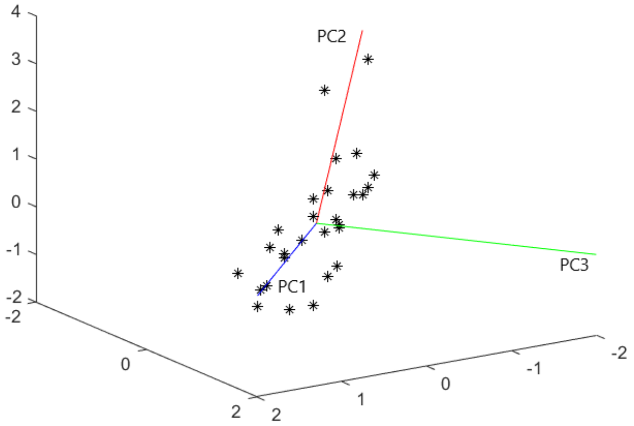


Figure 1.2. A plot of three principal components.

We then drop the inconsequential principal component to obtain a 2-dimensional case with two principal components.This results in PC1 and PC2 becoming the new axes to our data, as shown in Figure 1.3.

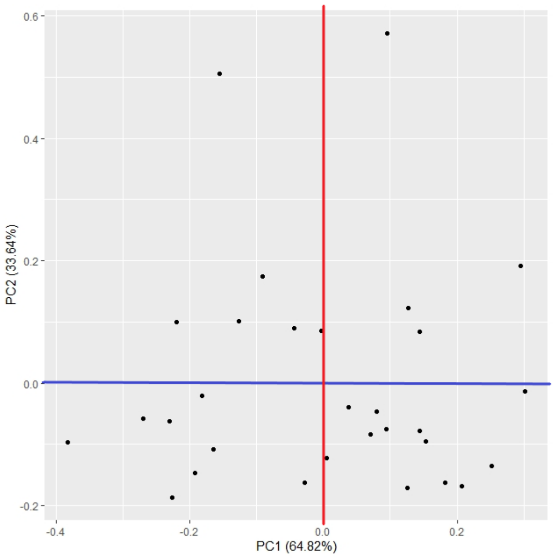


Figure 1.3. A plot of two principal components.

This gives us insight into what the transformed data looks like. Specific computations show that the first principal component explains 64.82% of the variance in the data (the points are clustered around the horizontal axis), and the second principal component explains 33.64% of the variance. This means that the first two principal components explain more than 95% of the variance in the data. More about PCA can be found in [19, 20, 21, 22].

In contrast to PCA, the active subspace method helps to approximate a scalar-valued function of several variables. In order to do that, we use the eigenvectors of an uncentered covariance of the gradient. However, we are not interested in constructing a low-dimensional, linear parameterisation of the gradient such that its covariance is well-approximated, like in PCA. Instead, we use those eigenvectors to identify directions along which the scalar-valued function (non-existent in PCA) changes more, on average [8, 14]. This will be illustrated in Chapter 2.

1.3 Network and Measurement Science

Throughout this thesis, we illustrate the use of the active subspace method motivated by problems in network and measurement science. In order to acquaint the reader with both fields, we give short introductions in this chapter.

1.3.1 Network Science Context

We live in a world where everything around us is connected, for example, information systems (such as the internet), electrical grids and transportation systems. Understanding, controlling or designing of such systems or networks is a huge challenge today. Many different complex systems consisting of thousands or millions of connected elements are studied in disciplines as diverse as sociology, economy, biology, neuroscience and physics [24].

Complex systems can often be usefully represented as networks. Each network component is represented by a node and nodes are linked when components are connected. This way we can study the connected elements of a system. Furthermore, networks are often represented as matrices (see Chapter 3). We can then apply statistical inference techniques to these systems more easily [25, 26]. For example, we can identify central nodes or edges in a given network. This is often achieved by considering a centrality measure [27, 28, 29], which assigns a value to each node to indicate importance or relevance in a given network [30]. We discuss networks and calculations of centralities in more detail in Chapter 3. We then apply the active subspace method on networks in Chapter 4.

1.3.2 Measurement Science Context

Measurement is one of the first human intellectual achievements. It is believed that people learned to measure centuries before they learned how to write. Moreover, people learned how to count through measurement [31]. It is a process of observing and recording objects or events (i.e., comparing objects, recording results). Measurement capabilities have increased drastically since the 1960s, making measurements faster and more precise. Instruments such as laser trackers, which were invented in 1987, are now capable of providing us with the 3-dimensional co-ordinates of an object (i.e., distance, azimuthal angle and elevation) [32, 33].

We discuss measurement science (more specifically, in design of experiments) and how it fits with the active subspace method in more detail in Chapter 6. We also consider laser trackers in more detail below since the problems in Chapter 6 are closely related to problems involving laser tracker measurements. Needless to say, NPL is interested and invested in measurement science and, in particular, laser trackers, which is why we highlight this topic in this thesis.

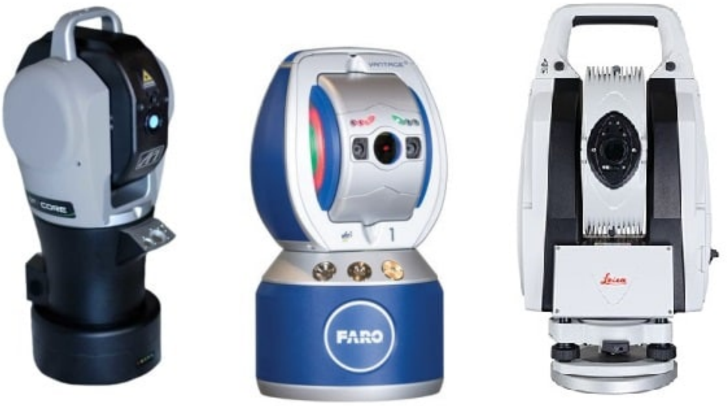


Figure 1.4. Examples of a laser tracker. From left to right: API's OT2 Core, FARO's Vantage, Leica Hexagon AT403 [38]

1.3.2.1 Introduction to Laser Trackers

Industries such as aeronautical engineering and space exploration are in need of large-scale measurement systems [32, 33, 34, 35] for problems where the dimensions and positioning of large parts require accurate measuring devices [35]. It is known that using laser tracker offers precision and relatively fast data acquisition for measuring large objects in industrial environments [34]. The first laser tracker was invented in 1987 by Kam Lau; see Figure 1.4 for examples of laser trackers today [36, 37].

A lot of research and development has been done during the last 30 years in modeling of laser tracker error sources, estimation of uncertainty, improvement of accuracy, design of laser trackers and performance evaluation methods [33, 39]. Because of this, the applicability of laser trackers has increased over time. The laser tracker is now an attractive measurement tool for a number of applications. Typical examples include: robot metrology; manufacture and assembly of large components such as aircraft wings or structures of a ship; development of machine tools; providing reference measurements for large volume laser scanners and distributed metrology systems (such as indoor positioning systems, i.e., indoor GPS); alignment of large optics and structures for astronomy; and the nuclear industry [40, 41]. Furthermore, laser trackers can be used not only for part inspection but also for monitoring the condition of a tool or a manufactured part over time. If a fault is found during the monitoring period, this can prevent other tools or parts from being manufactured that could, in general, lead to faulty equipment (parts might not suit one another if faulty) [38].

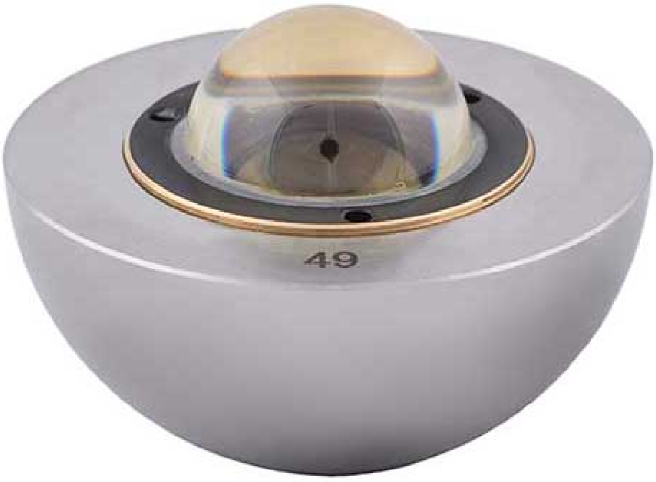


Figure 1.5. An example of a reflector [43]

A laser tracker is a portable coordinate measuring system. It measures a three-dimensional coordinate position of a target such as a spherically mounted retroreflector. In other words, the device records the distance to the target (say, d) along with two angles, providing position data in a spherical coordinate system. With this information, Cartesian (X, Y, Z) coordinates of the reference target can also be calculated [39, 40, 42]. The distance between the laser tracker and the reflector is usually denoted by d, and the inclication angle and azimuthal angle (rotation of the laser tracker) are denoted by θ and φ, respectively. An example of a reflector is presented in Figure 1.5.

The distance d can be measured by an interferometer or an absolute distance measurement technique [34]. An interferometer is a combination of two or more sources of light that create an interference pattern. This can then be measured and analysed. However, there is a huge issue with using an interferometer to calculate d: an interferometer can be used only when the beam is not interrupted (from the laser tracker to the reflector) [38, 44]. For example, the beam may get interrupted when experiments or calculations take place outdoors, or due to space-borne satellites in orbit, etc. However, this issue can be overcome by using the absolute distance measurement, which can provide the distance d even if the beam is interrupted. Examples of absolute distance measurement principles include time-of-flight measurement, multi-wavelength interferometry, wavelength scanning interferometry and intensity or frequency modulation. Although the absolute distance measurement is suitable when the beam gets interrupted, the interferometer

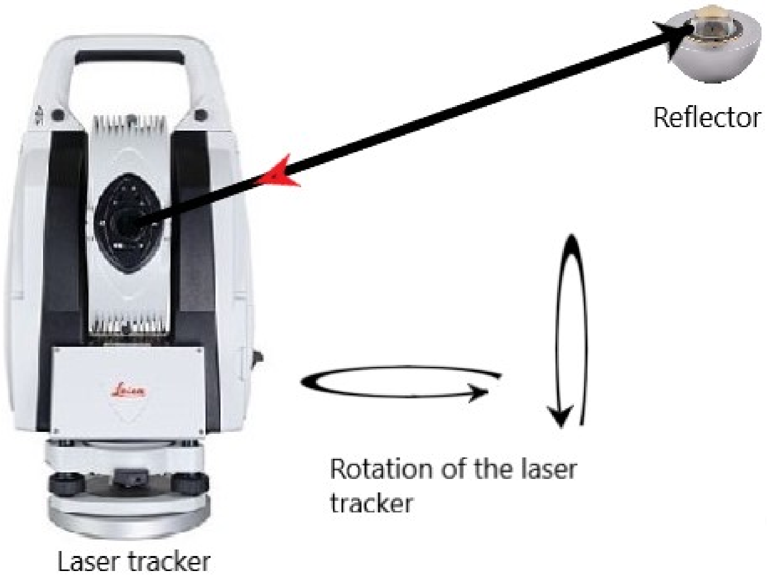


Figure 1.6. The working principle of the laser tracker.

measurement is faster with more accurate measurements of distance d [38, 44].

The inclination angle θ and azimuthal angle φ are measured by two angular encoders [34, 42]. Figure 1.6 displays the basic functionality idea of the laser tracker. The arrow from the laser tracker to the reflector denotes the beam with distance d. The basic idea is that the laser tracker sends out the laser beam to the reflector, then the reflector returns the laser beam. If the beam is detected to be off-course, then this causes movement of the laser tracker. This is done so that the beam is incident on or perpendicular to the optical center of the reflector [34], which means that the laser tracker continually monitors the position of the reflector.

However, there are some limitations to the laser tracker [41]. For example, air temperature, pressure and humidity can affect measurements. Also, the laser tracker is an expensive, relatively large and heavy instrument, which can make it unsuitable for some applications.

Consider the simple example displayed in Figure 1.7 so that we can better understand the capabilities of a laser tracker. We assume here a few things for the hypothetical aircraft in Figure 1.7: the aircraft is stationary, the parts of the aircraft are manufactured (e.g., the parts of the aircraft are denoted by the green lines are intersections in Figure 1.7) but have yet to be put together or built. In a case like this, the laser trackers are mostly used to inspect the parts of the aircraft (for things like alignment, stability and portability).

We now assume that the aircraft is put together. One of the main tasks for the engineering team is to inspect the aircraft for all instabilities, misalignments of parts and the symmetry of the aircraft. This can also be done by using laser trackers (they are especially used for inspecting the wings of the aircraft). It is a must for an aircraft to be built with as little error as possible (in the interests of important qualities such as stability and balance). Note that multiple laser trackers are usually combined to get a single target location in the aircraft industry (although the same concept applies to many industries or applications). The reason for doing this is that a single laser tracker can have errors and using multiple laser trackers to target a single target location can reduce that error.

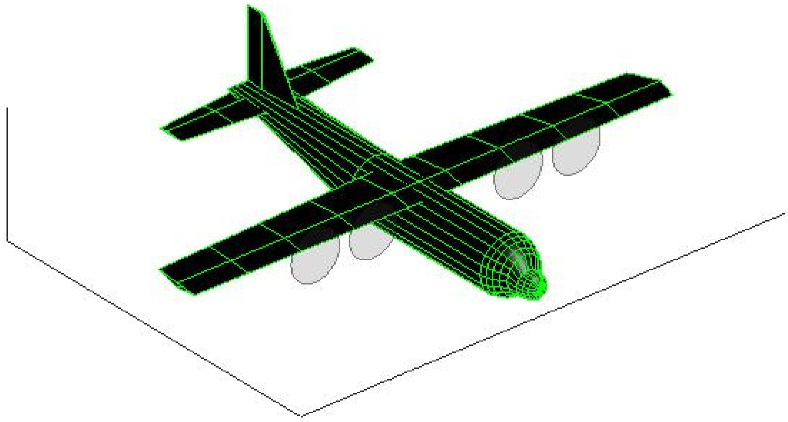


Figure 1.7. An example of an aircraft with target locations to be determined by the laser tracker.

1.4 Summary of Thesis

Chapter 2 contains the main mathematical background material for this thesis, where we introduce the method of the active subspace with theory, a practical algorithm and examples. In Chapter 3 we discuss the background theory of networks.

In particular, we present a new analytical derivation for the gradient of Katz centrality in § 3.3 (Lemma 3). In Chapter 4, we apply the theory of networks and the ideas of the active subspace method, and perform a number of experiments. It is important to note that, as far as we are aware, this is the first time the active subspace method has been used in network science. After illustrating some ideas on synthetic networks, we consider two important more realistic networks in § 4.4 and § 4.5, which have been previously studied by network science researchers, namely, Zachary's Karate Club network and a protein-protein interaction (PPI) network. The results we obtained using the active subspace method generally match those published in the research literature and also add further insights. In Chapter 5, we consider Sobol indices and make comparisons with the active subspace method. Despite the fact that the Sobol method is far more popular, the active subspace method is found to offer some advantages. In Chapter 6, we introduce the idea of design of experiments, and apply the active subspace method to some examples that are of interest to NPL. This includes a "black-box" example, where the experimental processs is partly known. The input parameters and output quantity of interest are supplied but unknown. The thesis is then concluded with a summary of the work presented and some remarks on possible future work.

Chapter 2

The Method of Active Subspaces

2.1 Introduction

As we have seen in Chapter 1, for a complicated physical model that has multiple input parameters, standard methods (e.g., uncertainty quantification, optimisation, integration, etc) are usually not practical for parameter studies due to the limited computational budget. The active subspace method offers an appealing approach to deal with this problem. In other words, if the underlying model is computationally expensive, the method of active subspaces can be used to deal with the curse of dimensionality. The method helps us to understand the importance of parameters in a given problem (in similar way to the PCA method). The active subspace method discovers a **linear combination** (or combinations) of important parameters instead of just identifying how important each individual parameter is to the model. These linear combinations of parameters are usually referred to as a set of important directions of the inputs. Once a set of important directions has been identified, it is then possible to use that information to perform parameter studies (e.g., response surfaces, inversion) based on a smaller number of parameters. The main reference for this chapter is [8].

There are a number of points that need to be considered in order to use an active subspace method. In this chapter we illustrate how to generate an active subspace for a general problem. To set notation, let $\mathbf{x}$ represent a vector of input parameters (or model parameters) sampled from a distribution with a probability density function, say ρ, and a function $f : \mathbb{R}^m \to \mathbb{R}$ be a map from (normalised) inputs to the simulation's quantity of interest. Consequently, $f(\mathbf{x})$ is a **scalar** quantity that depends on the input parameters $\mathbf{x}$. Furthermore, input parameters need to be normalised, i.e. centred around 0 with equal ranges. It is important to note that we will assume a certain degree of smoothness for f in this thesis (notably that its gradient exists).

2.2 Background Theory

Suppose we have an $m \times 1$ vector $\mathbf{x}$ containing the m input parameters, and a function f as defined in § 2.1, so that $f(\mathbf{x})$ is a scalar. We begin by introducing a matrix $\mathbf{C}$, which is the mean (with respect to the probability density function ρ [45]) of the outer product of the gradient of the function f, denoted by $\nabla_x f$, with itself, that is,

$$\mathbf{C} = \int (\nabla_x f)(\nabla_x f)^T \rho d\mathbf{x}. \tag{2.1}$$

The density function is usually chosen to be uniform or Gaussian, because these density functions are easy to implement and work with. Each entry of $\mathbf{C}$ is therefore the average of a product of partial derivatives

$$\mathbf{C}_{ij} = \int \left(\frac{\partial f}{\partial x_i}\right)\left(\frac{\partial f}{\partial x_j}\right) \rho d\mathbf{x},$$

where $\mathbf{C}_{ij}$ is the (i, j) element of $\mathbf{C}$ and x_i is the i^{th} entry of the vector $\mathbf{x}$. For now, we assume that partial derivatives of f are easily available (that is, gradients or approximation of the gradients). Here $\mathbf{C}$ is an $m \times m$ matrix. Furthermore, the matrix $\mathbf{C}$ is symmetric and positive semi-definite [8, Equation (3.4)], which implies that it has a real eigenvalue decomposition. We may therefore write

$$\mathbf{C} = \mathbf{W}\boldsymbol{\Lambda}\mathbf{W}^T, \qquad \Lambda = diag(\lambda_1, ..., \lambda_m), \tag{2.2}$$

where $\mathbf{W}$ is the $m \times m$ orthogonal matrix whose columns $\{\mathbf{w}_1, ..., \mathbf{w}_m\}$ are the eigenvectors of $\mathbf{C}$. We will assume without loss of generality that $\lambda_1 \geq ... \geq \lambda_m \geq 0$.

We now give a lemma that quantifies the relationship between the gradient of f and the eigenvalue decomposition of matrix $\mathbf{C}$ ([8], pp. 23).

Lemma 2.2.1. *The mean-squared directional derivative of f with respect to the eigenvector $\mathbf{w}_i$ is equal to the corresponding eigenvalue, that is,*

$$\int ((\nabla_x f)^T \mathbf{w}_i)^2 \rho d\mathbf{x} = \lambda_i, \qquad i = 1, ..., m.$$

Proof. Since the matrix $\mathbf{C}$ has the real eigenvalue decomposition given in (2.2),

$$\lambda_i = \mathbf{w}_i^T\mathbf{C}\mathbf{w}_i = \mathbf{w}_i^T\left(\int(\nabla_{\mathbf{x}}f)(\nabla_{\mathbf{x}}f)^T\rho d\mathbf{x}\right)\mathbf{w}_i = \int((\nabla_{\mathbf{x}}f)^T\mathbf{w}_i)^2\rho d\mathbf{x}.$$

□

By inspecting Lemma 2.2.1, we see that on average the change in f along eigenvector $\mathbf{w}_i$ is proportional to the size of eigenvalue λ_i. For example, if the smallest eigenvalue λ_m is zero, then the change in f along the eigenvector $\mathbf{w}_m$ is zero, which means that the directional derivative $(\nabla_x f)^T\mathbf{w}_m$ is zero.

Based on this information, we now separate the eigenvalues and eigenvectors of $\mathbf{C}$ into two sets, where perturbations in the first set change f more compared with perturbations in the second set, on average. That is, we write

$$\Lambda = \begin{bmatrix}\Lambda_1 & \\ & \Lambda_2\end{bmatrix}, \qquad \mathbf{W} = [\mathbf{W}_1 \quad \mathbf{W}_2],$$

where $\Lambda_1 = diag(\lambda_1, ..., \lambda_n)$ with the first $n < m$ eigenvalues and $\mathbf{W}_1$ contains the first n eigenvectors. From Lemma 2.2.1, the first n eigenvalues and eigenvectors (in $\Lambda_1, \mathbf{W}_1$) should have more impact on f than the remaining $m-n$ eigenvalues and eigenvectors (in $\Lambda_2, \mathbf{W}_2$). Strategies for choosing n will be discussed in § 2.5. By separating eigenvectors and eigenvalues in this way, we can define new variables $\mathbf{y}$ and $\mathbf{z}$ as

$$\mathbf{y} = \mathbf{W}_1^T\mathbf{x} \in \mathbb{R}^n, \qquad \mathbf{z} = \mathbf{W}_2^T\mathbf{x} \in \mathbb{R}^{m-n},$$

where changes in the $\mathbf{y}$ variables impact f more on average than changes in $\mathbf{z}$ variables. Note that $\mathbf{y}$ and $\mathbf{z}$ are linear combinations of the original parameters in $\mathbf{x}$. It is also worth noting that $\mathbf{x}$ can be expressed directly in terms of $\mathbf{y}$ and $\mathbf{z}$ as

$$\mathbf{x} = I\mathbf{x} = \mathbf{W}\mathbf{W}^T\mathbf{x} = \mathbf{W}_1\mathbf{W}_1^T\mathbf{x} + \mathbf{W}_2\mathbf{W}_2^T\mathbf{x} = \mathbf{W}_1\mathbf{y} + \mathbf{W}_2\mathbf{z},$$

$$\Rightarrow f(\mathbf{x}) = f(\mathbf{W}_1\mathbf{y} + \mathbf{W}_2\mathbf{z}),$$

where I is the identity matrix. By using the chain rule,

$$\nabla_{\mathbf{y}}f(\mathbf{x}) = \nabla_{\mathbf{y}}f(\mathbf{W}_1\mathbf{y} + \mathbf{W}_2\mathbf{z}) = \mathbf{W}_1^T\nabla_{\mathbf{x}}f(\mathbf{W}_1\mathbf{y} + \mathbf{W}_2\mathbf{z}) = \mathbf{W}_1^T\nabla_{\mathbf{x}}f(\mathbf{x}),$$

$$\nabla_{\mathbf{z}}f(\mathbf{x}) = \nabla_{\mathbf{z}}f(\mathbf{W}_1\mathbf{y} + \mathbf{W}_2\mathbf{z}) = \mathbf{W}_2^T\nabla_{\mathbf{x}}f(\mathbf{W}_1\mathbf{y} + \mathbf{W}_2\mathbf{z}) = \mathbf{W}_2^T\nabla_{\mathbf{x}}f(\mathbf{x}).$$

We can see that the active subspace is defined to be the range of the eigenvectors in $\mathbf{W}_1$ and the inactive subspace is defined to be the range of eigenvectors in $\mathbf{W}_2$. The equalities above tell us that the changes in $\mathbf{y}$ variables are associated with the active subspace (the span of the eigenvectors in $\mathbf{W}_1$). The changes in $\mathbf{z}$ variables are associated with the inactive subspace (the span of the eigenvectors in $\mathbf{W}_2$). Having this information, the result in Lemma 2.2.1 with respect to the new variables $\mathbf{y}$ and $\mathbf{z}$ can be shown as follows ([8], pp. 24).

Lemma 2.2.2. *The mean-squared gradients of f with respect to $\mathbf{y}$ and $\mathbf{z}$ satisfy*

$$\int (\nabla_y f)^T (\nabla_y f) \rho d\mathbf{x} = \lambda_1 + ... + \lambda_n,$$

$$\int (\nabla_z f)^T (\nabla_z f) \rho d\mathbf{x} = \lambda_{n+1} + ... + \lambda_m.$$

Proof. Using the linearity of the trace,

$$\begin{aligned}\int (\nabla_y f)^T (\nabla_y f) \rho d\mathbf{x} &= \int trace\left((\nabla_y f)(\nabla_y f)^T\right) \rho d\mathbf{x} \\ &= trace\left(\int (\nabla_y f)(\nabla_y f)^T \rho d\mathbf{x}\right) \\ &= trace\left(\mathbf{W}_1^T \left(\int (\nabla_y f)(\nabla_y f)^T \rho d\mathbf{x}\right) \mathbf{W}_1\right) \\ &= trace\left(\mathbf{W}_1^T \mathbf{C} \mathbf{W}_1\right) \\ &= trace\left(\Lambda_1\right) \\ &= \lambda_1 + \ldots + \lambda_n.\end{aligned}$$

The result for the $\mathbf{z}$ components can be derived in a similar way. □

Lemma 2.2.2 tells us that perturbations in the active variables $\mathbf{y}$ change f more than perturbations in inactive variables $\mathbf{z}$, on average. This is a similar observation to Lemma 2.2.1 and it is a particularly important remark for the active subspace method. From Lemma 2.2.2, we can see that the sum of eigenvalues which correspond to $\mathbf{W}_2$ is related to changes in $\mathbf{z}$. Moreover, the sum of eigenvalues which correspond to $\mathbf{W}_1$ is related to changes in $\mathbf{y}$. Note that, if $\lambda_{n+1} = ... = \lambda_m = 0$, the gradient with respect to $\mathbf{z}$ is zero everywhere. Studying $\lambda_1, ..., \lambda_n$ instead of $\lambda_1, ..., \lambda_m$ is the essential idea of the active subspaces method.

The main aim of the active subspace method is to find a linear combination of parameters that can be used to approximate f by a function of $n < m$ variables in order to enable parameter studies. In the context of the active subspace method, a linear combination of parameters gives the variables $\mathbf{y} = \mathbf{W}_1^T\mathbf{x}$. As mentioned before, this linear combination of parameters can be used to plot a sufficient summary plot (of the quantity of interest $f(\mathbf{x})$ against the active variable $\mathbf{y}$). The purpose of plotting a sufficient summary plot is to see if there exists a low-dimensional structure in the original problem.

In § 2.3, we discuss a practical algorithm to obtain an active subspace. But before we do that, there is a very important matter of the active subspace method which needs to be discussed. We have already considered the fact that given a smooth function f and a vector of input parameters $\mathbf{x}$, the calculation of the quantity of interest $f(\mathbf{x})$ and $\nabla_{\mathbf{x}} f(\mathbf{x})$ is straightforward. However, the calculation of these quantities are usually expensive. To perform a parameter study (e.g. response surfaces, integration, etc), it is essential to have as few calls of the simulation as possible. This is the reason we are interested in the active subspace method. The method produces a structure, which can be exploited efficiently in the parameter studies. By reducing the dimension of the underlying problem, we can focus on only the most important parameters, therefore saving computational time. In other words, we focus on the active variables to get faster results. In order to do that we may construct a function $h = h(\mathbf{y})$, which depends on the $n < m$ (components of $\mathbf{y}$) so that

$$f(\mathbf{x}) \approx h(\mathbf{y}) = h(\mathbf{W}_1^T\mathbf{x}), \tag{2.3}$$

where $\mathbf{W}_1^T\mathbf{x}$ is a linear combination of the important eigenvectors in $\mathbf{W}_1$ (columns of $\mathbf{W}_1^T$) and the input parameters $\mathbf{x}$. Thus, we need to construct a model so that h approximately represents f but with fewer simulation calls. In this thesis, we focus our attention on the calculation of the active subspace and its application. A discussion of exploitation of the active subspace (building a model so that parameter studies are feasible) can be found in [8, Chapter 4].

2.3 Practical Algorithm

The previous section of this chapter described the theory behind generating an active subspace. In this section we consider implementing these ideas to obtain a practical algorithm. It is based on constructing an approximation to the matrix $\mathbf{C}$ in (2.1), which lies at the core of defining an active subspace. The general algorithm is presented in Algorithm 1.

Step 1: Draw M independent samples $\{\mathbf{x}_i\}, i = 1, \ldots, M$ according to the sampling density ρ.

Step 2: For each sample $\mathbf{x}_i$, compute the gradient $\nabla_x f_i = \nabla f(\mathbf{x}_i)$ and the quantity of interest $f(\mathbf{x}_i)$.

Step 3: Compute the matrix $\hat{\mathbf{C}}$ and its associated eigenvalue decomposition,

$$\hat{\mathbf{C}} = \frac{1}{M} \sum_{i=1}^{M} \nabla_x f_i \nabla_x f_i^T = \hat{\mathbf{W}} \hat{\mathbf{\Lambda}} \hat{\mathbf{W}}^T, \tag{2.4}$$

where $\hat{\mathbf{W}}$ is the matrix of eigenvectors, and $\hat{\Lambda} = diag(\hat{\lambda}_1, ...\hat{\lambda}_m)$ is the diagonal matrix of eigenvalues ordered in decreasing manner.

Algorithm 1. The general algorithm for calculating active subspaces.

The $m \times m$ matrix $\hat{\mathbf{C}}$ is a Monte Carlo approximation to the matrix $\mathbf{C}$ in (2.1), which is the average of the outer product of the gradient with itself. Step 1 of the algorithm focuses on drawing M independent samples. The choice of M will affect the accuracy of the approximation: clearly, taking M large will improve the approximation quality, but it will make the computation more expensive. The recommended value in [8] is $M = ak \log m$, where a is the oversampling factor, k is the largest dimension we are interested in and m is the number of input parameters. It is possible to fit a linear model with $m + 1$ evaluations of the quantity of interest. However, random sampling algorithms in the literature suggest to use an oversampling factor of $a \in [2, 10]$. The largest dimension k corresponds to the fact that we can inspect k eigenvalues instead of m. Suppose that before we apply the active subspace method to a problem, we suspect that we have a 3-dimensional active subspace, and so you would choose $k = n + 1 = 4$. This way you can save computational time, however, if you are wrong about your prediction, you will need to increase $k \in [1, m + 1]$.

Imagine having a large computational budget, then the model can be run without performing a dimension reduction technique. However, this is rarely the case when working on real-world problems. With a limited computational budget, it is usually essential to decide on M in order to obtain efficient results. The number of Monte Carlo samples M is usually chosen to be large (e.g., 10^4) [46]. The reason for choosing M much larger than $ak \log m$ is that the numerical examples we are studying are computationally inexpensive and we wish to evaluate the method in the absence of this extra source of error.

The focal point of Step 2 of the algorithm is computing the gradient $\nabla f(\mathbf{x}_i)$ and the quantity of interest $f(\mathbf{x}_i)$ for each sample $\mathbf{x}_i$. Note here that $f(\mathbf{x}_i)$ is necessary here only for calculating $\nabla f(\mathbf{x}_i)$. Having this information, we can compute gradient samples $\nabla_{\mathbf{x}} f(\mathbf{x})$. Gradient samples simply depend on the input parameters $\mathbf{x}$, a smooth function $f(\mathbf{x})$, the number of samples and the density of sampling (described earlier). However, gradient samples are not always available. To overcome this problem, where necessary we use finite differences in this thesis to obtain an approximation of the gradient. Step 3 focuses on the computation of the matrix $\hat{\mathbf{C}}$ and its eigenvalue decomposition. In § 2.5, we will use examples to show how the algorithm works.

It is important to note that analysing $n < m$ eigenvalues and eigenvectors will cause subspace errors to occur. The source of that error will be discussed in § 2.5. Assume that we have true eigenvalues and eigenvectors obtained from (2.1), and that we have estimated eigenvalues and eigenvectors obtained from (2.4). Essentially, the subspace error is found by comparing $\mathbf{W}$ with $\hat{\mathbf{W}}$. In other words, we can quantify this error by finding the distance between the subspace defined by the range of $\mathbf{W}_1$ (true) and the subspace defined by the range of $\hat{\mathbf{W}}_1$ (estimated). The subspace error formula is

$$dist(ran(\mathbf{W}_1), ran(\hat{\mathbf{W}}_1)) = \|\mathbf{W}_1\hat{\mathbf{W}}_1^T - \hat{\mathbf{W}}_1\hat{\mathbf{W}}_1^T\| = \|\mathbf{W}_1^T\hat{\mathbf{W}}_2\|, \tag{2.5}$$

where $ran(X)$ is shorthand for the range of the columns of the matrix X, $\hat{\mathbf{W}}_2$ is the estimated inactive subspace of the eigenvectors as defined in § 2.2 and the L^2 norm is used as in [8, 47]. An example of the subspace error is presented in § 2.5.2.

2.4 Monte Carlo and Bootstrapping Methods

Two techniques which are very commonly used for large problems involving lots of data are Monte Carlo and bootstrapping methods.

The Monte Carlo method is a numerical method that uses random samples to tackle integration problems for which an analytical solution is not known or is not practical to compute. Although it is known that the simulation of random samples is computationally expensive, as technology has advanced, the Monte Carlo method has become more attractive [48]. The Monte Carlo method is a simulation technique that mainly relies on two things: repeated random sampling and statistical analysis. The first key ingredient is a probability distribution ρ (or probability distributions), which is used as the basis distribution for the input parameters $\mathbf{x}$. Random samples $\mathbf{x}_i, i = 1, \ldots, M$, are then drawn from the probability distribution ρ. For each $\mathbf{x}_i$, we calculate the quantity of interest $f(\mathbf{x}_i)$. Each output or a quantity of interest is a single outcome scenario in the simulation. In a way, we are approximating the quantity of interest $f(\mathbf{x})$. The output values are then collected in order to perform a certain statistical analysis [49]. The three main steps of the Monte Carlo method are summarised in Figure 2.2.

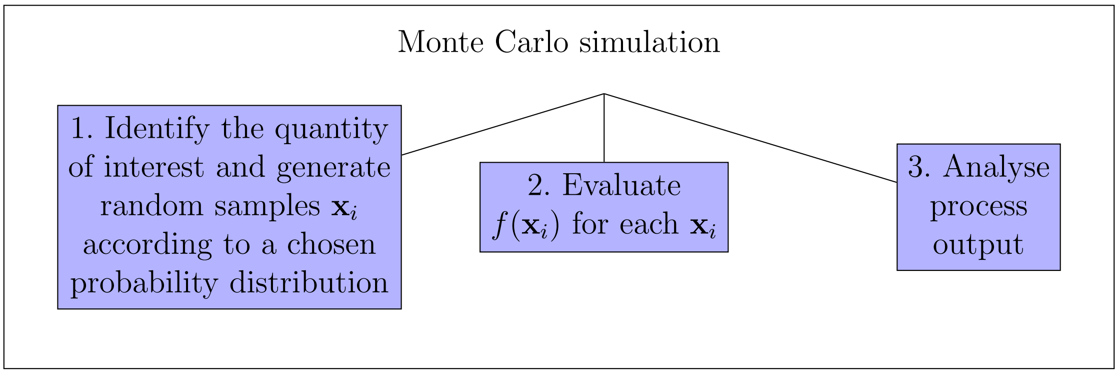

Monte Carlo simulation
1. Identify the quantity of interest and generate random samples $\mathbf{x}_i$ according to a chosen probability distribution
2. Evaluate $f(\mathbf{x}_i)$ for each $\mathbf{x}_i$
3. Analyse process output

Figure 2.1. The Monte Carlo simulation: basic process.

Suppose we have a random variable $\mathbf{x}$ with unknown expected value $\mathbb{E}(\mathbf{x}) = a$ and variance $Var(\mathbf{x}) = b$. We are interested in computing an approximation to a (and possibly b) by taking independent samples of $\mathbf{x}$ and using a pseudo-random number generator. If $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M$ are independent random variables with the same distribution as $\mathbf{x}$, then we expect

$$a_M = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i$$

to be a good approximation to a. The variance of x is approximated by the sample variance

$$b_M^2 = \frac{1}{M-1}\sum_{i=1}^{M}(\mathbf{x}_i - a_M)^2.$$

This leads us to the basic Monte Carlo method for approximating a. We compute M independent samples and form a_M. In order to monitor the error, we also compute the variance approximation b_M^2. Having b_M allows us to compute the 95% confidence interval [50]:

$$\left[a_M - \frac{1.96 \times b_M}{\sqrt{M}}, a_M + \frac{1.96 \times b_M}{\sqrt{M}}\right].$$

In Figure 2.2 we consider a result from a Monte Carlo simulation of $\mathbb{E}(e^{\mathbb{Z}})$, where $\mathbb{Z} \sim N(0,1)$. Analytically, this results in

$$\begin{aligned}
\mathbb{E}(e^{\mathbb{Z}}) &= \int_{-\infty}^{\infty} e^z \frac{1}{2\pi} e^{-z^2/2} dz \\
&= \int_{-\infty}^{\infty} \frac{1}{2\pi} e^{-(z^2-2z)/2} dz \\
&= \int_{-\infty}^{\infty} \frac{1}{2\pi} e^{-(z^2-2z+1)/2} e^{1/2} dz \\
&= e^{1/2} \int_{-\infty}^{\infty} \underbrace{\frac{1}{2\pi} e^{-(z-1)^2/2}}_{N(-1,1)} dz \\
&= e^{1/2} \\
&= \sqrt{e}.
\end{aligned}$$

We used 15 different sample sizes, $M = 2^3, 2^4, 2^5, 2^6, \ldots, 2^{17}$. We then computed a_M and the corresponding 95% confidence interval for each sample. By inspection, we see that as M increases the computed mean becomes more accurate and the confidence interval shrinks.

The accuracy of a Monte Carlo simulation is not deterministic. To deal with this issue, we can post-process the Monte Carlo results using a bootstrapping method. The main property of bootstrapping is that, unlike Monte Carlo, the method does not require information about the distribution of input parameters.

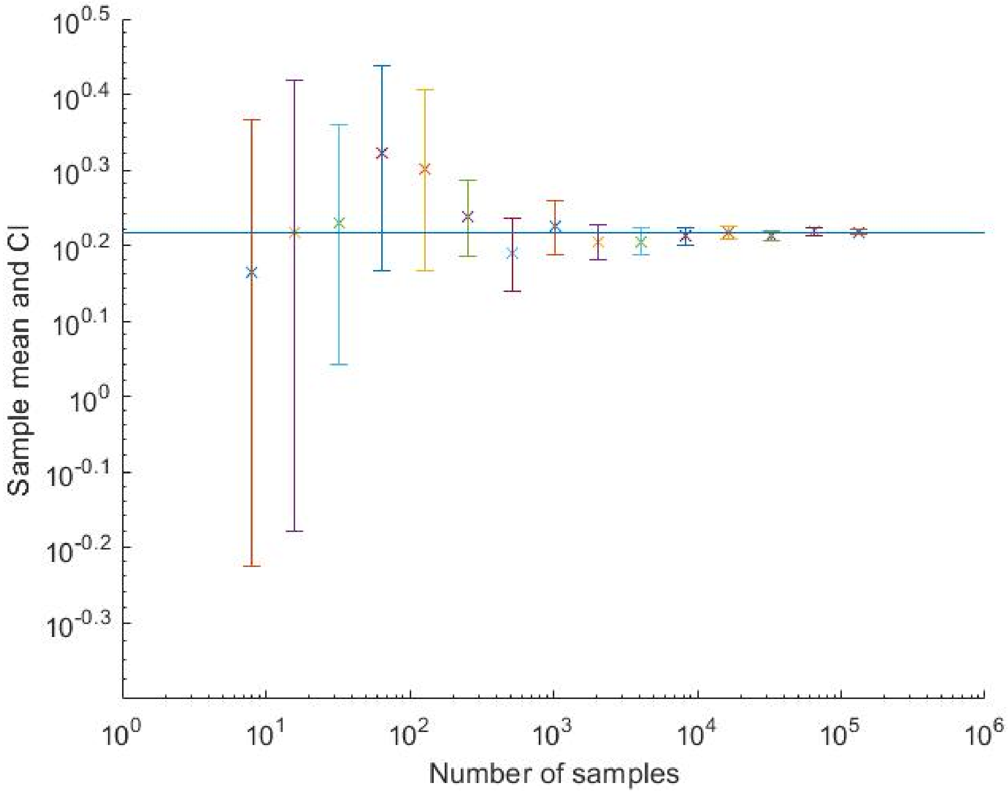


Figure 2.2. Monte Carlo approximation to $\mathbb{E}(e^{\mathbb{Z}})$, where $\mathbb{Z} \sim N(0,1)$. Crosses are the approximations, vertical line are 95% confidence intervals. Horizontal line is at height $\sqrt{e}$.

The main idea of bootstrapping is that it samples with replacement and extracts more information from a fixed data set. In other words, we draw a large number of resamples $\mathbf{x}_i$ from the original sample $\mathbf{x}$ (usually between 10^2 to 10^4 resamples) with replacement. For example, consider a sample $\mathbf{x}$, where

$$\mathbf{x} = (x_1, x_2, x_3, \ldots, x_m).$$

The main step of the bootstrapping method is to generate a bootstrap sample,

$$\mathbf{x}^b = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_b).$$

Bootstrap samples in $\mathbf{x}^b$ have the same dimension or length as $\mathbf{x}$ and bootstrap samples in $\mathbf{x}^b$ are randomly drawn from $\mathbf{x}$. Moreover, a bootstrap sample could include original input parameters more than once and some of the input parameters might not be included. For example, consider $\mathbf{x} = (1,2,3,4,5,6,7,8,9,10)$, then one of the bootstrap samples could be $\mathbf{x}_1 = (1,1,1,5,4,9,9,10,3,2)$. Having this information, we can calculate $f(\mathbf{x}^b)$ for each bootstrap sample in $\mathbf{x}^b$. Since the bootstrap samples in $\mathbf{x}^b$ vary slightly from $\mathbf{x}$, the statistic or a quantity of interest $f(\mathbf{x}^b)$ will also vary [51]. Note that the Monte Carlo method can provide confidence intervals without bootstrapping. However, the method of bootstrapping gives us a better efficiency for a given set of data (see Chapter 3). For more details about bootstrapping, review a short but very informative paper [52].

2.5 Numerical Examples

Following [8], we present two numerical examples in this section to illustrate key ideas of the active subspace method. The first example considers a comparison between the active subspace method from § 2.3 which needs gradients, and an active subspace method using a linear model. This gives a deeper understanding of the active subspace method and the approximation algorithm. The second example considers a quadratic model, where eigenvalues and eigenvectors are known analytically, so some error analysis can be carried out.

2.5.1 Comparison with a Global Linear Model

Consider the smooth function

$$f(x_1, x_2) = \sin(4\pi x_1) + 4x_2,$$

with input parameters $\mathbf{x} = (x_1, x_2)^T$, gradients $\nabla f(x_1) = 4\pi \cos(4\pi x_1)$ and $\nabla f(x_2) = 4$. The first task of the method is to choose the number of samples and the density function. For this particular example, we use $M = 208$ samples as in [8] that are uniformly distributed between -1 and 1. For the active subspace method as described in § 2.3, we follow the steps in Algorithm 1 and calculate gradient samples $\nabla f(\mathbf{x}_i)$ and the quantities of interest $f(\mathbf{x}_i)$ for each sample $\mathbf{x}_i$, $i = 1, \ldots, M$, and compute an eigenvalue decomposition to find estimated eigenvalues and eigenvectors.

We now compare this method with an active subspace method which uses a linear model. For a global linear model, we have the same number of samples, sampling density and the same function. However, in this case we compute only the quantity of interest $q_i = f(\mathbf{x}_i)$ for each sample $\mathbf{x}_i$ instead of also computing sample gradients. We then fit a global linear model

$$q_i = c + \mathbf{b}^T \mathbf{x}_i, \quad i = 1, ..., M, \tag{2.6}$$

using a least squares method to compute c and $\mathbf{b}$. The method of least squares is a standard approach in regression analysis to approximate solution of overdetermined systems [53, 54]. The normalised gradient is then

$$\hat{\mathbf{w}} = \frac{\mathbf{b}}{\|\mathbf{b}\|}. \tag{2.7}$$

It is worth noting that this least squares approach gives us only one direction, which is $\hat{\mathbf{w}}$. The reason why this is important will become clear later.

Since we have the leading eigenvector, $\hat{\mathbf{w}}$ (the direction $\hat{\mathbf{w}}$ for the second example), random samples $\mathbf{x}_i$ and quantities of interest q_i for both cases, we can now construct sufficient summary plots. Figure 2.3 shows sufficient summary plots for the active subspace estimation with gradients (left) and for the active subspace estimation with a global linear model (right). The first plot was produced by using the first (leading) eigenvector $\mathbf{w}_1$ and the second plot was produced by using the direction $\hat{\mathbf{w}}$ in (2.7). By looking at the plots it seems that the linear model identifies the dimension reduction space better than the method with gradients (one plot has scattered points whereas the other has a pattern). The gradient-based method fails here because we have a low-dimensional monotonic function,

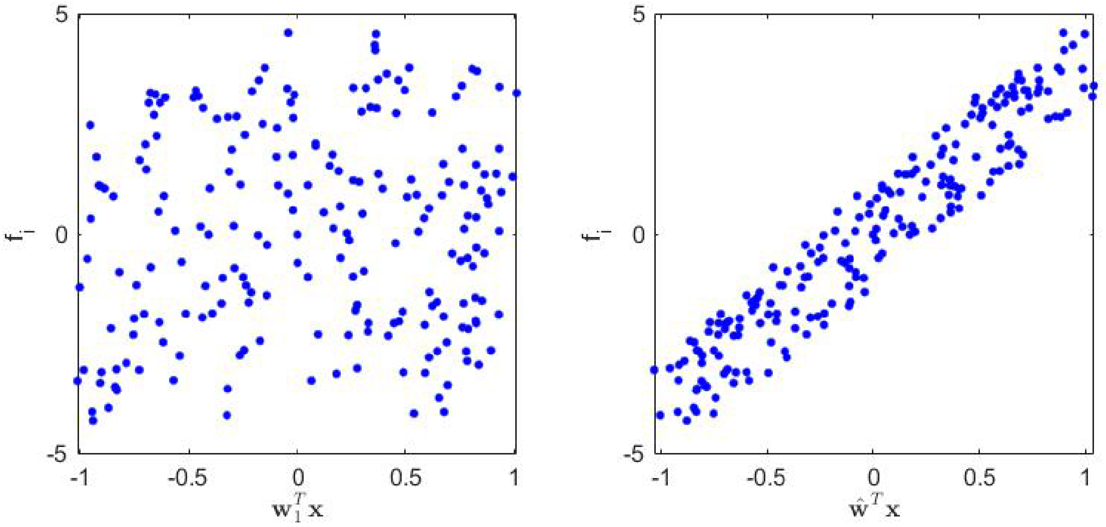


Figure 2.3. Left: Sufficient summary plot for the active subspace estimation with gradients for the function $f(x_1, x_2) = \sin(4\pi x_1) + 4x_2$. Right: Sufficient summary plot for the active subspace estimation with a global linear model.

i.e., fails to identify the underlying behaviour of the function. If the function is not monotonic, that is, if the sign of the partial derivatives changes over the input space, then the global linear model is not able to determine the dimension reduction space [47]. We illustrate this by using the same method above with a different function that is not monotonic:

$$f(x_1, x_2) = \frac{1}{2}(0.7x_1 + 0.3x_2)^2.$$

All the calculations are done in the same way as described earlier and the sufficient summary plots for this case are presented in Figure 2.4. The plot shows that the global linear model for the function $f(x_1, x_2) = \frac{1}{2}(0.7x_1 + 0.3x_2)^2$ fails to identify the dimension reduction space (a scatter plot) whereas the gradient-based method appears to identify the dimension reduction space. In other words, the active subspace method identified the underlying behaviour of the function (quadratic shape). Constantine in [8] also mentions that many 'engineering quantities of interest' are mononotic and that he has not yet encountered non-monotonic 'engineering quantities of interest' cases in practice.

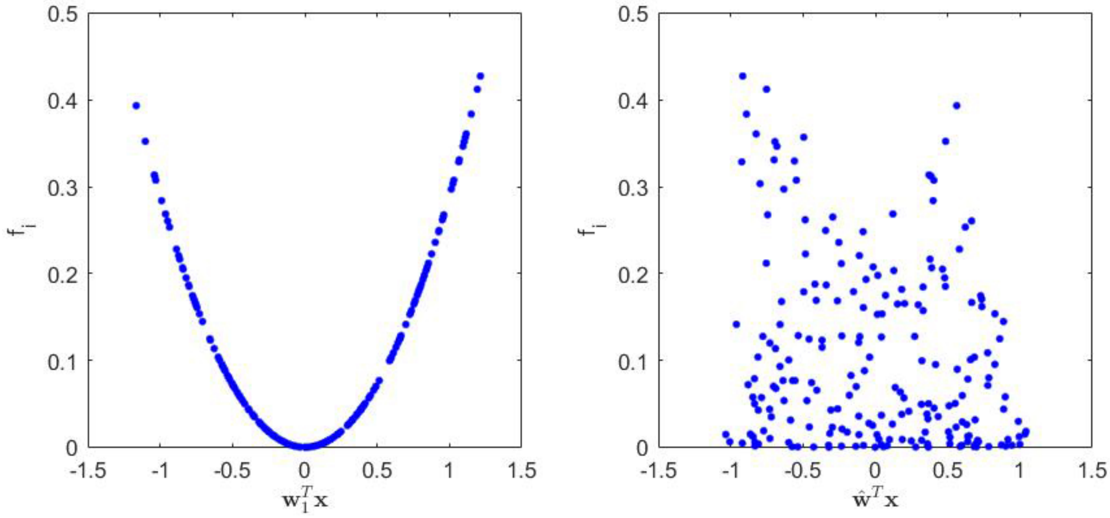


Figure 2.4. Left: Sufficient summary plot for the active subspace estimation with gradients for the function $f(x_1, x_2) = \frac{1}{2}(0.7x_1 + 0.3x_2)^2$. Right: Sufficient summary plot for the active subspace estimation with a global linear model.

2.5.2 A Quadratic Model

We now consider an example where we know true eigenvalues and eigenvectors. Here we first define the function $f(\mathbf{x})$ to be a smooth quadratic function in 10 variables as

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x},$$

where $\mathbf{A}$ is a 10×10 symmetric positive definite matrix, $\mathbf{x}$ is a 10×1 vector and the entries of $\mathbf{x}$ are uniformly distributed on the interval $(0, 1)$. Note that for this example, we can calculate the gradient directly as $\nabla_x f(\mathbf{x}) = \mathbf{A}\mathbf{x}$. We now choose the number of samples and the density function. In this case we will work again with a uniform distribution and $M = 28$ samples of 10 uniformly distributed numbers between -1 and 1. The number of independent samples is $M = ak\log(m) = 28$, where $a = 2$, $k = 6$ and $m = 10$ as in [8]. By using Equation (2.1), we get that

$$\mathbf{C} = \mathbf{A}\left(\int \mathbf{x}\mathbf{x}^T \rho d\mathbf{x}\right)\mathbf{A}^T = \frac{1}{3}\mathbf{A}^2,$$

where $\left(\int \mathbf{x}\mathbf{x}^T \rho d\mathbf{x}\right) = \frac{1}{3}I$. We see that the eigenvalues of $\mathbf{C}$ are the squared eigenvalues of $\mathbf{A}$ divided by 3 and that the eigenvectors of $\mathbf{C}$ are the eigenvectors of $\mathbf{A}$.

In order to illustrate the method, we use three different 10×10 matrices. Each matrix has the same eigenvectors, but the eigenvalues are different. Plots of the eigenvalues of three cases are presented in Figure 2.5. In the first case, there is no considerable separation between the eigenvalues. In the second case, there is a visible a gap between the first and second eigenvalues. In the third case, there is a separation between the third and fourth eigenvalues. A Monte Carlo algorithm was used to find estimated eigenvalues and eigenvectors for all three cases. Then a bootstrap method (iterating for 100 times and randomly choosing 28 samples)

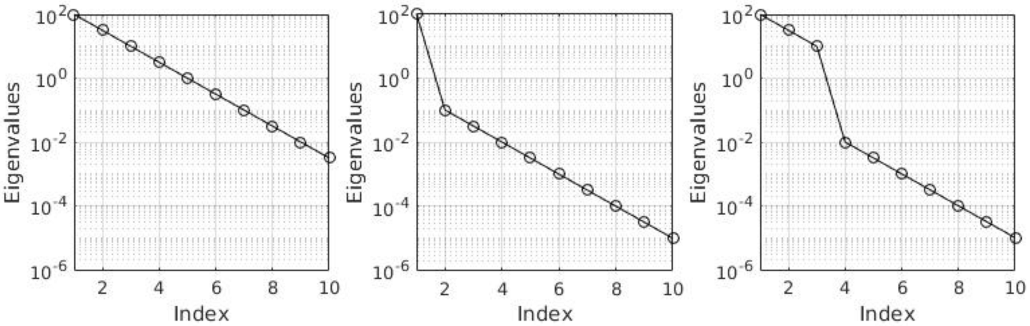


Figure 2.5. Three choices of eigenvalues for matrix $\mathbf{A}$.

was used to find bootstrap (confidence) intervals. In other words, the bootstrap intervals are used to measure the variability of the method or to see how well it performs. A description of the Monte Carlo method and bootstrapping was given in § 2.4.

Figure 2.6 represents three cases of true and estimated eigenvalues with bootstrapping for matrix **A**. Notice that the plots contain only the first 6 eigenvalues instead of 10 eigenvalues. In the first plot in Figure 2.6 a gap or a considerable separation is not present between the eigenvalues. The most important property of the second plot can be seen by looking at the first two eigenvalues. It is obvious that there is a significant separation between the first and second eigenvalues. Moreover, bootstrap intervals tell us that the first two estimated eigenvalues are evaluated more accurately than the other eigenvalues (bootstrap intervals are narrow between the first and second eigenvalues). This means that we may use a one-dimensional active subspace. If there was a considerable separation between the second and the third eigenvalues, then we would use a two-dimensional active subspace. In this case we have that examining a one-dimensional active subspace can be considered a more efficient approach than examining all the dimensions. For the third case, there is a considerable separation between the third and the fourth eigenvalues. This means that a three-dimensional active subspace may be used. The formula for the distance between subspaces given in Equation (2.7) allows us to quantify subspace errors using eigenvectors. The results can be observed in Figure 2.7. In the first plot of Figure 2.7, the subspace error is similar in all 6 dimensions. On the other hand, in the last two plots of Figure 2.7, the subspace error is smaller when we have a gap before the remaining eigenvalues. In other words, a large eigenvalue gap indicates a good place to define the active subspace dimension.

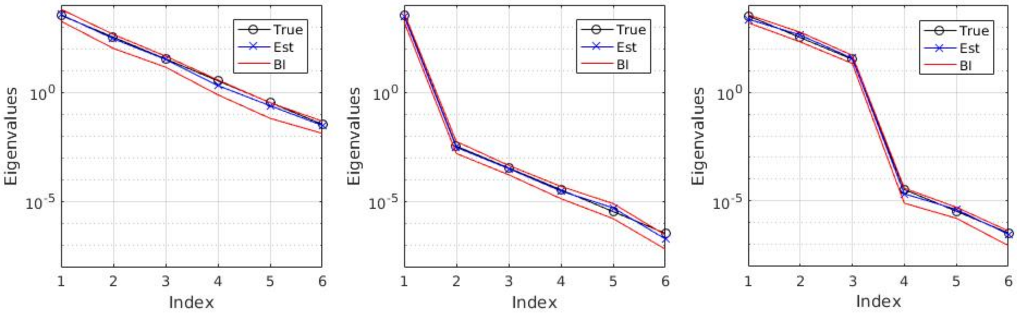


Figure 2.6. True and estimated eigenvalues along with the bootstrap intervals.

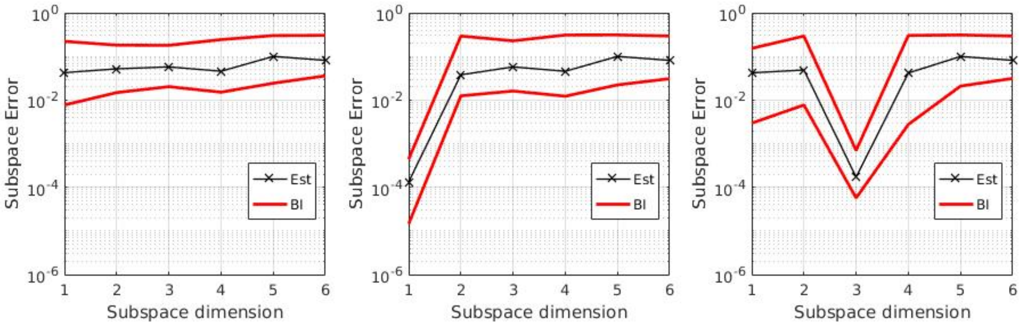


Figure 2.7. Estimated subspace error along with bootstrap intervals.

2.6 Summary

In this chapter, we introduced the method of active subspaces. In § 2.2 we discussed the underlying theory of the method. A practical algorithm for implementing the active subspace method then followed in § 2.3. We then discussed the basic principles of the Monte Carlo and Bootstrapping methods in § 2.4. Finally, we presented two numerical examples in § 2.5. The first example was a comparison between the active subspace method with gradients available and the active subspace method with a global linear model (gradients unavailable). We gave reasons why the active subspace method with gradients is superior to the method with a global linear model. The chapter finished with second example, where eigenvalues and eigenvectors were known analytically, so we could calculate errors to see whether the method performs well or not. It was found that larger eigenvalue gaps give us much smaller errors, which means that a good approximation is available if we have a gap between the eigenvalues.

Chapter 3

Background Theory on Networks

3.1 Introduction to Network Theory

Networks are everywhere. Intuitive examples are family and friends networks, the internet, airline networks, social networks, the banking network, telephone networks, train networks, etc. Networks allow us to share information and resources, which can be used to save valuable time or money, create income and so on [55].

Networks are also known as graphs. They consist of entities or nodes (or actors) with interactions or connections between the nodes. The nodes may be individuals, groups, organisations, societies, etc. Edges or links may represent social relationships between individuals, communication connections (talking or sharing information with one another), distribution routes, urban streets, airline routes, protein-protein interactions and so on [56]. A network with nodes and edges can represent a complex system, which enables us to do analysis.

Let's consider the social network created by Facebook. Figure 3.1 displays the friendship connections between 500 million people using Facebook in 2010 [57]. This is an example of a complex network, which can be analysed to answer questions of interest. For example, we might want to know which two cities had the most friendships or relationships between them. We might also wish to find the most influential or authoritative accounts. In order to try and answer questions like these, we need to introduce mathematical terminology and definitions of networks.

Figure 3.1. Facebook network in 2010.

We begin with a formal definition of a network.

Definition 3.1.1. *A network, G, is a pair (V, E), where V is called the vertex set of G and E is called the edge set of G.*

The elements of set V are vertices or nodes of G and the elements of E are pairs of nodes. It is important to note that edges can be directed or undirected, weighted or unweighted. Examples of these cases will be shown later in this section. It is also worth mentioning that if edges or links between the nodes are symmetric (see Figure 3.2), then G is an undirected network. If the edges are nonsymmetric, then G is a directed network (see Figure 3.3). A graph that is both edge and vertex-transitive is called a symmetric graph [58]. In this thesis we simply say that a graph is symmetric if it is undirected and nonsymmetric if it is directed.

Before we start illustrating and analysing examples of adjacency matrices, we need to introduce a few more general but rather important concepts from graph theory. Namely, the definition of a *walk* in a graph, a *closed walk*, a *path*, the *shortest path*, a *cycle*, a *simple* graph, and a *loop*. A *walk* on a graph G is a sequence of vertices connected by edges. A *walk* can repeat vertices and edges, which means that a *walk* may move forward or backward through the graph multiple times. A *closed walk* is a walk when node i is the starting and finishing node. A *path* is a walk with no repeated vertices or edges. A *loop* is an edge from node i to itself. A *shortest path* from one vertex to another uses the least number of edges or the least total weight. A *simple* graph is a graph with no multiple, unweighted edges or loops. A cycle is a walk in which the only repeated nodes are the first and the last

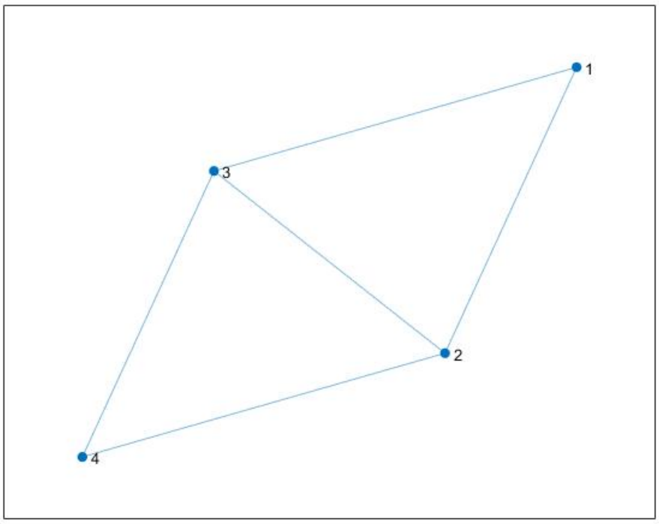
[59]. Definition 3.1.2 helps us to establish a useful mathematical representation of a network. If node i is connected to node j, then the connection between these two nodes is represented by 1 in the adjacency matrix, otherwise the entry is 0.

Definition 3.1.2. *Let $G = (V, E)$ be a simple network (that is, a network with a simple graph) where $V = \{1, \ldots, N\}$ is a finite set of vertices. For $1 \leq i, j \leq N$ define*

$$a_{i,j} = \begin{cases} 1, & (i,j) \in E, \\ 0, & (i,j) \notin E. \end{cases}$$

Then the square matrix $A = (a_{i,j})$ is called the adjacency matrix of G.

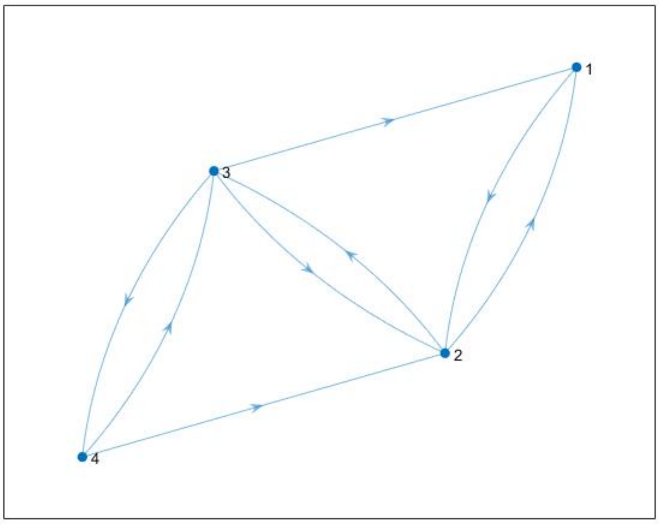
Every network can be represented as an adjacency matrix. We now demonstrate this concept with some small graphs. Figure 3.2 represents a graph with 4 nodes and 5 edges (or 10 edges if we consider the graph to be symmetric). Here we have a symmetric graph and a symmetric adjacency matrix.



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

Figure 3.2. A simple symmetric graph (left) and its adjacency matrix (right).

Figure 3.3 represents a directed graph with 4 nodes and 8 edges. Here, the adjacency matrix A is nonsymmetric. Figure 3.4 represents a weighted symmetric graph with 4 nodes and 5 edges. We chose the weights by sampling from a uniform distribution on the interval $(0, 1)$. We only consider positive weights on the edges throughout this thesis for simplicity. However, there are cases when considering

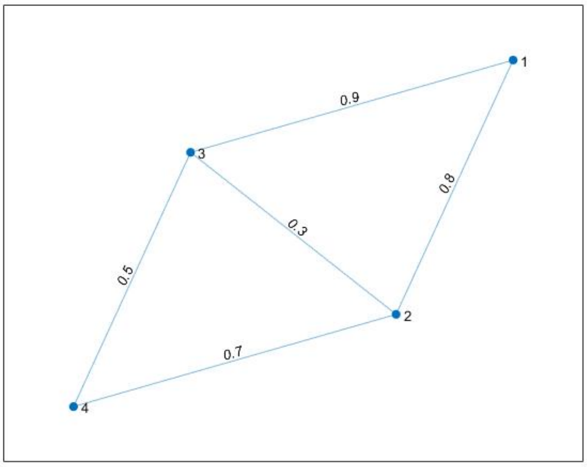


$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Figure 3.3. A directed graph (left) and its adjacency matrix (right).

negative weights is beneficial. One possible example of a negative weight on the edge is transferring money from one account into another. Futhermore, consider a social network such as Facebook, and say John likes Erica: this could be represented with a positive edge weight. If John dislikes Erica, this could be represented with a negative edge weight in a graph. On the other hand, assigning weights on these edges could be tricky since it is not easy to determine the level of relationship intensity between people. For more information about negative weights on the edges, see [60].

Continuing with Figure 3.4, the adjacency matrix A is the same as in Figure 3.2 but with weights. In our context, weights will be assigned at random. Consider a sample $\mathbf{x} = (x_1, x_2, x_3, \ldots, x_m)$ drawn from $U(0, 1)$ with m input parameters or weights. The number of weights m matches the number of edges between the nodes. Moreover, we transfer the vector of weights $\mathbf{x}$ into the adjacency matrix and replace ones with the corresponding weights. For example, Figure 3.2 has an adjacency matrix with ones and zeros. Since the graph is symmetric, we only require $m = 5$ random weights and replace ones in the adjacency with $\mathbf{x} = (0.8, 0.9, 0.3, 0.7, 0.5)$ row-wise. In other words, an edge $a_{1,2}$ is assigned a weight of $x_1 = 0.8$ since it is the first nonzero entry of the adjacency matrix in Figure 3.2, an edge $a_{1,3}$ is assigned a weight of $x_2 = 0.9$ since it is the second nonzero entry of the adjacency matrix in Figure 3.2 and so on. Additionally, $a_{i,j} = a_{j,i}$ in the adjacency matrix due to the

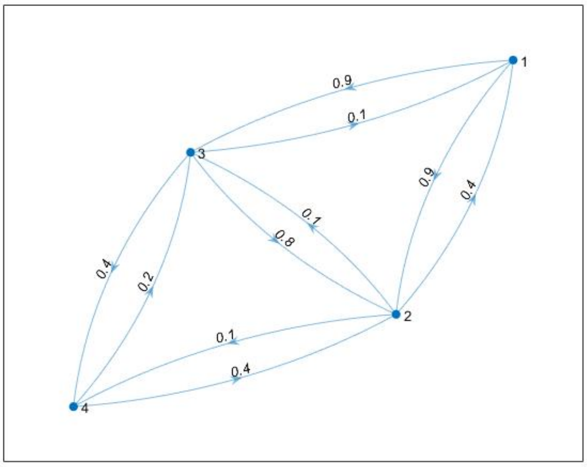


$$A = \begin{bmatrix} 0 & 0.8 & 9 & 0 \\ 0.8 & 0 & 0.3 & 0.7 \\ 0.9 & 0.3 & 0 & 0.5 \\ 0 & 0.7 & 0.5 & 0 \end{bmatrix}$$

Figure 3.4. A weighted symmetric graph (left) and its adjacency matrix (right).

fact that the graph is symmetric. This means that when $a_{1,2}$ is assigned a weight x_1, $a_{2,1}$ is also assigned a weight of x_1.

Note that weighted graphs can also be nonsymmetric, if different weights are allocated to each directed edge between two nodes (see Figure 3.5). Figure 3.5 represents a case when a graph is symmetric in structure but not symmetric by inspecting edges and its weights.



$$A = \begin{bmatrix} 0 & 0.9 & 0.9 & 0 \\ 0.4 & 0 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0 & 0.4 \\ 0 & 0.4 & 0.2 & 0 \end{bmatrix}$$

Figure 3.5. A weighted nonsymmetric graph (left) and its adjacency matrix (right).

3.2 Centrality Measures

In this section, we discuss centrality measures, which are used to identify important nodes. The main reference for this section is [61].

Networks have been a field of interest for a long time and centrality is a key ingredient that offers access to useful information. For example, displaying the Facebook network in Figure 3.1 gives us no obvious information about the network. However, centrality can reveal certain characteristics of the network, such as which person is more communicative or has more "important" relationships compared to other people. In other words, centrality measures the importance of each node [62]. Centrality can be thought of as a way of describing the potential for communication with other nodes. In this thesis, we will consider three popular centrality measures: degree centrality, eigenvector centrality and Katz centrality [63]. These centrality measures are widely used and have certain similarities. However, before we explain how they are related, we first need to define these centrality measures.

3.2.1 Degree Centrality

Degree centrality simply corresponds to the degree of a node. The degree of a node is defined as the number of connections the node has (that is, the number of edges involving the node). If we have an undirected network G and its $N \times N$ adjacency matrix A, then the degree of node i is defined as

$$d_i = \sum_{j=1}^{N} a_{i,j} = (\mathbf{1}^T A)_i = (A\mathbf{1})_i, \tag{3.1}$$

where $\mathbf{1}$ is $N \times 1$ column vector of ones. If $d_i > d_j$, then node i is more degree central than node j. Moreover, $max(d_i)$ gives us the most central node in the network. Equation (3.1) is valid if and only if $A = A^T$. For an undirected network, the expression $(\mathbf{1}^T A)_i$ in (3.1) corresponds to the in-degree centrality. The expression $(A\mathbf{1})_i$ in (3.1) corresponds to the out-degree centrality. Both are degree centralities, however, they measure different features. The calculation of the in-degree centrality simply relies on summing column-wise whereas out-degree centrality relies on summing row-wise. By definition, the in-degree centrality counts the number of links or connections that are "coming in" to node i whereas out-degree centrality counts the number of links or connections that are "going out" from node i. In other words, in-degree and out-degree centralities correspond to "receiving" information and “broadcasting” information, respectively.

3.2.2 Katz Centrality

Katz centrality may be viewed as an extended version of degree centrality. This is due to the fact that it takes account of all walks from node i to any other node in the network whereas degree centrality only counts the number of edges involving node i. The Katz centrality measure [61] is defined as

$$K_i = ((\alpha^0 A^0 + \alpha A + \alpha^2 A^2 + \ldots + \alpha^d A^d + \ldots)\mathbf{1})_i = \left(\sum_{d=0}^{\infty} (\alpha^d A^d)\mathbf{1} \right)_i . \tag{3.2}$$

Here, the parameter α is known as the attenuation factor. If the attenuation factor is too large (i.e., greater than the reciprocal of the spectral radius $\rho(A)$ of A), then the infinite sum in Equation (3.2) doesn't converge. If the attenuation factor is less than the reciprocal of the spectral radius $\rho(A)$ of A, then the geometric series converges [61, pp. 116-117]. In other words, if $\alpha < 1/\rho(A)$ then

$$K_i = \left((I - \alpha A)^{-1} \mathbf{1} \right)_i . \tag{3.3}$$

Equation (3.3) represents “broadcasting” out information. The other case for “receiving” information is

$$K_i = \left(\mathbf{1}^T (I - \alpha A)^{-1} \right)_i . \tag{3.4}$$

As mentioned before, Katz centrality of a node i counts all walks beginning at node i such that the longer walks are penalized through the attenuation factor α. We also note that A^d counts walks of length d. The immediate neighbours, i.e. walks of length 1, are given the value α^1, whereas the farther neighbours, i.e. walks of length d, are assigned α^d value. Thus, the farther the neighbours from the node in consideration are, the lesser influence they have [64]. Note that, as the “broadcasting” and “receiving” values in Equations (3.3) and (3.4) represent the sum of entries in row i and column i, respectively, these will be equal if matrix A is symmetric. It is also worth mentioning that Equation (3.3) can be written as

$$K_i = \left((I - \alpha A)^{-1} \mathbf{1} \right)_i = \mathbf{e}_i^T (I - \alpha A)^{-1} \mathbf{1},$$

where $\mathbf{e}_i$ is the i^{th} column of $N \times N$ identity matrix I.

3.2.3 Eigenvector Centrality

Another similar centrality measure to degree is eigenvector centrality [59, 63, 65, 66]. In degree centrality, all node connections are credited with equal importance. However, each node may differ in importance. For instance, a node which is connected to highly important nodes is itself an important node. Eigenvector centrality assigns a relative score to each node depending on the type of nodes it is connected to [64].

If G is a simple undirected graph, then the corresponding adjacency matrix A is symmetric with zeros on the diagonal. By the Perron-Frobenius theorem [67, 68], we can guarantee that the largest eigenvalue of the adjacency matrix A is real. Furthermore, if we let the eigenvalues of A be $\lambda_1 > \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_n$, then, if graph G is connected, $\lambda_1 > \lambda_2$ (again by the Perron-Frobenius theorem). If the matrix A is symmetric, then the decomposition of matrix A can be obtained as

$$A = Q\Lambda Q^T,$$

where $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_n)$ are eigenvalues in decreasing order and $Q = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n)$ is orthogonal with each column eigenvector corresponding to the ordered eigenvalue [59]. In other words, the biggest or the leading eigenvalue λ_1 corresponds to the dominant eigenvector $\mathbf{q}_1$ (with $\mathbf{q}_1 \geq \mathbf{0}$ by the Perron-Frobenius theorem). The eigenvector centrality measure is then defined as

$$C_{ev}(i) = \mathbf{e}_i^T \mathbf{q}_1 = q_1(i), \tag{3.5}$$

where $\mathbf{q}_1$ is the dominant eigenvector of A that corresponds to the leading eigenvalue λ_1 [59].

Bonacich and Lloyd showed in [66] that when $\alpha \to \dfrac{1}{\lambda_1}$ in (3.3), we have that Katz centrality converges to eigenvector centrality [59, pp. 17-18]. Estrada and Knight also give a proof in [61, pp. 158-160]. On the other hand, when $\alpha \to 0$, we have that degree centrality is a limiting case of Katz centrality.

In summary, we arrive at the following relationship between degree, Katz and eigenvector centralities. When the attenuation factor α is close to the inverse of spectral radius of A, Equation (3.3) represents a limiting case for the eigenvector centrality. When α is close to 0, Equation (3.3) represents a limiting case for the degree centrality. In this sense, Katz centrality interpolates between degree and eigenvector centralities.

3.2.4 Edge Betweenness Centrality

The last centrality measure that we consider in this thesis is edge betweenness centrality. This is defined as the number of the shortest paths that go through an edge in a graph or a network [69]. Each edge in the network can be associated with an edge betweenness centrality value, with an edge with a high edge betweenness centrality score representing a bridge-like connector between two parts of a network, the removal of which may affect the communication between many pairs of nodes through the shortest paths between them. The edge betweenness centrality can therefore be thought of as representing the information flow being propagated through a particular link in a complex network.

More precisely, let $G = (V, E)$ be a connected undirected graph, with v_i and v_j representing two nodes in G, respectively. The betweenness centrality of an edge $e \in E$ is then defined as

$$EB(e) = \sum_{v_i \in V} \sum_{v_j \in V} \frac{\sigma_{v_i v_j}(e)}{\sigma_{v_i v_j}}, \tag{3.6}$$

where $\sigma_{v_i v_j}$ is the total number of shortest paths from node v_i to node v_j and $\sigma_{v_i v_j}(e)$ is the number of those paths that pass through an edge e.

We will apply the edge betweenness centrality in some of our examples in Chapter 4. The code used to calculate the edge betweenness centrality in these examples was taken from [70].

3.3 Gradient of Katz Centrality

The reason why we are studying network theory along with centrality measures is that we want to apply the active subspace method to networks. We use centrality measures to determine which nodes are the most central or important to a certain network. We then use the active subspace method to determine the combination (or combinations) of edges that has the most influence.

As discussed in Chapter 2, the method of active subspaces requires the gradient of the quantity of interest $f(\mathbf{x})$ or an approximation of the gradient. In practice, the gradient is rarely available, which means we usually need to approximate the gradient. In order to use the active subspace method in our network setting, we need the gradient of Katz centrality. We could use finite differences, as said before. However, we can in fact find an analytic expression for the gradient of Katz centrality in (3.3). As far as we are aware, this analytic derivative has not previously been found.

Lemma 3.3.3. *Given a matrix A and a parameter α (with $0 < \alpha < 1/\rho(A)$), the Katz centrality $K_i = \left((I-\alpha A)^{-1}\mathbf{1}\right)_i$ can be analytically differentiated with respect to each $a_{i,j}$ as*

$$\frac{\partial K_i}{\partial a_{i,j}} = \left((I-\alpha A)^{-1}\alpha E_{i,j}\,(I-\alpha A)^{-1}\mathbf{1}\right)_i, \tag{3.7}$$

where $E_{i,j}$ is the zero matrix except for a 1 in position (i,j).

Proof. We have $K_i = \left((I-\alpha A)^{-1}\mathbf{1}\right)_i$, where I is the identity matrix, A is a weighted adjacency matrix, $\mathbf{1}$ is a vector of ones, and α lies between 0 and $1/\rho(A)$. By definition

$$\frac{\partial K_i}{\partial a_{i,j}} = \lim_{\epsilon\to 0}\frac{((I-\alpha(A+\epsilon E_{i,j}))^{-1}\mathbf{1})_i - ((I-\alpha A)^{-1}\mathbf{1})_i}{\epsilon}. \tag{3.8}$$

We now expand the first term in the expression in the right-hand side of this equation, $(((I-\alpha(A+\epsilon E_{i,j}))^{-1}\mathbf{1})_i)$, in terms of powers of ϵ. First we note that

$$(I-\alpha(A+\epsilon E_{i,j}))^{-1} = (I-\alpha A-\epsilon\alpha E_{i,j})^{-1} = \left((I-\alpha A)\left(I-\epsilon(I-\alpha A)^{-1}\alpha E_{i,j}\right)\right)^{-1}.$$

Since $(XY)^{-1} = Y^{-1}X^{-1}$, we obtain

$$(I-\alpha(A+\epsilon E_{i,j}))^{-1} = \left(I-\epsilon(I-\alpha A)^{-1}\alpha E_{i,j}\right)^{-1}(I-\alpha A)^{-1}.$$

We now observe that for a small h and given matrix B [71, pp. 58-59],

$$(I+hB)^{-1} = I-hB+O(h^2).$$

Applying this result to our expression gives

$$(I-\alpha(A+\epsilon E_{i,j}))^{-1} = \left(I+\epsilon(I-\alpha A)^{-1}\alpha E_{i,j}+O(\epsilon^2)\right)(I-\alpha A)^{-1}.$$

Hence,

$$\left(I - \alpha\left(A + \epsilon E_{i,j}\right)\right)^{-1} = \left(I - \alpha A\right)^{-1} + \epsilon\left(I - \alpha A\right)^{-1} \alpha E_{i,j}\left(I - \alpha A\right)^{-1} + O(\epsilon^2).$$

Substituting this into the expression for the derivative in (3.8) gives

$$\frac{\partial K_i}{\partial a_{i,j}} = \lim_{\epsilon \to 0}\left(\frac{\left(\left(I - \alpha A\right)^{-1}\mathbf{1}\right)_i + \left(\epsilon\left(I - \alpha A\right)^{-1}\alpha E_{i,j}\left(I - \alpha A\right)^{-1}\mathbf{1}\right)_i + O(\epsilon^2) - \left(\left(I - \alpha A\right)^{-1}\mathbf{1}\right)_i}{\epsilon}\right).$$

If we simplify and take the limit as ϵ goes to zero, we obtain the final result

$$\frac{\partial K_i}{\partial a_{i,j}} = \left(\left(I - \alpha A\right)^{-1}\alpha E_{i,j}\left(I - \alpha A\right)^{-1}\mathbf{1}\right)_i$$

as required. □

We will use this result in the next chapter when applying the active subspace method to networks using Katz centrality. As part of our numerical study, we will investigate the effect of using finite differences as compared to using the exact expression for the derivative of Katz centrality in (3.7) to obtain gradient samples.

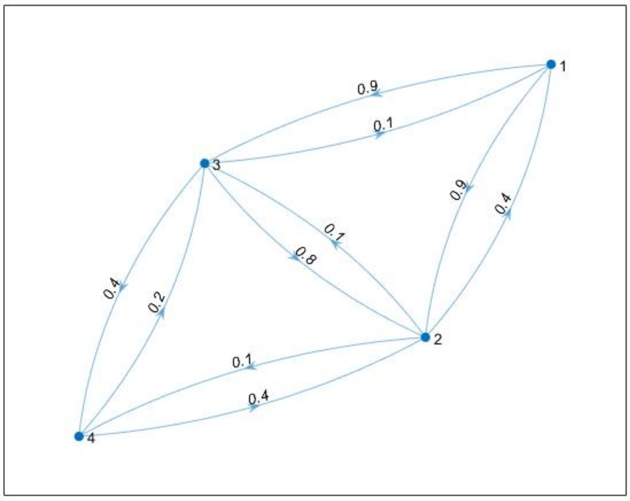
Chapter 4

Active Subspaces in Network Science

4.1 Overview

In Chapter 2 we described the method of active subspaces, and in Chapter 3 we introduced some basic concepts involved in the mathematical modelling of networks. In this chapter, we combine these ideas by applying the active subspace method to a problem in network theory. Specifically, we use a centrality measure as the smooth function f and weights on network edges as parameters $\mathbf{x}$ in the active subspace method. Note that f must be chosen in such a way that we obtain a scalar quantity of interest $f(\mathbf{x})$ so that it is suitable for the active subspace method. Candidates for $f(\mathbf{x})$ are therefore a) the centrality of a particular node of interest, b) some norm of the centrality vector, describing the overall level of centrality and c) the ratio of centralities between a pair of nodes of interest, describing their relative importance. For a given graph, we will use the Katz centrality of a particular node of interest to be $f(\mathbf{x})$ (the quantity of interest). Also, as the aim of applying the active subspace method is to discover a combination of **edges** that strongly influences the quantity of interest, we will also consider the edge betweenness centrality, which also focuses on identifying important edges. These experiments will give some insight into the question "which interactions are important in terms of centrality?". Addressing this question adds value to the existing network centrality literature.

We now describe our application of the active subspace method to networks in more detail. The graphs we consider are always undirected and unweighted at first, such as the graph in Figure 3.2 which is symmetric and undirected (each undirected edge is regarded as a pair of directed edges), with an adjacency matrix A consisting of ones and zeros. To apply the active subspace method, we construct a parameter vector $\mathbf{x}$ whose entries correspond to weights on the graph edges. To do this, we generate $\mathbf{x}$ with entries drawn independently from a uniform distribution on $(0, 1)$. Treating this as a vector of weights leads to a weighted version of the graph, as illustrated in Figure 4.1. We can also use the weight vector $\mathbf{x}$ to create the adjacency matrix A as shown in Chapter 3 (Figure 3.2 and Figure 3.3). This procedure can be repeated using as many random samples (weight vectors), M say, as required.



$$A = \begin{bmatrix} 0 & 0.9 & 0.9 & 0 \\ 0.4 & 0 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0 & 0.4 \\ 0 & 0.4 & 0.2 & 0 \end{bmatrix}$$

Figure 4.1. Left: A weighted graph with 4 nodes and 10 edges. Right: A weighted adjacency matrix.

Note that in [8], Constantine normalises the input parameters $\mathbf{x}$ to be centered around 0 with equal ranges. This is done to ensure that large values in $\mathbf{x}$ do not affect the analysis. If we normalise the parameters or a weight vector such that $\mathbf{x}$ is centered around 0 with equal ranges, this means that some input parameters are assigned negative weights, which does not make sense in the context of our network analysis. In this thesis, we therefore use positive weights only.

In order to apply the active subspace method in a network setting, as well as a parameter or weight vector $\mathbf{x}$ we require a smooth function f and a scalar $f(\mathbf{x})$ for our quantity of interest. Consider the smooth Katz centrality as defined in Equation (3.3), which we restate here for convenience:

$$K_i = ((I - \alpha A)^{-1}\mathbf{1})_i, \tag{4.1}$$

where I is the $N \times N$ identity matrix and A is weighted or unweighted adjacency matrix. This gives us a centrality index for each node (the importance of each node in the graph). Recall that we are interested here in edges and how much each edge or connection can affect the centrality of a node of interest. In order to proceed, we need to choose a node that we are interested in. Having defined $\mathbf{x}$ and $f(\mathbf{x})$ we then proceed with the active subspace method, which allows us to determine important edges that affect K_i for a given i (discussed in the following sections).

4.2 Finite Difference Method Comparison with Exact Derivative

Before we present our experiments illustrating the effectiveness of the active subspace method for networks, we recall that as part of the active subspace method, we also require the gradient of the quantity of interest. Although this can be calculated using a finite difference approach, here we have an exact expression for the derivative of the Katz centrality at a given node, namely, the result in Chapter 3, § 3.3 (Lemma 3.3.3). This gives us the opportunity to check how accurate a simple finite difference approximation is in this setting.

We consider the simple network displayed in Figure 4.2. Let A be a weighted adjacency matrix for this graph with weights $a_{i,j}$ on the edges drawn from a uniform distribution between 0 and 1. To fix ideas, we will consider $i = 1$ and let $j = 2$ (so that we are looking at the edge from node 1 to node 2). We set the parameter α in the Katz centrality to be $\alpha = 1/5$, which satisfies the restriction that α must be less than $\dfrac{1}{\rho(A)} = 0.3536$.

With these values, the expression (3.7) in Lemma 3.3.3 becomes

$$\frac{\partial K_1}{\partial a_{1,2}} = \left(\left(I - \frac{1}{5} \cdot A\right)^{-1} \cdot \frac{1}{5} \cdot E_{1,2} \left(I - \frac{1}{5} \cdot A\right)^{-1} \cdot \mathbf{1}\right)_1 . \tag{4.2}$$

To construct a finite difference approximation to $\dfrac{\partial K_1}{\partial a_{1,2}}$, we therefore need to calculate

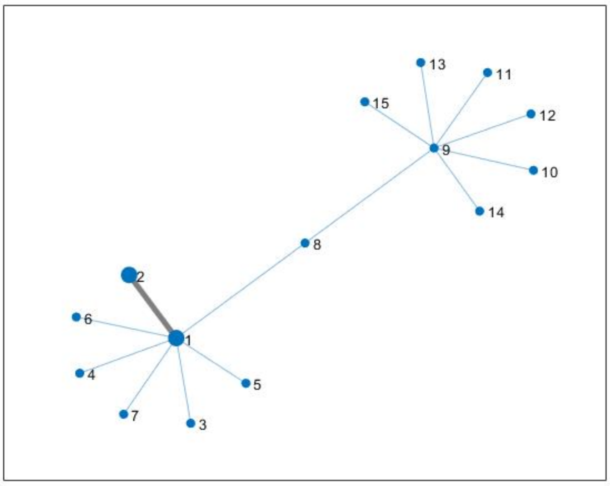


Figure 4.2. A simple network with 15 nodes and 14 edges.

$$\Delta K_1(a_{1,2}) = \frac{\left(\left(I - \alpha \left(A + hE_{1,2}\right)\right)^{-1} \mathbf{1}\right)_1 - \left(\left(I - \alpha A\right)^{-1} \mathbf{1}\right)_1}{h}$$

for some step size h. In this example, we consider five different step sizes: $10^{-5}, \ldots, 10^{-1}$. From Taylor's theorem

$$\frac{f(x+h) - f(x)}{h} = f'(x) + \frac{h}{2} f''(x) + \cdots ,$$

If we move $f'(x)$ to the left hand side, we get that

$$\frac{f(x+h) - f(x)}{h} - f'(x) = \frac{h}{2} f''(x) + \cdots ,$$

which shows that the difference between the finite difference approximation and the exact derivative should be proportional to h (step size) for small h.

We define the error ϵ to be the absolute value of the difference between the finite difference approximation and the exact derivative, namely

$$\epsilon = \left| \Delta K_1(a_{1,2}) - \frac{\partial K_1}{\partial a_{1,2}} \right| . \tag{4.3}$$

The values of this error for five difference step sizes are shown in Table 4.1. We can see that decreasing the step size h improves the accuracy of the finite difference approximation, with the approximate value approaching the value of $\dfrac{\partial K_1}{\partial a_{1,2}} = 0.290515$. The error is also plotted in Figure 4.3, which provides us with

h	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
$\Delta K_1(a_{1,2})$	0.291711	0.290634	0.290527	0.290516	0.290515
ϵ/h	0.119627	0.119185	0.119141	0.119137	0.119124

Table 4.1. Comparison of finite differences against exact derivative to 6 decimal places.

confirmation that the error decreases as we decrease the step size. A reference line with linear slope is also shown in the figure to illustrate that the error is proportional to h as expected.

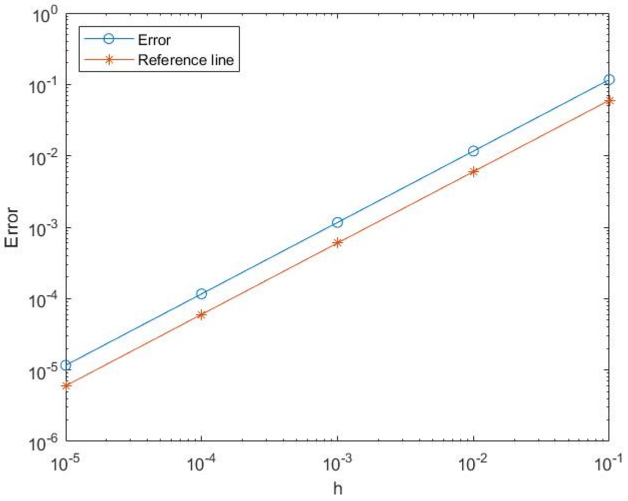


Figure 4.3. Difference between exact derivative and finite difference approximation.

4.3 Examples of Synthetic Networks

We are now ready to illustrate the effectiveness of the active subspace method applied to some examples of synthetic networks. These synthetic networks have been chosen to have clearly-defined structures, allowing us to judge the results. Applications to real-world networks will be considered later in the chapter.

As mentioned in previous sections, given a complex network, a key task is to determine the most important nodes or edges. Typically, the importance of each node is determined by using a centrality measure, such as the Katz centrality described in § 3.2, § 3.3, § 4.1 and § 4.2. By using random weights for a network's edges, the active subspace method will help us determine which edges are important in the network and which edges impact the quantity of interest the most (i.e. $f(\mathbf{x})$). The method can discover a linear combination or combinations of important edges (parameters). In our first example, we consider the graph in Figure 4.2 but now with directed edges (see Figure 4.4). We examine this simple graph to illustrate the steps needed to apply the active subspace method to a network.

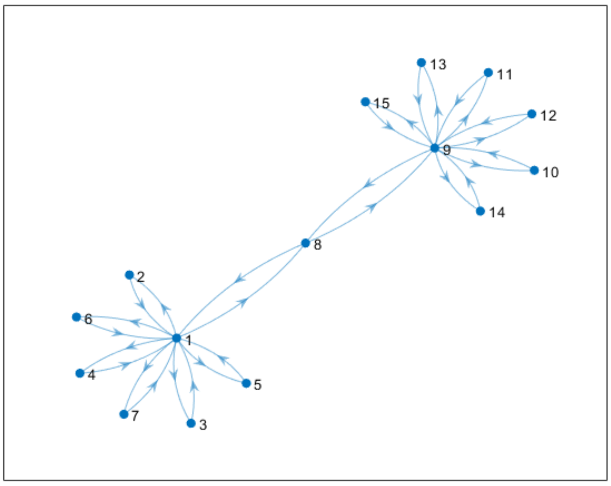


Figure 4.4. A synthetic network with 15 nodes and 28 edges.

This network has 15 nodes and 28 edges: note that certain edges are topologically equivalent (e.g., 1 → 2, 1 → 6, etc). The network has two qualities, which we can see by inspection: the first is that two nodes are highly connected to other nodes (nodes 1 and 9) and, secondly, there is one node that acts as a link between the important nodes (node 8). Our task is to use the active subspace method to determine the most important edges, or combination(s) of edges, that impact our quantity of interest $f(\mathbf{x})$ the most. The key point here is that we want to see whether the results agree with our intuition. For example, if we look at node 1, we expect that the most important edges for this node are the surrounding edges with node 8 being the most important neighbour.

For our quantity of interest $f(\mathbf{x})$, we consider the Katz centrality of a given node (4.1) with $\alpha = 0.2$ (which satisfies the condition $0 < \alpha < 1/\rho(A)$). Table 4.2 shows the centralities of the unweighted version of graph in Figure 4.4. The choice of α depends on the inverse of the spectral radius, i.e., $1/\rho(A)$ of matrix A, which is 0.3536 in this case.

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Centrality	**3.5294**	1.7059	1.7059	1.7059	1.7059	1.7059	1.7059	**2.4118**	**3.5294**	1.7059	1.7059	1.7059	1.7059	1.7059	1.7059

Table 4.2. Katz centralities for the network in Figure 4.4 with all edge weights equal to 1.

Looking at Table 4.2, we can see that nodes 1, 8, 9 are rated as the most central nodes of this network (as expected). Nodes 1 and 9 are the most central nodes largely because they have more connections compared with other nodes. However, a certain node can have a fairly high centrality with few connections. The reason is that being connected to a few important nodes make a node more central (node 8).

To consider sensitivity, we now allow the edges to be weighted. The zeros in matrix A are symmetric, however, we will allow the weights on the edges to be different for each edge. For example, an edge $1 \to 2$ and an edge $2 \to 1$ can have different weights. In this example, our weights $\mathbf{x} \sim U(0, 1)$. It is also important to note that $\mathbf{x} \in \mathbb{R}^{28}$. The reason for this is that we have 28 edges in the graph and so we have 28 parameters.

To proceed with an active subspace method, we need to choose the number of samples M we want to work with and a scalar quantity of interest $f(\mathbf{x})$. We choose M to be large since the problems we are studying are computationally cheap and so we can work with at least $M = 10^3$ samples in most cases. However, it is not an easy task to decide on the number of samples to use or if a given problem is computationally expensive, it is advisable to use a systematic rule of thumb that calculates the number of samples needed to get efficient results (this is discussed in § 2.3). It is also advisable to repeat the experiment with a larger M and check that the results are consistent.

We now let the Katz centrality of node 1 be the quantity of interest $f(\mathbf{x})$ with input parameters $\mathbf{x}$, so that

$$f(\mathbf{x}) = K_1 = \left(\left(I - \alpha A\left(\mathbf{x}\right)\right)^{-1}\mathbf{1}\right)_1. \tag{4.4}$$

Again, we note here that we want to find which edges or linear combination(s) of edges impact $f(\mathbf{x})$ the most. Once we describe our quantity of interest $f(\mathbf{x})$, parameters $\mathbf{x}$ and the number of samples M, we then calculate (or approximate) partial derivatives with respect to Equation (4.4). Note that an analytical differentiation of Katz centrality equation is performed in the previous chapter to calculate gradients. However, before we use an analytical expression of Katz centrality (see Equation (3.7)) in Lemma 3.3.3, we need to calculate our quantity of interest $M = 10^3$ times (calculating $f(\mathbf{x})$ for each sample).

Using the active subspace method we then apply Monte Carlo to generate an eigenvalue decomposition (discussed in Chapter 2, Algorithm 1) in order to obtain eigenvectors and eigenvalues. A plot of eigenvalues is presented in Figure 4.5.

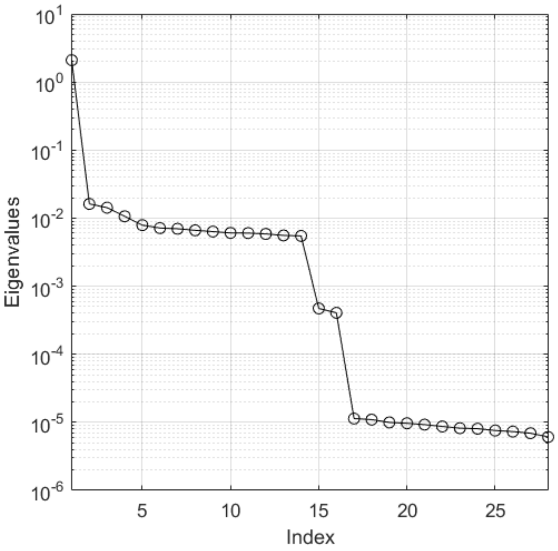


Figure 4.5. The eigenvalues obtained by the active subspace method for the network in Figure 4.4.

A gap between the first and second eigenvalues is present, which suggests that there is a 1-dimensional active subspace (Algorithm 1). Since $\mathbf{x} \in \mathbb{R}^{28}$, we have 28 eigenvector components (each component of the eigenvector represents the contribution of each edge) of matrix $\mathbf{C}$ (see Algorithm 1). The components of the eigenvector are shown in Figure 4.6 and the ordering of the components is presented in Table 4.3.

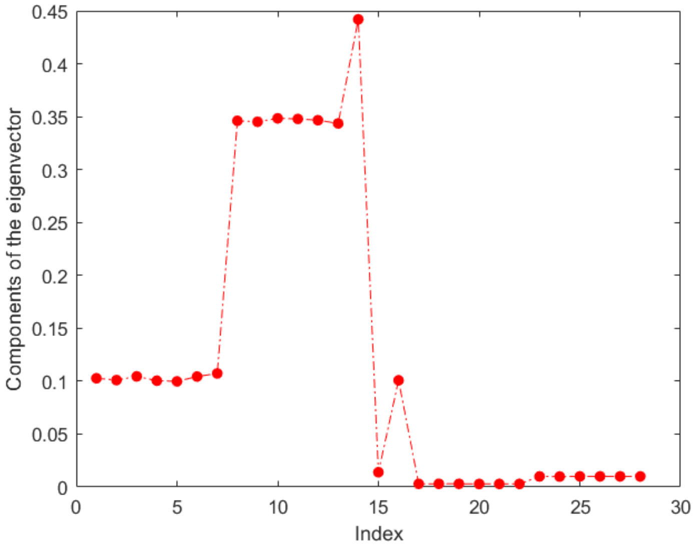


Figure 4.6. The leading eigenvector for the network in Figure 4.4.

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Edge	2 → 1	3 → 1	4 → 1	5 → 1	6 → 1	7 → 1	8 → 1	1 → 2	1 → 3	1 → 4	1 → 5	1 → 6	1 → 7	1 → 8
Index	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Edge	9 → 8	8 → 9	10 → 9	11 → 9	12 → 9	13 → 9	14 → 9	15 → 9	9 → 10	9 → 11	9 → 12	9 → 13	9 → 14	9 → 15

Table 4.3. Ordering of the edges in Figure 4.6.

Recall that the magnitudes of the eigenvector components tell us how sensitive the quantity of interest is to those parameters. The largest component in Figure 4.6 corresponds to a connection between the first and eighth nodes, which is not a surprise since our quantity of interest is the Katz centrality of node 1 and we are explicitly searching for edges that have the most impact on $f(\mathbf{x})$. This means that any changes to this edge impact $f(\mathbf{x})$ the most, on average. The other 6 edges that seem to be fairly important to our quantity of interest are the edges surrounding node 1 (1→2, 1→3, 1→4, 1→5, 1→6, 1→7). It is important to note that all the edges that are important to node 1 are outgoing edges ('sending out' information). This is not surprising due to the fact that Katz centrality is based on broadcasting information out. The values of the remaining edges are close to zero. This represents the fact that these edges do not have as much impact on $f(\mathbf{x})$ as the other edges in the network.

As mentioned before, there seems to be a 1-dimensional active subspace since there is a gap between the first and second eigenvalues. This has the implication that there is a univariate trend. Sufficient summary plots are a useful way to inspect such cases. In Chapter 2 we explained that a sufficient summary plot helps us to identify a low-dimensional structure. It is a plot of the quantity of interest against a linear combination or combinations of the input parameters. In this particular case, it is a plot of $f(\mathbf{x})$ against a linear combination of parameters $\hat{\mathbf{w}}_1^T\mathbf{x}$, where $\hat{\mathbf{w}}_1$ is an $m = 28$ vector (the first eigenvector) and $\mathbf{x}$ is a sample vector of size m. Since we have $M = 10^3$ samples, we have $M = 10^3$ quantities of interest $f(\mathbf{x})$ and $M = 10^3$ values of $\hat{\mathbf{w}}_1^T\mathbf{x}$. The result is shown in Figure 4.7. The sufficient summary plot in Figure 4.7 shows that there is a univariate trend. The main point of this is that parameter studies can be performed (see Chapter 2) knowing this information. In other words, $f(\mathbf{x})$ can be approximated with a simpler function, say

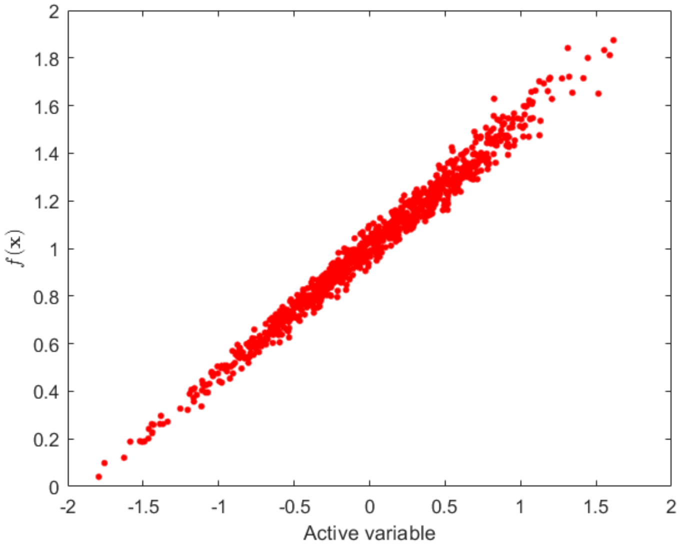


Figure 4.7. Sufficient summary plot of $f(\mathbf{x})$ (Katz centrality of node 1 for each sample) against the active variable ($\hat{\mathbf{w}}_1^T\mathbf{x}$).

$$f(\mathbf{x}) \approx h(\hat{\mathbf{W}}_1^T\mathbf{x}),$$

which depends on the linear combination or combinations of the inputs ($\hat{\mathbf{W}}_1^T$ represents the span of the first k eigenvectors mentioned in Chapter 2). In this particular example, $k = 1$ and $f(\mathbf{x})$ is computationally cheap. We denoted the linear combination of parameters by $\hat{\mathbf{w}}_1^T\mathbf{x}$ instead of $\hat{\mathbf{W}}_1^T\mathbf{x}$ simply because we identified a one-dimensional active subspace, which involves only 1 eigenvector. However, real-world problems/networks are often more complex and computationally expensive. Finding important directions in the parameter space of a given problem and using that information to build a model around those directions can reduce the computational costs significantly. In this way, the active subspace method attempts to deal with the curse of dimensionality.

As a second example, we consider another network shown in Figure 4.8, which is similar to Figure 4.4. However, this time we have more nodes, and the edges between the nodes are undirected. We want to illustrate how the centralities vary and how the active subspace method performs in this case. By looking at Figure 4.8, we can see that there are 20 nodes and 19 undirected edges. We note that five nodes are highly connected to other nodes (namely, 1, 8, 9, 17, 18). The Katz centralities (Equation (4.1)) for the network in Figure 4.8 are presented in Table 4.4 (with $\alpha = 0.2$). Looking at Table 4.4 we can see that nodes 1, 8, 9, 17 and 18 are the most central nodes of this network. Node 1 is the most central node largely because it has more connections compared to other nodes. For this example we let the Katz centrality of node 1 be the quantity of interest, that is, Equation

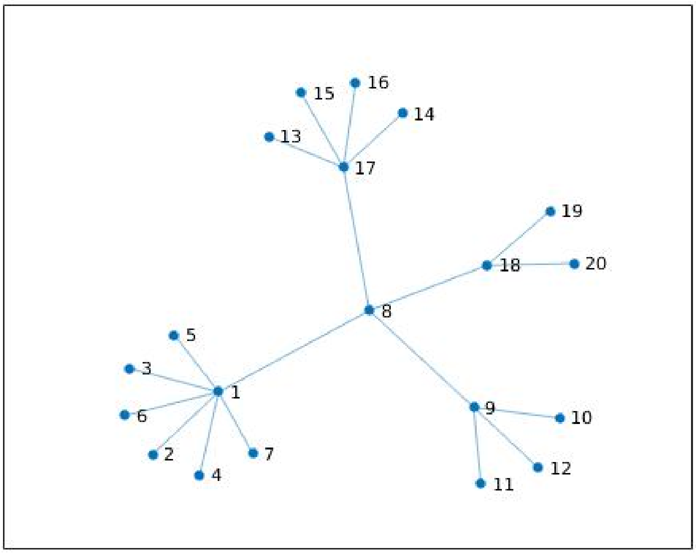


Figure 4.8. A network with 20 nodes and 19 edges.

Centrality number	1	2	3	4	5	6	7	8	9	10
Centralities	**3.0454**	1.5270	1.5270	1.5270	1.5270	1.5270	1.5270	**2.8588**	**2.1745**	1.3763
Centrality number	11	12	13	14	15	16	17	18	19	20
Centralities	1.3763	1.3763	1.4231	1.4231	1.4231	1.4231	**2.4451**	**1.9212**	1.3324	1.3324

Table 4.4. Katz centralities for the network in Figure 4.8.

(4.4). We again choose weights $\mathbf{x}$ to follow a uniform distribution on the interval $(0, 1)$. We then choose $M = 10^3$ samples and finite differences. Using the active subspace method we then obtain eigenvalues and eigenvectors. The components of the leading eigenvector are shown in Figure 4.9 and the ordering of the components is presented in Table 4.5. By looking at the leading eigenvector, it was found that 17 edges (7 edges, in particular, see Figure 4.9 and Table 4.5) between nodes were more important than the others and they are highlighted in Figure 4.10. Namely, edges with an index $1 - 8, 20 - 26, 16$ and 17. The network in Figure 4.8 is symmetric and undirected, which means that edges such as $1 \to 2$ and $2 \to 1$ are equivalent (equal weights). We can also see that these edges are not equally important to the quantity of interest. We note here again that the Katz centrality is based on broadcasting information out, which causes the differences between equivalent edges seen in Figure 4.9.

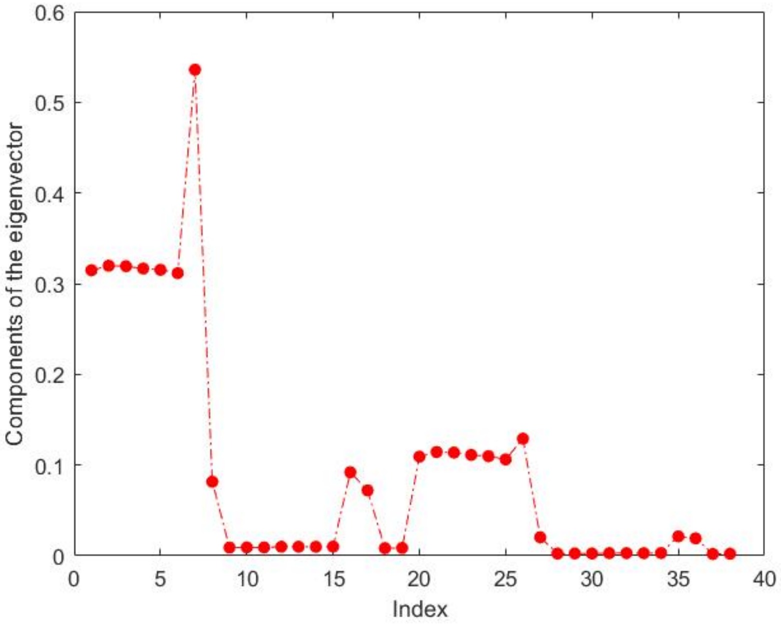


Figure 4.9. The leading eigenvector for the network in Figure 4.8.

Index	1	2	3	4	5	6	7	8	9	10	11	12	13
Edge	$1 \to 2$	$1 \to 3$	$1 \to 4$	$1 \to 5$	$1 \to 6$	$1 \to 7$	$1 \to 8$	$8 \to 9$	$9 \to 10$	$9 \to 11$	$9 \to 12$	$17 \to 13$	$17 \to 14$

Index	14	15	16	17	18	19	20	21	22	23	24	25	26
Edge	$17 \to 15$	$17 \to 16$	$8 \to 17$	$8 \to 18$	$18 \to 19$	$18 \to 20$	$2 \to 1$	$3 \to 1$	$4 \to 1$	$5 \to 1$	$6 \to 1$	$7 \to 1$	$8 \to 1$

Index	27	28	29	30	31	32	33	34	35	36	37	38
Edge	$9 \to 8$	$10 \to 9$	$11 \to 9$	$12 \to 9$	$13 \to 17$	$14 \to 17$	$15 \to 17$	$16 \to 17$	$17 \to 8$	$18 \to 8$	$19 \to 18$	$20 \to 18$

Table 4.5. Ordering of the edges in Figure 4.9.

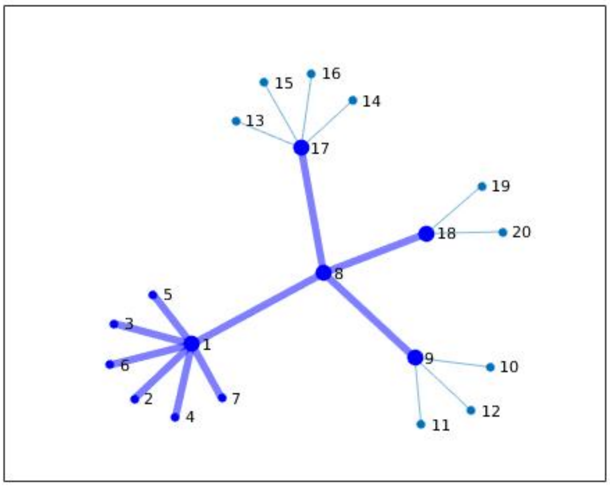


Figure 4.10. Highlighted dominant edges.

Consider the next synthetic network shown in Figure 4.11. This is a similar network to the network in the previous example. However, here we have removed node 1 and its edges from the network in order to illustrate what happens to the rest of the network.

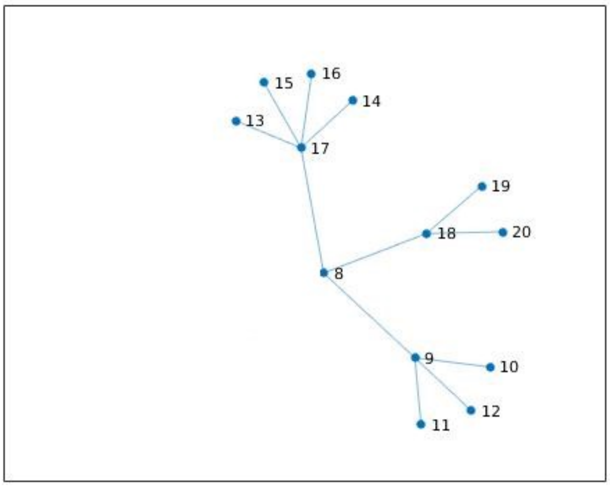


Figure 4.11. A network with 13 nodes and 12 edges.

The Katz centralities for the network in Figure 4.11 are given in Table 4.6. Note that the centralities in Table 4.6 are calculated using an unweighted adjacency matrix A. Looking at Table 4.6 we can see that nodes 8, 9, 17, 18 have the highest centralities. Again, we choose our weights on edges to follow a uniform distribution on $(0, 1)$, the number of samples to be $M = 10^3$ and use finite differences for the approximation of partial derivatives.

Centrality number	8	9	10	11	12	13	14	15	16	17	18	19	20
Centralities	**2.4367**	**2.3774**	1.4768	1.4768	1.4768	1.5476	1.5476	1.5476	1.5476	**2.7304**	**2.0553**	1.4122	1.4122

Table 4.6. Katz centralities for the network in Figure 4.11.

We also let our quantity of interest be

$$f(\mathbf{x}) = \left((I - \alpha A(\mathbf{x}))^{-1} \mathbf{1}\right)_{18}. \tag{4.5}$$

Equation (4.5) represents a quantity of interest, which is the Katz centrality of node 18, in this case. Following the procedure of the active subspace method as in previous examples, we examine the leading eigenvector, which corresponds to the biggest eigenvalue. The components of the leading eigenvector are shown in Figure 4.12 and the ordering of the components is presented in Table 4.7.

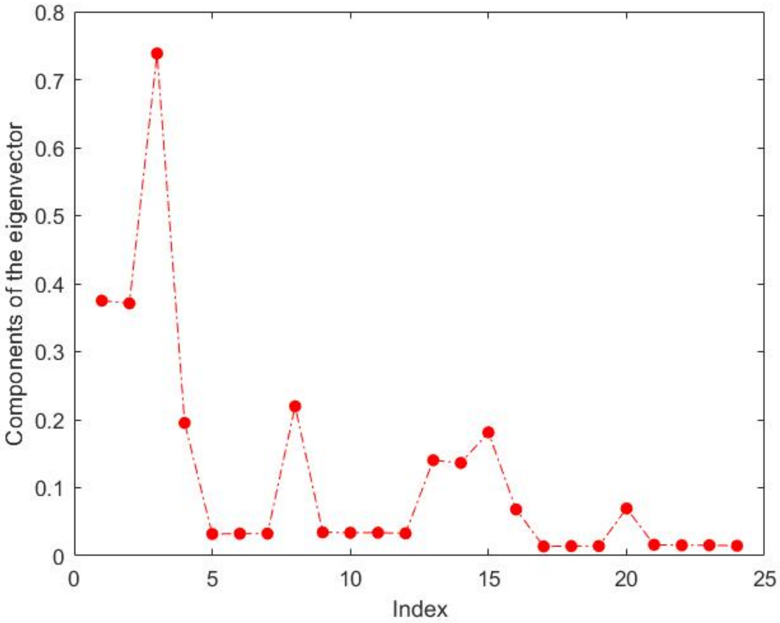


Figure 4.12. The leading eigenvector for the network in Figure 4.11.

Index	1	2	3	4	5	6	7	8	9	10	11	12
Edge	18 → 19	18 → 20	18 → 8	8 → 9	9 → 10	9 → 11	9 → 12	8 → 17	17 → 13	17 → 14	17 → 15	17 → 16

Index	13	14	15	16	17	18	19	20	21	22	23	24
Edge	19 → 18	20 → 18	8 → 18	9 → 8	10 → 9	11 → 9	12 → 9	17 → 8	13 → 17	14 → 17	15 → 17	16 → 17

Table 4.7. Ordering of the edges in Figure 4.12.

By looking at the components of the leading eigenvector, it was found that the edges between nodes $(18, 19)$, $(18, 20)$, $(18, 8)$ and $(8, 17)$ are the most important edges in this case. Note that the edge between nodes 8 and 9 is not as important as in the previous examples since node 8 lost an important node from the previous case (i.e., node 1). If we investigate edges with an index $1 - 3$ and $13 - 15$ further using Figure 4.12 and Table 4.7, we can see that these are equivalent edges. However, they are clearly not equally important to the quantity of interest due to the nature of Katz centrality.

We now consider a final network, taken from [72]. Unlike our previous examples, this network (shown in Figure 4.13) does not have a tree structure. As before, we

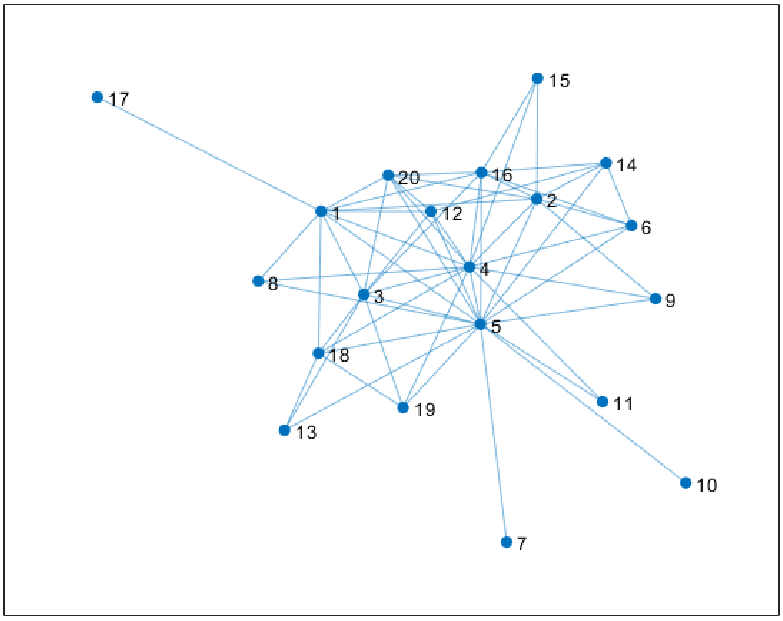


Figure 4.13. A network with 20 nodes and 59 edges.

choose weights on the edges to follow a uniform distribution on $(0, 1)$, the number of samples to be $M = 10^3$ and use finite differences for the approximation of partial derivatives. We also let our quantity of interest be

$$f(\mathbf{x}) = \left((I - \alpha A(\mathbf{x}))^{-1} \mathbf{1}\right)_{19}, \tag{4.6}$$

that is the Katz centrality of node 19. We have chosen this node because, looking at Figure 4.13 we see that there are four connections to node 19, namely, nodes 18, 3, 4 and 5. We therefore expect the active subspace method to identify the links between these connections as important.

Following the procedure of the active subspace method, we examine the leading eigenvector, which corresponds to the biggest eigenvalue. The components of the leading eigenvector are shown in Figure 4.14. The components corresponding to the edges $(19, 3)$, $(19, 4)$, $(19, 5)$ and $(19, 18)$, which are components $27, 37, 58, 107$, respectively, have again been identified by the active subspace method. This provides some evidence that we can also apply the active subspace method successfully to non-tree networks.

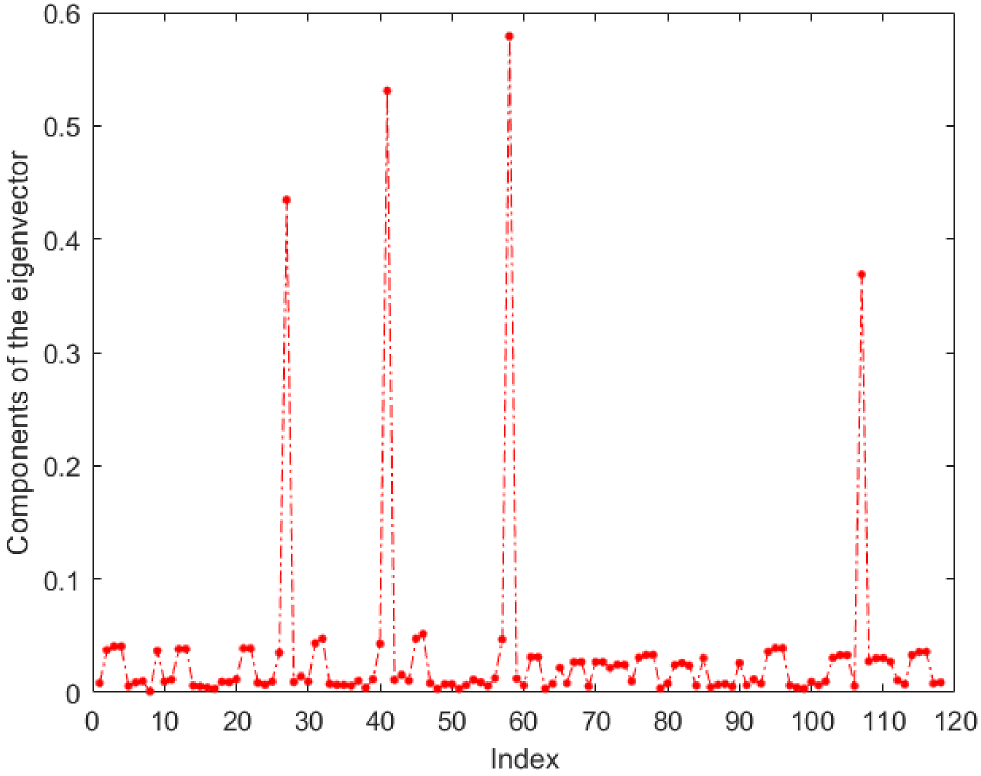


Figure 4.14. The leading eigenvector for the network in Figure 4.13.

In § 3.2.4 we briefly discussed edge betweenness centrality. We use an unweighted and undirected Figure 4.13 network, and calculate the edge betweenness centralities in order to compare the results against the active subspace method. This is done to see whether there are any similarities and/or differences between our method and the edge betweenness centrality. We note, however, that the active subspace method and the edge betweenness centrality are answering different questions: we use the active subspace method to find important edges with respect to the quantity of interest (i.e., the centrality of a node of interest) whereas the edge betweenness centrality identifies the most sensitive edges that affect the communication the most in the network.

Edge	(1, 17), (17, 1)	(5, 7), (7, 5)	(5, 10), (10, 5)	(5, 13), (13, 5)	(5,11), (11, 5)
EB Centrality	19	19	19	12.3	11

Table 4.8. Ten largest edge betweenness centralities for the network in Figure 4.13.

The ten largest edge betweenness centralities for the network in Figure 4.13 (calculated using code taken from [70]) are shown in Table 4.8. These results are quite different than the results obtained above using the active subspace method: this is because we used the active subspace method to identify important edges with respect to Katz centrality of node 19, whereas the edge betweenness centrality indicates the most sensitive edges that affect the communication the most in the whole network. We can also produce a single plot with results obtained by both methods to better understand the differences (see Figure 4.15). We can clearly see that the edges identified as important by the active subspace method differ from the edges identified as important by the edge betweenness centrality. To make a better comparison, we now use the active subspace method with respect to the node with the largest Katz centrality in the network. A list of Katz centralities can be seen in Table 4.9 and we see that node 5 has the largest Katz centrality.

Centrality number	1	2	3	4	5	6	7	8	9	10
Centralities	3.2946	3.0662	3.1203	3.9279	**4.2908**	2.3340	1.3433	1.9211	1.9028	1.3433
Centrality number	11	12	13	14	15	16	17	18	19	20
Centralities	1.6575	2.5794	1.7915	2.2341	1.8120	3.1564	1.2636	2.4825	2.1057	2.8748

Table 4.9. Katz centralities for the network in Figure 4.13.

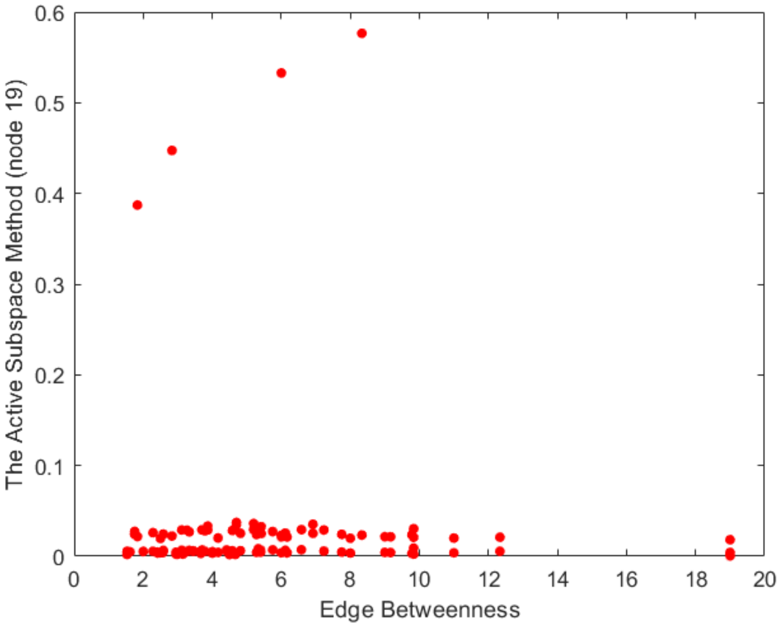


Figure 4.15. Comparison of the results from the active subspace method with respect to node 19 and edge betweenness centrality for the network in Figure 4.13.

We use the active subspace method with respect to Katz centrality of node 5 and apply symmetric weights on the edges in order to further reduce the bias between the two methods (we use edge betweenness centrality on an unweighted symmetric network). We showcase the results of both methods in a single plot. In Figure 4.16, we now see that there is some correlation in the results of both methods. The least important edges identified by the active subspace method are also identified by edge betweenness centrality. We also see that the important edges with respect to the active subspace method overlap with the important edges identified by the edge betweenness centrality. However, there are some edges that were identified as important by the edge betweenness centrality but not by the active subspace method, namely, edges $(1, 17)$, $(17, 1)$, $(4, 11)$, $(11, 4)$, $(4, 15)$, $(15, 4)$. We note that these edges do not involve node 5, which was the focus of the active subspace method.

When comparing the results of the active subspace method and edge betweenness centrality, there was no evidence to suggest that, in general, there are similarities in the results of the two methods. However, given the correct setting (focusing on the most Katz-central node and using symmetric weights on the edges for the active subspace method), we find some similarities in the results of the two methods.

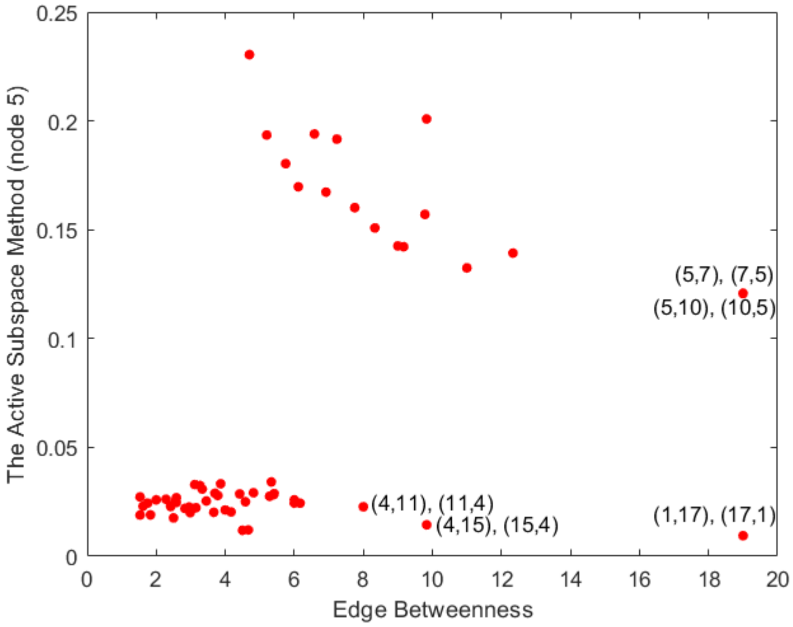


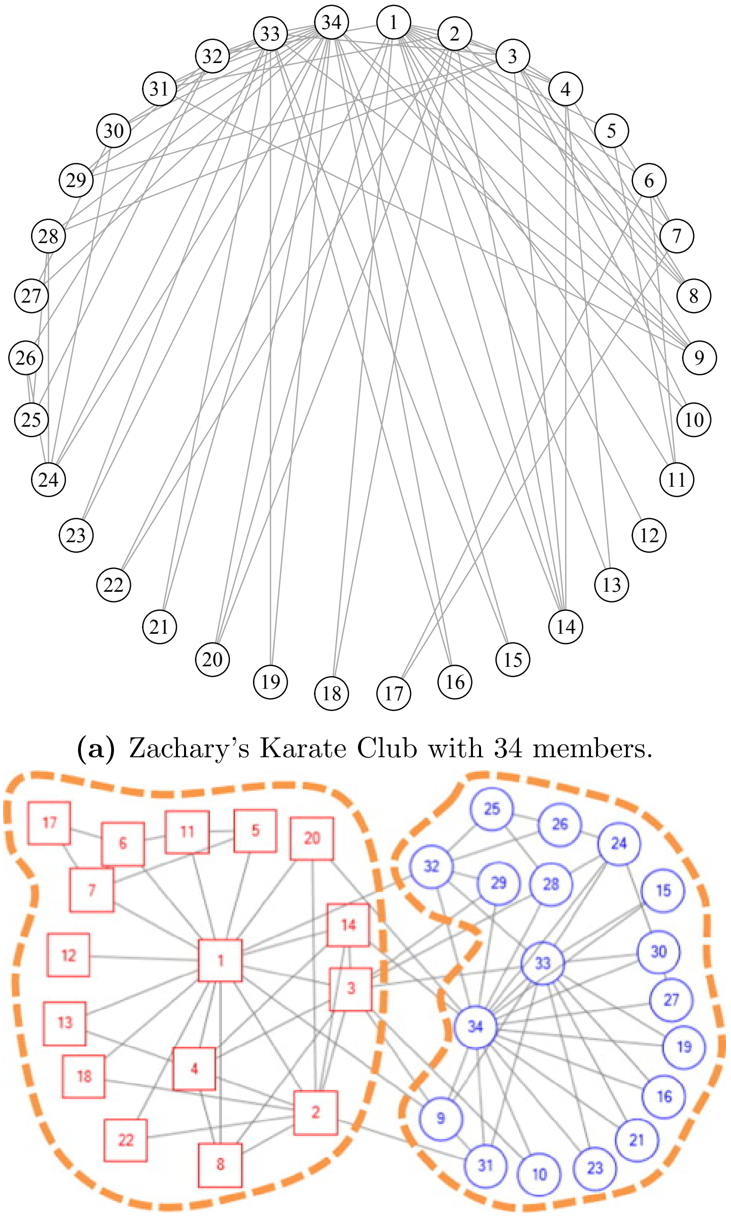
Figure 4.16. Comparison of the results from the active subspace method with respect to node 5 and edge betweenness centrality for the network in Figure 4.13.

4.4 Zachary's Karate Club Network

In this section, we consider a social network model of relationships in Zachary's Karate Club, which is a well-known example in the literature of social networks [73]. Its origins can be summarised as follows.

Zachary's Karate Club was a club with 34 members and 78 pairwise links between members who interacted outside the club (Figure 4.17a). The karate club was studied from 1970 till 1972 and towards the beginning of the study there was a conflict between the club president (say, John A) and the karate instructor (say, Mr. Hi) [73]. The disagreement between the two main figures in the club arose due to the price of karate lessons. Mr. Hi wanted to raise the price of lessons and set his own fees but the club president (John A) did not approve (he had the authority to set the fees). As time went by, the club became divided because of the disagreement between Mr. Hi and John A. The instructor attempted to raise the fees for lessons without permission and was fired by the club president. After these events, two groups formed: people who supported Mr. Hi and people who supported John A. This situation is represented in Figure 4.17b (club after fission), where Node 1 represents Mr. Hi and Node 34 represents John A.

The Karate Club data set has been widely studied because of the availability of additional information, which allows the results of clustering or centrality algorithms to be judged and compared.



(a) Zachary's Karate Club with 34 members.

(b) Zachary's Karate Club after fission.

Figure 4.17. Graphical representation of Zachary's Karate Club.

By looking at Figure 4.17a, we can immediately see that Nodes 34 and 1 (and possibly node 33) are the most dominant ones. Our task is to determine which edges (links) have the most influence on these nodes (i.e., which connections are the most important to John A and Mr. Hi). Since we have 34 nodes, we have a

34×34 adjacency matrix. As before, we will add weights on the edges, which follow a uniform distribution between 0 and 1. The matrix A is symmetric in nonzero structure but not symmetric in general. This means that if node 1 is connected to node 2, then node 2 is connected to node 1 and so on. However, the weights on the edges are different. In this case, we have 156 links with different weights. We will consider two quantities of interest: Katz centrality of Node 34 and Katz centrality of Node 1.

To use the active subspace method, we need to calculate partial derivatives. In order for us to use expression (3.7) in Lemma 3.3.3, we need to have a fixed α. The examples in previous sections consider a value $\alpha < 1/||A_{adj}||_2$ (or $\alpha < 1/\rho(A_{adj})$), where A_{adj} is the adjacency matrix of A and $|| \cdot ||_2$ is the Euclidean norm. And so we fix $\alpha = 1/8$ which satisfies the condition $\alpha < 1/||A_{adj}||_2 = 0.1487$.

Once we define our quantities of interest, fix α, choose the number of samples M (say, $M = 10^3$) and form weighted matrices A, we are then able to obtain the eigenvalue decomposition (see Algorithm 1) and find the leading eigenvector with respect to each quantity of interest. These two eigenvectors are plotted against each other in Figure 4.18. Each eigenvector corresponds to a quantity of interest (Katz centralities of Node 1 and Node 34). The entries of the eigenvectors indicate the importance of each edge to each quantity of interest.

In Figure 4.18 the x-axis is represented by John A (Node 34) and y-axis is represented by Mr. Hi (Node 1). We have also divided the scatter-plot points into categories. We define group A as the group of people who joined Mr. Hi after fission and we define group B as the group of people who joined John A after fission. We emphasize that the algorithm did not use this information.

If a node from group A is connected to a node from group A, then that connection (edge or link) is plotted with a blue colour. If a node from group A is connected to a node from group B, then that connection is plotted with a red colour. If a node from group B is connected to a node from the same group, then that link is plotted with a green colour. The reason for doing this is that we want to judge how well the active subspace method can determine the most important links and how well it can explain the grouping of members after fission. We also draw green and blue dashed lines to separate five highest values from either group A or group B. These represent the edges $1 \rightarrow 3, 1 \rightarrow 2, 1 \rightarrow 9, 1 \rightarrow 14, 1 \rightarrow 4$ (Mr.

Hi) and $34 \to 33, 34 \to 32, 34 \to 31, 34 \to 24, 34 \to 16$ (John A).

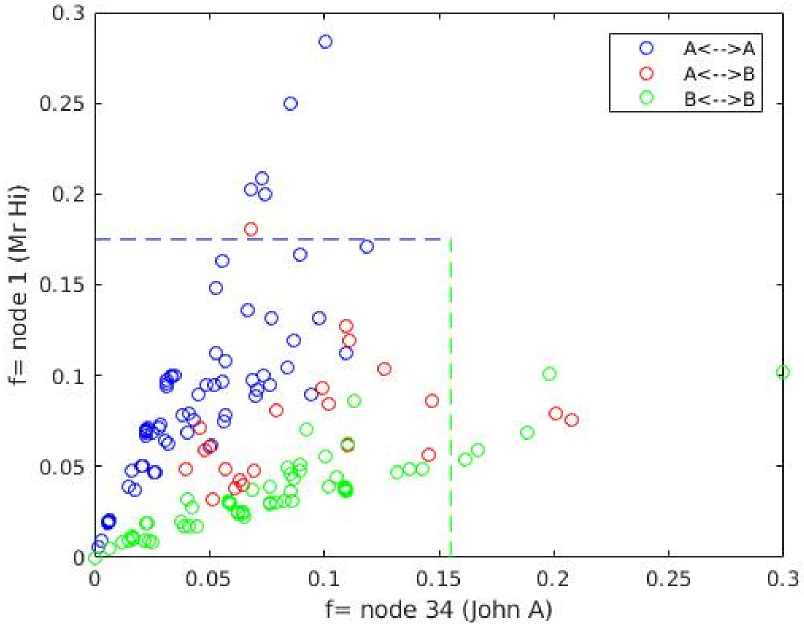


Figure 4.18. Two leading eigenvectors arising from quantities of interest based on the centrality of Node 34 (John A) and Node 1 (Mr. Hi).

The scatter-plot in Figure 4.18 indicates that there are quite a few strong links to John A and there are quite a few strong links to Mr. Hi (which lie outside separated with dashed lines). The other connections closer to the origin represent weak links to John A, Mr. Hi or links between other karate members. The green points further along the x-axis are the links of people with the club president that supported him and joined his side after the fission. Similarly, the blue points further on the y-axis are the links of people with the karate instructor that supported him and joined his side after the fission. Consequently, these links have the most influence for either Node 34 or Node 1. In other words, this means that these connections affect our quantities of interest the most (either the Katz centrality of Node 34 or the Katz centrality of node 1). The red points furthest on the right on the x-axis and furthest up on the y-axis are "outliers". We call these points "outliers" here because they appear to be important to both Node 1 and Node 34. In addition to this, the red points furthest on the right on the x-axis seem to be important links to Node 34, however, these links represent the people that joined Mr. Hi after the fission. The outlying red point on the y-axis can be interpreted in a similar manner. Nevertheless, this might represent the fact that these links were important to either John A or Mr. Hi in building their separate clubs and that these connections could have been deciding factors for some other members of the karate club (i.e., the ones that were undecided who to join).

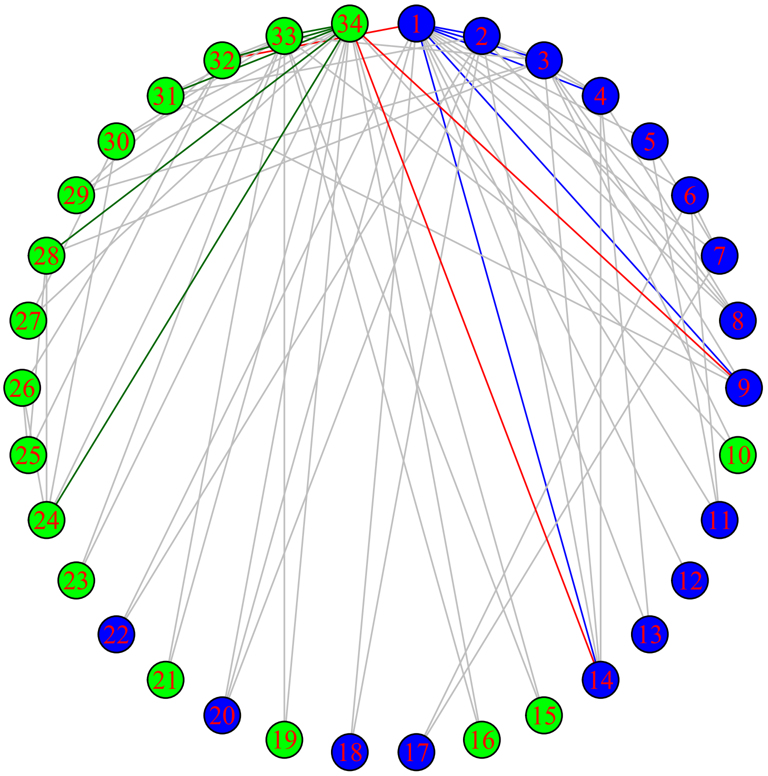


Figure 4.19. The first five important links to either Node 34 (five green links) or Node 1 (five blue links) for Zachary's Karate Club network.

Figure 4.19 illustrates the most important links (the first five influential links from Figure 4.18 are highlighted) to Node 34 (John A) and Node 1 (Mr. Hi). The edges which are important to both leaders as identified by active subspace analysis are coloured in red. We emphasize the fact that nodes 9 and 14 have important connections to either Node 1 or 34. This means that we are not able to predict with confidence whether these club members are joining Mr Hi (Node 1) or John A (Node 34).

Zachary used the Netflow computer language program [73] to carry out the maximum flow-minimum cut labeling procedure [74] in order to predict if a member of a karate club would join either Mr. Hi or John A after fission. He managed to predict correctly each person except for one, i.e., node 9 (see Figure 4.17b). Our active subspace algorithm has highlighted important links to the key players (Nodes 1 and 34) that generally match the results obtained by Zachary. The difference in results is due to the fact that we are trying to answer a slightly different question. Zachary specifically carried out a prediction procedure to find out which club members are joining either Mr Hi or John A. On the other hand, we use the active subspace method to find the most important or impactful connections between the club members.

It is sensible to assume that the most important connections (green points further on the x-axis and blue points further on the y-axis) could have influenced other people to join either Mr. Hi or John A after the fission. The active subspace method has allowed us to find important parameters (links in this case) or rather a combination of parameters. In this example we established that five links to either Node 34 or Node 1 are the most important links.

We also use the Zachary's Karate Club network and calculate the edge betweenness centralities to make a comparison with the active subspace method. The twenty largest edge betweenness centralities for Figure 4.17a are shown in Table 4.10.

Edge	(1, 32), (32, 1)	(1, 6), (6, 1)	(1, 7), (7,1)	(1, 3), (3, 1)	(1, 9), (9, 1)
EB Centrality	71.4	43.8	43.8	43.6	41.7
Edge	(3, 33), (33, 3)	(14, 34), (34, 14)	(20, 34), (34, 20)	(27, 34), (34, 27)	(32, 34), (34, 32)
EB Centrality	38.7	38.1	33.3	30.1	30

Table 4.10. Twenty largest edge betweenness centralities for the network in Figure 4.17a.

We will use the active subspace method with respect to both quantities of interest, i.e., Katz centrality of node 34 and Katz centrality of node 1. Also, we will use symmetric weights on the edges to reduce the bias in our analysis. Figure 4.20 shows the results obtained by the active subspace method with respect to Katz centrality of node 34 and edge betweenness centrality. We see a relationship in the results emerge, i.e., the important edges identified by the active subspace method are also identified by edge betweenness centrality. However, we also see that there are some edges that were identified as important by edge betweenness centrality but not by the active subspace method, namely, edges $(1, 32)$ and $(32, 1)$. On the other hand, we see that there are some edges that were identified as important by the active subspace method but not by edge betweenness centrality, namely, edges

(33, 34) and (34, 33).

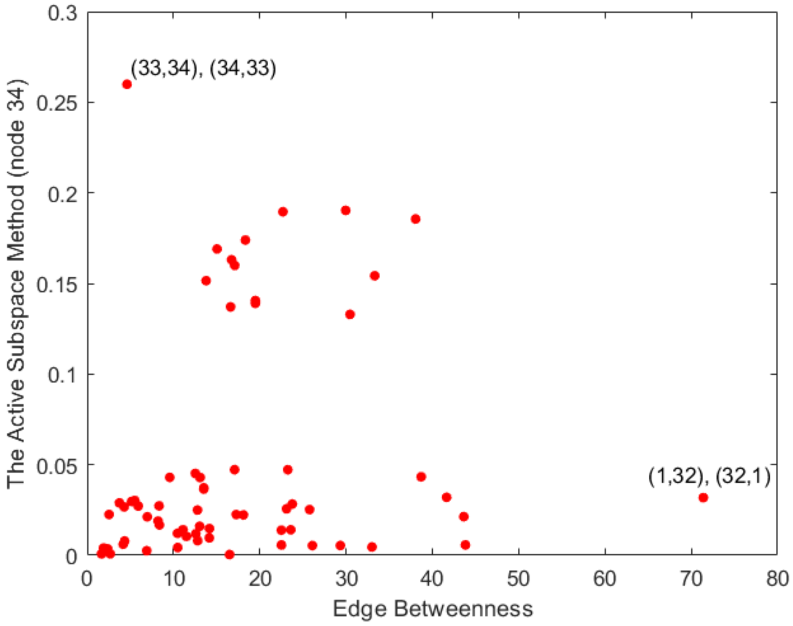


Figure 4.20. Zachary's Karate Club network: comparison of the results from the active subspace method with respect to node 34 and edge betweenness centrality for the network in Figure 4.17a.

Figure 4.21 shows the results obtained by the active subspace method with respect to Katz centrality of node 1 and edge betweenness centrality. We can see there is an stronger relationship in the results than in Figure 4.20. In other words, we see that the edges identified as important by the active subspace method are also identified by edge betweenness centrality. The reason why we see such differences in the results in Figure 4.20 and Figure 4.21 is because of the way we choose our quantity of interest for the active subspace and because both methods are trying to answer slightly different questions.

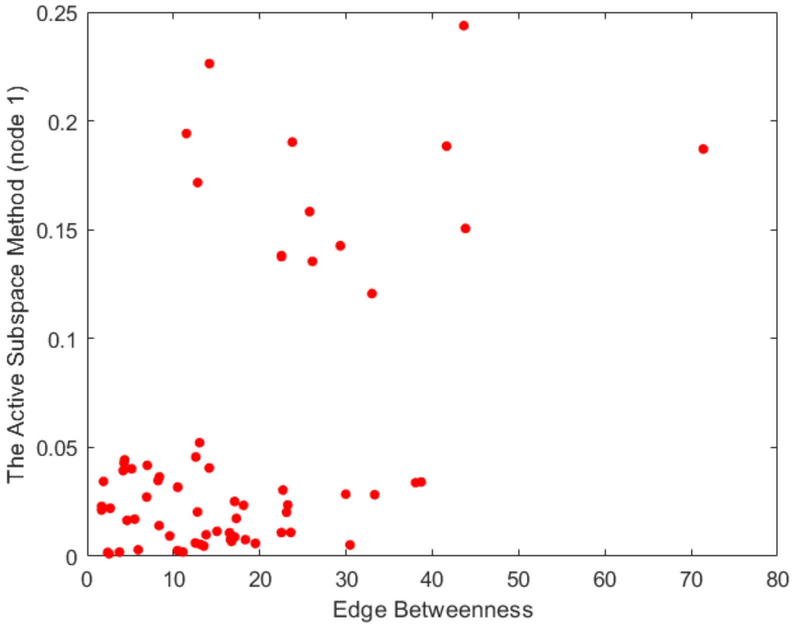


Figure 4.21. Zachary's Karate Club network: comparison of the results from the active subspace method with respect to node 1 and edge betweenness centrality for the network in Figure 4.17a.

4.5 Background on PPI Networks

Research on large-scale biological data sets has been evolving and progressing over the years [75]. It comes to no surprise that biological problems have been studied by computational scientists. In this section we focus on a protein-protein interaction (PPI) network. It is important to note that such networks usually do not do justice to biological reality. However, it has been found that focusing on a very high-level interaction network can lead to useful insights [75, 76, 77]. Nodes represent proteins and edges represent the experimental observation that two proteins can physically connect. In many biological cases, a PPI network has thousands of nodes and edges. Any network can be represented as a graph. It is then possible to represent the graph using an adjacency matrix $A \in \mathbb{R}^{N\times N}$, where $a_{ij} = a_{ji} = 1$ if proteins i and j have an interaction and $a_{ij} = a_{ji} = 0$ if proteins i and j do not have an interaction. In this thesis, we discuss the interaction network of yeast shown in Figure 4.22 [78].

In Figure 4.23 we show the adjacency matrix of the data set that will be used later. A nonzero in the matrix is represented as a dot in the figure. PPI data sets are available in the public domain. However, some experience is required to handle the data and draw out appropriate information [76], [77]. In the following subsection we will use such data to extract information using an active subspace method.

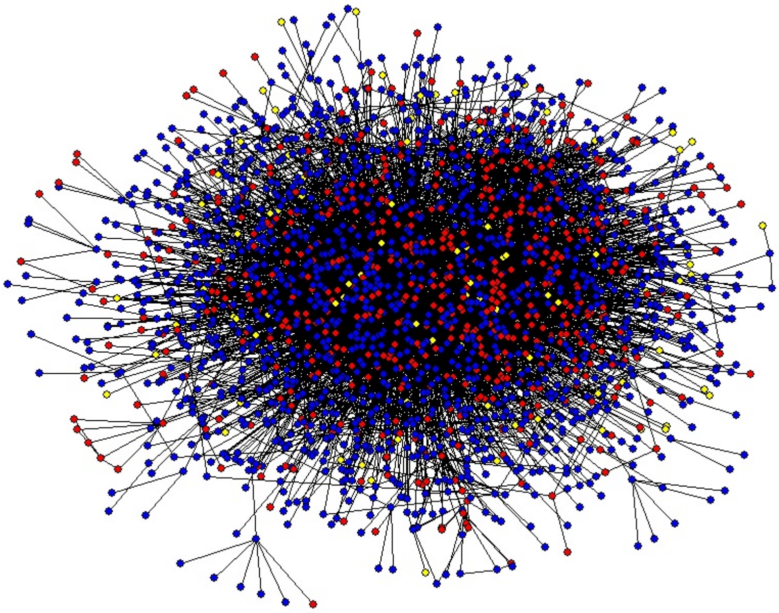


Figure 4.22. The complete protein interaction network of yeast showing essential proteins in red and non-essential ones in blue, yellow circles correspond to proteins with unknown essentiality [78].

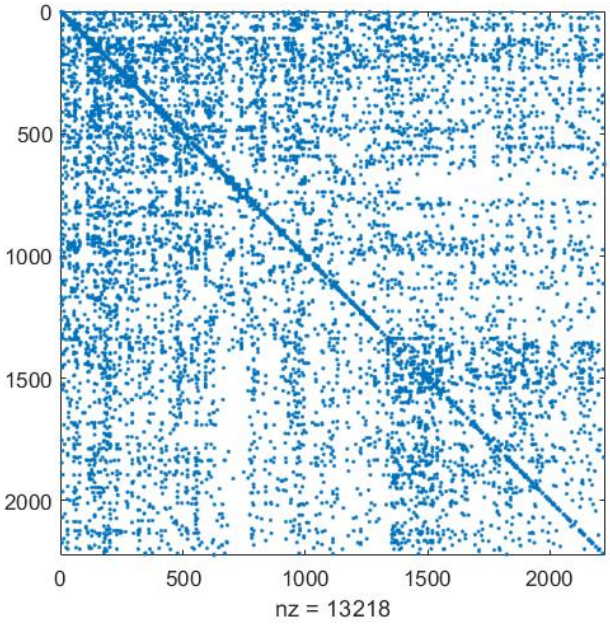


Figure 4.23. Adjacency matrix of a PPI network.

4.5.1 PPI Network and the Active Subspace Method

Our PPI network of interest has 2224 nodes with 13218 egdes. We also have additional information about the individual proteins. The nodes (proteins) are represented as either important (essential) or not (non-essential). The difference between essential and non-essential proteins is that non-essential proteins (or amino acids that combine to form proteins) are made by the body, while essential proteins can only be obtained through foods (e.g., beans, nuts, dairy, cottage cheese, fish, turkey) [79]. The active subspace method will be used to determine which connections have the most influence on the quantity of interest. In this particular case, we choose our quantity of interest to be the sum of 30 highest Katz centrality values of the network. The active subspace method was used with $M = 100$, i.e. we have 100 samples of size 13218 (number of edges). Samples follow a uniform distribution between 0 and 1. The method produces 2224 eigenvectors that correspond to eigenvalues and we analyse the leading eigenvector with the highest eigenvalue. It is important to note that the leading eigenvector gives us the order of importance of edges. The procedure then often is to plot the leading eigenvector and inspect which edges or a combination of edges are the most influential. However, that is hard to determine in this case due to the fact that we have 13218 entries in the eigenvector.

One way to get past this problem is to use information we were given beforehand about this PPI network: we know which nodes (proteins) are either essential or non-essential. We can now form three groups or types of connections using this information. Let Type 1 represent a connection between an essential node that is connected to another essential node, Type 2 is for an essential node that is connected to a non-essential node and Type 3 is for a non-essential node that is connected to another non-essential node.

We are mainly interested in how well the active subspace method picks up Type 1 edges or connections between nodes. Since our leading eigenvector is ordered from the most important edge to the least important edge, we calculate the proportion of how many edges are Type 1, Type 2 or Type 3 (looking at each entry of the eigenvector and summing over) against the total number of Type 1, Type 2, Type 3 edges, respectively. Note that information about which proteins are essential is incomplete and inaccurate. Hence a method which can uncover information using only the interaction data is likely to be biologically useful.

We first consider randomly permuting the edges of the eigenvector and look at the proportion plots. We expect straight lines for all three cases, which are presented in Figure 4.24.

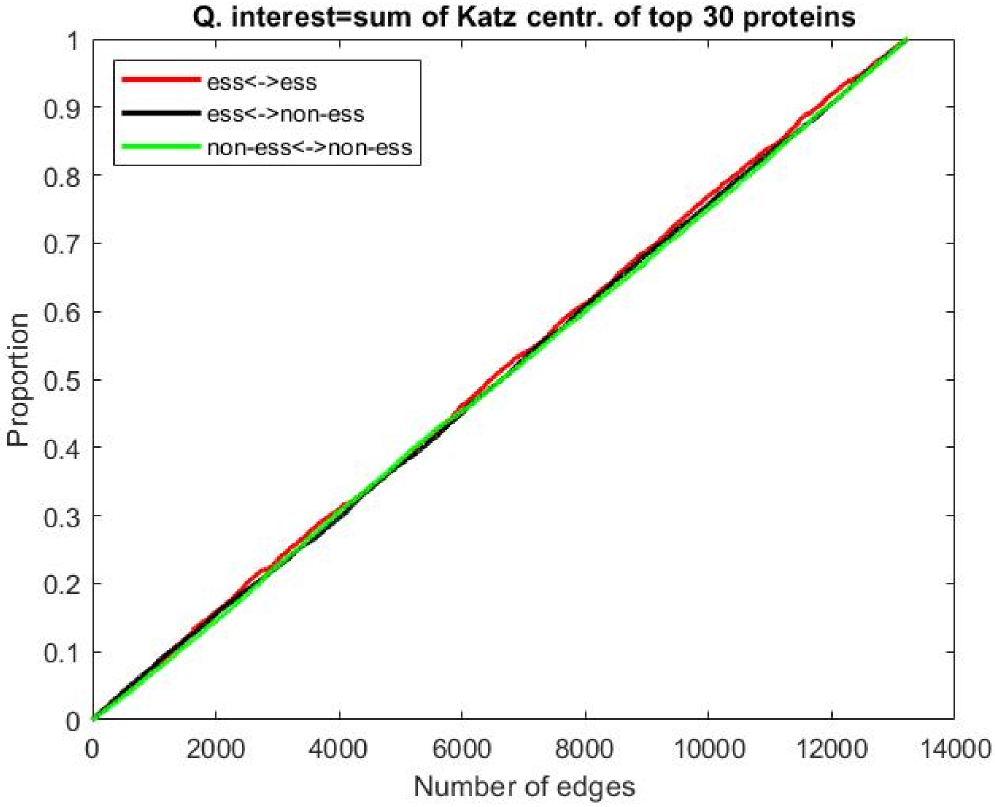


Figure 4.24. Proportions of three types of edges after permuting the eigenvector.

The reason for doing this is to understand how well the active subspace method is performing in comparison with a method that chooses links at random. Function "*randperm*" was used to permute the entries of the leading eigenvector and this was done 100 times in order to get satisfactory confidence intervals (or "envelopes").

Permuting an eigenvector 100 times requires us to calculate the proportion in the same way as described before. However, this time we get 100 values for each of 13218 times we calculate the proportion. Each time we pick the smallest and the highest of 100 permutations for all types of edges, which gives us lower and upper bounds. These are illustrated as dashed lines and can be observed in Figure 4.25. As expected, the dashed lines that represent confidence intervals for permuting the eigenvector are straight lines. This allows to say something about the active

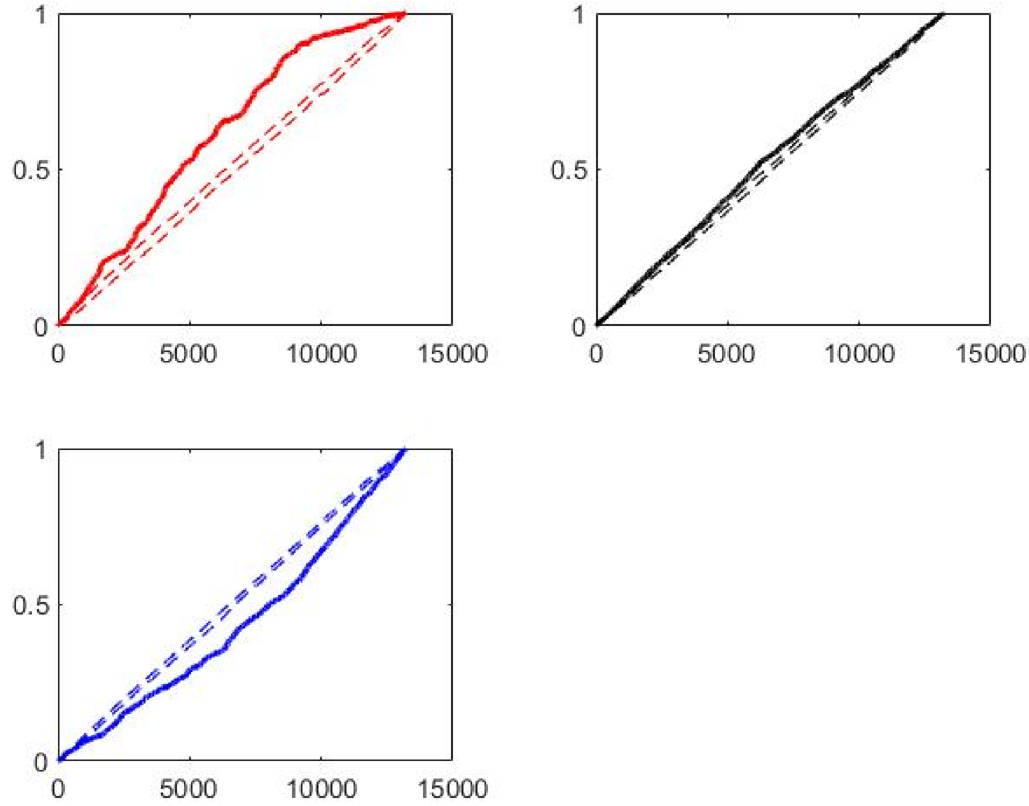


Figure 4.25. Proportions of three types of edges with confidence intervals (or "envelopes").

subspace method that was used for this particular problem. The first plot in Figure 4.25 has the solid line (the proportion of essential-essential protein connections) above the confidence interval, which suggests that the active subspaces method is doing well and that it picks up type 1 edges faster than the other types. The solid line in the second plot (the proportion of non-essential-essential protein connections) is close to the confidence interval which indicates the fact that type 2 edges are not impactful or important to our quantity of interest. The last plot in Figure 4.25 (the proportion of non-essential and non-essential protein connections) has the solid line below the confidence interval. This shows that the active subspace method picks up type 3 edges the slowest and specifies type 3 edges as the least important edges.

4.6 Summary

In this chapter, we studied the underlying theory of networks and centrality measures. The Katz centrality measure was studied in more detail compared with other centrality measures. This was done due to the fact that Katz centrality is a convenient and widely-used measure. Furthermore, we discussed gradient approximation when gradients are not available. Namely, we derived an exact derivative expression for Katz centrality in Chapter 3 and made a comparison with finite difference method in this chapter. In an example in § 4.2, it was shown and confirmed that the smaller step size h we have for the finite difference approximation, the closer we get to the true derivative. In practice we used the finite difference approximation because it was sufficiently accurate for our purposes.

Furthermore, we presented a number of numerical examples. The examples discussed in § 4.3 were illustrated in order to see if the active subspace method is applicable to synthetic networks. It was found that the active subspace method on networks was able to find important edge(s) or connections in a graph. In § 4.4 we analysed the famous Zachary's Karate Club network. It was found that the active subspace method correctly identified important links between club members. The results were consistent with the results obtained by Zachary. We also used the edge betweenness centrality on one of our synthetic networks and the Zachary's Karate Club network to see any similarities or differences between the results obtained using the active subspace method. We also saw that the results using the edge betweenness centrality on two of our test networks had some similarities with the active subspace results, but were not identical - this is explained by the fact that the two methods are addressing different questions. The last network we considered was a protein-protein interaction network. It was established that the connections or links between essential proteins were picked up faster by the active subspace method compared to non-essential connections or connections between essential and non-essential. This suggests that the active subspace approach is able to extract useful information about which proteins are essential using only topological information. Such information may be useful, for example, in the study of organisms where little is known about which proteins are essential.

Chapter 5

Comparison with Sobol Indices

5.1 Background Theory on Sobol Indices

The Sobol method or the use of Sobol indices (also known as variance-based sensitivity analysis) is a form of global sensitivity analysis [3, 4, 6, 15, 16, 17, 18, 80]. The idea of this method is that it decomposes the variance of the output of the model into fractions which can be assigned to inputs or sets of inputs. For example, given a model with two inputs and one output, one might find that 70% of the output variance is caused by the variance in the first input, 20% by the variance in the second, and 10% due to interactions between the two. These percentages can then be directly interpreted as measures of sensitivity. The Sobol method takes account of sensitivity across the whole input space (i.e., it is a global method). It can deal with nonlinear responses and it can measure the effect of interactions between the inputs [6, 81]. The purpose of the Sobol method is to determine the most important input parameter, or combination of parameters (i.e., interaction between parameters), that affects the output. The main idea of the method is to analyse the effect of varying one model input variable at a time while keeping all other variables fixed. Moreover, we then inspect variables that can have high influence on the output [81, 82]. One drawback of the method is that it is computationally expensive [7]. Nonetheless, it is a powerful tool for performing sensitivy analysis.

In this chapter we give a mathematical explanation of the Sobol method. We then apply the Sobol method to a number of synthetic examples in order to showcase the method. Finally, we apply the Sobol method to some of the examples considered in Chapter 4 with the active subspace method. We can then directly compare the results obtained from the Sobol method and the active subspace method.

5.2 Introduction to Variance-Based Methods

The idea of the variance-based methods is to quantify the amount of variance that each input parameter X_i contributes to the variance of the output $V(Y)$. The main references for this section are [16, 17, 18, 83].

Suppose we have a model

$$Y = f(\mathbf{X}), \tag{5.1}$$

where $\mathbf{X} = [X_1, \ldots, X_N]^T$ is a vector of inputs that follow a known probability distribution, N is the number of input parameters, Y is a scalar output and f is a "black-box". We want to rank the input parameters according to the amount of variance with respect to the output Y. We say that $V(Y|X_i = x_i)$ is the conditional variance of Y given $X_i = x_i$: it is obtained by taking the variance over all factors but X_i. However, we rarely know the value x_i for each X_i. This is why the average of this conditional variance for all possible values x_i of X_i is used, i.e., $E[V(Y|X_i)]$, which is the expectation value over the whole variation interval of the input X_i. Having the unconditional variance of the output $V(Y)$, assuming that X_i are independent and by using the property of total variance that

$$V(Y) = V[E(Y|X_i)] + E[V(Y|X_i)], \tag{5.2}$$

we obtain the variance of the conditional expectation $V_i = V(E[Y|X_i])$. This measure is sometimes called 'main effect' and is used as an indicator of the importance of X_i on the variance of Y. Normalising the main effect V_i by the unconditional variance of the output we obtain:

$$S_i = \frac{V[E(Y|X_i)]}{V(Y)}. \tag{5.3}$$

The ratio S_i was named the first order sensitivity index by Sobol [16, 17, 18, 83]. Various other names for this ratio can be found in the literature: 'importance measure', 'correlation ratio' and 'first order effect'.

The first order sensitivity index measures only the main effect contribution of each input parameter X_i on the output variance. It doesn't take into account the interactions between input factors. Two factors are said to interact if their total effect on the output isn't the sum of their first order effects. The effect of the interaction between two orthogonal factors X_i and X_j on the output Y, in terms of conditional variances, is

$$V_{ij} = V(E[Y|X_i, X_j]) - V(E[Y|X_i]) - V(E[Y|X_j]). \tag{5.4}$$

Here $V(E[Y|X_i, X_j])$ describes the joint effect of the pair (X_i, X_j) on Y . This effect is known as the second-order effect. Higher-order effects are expressed in a similar way. The total order sensitivity index S_{T_i} is defined as the sum of all indices relating to X_i (first and higher order).

Examples of variance-based methods include Sobol indices, Jansen's Winding Stairs technique, the Fourier Amplitude Sensitivity Test (FAST) and the Extended Fourier Amplitude Sensitivity Test (EFAST) [83]. In this thesis, we will consider only the Sobol indices in more detail. However, before we do that, we need to highlight a few assumptions.

The input factor space N is assumed to be the N-dimensional unit hypercube [83]:

$$\Omega^N = (\mathbf{X}|0 \leq X_i \leq 1; i = 1, \ldots, N). \tag{5.5}$$

The input factors are also assumed to be independent and orthogonal to each other, thus no correlation structure can be induced on the input factors. The expected value of the output $E(Y)$ can be evaluated by the N-dimensional integral:

$$E(Y) = \int_{\Omega^N} f(\mathbf{X})\rho(\mathbf{X})d\mathbf{X} = \int_{\Omega^N} f(\mathbf{X})d\mathbf{X},$$

where $\rho(\mathbf{X})$ is the joint probability density function, assumed to be uniform, for each input parameter.

5.3 Calculating Sobol Indices

Sobol [16, 17, 18, 83] decomposed the model function f into summands of increasing dimensionality as

$$f(\mathbf{X}) = f_0 + \sum_{i=1}^{N} f_i(x_i) + \sum_{l=1}^{N} \sum_{j=i+1}^{N} f_{ij}(x_i, x_j) + \cdots + f_{12\ldots N}(x_1, x_2, \ldots, x_N). \quad (5.6)$$

This representation of $f(\mathbf{X})$ holds if f_0 (which is the mean or the expectation of the output $E(Y)$) is a constant and the integrals of every summand in Equation (5.6) are orthogonal, i.e.,

$$\int_0^1 f_{i_s}(X_{i_1}, \ldots, X_{i_s}) dX_{i_N} = 0, \qquad 1 \leq N \leq s. \quad (5.7)$$

The total variance $V(Y)$ is defined as

$$V(Y) = \int_{\Omega^N} f^2(\mathbf{X}) d\mathbf{X} - f_0^2 \quad (5.8)$$

and the partial variances are computed from each of the terms in Equation (5.6) such that

$$V_{i_1 \ldots i_s} = \int_0^1 \cdots \int_0^1 f^2_{i_1 \ldots i_s}(X_{i_1}, \ldots, X_{i_s}) dX_{i_1} \ldots dX_{i_s}, \quad (5.9)$$

where $1 \leq i_1 < \cdots < i_s \leq N$ and $s = 1, \ldots, N$. The sensitivity indices are then obtained from

$$S_{i_1 \ldots i_s} = \frac{V_{i_1 \ldots i_s}}{V}.$$

The integrals in Equations (5.8) and (5.9) can be computed with a Monte Carlo method. For a given sample size M, the Monte Carlo estimate of f_0 is

$$\hat{f}_0 = \frac{1}{M} \sum_{m=1}^{M} f(\mathbf{X}_m), \quad (5.10)$$

where $\mathbf{X}_m$ is a sampled point in the input space Ω^N (sample matrix $\mathbf{X}$ is of size $M \times N$). The Monte Carlo estimate of the output variance $V(Y)$ is then

$$\hat{V}(Y) = \frac{1}{M} \sum_{m=1}^{M} f^2(\mathbf{X}_m) - \hat{f}_0^2. \quad (5.11)$$

The main effect of input factor X_i is then estimated as

$$\hat{V}_i = \frac{1}{M} \sum_{m=1}^{M} f(\mathbf{X}^{(H1)}_{m\sim i}, X^{(H1)}_{mi}) f(\mathbf{X}^{(H2)}_{m\sim i}, X^{(H1)}_{mi}) - \hat{f}_0^2.$$

We use two sampling matrices here, $\mathbf{X}^{(H1)}$ and $\mathbf{X}^{(H2)}$, both of size $M \times N$: note that $\mathbf{X}^{(H1)}_{m\sim i}$ means the full set of samples from $\mathbf{X}^{(H1)}$ except the ith one. Matrix $\mathbf{X}^{(H1)}$ is usually called the data base matrix while $\mathbf{X}^{(H2)}$ is called the resampling matrix [83]. And so, the first order sensitivity indices can be evaluated as

$$\hat{S}_i = \frac{\hat{V}_i}{\hat{V}(Y)}.$$

The Monte Carlo evaluation of the second order terms V_{ij} is shown in [84] to be

$$\hat{V}_{ij} = \frac{1}{M}\sum_{m=1}^{M} f(\mathbf{X}^{(H1)}_{m\sim i\sim j}, X^{(H1)}_{mi}, X^{(H1)}_{mj}) f(\mathbf{X}^{(H2)}_{m\sim i\sim j}, X^{(H1)}_{mi}, X^{(H1)}_{mj}) - \hat{V}_i - \hat{V}_j - \hat{f}_0^2.$$

This means that the second order sensitivity indices can be evaluated as

$$\hat{S}_{ij} = \frac{\hat{V}_{ij}}{\hat{V}(Y)}.$$

Higher order sensitivity indices can be derived in a similar fashion.

We can also calculate total order sensitivity indices which take into account first, second and higher order effects as in [12, 81, 82, 83, 84]:

$$\hat{T}_i = 1 - \frac{\hat{V}_{ci}}{\hat{V}(Y)}, \tag{5.12}$$

where $\hat{V}_{ci}$ is the 'complementary effect' of $\hat{V}_i$ and is calculated via

$$\hat{V}_{ci} = \frac{1}{M}\sum_{m=1}^{M} f(\mathbf{X}^{(H1)}_{m\sim i}, X^{(H1)}_{mi}) f(\mathbf{X}^{(H1)}_{m\sim i}, X^{(H2)}_{mi}) - \hat{f}_0^2. \tag{5.13}$$

In simpler terms, Equation (5.13) means that we calculate the 'complementary index' of X_i in order to obtain the total order sensitivity index in Equation (5.12).

5.4 Examples Using Sobol Indices

The Sobol method and the active subspace method can be used to tackle similar problems. Our aim here is to study and compare the two approaches. We will illustrate the Sobol method on examples from previous chapters where the active subspace method was used, and we will also compare both methods on some new problems. However, we will start with some simple cases in order to give a feel for how the Sobol method works.

5.4.1 Example 1: A Simple Function f

For the first simple example, we set

$$y = f(\mathbf{X}) = f(X_1, X_2, X_3) = X_1 + 5X_2X_3^2. \tag{5.14}$$

Intuitively, we expect X_2 and X_3 to have higher first order sensitivity indices compared to variable X_1. We would also expect X_3 to have the highest first order sensitivity index due to its exponent of 2. As the first variable does not interact with any other variables, we expect that X_1 should have a total order sensitivity index which is the same as its first order sensitivity index. Figure 5.1 shows the results of the Sobol method for (5.14) with 3 input parameters and $M = 10^4$ samples on a uniform distribution $(0, 1)$. The same data is presented in Table 5.1. The results indicate that parameters X_2 and X_3 are the most important parameters and that they are contributing the most to the output variance Y. Notice that total $\hat{T}_i \geq \hat{S}_i$ (i.e., the total sensitivity index is greater or equal to first order sensitivity index) for each variable and $\hat{T}_1 = \hat{S}_1$. This matches our intuitive predictions above. Figure 5.2 illustrates the second order sensitivity indices. The second order sensitivity indices are computed to be $\hat{S}_{1,2} = 0.0115$, $\hat{S}_{1,3} = 0.0115$, $\hat{S}_{2,3} = 0.1970$. This is consistent with the definition of f in (5.14), where we see X_2 and X_3 interact in a pairwise manner. It is worth mentioning that one of the properties of Sobol indices with respect to this example is that $1 = S_1 + S_2 + S_3 + S_{1,2} + S_{1,3} + S_{2,3} + S_{1,2,3}$. The

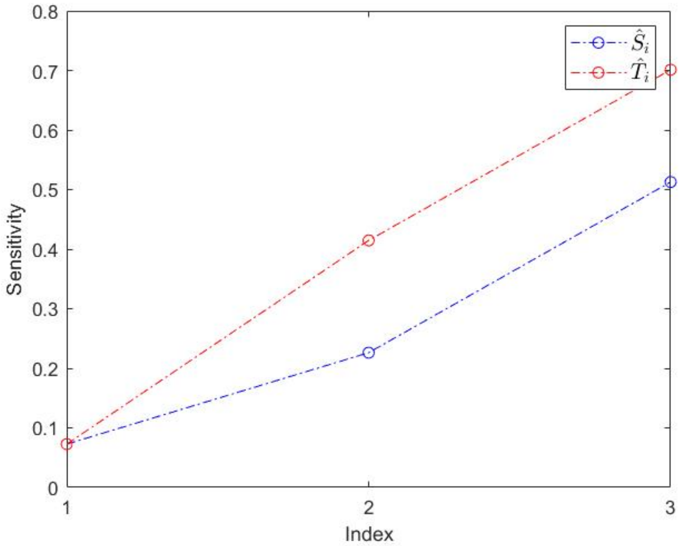


Figure 5.1. The importance of parameters to the output Y or contribution of each parameter to the variance of Y.

Sensitivity index	$\hat{S}_1$	$\hat{S}_2$	$\hat{S}_3$	$\hat{T}_1$	$\hat{T}_2$	$\hat{T}_3$
Value	0.0726	0.2262	0.5127	0.0726	0.4147	0.7012

Table 5.1. Approximate first and total order sensitivity indices for (5.14), as in Figure 5.1.

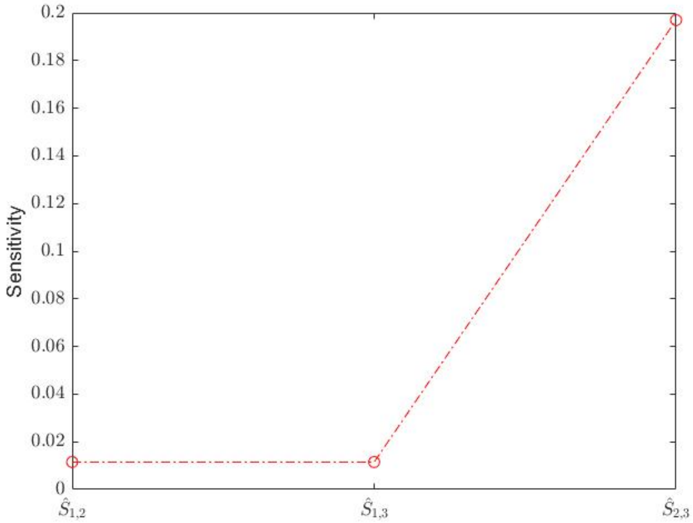


Figure 5.2. Second order sensitivity indices $\hat{S}_{1,2}$, $\hat{S}_{1,3}$, $\hat{S}_{2,3}$.

explanation of this equation is in [7, 82] and the proof is in [82]. This identity emphasizes that when we calculate first order indices, those indices usually do not sum up to one, as there are usually some higher-order interactions between the variables. Moreover, the total order sensitivity index should be higher than or equal to the first order sensitivity indices. For example, consider another Sobol indices property $T_1 = S_1+S_{1,2}+S_{1,3}+S_{1,2,3}$ (we stop at 3 because there are only 3 variables) [82]. This corresponds to the total order index with respect to variable X_1 (first order index with interactions that involve parameter X_1) . It is the sum of the first order index S_1 and interactions involving variable X_1. Intuitively, this means that T_1 has to be greater than or equal to S_1. The only time T_1 can be equal to S_1 is when we have no interactions with other variables, as in this example. However, in the later examples, we have cases when $\hat{S}_i > \hat{T}_i$ for the computational results. This is due to the fact that we are using Monte Carlo approximations. Further computation using more samples confirmed that the discrepancy is due to Monte Carlo error.

5.4.2 Example 2: Sobol G-function

In order to further demonstrate the Sobol method we consider a number of examples in this chapter. Consider the Sobol G-function

$$Y = \prod_{i=1}^{N} \frac{|4X_i - 2| + a_i}{1 + a_i}, \tag{5.15}$$

where $a_i = (0, 1, 4.5, 9, 99, 99, 99, 99)$ are the parameters that control the sensitivity of each variable X_i, $i = 1, \ldots, 8$. It is a well-known example in global sensitivity analysis algorithms [81]. In particular, exact values are known for the first order sensitivities (see Table 5.3). The importance of each input X_i is represented by the control parameters a_i. The smaller each control parameter a_i, the more impactful each variable X_i is. This is intuitive because when we have $a_1 = 0$, the scalar value of Y ranges between 0 and 2 if the variables of interest follow the uniform distribution on the interval $(0, 1)$. Similarly, when we have $a_8 = 99$, the scalar value of Y ranges between 0.99 and 1.02.

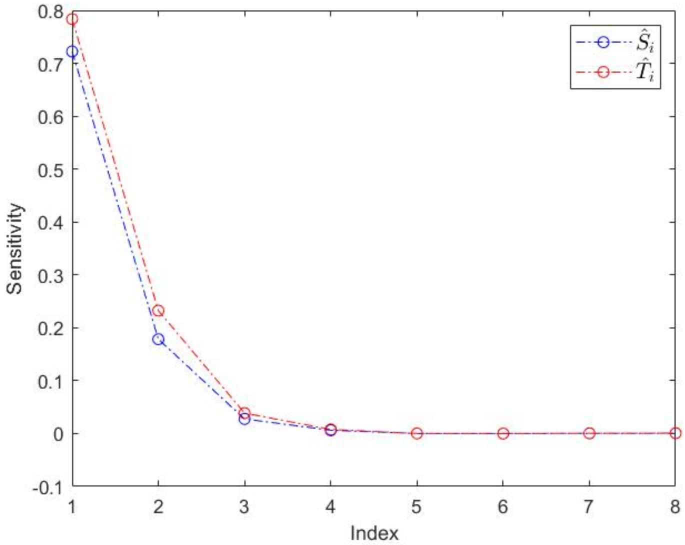


Figure 5.3. The importance of parameters: approximate first and total order sensitivity indices for Sobol G-function in (5.15).

As discussed in § 5.1, the total sensitivity indices for variables X_i are greater or equal to the first order sensitivity indices. This can be observed in Figure 5.3 and Table 5.2. Parameters X_1 and X_2 appear to have the most influence or contribute the most variance to the output Y, as expected. Figure 5.3 was obtained using 10^4 Monte Carlo samples. Figure 5.4 considers variable X_1 with 10^2, 10^3, 10^4, 10^5 and

Index i	1	2	3	4	5	6	7	8
$\hat{S}_i$	0.7223	0.1784	0.0276	0.0063	0.0002	0.0001	0.0005	0.0006
$\hat{T}_i$	0.7836	0.2324	0.0387	0.0078	0.0002	-0.0001	0.0006	0.0008

Table 5.2. Approximate first and total order sensitivity indices for (5.15).

10^6 samples. The reason for doing this is that we want to see how the accuracy changes when we increase the number of samples. Moreover, we compare these results with true analytical values of Sobol G-function, which are given in Table 5.3 and in [82], namely, the explicit formulas for V_i and V are

$$V_i = \frac{1}{3(1+a_i)^2}, \qquad V(Y) = \prod_{i=1}^{N}(1+V_i).$$

Figure 5.4 shows the error between analytical and estimated X_1 values for different number of samples, namely,

$$Error = |S_1 - \hat{S}_1|.$$

As expected, the difference or the error decreases when we increase the number of samples (see Figure 5.4). The reference line in Figure 5.4 is proportional to $1/\sqrt{M}$,

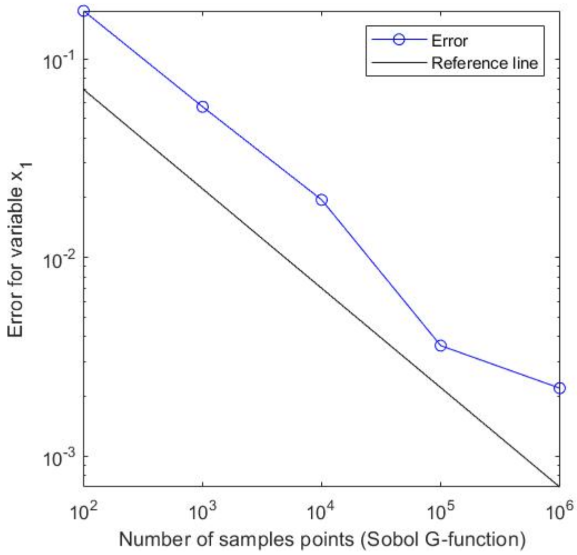


Figure 5.4. Difference between analytical and estimated values with Sobol method for S_1 on (5.15).

which indicates that the error is approximately proportional to $1/\sqrt{M}$ [85].

It is also worth mentioning that confidence intervals or bootstrap confidence intervals can be calculated if needed. The 95% confidence interval can be calculated as

$$\hat{S}_i \pm H_i = \hat{S}_i \pm 1.96 \left[\frac{\hat{\sigma}_{\hat{S}_i}}{\sqrt{M}} \right],$$

where $\hat{\sigma}_{\hat{S}_i} = \sqrt{\hat{S}_i}$ and H_i is the half-width of the confidence interval. In [7], it is recommended that if the half-width of the confidence interval is greater than 10% of the dominant parameter, the number of samples M needs to be increased. In other words, increase the number of samples M if

$$\frac{H_i}{\hat{S}_i} > 10\% \implies H_i > 0.1\hat{S}_i.$$

For example, the dominant variable X_1 of the Sobol G-function (5.15) has a first order sensitivity index of 0.7046 and the 95% confidence interval is $(0.5937, 0.8155)$ when $M = 10^2$. Also, $H_i = 0.1109$ which is greater than $0.1\hat{S}_1$. This suggests that we need to increase the number of samples M to get reasonable estimates of Sobol indices. When $M = 10^4$ we find $\hat{S}_1 = 0.7223$ and the 95% confidence interval is $(0.7110, 0.7336)$. In this case, the half-width of the confidence interval $H_i = 0.0113$ is less than $0.1\hat{S}_1 = 0.0722$. This suggests that we do not need to increase the number of samples M for this example.

5.4.3 Gaussian Test Function

For the next example, we will consider the Gaussian test function:

$$Y = \prod_{i=1}^{N} 1.2 \exp\left(\frac{-(X_i - b_i)^2}{c_i} \right), \tag{5.16}$$

where X_i, $i = 1, \dots, 20$ are the variables of interest, and

$$\begin{aligned} b_i &= (0.4, 0.3, 0.2, 0.1, 0.4, 0.3, 0.2, 0.1, 0.4, 0.3, 0.2, 0.1, 0.4, 0.3, 0.2, 0.1), \\ c_i &= (0.5, 0.5, 0.5, 0.5, 1, 1, 1, 1, 2.5, 2.5, 2.5, 2.5, 3, 3, 3, 3, 5, 5, 5, 5) \end{aligned}$$

are the control parameters [85]. These parameters control the importance of each variable X_i in a similar manner to the parameters a_i in the Sobol G-function (5.15). The control parameters c_i are set up in increasing order whereas b_i is a repeat of the vector $(0.4, 0.3, 0.2, 0.1)^T$ five times. Intuitively, this means that the first four variables of interest X_i will be more important for the quantity of interest Y.

Figure 5.5 illustrates first and total sensitivity indices for the Gaussian test function with $M = 10^4$ samples. We can see that parameters X_3 and X_4 have the highest first and total sensitivity indices as expected. Moreover, it seems that there is a pattern, i.e., every four indices there is an increase and a drop in values due to the way control parameters b_i and c_i were set up.

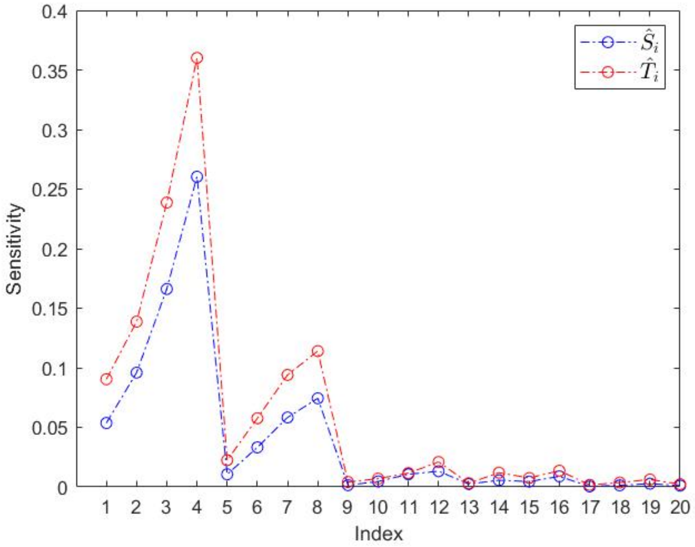


Figure 5.5. Approximate first and total order sensitivity indices for (5.16).

5.5 Comparison of the Sobol Method with Active Subspaces

Previous sections of this chapter defined and illustrated the Sobol method. Now we consider a number of examples where we compare the results of applying the Sobol method to identify important variables with those obtained from using the active subspace method.

5.5.1 Comparison of the Active Subspace Method with Sobol Method on the Sobol G-function

For the first comparison, we return to the Sobol G-function (5.15) and use the active subspace method instead of the Sobol method to determine influencial parameters. Figure 5.6 displays a plot of the eigenvalues and the leading eigenvector that corresponds to the dominant eigenvalue. Note that the calculations were done using $M = 10^3$ samples and a finite difference method. An exact derivative of the Sobol G-function is available, however, we want to test with finite differences as this is more realistic (i.e., we rarely have analytical expressions). We produce eigenvalues and a leading eigenvector in Figure 5.6. The plot on the right of Figure 5.6 indicates that parameter X_1 is the most important and influencial to the quantity of interest. This is consistent with the result obtained in Figure 5.3 in § 5.4.2.

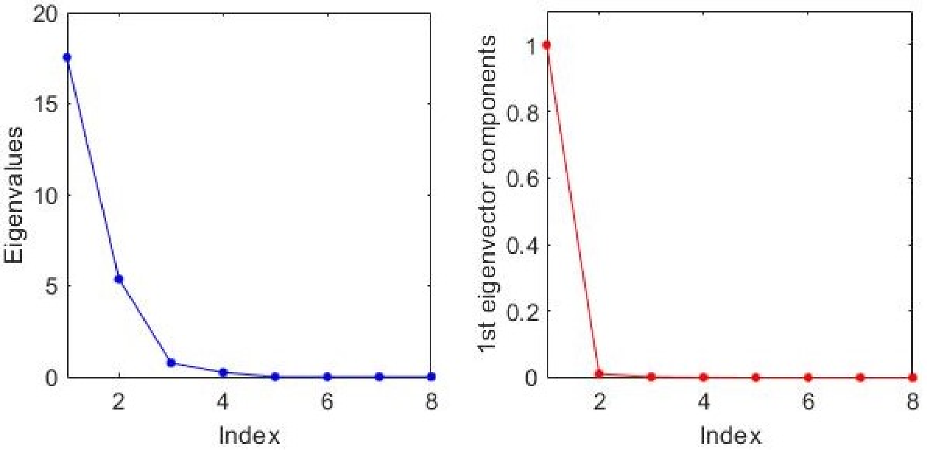


Figure 5.6. The active subspace method on Sobol G-function with finite difference method.

Table 5.3 and Table 5.4 present the results of the Sobol method and the active subspace method for a problem where Y is the Sobol-G function (5.15). Note that we are not comparing values of the results between the Sobol method and the active subspace method. If we compare approximated first order Sobol sensitivity indices with the analytical values, we can see that we get closer to the true values of first order indices when we increase the number of samples.

The Sobol Method				
# of samples M	$M = 10^2$	$M = 10^3$	$M = 10^4$	Analytical values of Sobol G-function
X_1	0.6566	0.7712	0.7252	0.7162
X_2	0.0237	0.1456	0.1882	0.1790
X_3	0.0453	0.0532	0.0284	0.0237
X_4	0.0111	0.0231	0.0159	0.0072
X_5	0.0027	0.0179	0.0023	0.0001
X_6	0.0010	0.0176	0.0015	0.0001
X_7	0.0035	0.0205	0.0021	0.0001
X_8	0.0004	0.0184	0.0023	0.0001

Table 5.3. The results of Sobol method on Sobol G-function in (5.15).

The Active Subspace Method			
# of samples M	$M = 10^2$	$M = 10^3$	$M = 10^4$
X_1	0.9791	0.9987	0.9999
X_2	0.1996	0.0501	0.0053
X_3	0.0326	0.0052	0.0003
X_4	0.0197	0.0016	0.0003
X_5	0.0012	0.0004	0.0001
X_6	0.0016	0.0003	0.0001
X_7	0.0013	0.0002	0.0001
X_8	0.0012	0.0008	0.0001

Table 5.4. The results of the active subspace method on Sobol G-function in (5.15).

Note that when the number of samples is $M = 10^2$, the Sobol method identifies X_1, X_3 and X_2 as the most important parameters (in order). The active subspace method identifies X_1, X_2 and X_3 as the most important parameters in order. The reason for this difference is that the Sobol method needs a considerable number of samples to get reasonable results. In other words, $M = 10^2$ is not enough for Sobol method to get reasonable results. We can calculate and check the size of confidence intervals to verify if the ordering of sensitivities is correct (as discussed in § 5.4.2). When we use the Sobol method with 10^2 to 10^3 samples, we often find that first order indices are higher than the total order indices, which is not feasible as discussed earlier in the chapter. The Sobol method usually needs a large number of samples to get sensible results, e.g., 10^4 or more. On the other hand, the active subspace method provides sensible results with only 10^2 samples. In other words, the active subspace method is more efficient at revealing the most important parameters in terms of the number of randomly generated samples required.

5.5.2 Comparison of Two Methods on a Synthetic Network

The next example we consider in order to compare the active subspace method against the Sobol method is the simple network we studied in Chapter 4, Figure 4.4 (a directed network with 15 nodes and 28 edges). For convenience, we reproduce Figure 4.4 here as Figure 5.7. We consider $f(\mathbf{X})$ to be the Katz centrality of node 1,

$$K_1 = ((I - \alpha A)^{-1}\mathbf{1})_1,$$

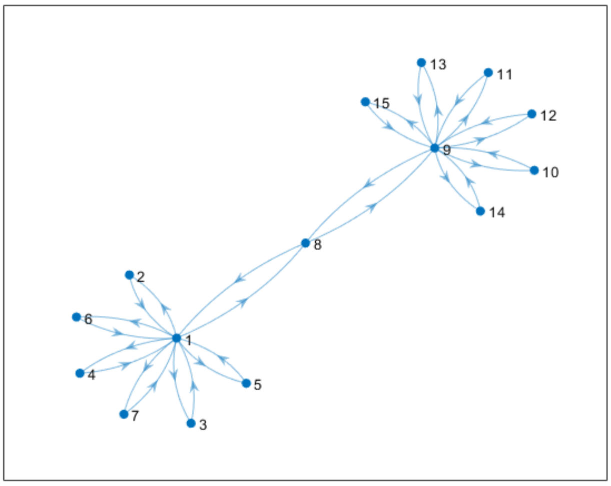


Figure 5.7. A simple network with 15 nodes and 28 edges.

where $\mathbf{X}$ represents the weights on the edges (as discussed in § 4.3). For the network in Figure 4.4, we have seen the results with the active subspace method and the leading eigenvector (recall Figure 4.6 and Table 4.3).

Now we use the Sobol method and calculate first and total order sensitivity indices with $M = 10^4$ samples. Inspecting the results in Figure 5.8, we can see

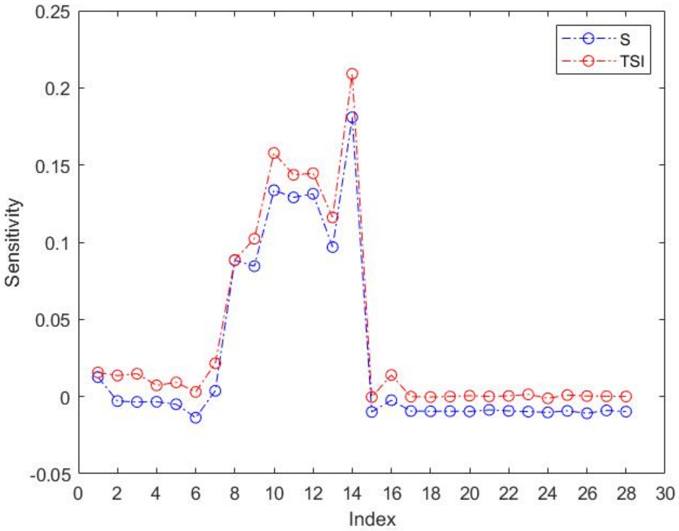


Figure 5.8. Simple synthetic network with 15 nodes and 28 edges using Sobol method.

that there are 7 parameters (edges) that impact the quantity of interest the most (refer back to Table 4.3 for the ordering of the edges in Figure 5.8). The "S" in the legend of Figure 5.8 represents approximate first order sensitivity indices whereas "TSI" represents approximate total order sensitivity indices. The results agree closely with those obtained by the active subspace method. We reproduce Figure 4.6 (used $M = 10^3$ samples) here as Figure 5.9 but with $M = 10^4$ samples to make the results obtained by both methods comparable. By inspecting the plots, it is clear that both methods produce similar results and identify the most important variables or edges for the quantity of interest $f(\mathbf{X})$ (the Katz centrality of node 1) consistently.

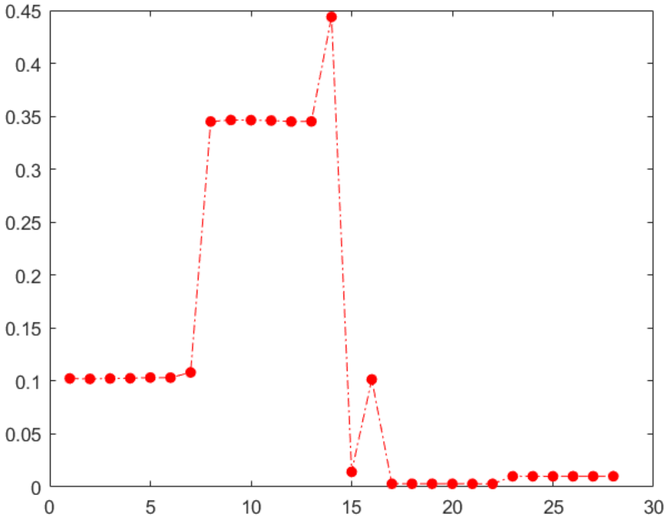


Figure 5.9. The leading eigenvector obtained by using the active subspace method with $M = 10^4$ samples.

We can also combine the results obtained by both methods into one simple visualisation. We note here that we are not comparing absolute values between the two methods. Instead, we are comparing trends. The x-axis in Figure 5.10 represents the edge sensitivities obtained by the active subspace method and the y-axis represents the edge sensitivities obtained by using the Sobol method for the network in Figure 5.7. By inspection, we can see that there are clusters of points. The reference line (least-squares fit to data) in Figure 5.10 indicates that there is a trend between the results of the active subspace method and the Sobol method. This suggests that both methods are producing similar results.

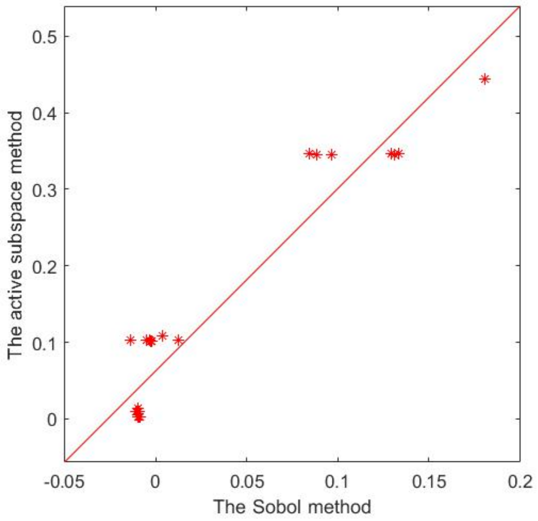


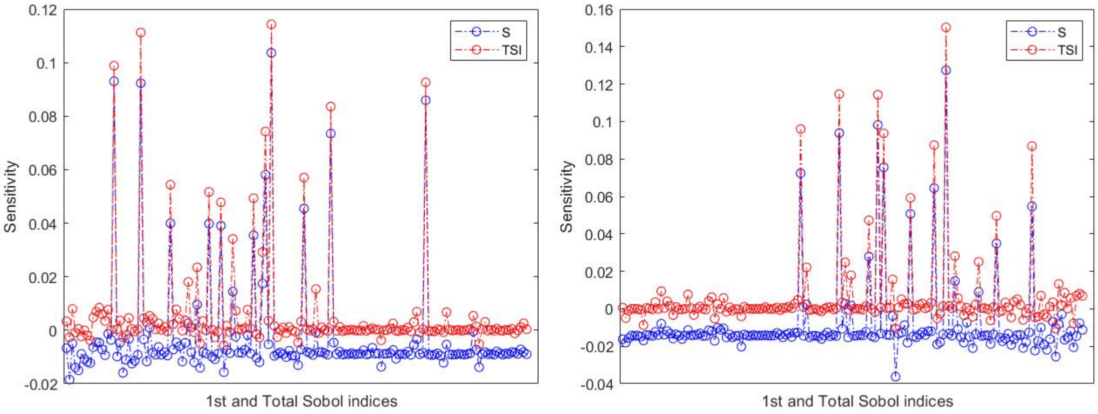
Figure 5.10. A combination of Figure 5.9 and the first order sensitivity indices in Figure 5.8.

5.5.3 Sobol Method and Zachary's Karate Club Network

We now return to Zachary's Karate Club network, which was considered in detail in § 4.4, and use the Sobol method to identify important parameters.

Note that for the active subspace method case we used $M = 10^3$ samples on Zachary's Karate Club network (§ 4.4), while here we use $M = 10^4$ samples for the Sobol method. From our observations in § 5.5.2, we are required to use (at least) $M = 10^4$ samples to calculate first and total order sensitivity indices in order to get reasonable results (we could also apply the condition about the confidence interval from § 5.4.2). If we use $M = 10^2$ to $M = 10^3$ samples, we get the first order sensitivity indices with higher values than the total sensitivity indices. The solution is simply to increase the number of samples at the expense of computational cost.

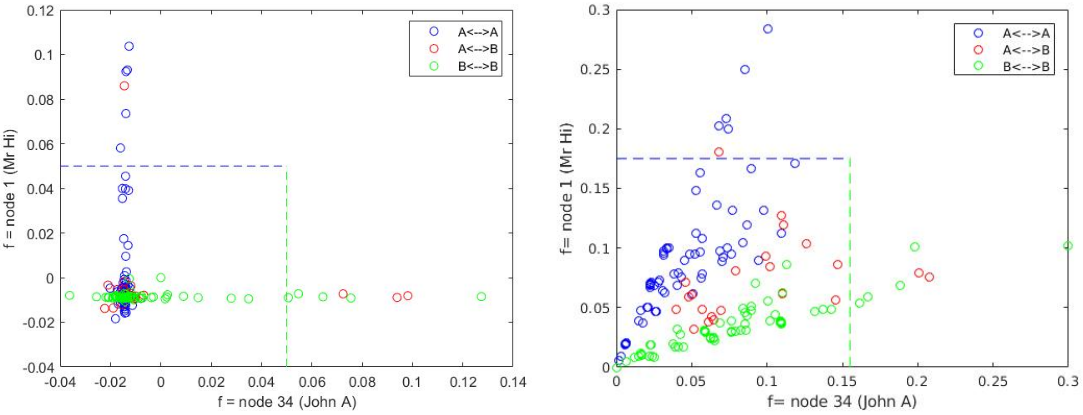
As for previous examples, we plot first and total order sensitivity indices (Figure 5.11). In this case, we have two plots for first and total order sensitivity indices due to the fact that we have two quantities of interest $f(\mathbf{X})$: Katz centrality of Node 1 and Katz centrality of Node 34 (as defined in § 4.3). Figure 5.11 shows that the Sobol method is able to capture important parameters to our quantity of interest (either Katz centrality of Node 1 or Katz centrality of Node 34). In order to compare these results directly with the active subspace method, we produce



(a) $f(\mathbf{X})$ = Katz centrality of Node 1 **(b)** $f(\mathbf{X})$ = Katz centrality of Node 34

Figure 5.11. The Sobol method for Zachary's Karate Club network.

a similar plot to Figure 4.18 for the Sobol method, namely Figure 5.12(a). We reproduce Figure 4.18 here as Figure 5.12(b) to make a direct comparison easier. Note that we compare the results of the active subspace method (Figure 5.12(b)) with only first order sensitivity indices (Figure 5.12(a)). Figure 5.12 allows us to compare the two methods. We also note here again that we are not comparing values of the results between the Sobol method and the active subspace method (the shapes differ in both plots due to the differences in the methods). As mentioned in § 4.4, the x-axis is represented by John A and y-axis is illustrated by Mr. Hi (two people of interest who had a disagreement). We defined group A as people who joined Mr. Hi after fission and we defined group B as people who joined John



(a) Using Sobol Method. **(b)** A.S. method (as in Figure 4.18).

Figure 5.12. Results from the Sobol method and the active subspace method on Zachary's Karate Club network.

A after fission.

The scatter-plot points fall into three categories. If a node from group A is connected to a node from group A, then we draw that connection (edge or link) with a blue colour. If a node from group A is connected to a node from group B, then we draw that connection with a red colour (and vice-versa). If a node from group B is connected to a node from the same group, then we draw that link with a green colour. The reason for doing this is that we want to know how well both methods can determine the most important links and in terms of how well they reflect the grouping of members after fission. It is important to note that both methods are using only information from **before** the fission.

As with the active subspaces method, we can see that the Sobol method highlights a few strong links to John A and there are a few strong links to Mr. Hi. Moreover, we have that the other connections closer to the origin represent weak links to John A, Mr. Hi or links between other karate club members. We note that, for the Sobol method, there are a number of parameters associated with negative values. As discussed earlier in this chapter, this is the consequence of the Monte Carlo approximation and is related to the number of samples M used. In practice, the negative values produced are fairly small in magnitude and so these values can be treated as 0 with little or no impact on our quantity of interest. If we use $M = 10^5$ or $M = 10^6$ samples for the Sobol method on this example, the number of first order sensitivity indices, which are negative, decreases considerably. However, the computation time increases greatly.

In both cases in Figure 5.12, the green points further on the x-axis represent the connections between people with the club president that supported him and joined his side after the fission. Blue points can be explained in a similar manner (the explanation of red points and more are discussed in § 4.4). These points have the most influence for either Katz centrality of Node 1 or Katz centrality of Node 34. Moreover, the points that are beyond the synthetic threshold are the most important. The synthetic threshold is set in a way that we consider only five important links or connections for both methods. On the other hand, we will discuss other connections or links that can have valuable information inside the threshold.

To further compare the results of the active subspace method and the Sobol method, we list the five most important links identified in each case in Table 5.5.

Quantity of interest $f(\mathbf{x})$	$f(\mathbf{x})$ = Katz centrality of Node 1 (Mr. Hi)	$f(\mathbf{x})$ = Katz centrality of Node 34 (John A)
Active subspace method	1 → 3, 1 → 2, 1 → 9, 1 → **14**, 1 → 4	34 → 33, 34 → 32, 34 → 31, 34 → 24, 34 → 16
Sobol method	1 → 3, 1 → 4, 1 → 9, 1 → 2, 1 → **7**	34 → 33, 34 → 32, 34 → 24, 34 → 31, 34 → 16

Table 5.5. Five most important links or connections identified as important by either the active subspace method or the Sobol method.

The second column of Table 5.5 shows the results with respect to Katz centrality of Node 1. We can see that four of the five most important connections or links between the Zachary's club members are identified by both methods. However, the active subspace method identified 1 → 14 connection as important whereas Sobol method identified 1 → 7 as important (blue points in Figure 5.12). On the other hand, when we look at the third column of Table 5.5, we can see all five important connections are identified by both methods (green points in Figure 5.12). However, the connections 1 → 14 and 1 → 7 can appear in the top 10 or 20 lists for either method. When we look at 10 most important links identified as important by either method, we find that eight connections overlap with respect to the Katz centrality of Node 1. Top 20 lists for either method have 15 overlaps. The connections 1 → 7 and 1 → 14 overlap when we analyse more edges of interest. As we inspect how well both methods perform, we see that more and more unimportant connections are not overlapped.

On the other hand, the top 10 connections with respect to the Katz centrality of Node 34 have 8 overlaps between the two methods. Even though Table 5.5 tells us that the first 5 links overlap, the next 5 links have only 3 overlaps. Top 20 lists for either method have 16 overlaps.

In the last two sections we discuss key differences and similarities we found so far between Sobol method and the active subspace method. Moreover, we discuss which method is superior for a given problem.

5.6 The Active Subspace Method vs The Sobol Method

The active subspace and Sobol methods are sensitivity analysis methods, which attempt to find the most influential input parameters on a given quantity of interest. Both methods are able to rank the inputs of a given function in terms of their output importance.

The examples comparing the active subspace method and the Sobol method in § 5.5 tell us that the methods are similar, in that both methods can determine how important a single parameter is to the quantity of interest. However, the Sobol method seems to do well and produce accurate results only when we consider sufficiently many Monte Carlo evaluations (e.g., at least $M = 10^4$ samples in our examples). The active subspace method seems to work well with fewer samples, e.g., $M = 10^3$ or less in some cases. The biggest difference between the two methods is that the Sobol method is capable of calculating sensitivities for interactions between individual combinations of variables (as discussed in § 5.4). The active subspace method seeks to approximate a function of many variables or parameters by a new function of a few **linear combinations** (important directions in the input space) of the variables (recall (2.3) in Chapter 2).

The Sobol method deals with single, multiple parameter sensitivities to the quantity of interest. The active subspace method can also be used to determine which parameters are important. However, the main point of an active subspace method is that it gives us linear combinations of parameters which can be used to approximate a new function. In this thesis, we are mainly interested in finding linear combinations of parameters.

5.7 Summary

In this chapter, we studied the underlying theory of the Sobol method. Furthermore, we introduced and solved a number of synthetic and real problems using the Sobol method. We showcased and explained the main properties of the Sobol method along with first, second, total order sensitivity indices.

In order to compare the active subspace method and the Sobol method (both global sensitivity methods), we considered several examples where both the Sobol and the active subspace methods were applied. It was found that both methods resulted in similar findings. However, it was discussed that the Sobol method needs a considerable number of samples to get reasonable results whereas the active subspace method performed well, in some cases with only $M = 10^2$ samples.

In § 5.5.3 we considered Zachary's Karate Club network using the Sobol method. We made calculations and compared them with the results obtained by using the active subspace method (as discussed in § 4.4). Figure 5.12 and Table 5.5 showed that the results obtained by both methods were relatively similar. It was shown that both methods can determine how important a single parameter is to the quantity of interest. However, the active subspace method seems to do well with fewer samples than the number required by the Sobol method.

Chapter 6

Applying the Active Subspace Method to the Design of Experiments

6.1 Introduction

The National Physical Laboratory (NPL) is the UK's national metrology institute and is concerned with all aspects of measurement science, including the statistical analysis of measurement data and the design of experiments. In this chapter we consider problems that are of interest to NPL. We note that NPL does not currently use the active subspace method. In this chapter, we show how the method could be useful for NPL scientists in the future by using the method on problems of interest to NPL in order to determine the most important parameters or measurements in a given setting.

Design of Experiments (DOE) is a branch of applied statistics used for analysing a system, process or product, where input variables are manipulated in order to investigate a particular quantity of interest. DOE is used for planning, conducting, analysing and interpreting experimental results [86]. In addition to this, DOE tries to design an experiment so that observations or measurements are obtained to answer the question of interest in a valid and efficent way. DOE is used in a wide range of disciplines, such as engineering, administration, marketing, hospitals, pharmaceutical [87], food industry [88], energy and architecture [89, 90]. It is known that DOE can be applied to computer simulation models as well as physical processes [91].

The main task of DOE is the identification of important input variables and deciding how important they are to the response variable or response function. The most common DOE types are variable screening (i.e., selecting important input factors of a system), system optimisation (i.e., optimal setting of the input variables) and robust design (i.e., reduction of variance in the system) [86, 92]. The aim of this chapter is to investigate the potential for the active subspace method to be used in a DOE setting of interest to NPL.

The general steps for planning and conducting DOE [86] may be summarised as follows.

1. Objectives: Determine the problems to be investigated.

2. Response variable definition: Define the response function or the outcome of the experiment. In our active subspace terminology, this is the "quantity of interest".

3. Experimental design: Decide on the experimental design to use, i.e., screening design, optimisation design, robust design, etc.

4. Perform an experiment. Generate data involving the quantity of interest.

5. Data analysis and conclusions: Using statistical methods, analyse the data and give conclusions along with graphical representation of the results.

In this thesis we consider a DOE problem proposed by NPL researchers arising in dimensional metrology, the measurement of the geometry of artefacts. A laser tracker is a portable measuring instrument and measures the location of a target in 3d in spherical coordinates. The distance measurement is provided by a laser interferometric system while the angle measurements are supplied by angle encoders [32, 33, 93]. Laser trackers are subject to geometric and misalignment errors associated with the rotating axes systems and have to be calibrated to account for these errors (see Chapter 1 for more information about laser trackers). This chapter is concerned with designing experiments to perform this calibration.

6.2 The Design of a Calibration Experiment

The beam steering mechanism and angular encoders within the laser tracker are subject to misalignments, offsets and eccentricities that lead to errors in the measured coordinates. For this reason, all laser tracker manufacturers provide online correction of these systematic effects, usually by software running in the laser tracker control system. The correction software relies on a model that describes the beam steering mechanism and its errors. The parameters of the model are usually derived from a combination of calibrations performed at the factory during manufacture and simple procedures performed by the user on the shop floor, prior to using the instrument [54, 94, 95]. In this section, we discuss the procedure of how to design an experiment to calibrate a laser tracker.

In design of experiments and measurement science we often need to calibrate instruments or parameters to get reasonable measurements. Having a design with "proper" characteristics increases the chances of the actual measuring instruments performing as expected. The purpose of calibration is to sustain the quality of measurement and to ensure the proper functionality of a particular instrument [96].

In executing a calibration experiment we need to decide i) how many targets to use and where to place them, and ii) how many positions and what positions to use for the tracker. Increasing the numbers of targets and positions will increase the time and resources required to complete the calibration. In practice, there will be a limit on resources and we wish to design an experiment that maximises the information gain for the resources available. Here, we concentrate on the problem of determining which targets to use from a set of potential targets, assuming that station positions have already been fixed [54]. We describe some experiments NPL has done and then apply the active subspace method in § 6.3.

6.2.1 Least Squares Problem and Aggregate Measures of Uncertainty

We discuss the mathematics first for designing an experiment to calibrate a laser tracker before we look at some examples. The main references for this section are [54, 96, 97, 98].

Suppose we have an $m \times n$ observation matrix C and an $m \times 1$ data vector $\boldsymbol{y}$ generated according to

$$\boldsymbol{y} = C\boldsymbol{\alpha} + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \in N(\boldsymbol{0}, I), \tag{6.1}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ are the parameters of the model to be fitted or estimated from the observations, $\boldsymbol{\epsilon}$ represents an independent random effect and I is an $n \times n$ identity matrix. Then the least squares estimate $\boldsymbol{a}$ of $\boldsymbol{\alpha}$ is the solution of the problem

$$\min_{\boldsymbol{\alpha}} (\boldsymbol{y} - C\boldsymbol{\alpha})^T (\boldsymbol{y} - C\boldsymbol{\alpha}).$$

Using a QR decomposition on C we can write

$$C = Q_1 R_1,$$

where Q_1 is an $m \times n$ matrix with orthonormal columns and R_1 is an $n \times n$ upper triangular matrix. From this we can rewrite the least squares solution as

$$\boldsymbol{a} = (C^T C)^{-1} C^T \boldsymbol{y} = R_1^{-1} Q_1^T \boldsymbol{y}.$$

The corresponding model approximant is then

$$\hat{\boldsymbol{y}} = C\boldsymbol{a} = C(C^T C)^{-1} C^T \boldsymbol{y} = Q_1 Q_1^T \boldsymbol{y}.$$

If the diagonal variance matrix associated with $\boldsymbol{y}$ is $V_{\boldsymbol{y}}$, then the variance matrix associated with $\boldsymbol{a}$ is

$$V_{\boldsymbol{a}} = (R_1^{-1} Q_1^T) V_{\boldsymbol{y}} (R_1^{-1} Q_1^T)^T.$$

In the case when $V_{\boldsymbol{y}} = I$, then

$$V_{\boldsymbol{a}} = V = (R_1^{-1} Q_1^T) I (R_1^{-1} Q_1^T)^T = (R_1^{-1} Q_1^T)(R_1^{-1} Q_1^T)^T = (R_1^T R_1)^{-1} = (C^T C)^{-1}.$$

The last equality is true due to the fact that Q_1 is orthogonal, so $Q_1^T Q_1 = Q_1^{-1} Q_1 = I$.

When $V_{\boldsymbol{y}}$ is not simply an identity matrix, we have a weighted least squares problem. For example, suppose $V_{\boldsymbol{y}}$ is a diagonal matrix with diagonal entries $v_i =$

u_i^2, and W is a diagonal weight matrix with diagonal entries $w_i = \dfrac{1}{u_i^2}$, so that $V_{\boldsymbol{y}} = W^{-1}$. If we perform the same calculations as above, we get that

$$V_{\boldsymbol{a}} = (C^T W C)^{-1}. \tag{6.2}$$

We now look at some of the main aggregate measures of uncertainty with respect to $V_{\boldsymbol{y}}$. Let $V_{\boldsymbol{a}}$ be an $n \times n$ variance matrix with eigenvalues $\lambda_j, j = 1, \ldots, n$. Also, let $\boldsymbol{b} = B\boldsymbol{a}$, where B is $n \times n$ and has full rank, so that $\boldsymbol{b}$ can be regarded as an alternative parametrisation of the problem. We have

$$V_{\boldsymbol{b}} = B V_{\boldsymbol{a}} B^T.$$

The following are aggregate measures of uncertainty associated with $V_{\boldsymbol{y}}$ or simply V.

- **A-measure**

  The first aggregate measure of uncertainty we consider is known as the A-measure:

  $$trace(V_{\boldsymbol{a}}) = \sum_{j=1}^{n} \lambda_j, \tag{6.3}$$

  where $trace(X)$ is shorthand for the trace of some square matrix X. In general,

  $$trace(V_{\boldsymbol{b}}) = trace(B V_{\boldsymbol{a}} B^T) = trace(B^T B V_{\boldsymbol{a}}) = trace(V_{\boldsymbol{a}} B^T B) \neq trace(V_{\boldsymbol{a}}).$$

  This means that the way we paramaterise $\boldsymbol{b}$ is important. In other words, the A-measure is sensitive to the choice of parametrisation.

- **D-measure**

  The second aggregate measure of uncertainty we consider is known as the D-measure:

$$|V_{\boldsymbol{a}}| = \prod_{j=1}^{n} \lambda_j. \tag{6.4}$$

We then have that

$$|V_{\boldsymbol{b}}| = \left|BV_{\boldsymbol{a}}B^T\right| = |B|^2 \, |V_{\boldsymbol{a}}| \, .$$

This tells us that unlike measure A, measure D is invariant in terms of the parametrisation.

- **E-measure**

  A third aggregate measure of uncertainty is the E-measure:

  $$\max_j \lambda_j. \tag{6.5}$$

  It can be shown that this measure is similar to the A-measure in the sense that it is sensitive to parametrisation.

- **G-measure**

  The last aggregate measure of uncertainty we consider is the G-measure:

  $$trace(DV_{\boldsymbol{a}}D^T), \tag{6.6}$$

  where D is a $r \times n$ matrix ($r \geq n$), for example, used to provide model predictions $z = D\boldsymbol{a}$. More information on this measure can be found in [54].

6.3 NPL Approach to Designing a Calibration Experiment

In this section, we describe an NPL experiment, which will be used for comparison when we apply the active subspace method on the same problem in § 6.4. The main reference for this section is [54].

We consider a measure of the goodness of design given by the quantity

$$f = trace(DV_{\boldsymbol{a}}D^T), \tag{6.7}$$

where $V_{\boldsymbol{a}}$ is the variance matrix associated with the least squares estimate $\boldsymbol{a}$ of $\boldsymbol{\alpha}$ as in § 6.2.1. The matrix D is constructed so that f is related to the uncertainties of a set of measurements undertaken by a calibrated tracker. The better the design, the more accurately the tracker is able to measure these targets through minimising the uncertainties associated with the estimated geometric errors $\boldsymbol{\alpha}$.

One approach is to fix the number of targets n, based on the resources available, and try to solve the optimisation problem

$$\min_{\{\xi_j\}} f, \qquad \xi_j \in \mathbb{R}^3, \tag{6.8}$$

where ξ_j represents the location of the j^{th} target [54]. This problem tends to be ill-posed and diffcult to solve, but can be implemented for small n. A second approach is to limit the ξ_k to a finite set of points in $\mathbb{R}^3, j = 1, \ldots, m, m > n$, and then choose the subset of n targets that minimises f over all subsets. For m much bigger than n, this approach becomes computationally expensive as there are $\begin{pmatrix} m \\ n \end{pmatrix}$ cases to consider.

The selection problem can be posed in terms of determining optimal weights $w_j \in \{0, 1\}$, where $w_j = 1$ if the jth target is used and $w_j = 0$, otherwise, $j = 1, \ldots, m$. A convex relaxation of the problem is to determine optimal weights $0 \leq w_j, \sum_j w_j = m$ where the jth target is given weight w_j. The general approach is as follows. Suppose m experiments could be performed with the jth experiment generating data according to

$$\boldsymbol{y}_j = C_j\boldsymbol{\alpha} + \boldsymbol{\epsilon}_j, \qquad \boldsymbol{\epsilon}_j \in N(\mathbf{0}, I), \qquad j = 1, \ldots, m, \tag{6.9}$$

where C_j is the $m \times n$ observation matrix associated with the jth experiment. Let

$$C = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \end{bmatrix}, \qquad \boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_m \end{bmatrix} \tag{6.10}$$

and $W = W(\boldsymbol{w})$ be the diagonal weighting matrix with w_jI in the jth set of diagonal elements, where I is the $m \times m$ identity matrix. The least squares solution $\boldsymbol{a}$ of $\boldsymbol{\alpha}$ and its associated variance matrix $V_{\boldsymbol{a}}$ of the weighted least squares problem is then given by

$$\boldsymbol{a} = \boldsymbol{a}(\boldsymbol{w}) = (C^T W C)^{-1} C^T W \boldsymbol{y}, \qquad V_{\boldsymbol{a}} = V_{\boldsymbol{a}}(\boldsymbol{w}) = (C^T W C)^{-1}.$$

Non-negativity and cost constraints can be imposed in the form

$$\sum_{i=1}^{m_j} \gamma_j w_j \leq \Gamma, \qquad w_j \geq 0, \qquad j = 1, \ldots, m, \tag{6.11}$$

where Γ is the number of target locations, the coeffcients γ_j are chosen to reflect the fact that some experiments may use more resources than others, i.e., they are computationally more expensive. Given D, the G-optimal weighting solves

$$\min_{\boldsymbol{w}} trace(D V_{\boldsymbol{a}}(\boldsymbol{w}) D^T), \tag{6.12}$$

subject to constraints in (6.11).

A simple example of this approach is the optimal design for fitting a straight line $y = a + bx$ to data (x_j, y_j), where $L \leq x_j \leq U$ (lower and upper bounds), and each y_j is regarded as a separate experiment so that C_j is the 1×2 matrix $[1 \quad x_j]$. The optimal design places the weights at the two extreme values for x_j and all other weights zero. This is intuitive because we require 2 points to determine the slope and the intercept of a straight line and the further these points are apart, the better the slope is determined.

6.4 Applying the Active Subspace Method

The main aim of this chapter is to use the active subspace method to find the most influential measurements in a particular experimental design setting. That is, we are not interested in allocating appropriate weights to get optimal measurements: our main focus is on finding which measurements affect the quantity of interest the most. This will be shown in the examples later in this section.

To use the active subspace method, we need to identify a quantity of interest $f(\mathbf{x})$ with respect to the general measurement problem. We first recall Algorithm 1 with a slight modification, for application in this new setting. The only difference between Algorithm 1 in Chapter 2 and Algorithm 2 here is that we have an extra step (Step 2). This is because of the form of the weights. In previous chapters, the weights we considered were simply uniform. The reason behind the change in the definition of weights is to match Forbes [54, 99, 100]. This means that the way Step 2 is set up is the consequence of the examples we are about to consider. Step 2 of Algorithm 2 is discussed more in § 6.2.

Step 1: Draw M independent samples $\{\mathbf{x}_i\}, i = 1, \ldots, M$ (vector $\mathbf{x}$ has m entries) according to the sampling density ρ.

Step 2: Compute $\boldsymbol{w}_i = e^{(-\mathbf{x}_i/2)}$ (here we assume the sampling density ρ for the $\{\mathbf{x}_i\}$ to follow a uniform distribution).

Step 3: For each sample $\boldsymbol{w}_i$, compute the gradient $\nabla_w f_i = \nabla f(\boldsymbol{w}_i)$ and the quantity of interest $f(\boldsymbol{w}_i)$.

Step 4: Compute the matrix $\hat{\mathbf{C}}$ and its associated eigenvalue decomposition,

$$\hat{\mathbf{C}} = \frac{1}{M} \sum_{i=1}^{M} \nabla_w f_i \nabla_w f_i^T = \hat{\mathbf{W}} \hat{\mathbf{\Lambda}} \hat{\mathbf{W}}^T, \tag{6.13}$$

where $\hat{\mathbf{W}}$ is the matrix of eigenvectors, and $\hat{\Lambda} = diag(\hat{\lambda}_1, ...\hat{\lambda}_m)$ is the diagonal matrix of eigenvalues ordered in decreasing manner.

Algorithm 2. The active subspace algorithm for experimental design.

The quantity of interest we consider in this chapter is the D-measure from § 6.2.1 with a slight modification:

$$f(\boldsymbol{w}) = \log |C^T W(\boldsymbol{w}) C|. \tag{6.14}$$

We note here that exponentially perturbed weights $\boldsymbol{w}$ are used to row-scale matrix C (as in Equation (6.14)). We also need to note that in [99], $W(\boldsymbol{w})$ is used to multiply blocks of the observation matrix C, where each block in C represents measurements from each sensor. In other words, a block or blocks of measurements from each sensor are multiplied with a particular weight. In the active subspace approach and in our examples, we choose to apply row-scaling to the observation matrix C instead of block-scaling. The reason for this is that we are more interested in finding which individual measurements have more leverage on the quantity of interest than studying a block of measurements. We could, however, easily adapt our procedure to find an important block or blocks of measurements (instead of assigning weights to each measurement, assign one weight to each block of measurements).

6.4.1 Example: Fitting a Straight Line to Given Data

Imagine we want to fit a straight line $y_j = \alpha_1 + \alpha_2 x_j + \epsilon_j$ to some data (x_j, y_j), where $\epsilon_j \in N(0, \sigma^2)$ (drawn from a Gaussian distribution). Let's suppose that we have one sensor or an instrument that records m co-ordinate measurements $(x_j, y_j), j = 1, \ldots, m$, and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T$ contains the parameters to be fitted (the intercept and the slope of the line). Using the information from (6.9) we obtain

$$\begin{aligned} y_1 &= \alpha_1 + \alpha_2 x_1 + \epsilon_1 \\ y_2 &= \alpha_1 + \alpha_2 x_2 + \epsilon_2 \\ &\vdots \\ y_m &= \alpha_1 + \alpha_2 x_m + \epsilon_m \end{aligned} \quad \Rightarrow \quad \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}. \tag{6.15}$$

Suppose we now want to know which measurements $(x_j, y_j)^T$ have the most leverage or influence on a given quantity of interest by using the active subspace method. To illustrate this, let $\boldsymbol{x} = [1, \ldots, 10]$, α_1 represent the slope of a straight line and α_2 represent the intercept, so that

$$C = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ \vdots & \vdots \\ 1 & 10 \end{bmatrix}. \tag{6.16}$$

The matrix C in the matrix in (6.16) is called the observation matrix. The main task is then to find which measurements impact or have more leverage on $f(\boldsymbol{w})$. Recall that the quantity of interest is

$$f(\boldsymbol{w}) = \log |C^T W(\boldsymbol{w}) C|.$$

In order to calculate our quantity of interest $f(\boldsymbol{w})$ for this example, we let $m = 10$ (the number of input parameters or the rows of matrix C) and $\boldsymbol{w}$ be the vector of weights, where each entry of vector $\boldsymbol{w}$ corresponds to a particular measurement. The idea now is to find which measurements (x_j, y_j) affect $f(\boldsymbol{w})$ the most.

For the active subspace approach, we also require gradient samples. As in previous chapters, we use finite differences to obtain an approximation to the derivative of f. Let $M = 100$ be the number of samples so that we generate $\boldsymbol{w}_i, i = 1, \ldots, M$ vector samples, where $\boldsymbol{w}_i = e^{(-\mathbf{x}_i/2)}$ (exponentially perturbed weights) and $\mathbf{x}_i$ follows a uniform distribution on the interval $(0, 1)$. Then we take each sample and calculate $f(\boldsymbol{w}_i)$. The next standard procedure of the active subspace method is to use finite differences to get an approximated version of gradient samples. It is then followed by performing an eigenvalue decomposition and analysing the dominant eigenvector that corresponds to the leading eigenvalue (see Algorithm 2). Components of the dominant eigenvector are shown in Figure 6.1. The computation shows that the first 3 or 4 measurements and the last measurement are the most important, which makes sense since these measurements affect the slope and intercept the most. When fitting a straight line to data, the first point or the first few points heavily impact the intercept and the slope whereas the last point heavily impacts only the slope. In other words, these measurements have the most leverage on the variance of α_1 and α_2.

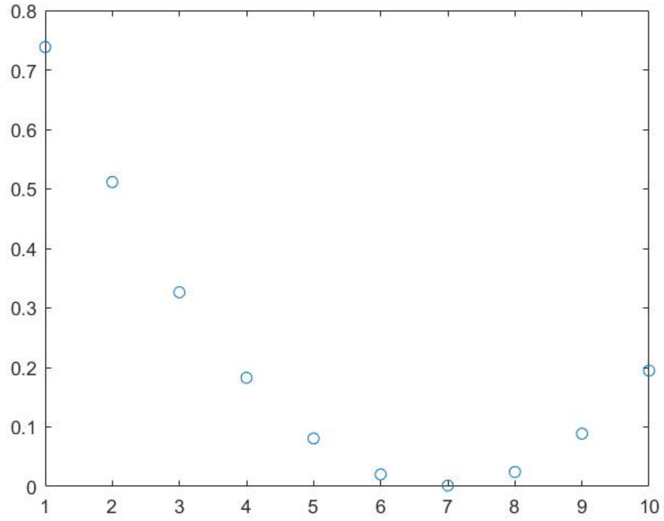


Figure 6.1. The dominant eigenvector with respect to the model in (6.15) and (6.16).

We now consider a very similar example to (6.16), but we let $\boldsymbol{x} = [-4.5, \ldots, 4.5]$ so that our data points are now symmetric about $(0, 0)$. The new observation matrix is given by

$$C = \begin{bmatrix} 1 & -4.5 \\ 1 & -4 \\ \vdots & \vdots \\ 1 & 4.5 \end{bmatrix}. \tag{6.17}$$

Using the same procedure as before we obtain the dominant eigenvector shown in Figure 6.2. Figure 6.2 indicates that the most important measurements are the first few and the last few. For $\boldsymbol{x} = [1, \ldots, 10]$, the measurements near 1 have much more leverage in determining the intercept compared to the points near 10, while points near 1 and near 10 have equal leverage in determining the slope. For $\boldsymbol{x} = [-4.5, \ldots, 4.5]$, the points at either end have the same leverage for determining both the intercept and the slope. The aggregate measure of uncertainty depends on the uncertainty contributions from both parameters and the active subspace approach differentiates between the two designs appropriately.

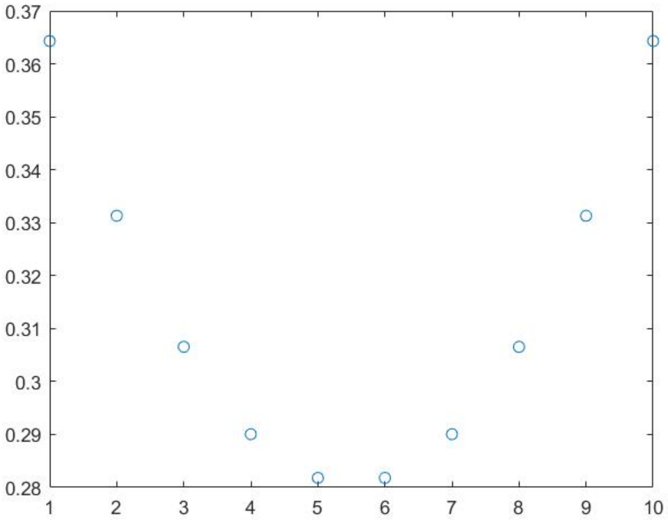


Figure 6.2. The dominant eigenvector with respect to the observation matrix in (6.17).

6.4.2 Example: Fitting a Straight Line to Data with Extra Information

The next test problem is slightly different from the two problems we just encountered in § 6.4.1. The distinction is that we add a third column to the observation matrix C:

$$C = \begin{bmatrix} 1 & -4.5 & 0 \\ 1 & -4 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 4.5 & 1 \end{bmatrix}. \tag{6.18}$$

Recall that in the first two examples we wanted to fit a straight line $y_j = \alpha_1 + \alpha_2 x_j + \epsilon_j$ to some data (x_j, y_j), where $\epsilon_j \in N(0, \sigma^2)$, $j = 1, \ldots, m$, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T$ (the intercept and the slope of the line) and $\mathbf{c}_j^T = (1, x_j)^T$. The observation matrix in (6.18) represents the fact that we want to fit a straight line $y_j = \alpha_1 + \alpha_2 x_j + \epsilon_j$ to some data (x_j, y_j), where $j = 1, \ldots, m-1$ and

$$y_m = \alpha_1 + \alpha_2 x_m + e + \epsilon_m. \tag{6.19}$$

The difference here is that when measuring y_m we supppose that we are, for example, using a different instrument in (6.18), namely, we have an additional factor e (along with α_1 and α_2), which we call an 'offset'. This represents a situation where the instrument associated with (6.19) is not calibrated and can have a significant effect on experiments. We expect the active subspace method to tell us that the last measurement has the most leverage on the quantity of interest. The reason for this is that the last entry of the third column of matrix C in (6.18) is a non-zero. If we don't perform the last measurement, then we don't have any information on what e is and the variance associated with e will be infinite. In a problem like this, we have to include the last measurement if we want to reduce the variance associated with fitted parameters (α_1, α_2, e). Moreover, we want to know how e can affect the quantity of interest. By performing standard measurements and the measurement involving an 'offset', we want to see how well the active subspace method picks up this last important measurement. Following the procedure shown in § 6.4.1, we obtain the dominant eigenvector of this problem, which is shown in Figure 6.3.

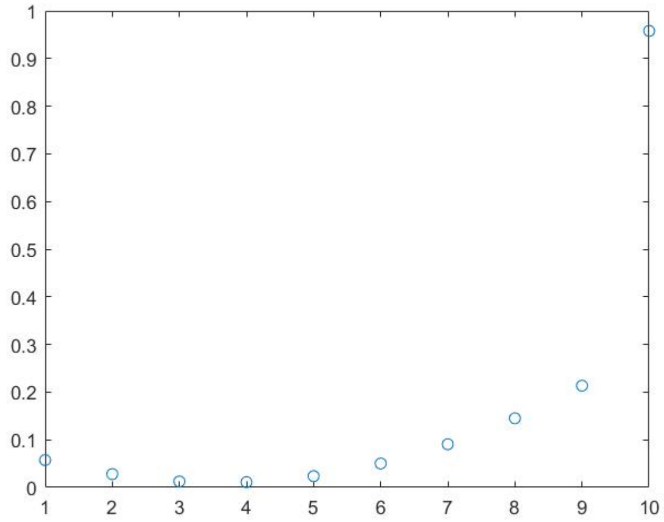


Figure 6.3. The dominant eigenvector with respect to the matrix in (6.18).

As expected, we can see in Figure 6.3 that the added information had a considerable impact on the problem. We can see that the first few measurements are not as important as in the previous examples (see Figures 6.1 and 6.2, however, these measurements are still important for determining the slope and the intercept (α_1 and α_2, respectively). The reason for the points increasing in Figure 6.3 is that, the measurements get more and more important at determining what e (offset) is. Clearly, the last measurement is the most important at determining e by observing Figure 6.3.

6.4.3 Laser Tracker Calibration Experiment

For the last example of this chapter, we have a laser tracker calibration experiment. This is a realistic problem supplied by NPL. We have 18 targets that are being measured by five laser trackers. The main aim is to find the targets that are most important in the minimisation problem

$$\min_{\boldsymbol{w}} trace(DV_{\boldsymbol{a}}(\boldsymbol{w})D^T).$$

Simply put, we want to find the optimal weighting discussed in § 6.3 for targets so that we minimise the aggregate measure of uncertainty in (6.20).

The results of optimal weighting supplied by Forbes [100] indicate that all but three of the weights are zero [54]. Specifically, his results indicate that measurements of targets 10, 12 and 16 are sufficient to estimate all the parameters. Moreover, it is more efficient (in terms of reducing the aggregated estimate of uncertainty) to re-measure these targets rather than measuring other targets. On the basis of the optimal weighting, he also concludes that if the design is constrained such that at most 18 target measurements are made, then a good strategy is to measure target 10 six times, target 12 four times and target 16 eight times. It is important to note that only three targets need to be measured, which means that the time required to set up and perform the experiments is significantly reduced [54].

So, the task now is to try and verify this result with the active subspace method. The only information provided by NPL was a code for calculating $f(\boldsymbol{w})$, making this a "black-box" problem where the details of the function were unknown. The usual procedure for identifying important parameters ($m = 18$ input parameters) is to calculate $f(\boldsymbol{w}_i)$ for each sample, $i = 1, \ldots, M$, apply the finite difference method to get an approximation of the gradient samples and perform an eigenvalue decomposition in order to examine the leading eigenvector (see Algorithm 1).

Using $M = 10^3$ samples and Algorithm 1, this calculation leads us to Figure 6.4, which represents the leading eigenvector that corresponds to the leading eigenvalue.

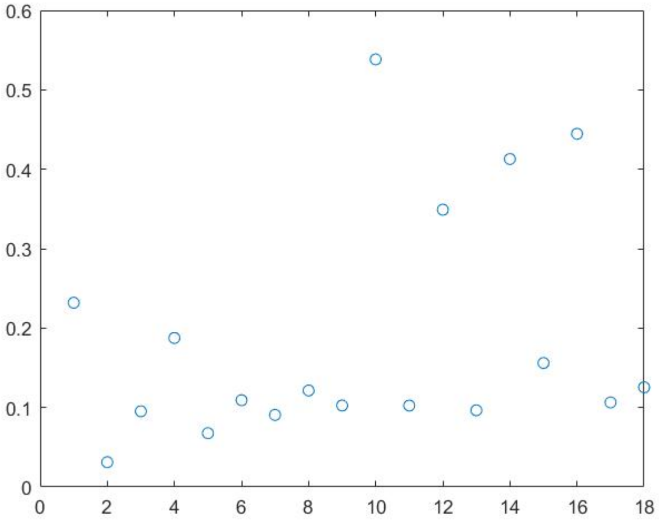


Figure 6.4. The leading eigenvector of a black-box problem obtained by the active subspace method.

It was found that parameters w_{10}, w_{12}, w_{14} and w_{16} are the most influential or have more leverage than other parameters on our unknown quantity of interest $f(\boldsymbol{w})$. These results are consistent with the results obtained by Forbes with three important measurements (w_{10}, w_{12} and w_{16}) matched. It was also found by Forbes that the steepest reduction in the quantity of interest from a unit weighting is achieved by increasing the weights associated with targets 10, 12, 14 and 16 [54] (and matches the result obtained by using the active subspace method). It is possible that the active subspace method is identifying important targets but the optimisation is selecting from the important targets the set of three that is most important. The optimisation and the active subspace method are asking slightly different questions and the answers are consequently slightly different. However, the active subspace method has identified important information that is consistent with the results obtained by Forbes [54].

6.5 Conclusion

In this chapter, we have introduced some ideas from design of experiments and measurement science, which are topics of interest to the National Physical Laboratory. We considered a general measurement model that can be used for linear cases such as fitting a straight line to data or the more general problem of calibrating a measurement instrument using a linear model. We then used the active subspace method to identify which measurements (x_j, y_j) affect the quantity of interest the most. The fact that we have an easy access to the quantity of interest means that we can comfortably employ the active subspace method.

We examined which measurements affect our slope and the intercept the most (straight line tests). In the first two examples we studied, we obtained sensible results. In other words, we identified measurements which were predicted to be important by our knowledge of the underlying problem. The last example we considered was a laser tracker calibration experiment, where the results of the active subspace are consistent with the results obtained previously by NPL. The two examples, fitting a straight line to data, and calibration of a laser tracker, showed that the active subspace method can be used to determine which of a set of potential measurements are most influential in determining the calibration parameters, even for complex design of experiment problems.

Chapter 7

Conclusion

7.1 Thesis Achievements

The introductory chapter of this thesis discussed the basics of sensitivity analysis and its examples (e.g., PCA, Sobol method, etc). Chapter 2 was devoted to the main mathematical background material of this thesis, where we introduced the method of the active subspace, and provided a practical algorithm with examples. In Chapter 3 we discussed the background theory of networks. We presented a new analytical derivation for the gradient of Katz centrality in § 3.3 (Lemma 3.3.3). In Chapter 4, we applied the theory of networks and the ideas of the active subspace method, and performed a number of experiments. Again, it is important to note that, as far as we are aware, this is the first time the active subspace method has been used in network science. After illustrating some ideas on synthetic networks, we considered two realistic networks in § 4.4 and § 4.5, which have been previously studied by network science researchers, namely, Zachary's Karate Club network and a protein-protein interaction (PPI) network. The results we obtained using the active subspace method generally match those published in the research literature. In Chapter 5, we considered Sobol indices and made comparisons with the active subspace method. Despite the fact that the Sobol method is far more popular, the active subspace method did well in comparison. In fact, the active subspace method was better in some situations. In Chapter 6, we introduced the idea of design of experiments, and applied the active subspace method to some examples that are of interest to NPL. This included a "black-box" example, where the experimental processs was partly known. The input parameters and output quantity of interest were supplied. In this case, we found that the active subspace method has identified important information that is consistent with the results obtained by NPL.

7.2 Future Work

The active subspace method is a fairly new application with many possibilities. Constantine in [8] mentions a number of applications where the active subspace method could be used and exploited further (e.g., pressurised water reactors, photovoltaic (PV) solar cells, turbine efficiency function, wind plant power, Onera-M6 fixed wing shape optimisation, etc). As far as we know, this is the first time the active subspace method is used in network and measurement sciences. Therefore, further research is required to examine the relevance and applicability of the active subspace method in comparison with other sensitivity analysis tools.

References

[1] Thomas Most. Efficient sensitivity analysis of complex engineering problems. *11th International Conference on Applications of Statistics and Probability in Civil Engineering*, 2011.

[2] Johannes Will, J. S. Moller, and Bernhard Bauer. Robustness evaluations of the NVH comfort using full vehicle models by means of stochastic analysis. *VDI Congress Berechnung und Simulation im Fahrzeugbau, VDI-Berichte 1846*, pages 505–525, 2004.

[3] A. van Griensven, Thomas Meixner, Sabine Grunwald, Tracey Bishop, M. E. Diluzio, and Raghavan Srinivasan. A global sensitivity analysis tool for the parameters of multi-variable catchment models. *Journal of Hydrology*, 324:10–23, 2006.

[4] Richard W. Katz. Extreme value theory for precipitation: sensitivity analysis for climate change. *Advances in Water Resources*, 23:133–139, 1999.

[5] Elmar Plischke, Emanuele Borgonovo, and Curtis L. Smith. Global sensitivity measures from given data. *European Journal of Operational Research*, 226:536–550, 2013.

[6] Hui Wan, Jun Xia, Liping Zhang, Dunxian She, Yang Xiao, and Lei Zou. Sensitivity and interaction analysis based on Sobol method and its application in a distributed flood forecasting model. *Water*, 7(6):2924–2951, 2015.

[7] Xy Zhang, Mirjam Trame, Larry Lesko, and Stephan Schmidt. Sobol sensitivity analysis: A tool to guide the development and evaluation of systems pharmacology models. *CPT: Pharmacometrics and Systems Pharmacology*, 4:69–79, 2015.

[8] Paul G. Constantine. *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies.* SIAM, Philadelphia, USA, 2015.

[9] Yanjun Gan, Qingyun Duan, Wei Gong, Charles Tong, Yunwei Sun, Wei Chu, Aizhong Ye, Chiyuan Miao, and Zhenhua Di. A comprehensive evaluation of various sensitivity analysis methods: A case study with a hydrological model. *Environmental Modelling and Software*, 51:269–285, 2014.

[10] Jerome Morio. Global and local sensitivity analysis methods for a physical system. *European Journal of Physics*, 32:1577–1583, 2011.

[11] Cibin Raj, Kp Subheer, and Indrajeet Chaubey. Sensitivity and identifiability of stream flow generation parameters of the SWAT model. *Hydrological Processes*, 24(9):1133–1148, 2010.

[12] Andrea Saltelli, Paola Annoni, Ivano Azzini, Francesca Campolongo, Marco Ratto, and Stefano Tarantola. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Computer Physics Communications*, 181(2):259–270, 2010.

[13] Amir Mokhtari and Henry C. Frey. Sensitivity analysis of a two-dimensional probabilistic risk assessment model using analysis of variance. *Risk Analysis*, 25:1511–1529, 2005.

[14] Active Subspaces. FAQ. ***http://activesubspaces.org/faq/***, [accessed 22.11.2020].

[15] Amandine Marrel, Bertrand Iooss, Beatrice Laurent, and Olivier Roustant. Calculations of the Sobol indices for the Gaussian process metamodel. *Reliability Engineering and System Safety*, 94:742–751, 2009.

[16] Ilya M. Sobol. Global sensitivity analysis indices for the investigation of nonlinear mathematical models. *Matematicheskoe Modelirovanie*, 19(11):23–24, 2007.

[17] Ilya M. Sobol. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55:271–280, 2001.

[18] Ilya M. Sobol. Sensitivity estimates for nonlinear mathematical models. *Matematicheskoe Modelirovanie*, 2:112–118, 1993.

[19] Sidharth P. Mishra, Uttam Sarkar, Subhash Taraphder, Sanjay Datta, Devi P. Swain, Reshma Saikhom, Sasmita Panda, and Menalsh Laishram. Multivariate statistical data analysis – Principal Component Analysis (PCA). *International Journal of Livestock Research*, 7(5):60–78, 2017.

[20] Lindsay Smith. A Tutorial on Principal Components Analysis. ***http://reflect.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf***, [accessed 27.05.2019], 2002.

[21] Jinhyun Ju, Jason Banfelder, and Luce Skrabanek. Quantitative understanding in biology Principal Component Analysis. *https://physiology.med.cornell.edu/people/banfelder/qbio/lecture_notes/3.4_Principal_component_analysis.pdf*, [accessed 29.03.2020], 2019.

[22] Analytics Vidhya. Understanding Principle Component Analysis(PCA) step by step. *https://medium.com/analytics-vidhya/understanding-principle-component-analysis-pca-step-by-step-e7a4bb4031d9*, [accessed 20.09.2021], 2020.

[23] Akash Dubey. The Mathematics Behind Principal Component Analysis. *https://towardsdatascience.com/the-mathematics-behind-principal-component-analysis-fff2d7f4b643*, [accessed 20.09.2021], 2018.

[24] Maarten van Steen. On the complexity of simple distributed systems. *IEEE Distributed Systems Online*, 5(5), 2004.

[25] Alessandro Vespignani. Modelling dynamical processes in complex socio-technical systems. *Nature Physics*, 8:32–39, 2012.

[26] Ingo Scholtes. Understanding complex systems: When big data meets network science. *IT – Information Technology*, 57(4):252–256, 2015.

[27] Ernesto Estrada and Desmond J. Higham. Network properties revealed through matrix functions. *SIAM Review*, 52(4):696–714, 2010.

[28] Francesca Arrigo, Peter Grindrod, Peter Grindrod, Desmond J. Higham, and Vanni Noferini. Non-backtracking walk centrality for directed networks. *Journal of Complex Networks*, 6:54–78, 2018.

[29] Francesco Tudisco, Francesca Arrigo, and Antoine Gautier. Node and layer eigenvector centralities for multiplex networks. *SIAM Journal of Applied Mathematics*, 78:853–876, 2018.

[30] Francesco Tudisco and Desmond J. Higham. Sparse matrix computations for dynamic network centrality. *Applied Network Science*, 17(2):17–38, 2017.

[31] David A. Katz. Scientific measurements. *http://chymist.com/Measurement.pdf*, [accessed 24.11.2019], 2019.

[32] Kam C. Lau and Robert J. Hocken. *Three and five axis laser tracking systems*. US Patent 4714339, 1987.

[33] Ben Hughes, Alistair B. Forbes, Andrew Lewis, Wenjuan Sun, Dan Veal, and Karim Nasr. Laser tracker error determination using a network measurement. *Measurement Science and Technology Journal*, 22(4), 2011.

[34] Javier Conte, Jorge S. Mazo, Ana C. Majarena, Agustin Brau, and Juan J. Aguilar. Identification and kinematic calculation of laser tracker errors. *Procedia Engineering*, 63:379–387, 2013.

[35] Ana C. Majarena, Javier Conte, Jorge Santolaria, and Raquel Acero. A new methodology for kinematic parameter identification in laser trackers. https://www.intechopen.com/books/kinematics/a-new-methodology-for-kinematic-parameter-identification-in-laser-trackers, [accessed 17.09.2019], 2017.

[36] Kam C. Lau, Robert J. Hocken, and William C. Haight. Robot performance measurements using automatic laser tracking techniques. *Robotics and Computer – Integrated Manufacturing*, 2(3/4):227–236, 1985.

[37] Kam C. Lau, Robert J. Hocken, and William C. Haight. Automatic laser tracking interferometer system for robot metrology. *Precision Engineering*, 8(1):3–8, 1986.

[38] Ian Wright. Laser trackers - from inspection to manufacturing. https://www.engineering.com/AdvancedManufacturing/ArticleID/13499/Laser-Trackers-From-Inspection-to-Manufacturing.aspx, [accessed 10.10.2019], 2016.

[39] Bala Muralikrishnan, Steve Phillips, and Daniel Sawyer. Laser trackers for large-scale dimensional metrology: A review. *Precision Engineering*, 44:13–28, 2013.

[40] Ian Wright. How to avoid 3 common mistakes when using laser trackers. https://apisensor.com/avoid-3-common-mistakes-using-laser-trackers/, [accessed 10.10.2019], 2019.

[41] Optical Metrology Centre. OMC technical brief - laser tracker. https://www.optical-metrology-centre.com/Downloads/Tech_Briefs/TechBrief_LaserTracker.pdf, [accessed 17.09.2019], 2001.

[42] Benjamin B. Gallagher. *Optical Shop Applications for Laser Tracker Metrology Systems.* PhD thesis, The University of Arizona, 2003.

[43] Advanced Machinery. Hexagon laser tracker systems. *https://advancedmachinery.com/hexagon-laser-tracker-systems-2/*, [accessed 10.10.2019], 2019.

[44] Yoon S. Jang, Wooram Kim, Heesuk Jang, and Seung W. Kiml. Absolute distance meter operating on a free-running mode-locked laser for space mission. *International Journal of Precision Engineering and Manufacturing*, 19(7):975–981, 2018.

[45] Nathaniel E. Helwig. Introduction to random variables. *University of Minnesota*, 2020.

[46] Vose Software. How many Monte Carlo samples are enough? *https://www.vosesoftware.com/riskwiki/Howmanyiterationstorun.php*, [accessed 22.09.2021], 2017.

[47] Mario Manuel Teixeira Parente. *Active Subspaces in Bayesian Inverse Problems.* PhD thesis, Technische Universitat Munchen, 2020.

[48] Gabriel A. Terejanu. *Tutorial on Monte Carlo techniques.* Department of Computer Science and Engineering, New York, USA, 2002.

[49] Samik Raychaudhuri. Introduction to Monte Carlo simulation. *Simulation Conference*, pages 91–100, 2008.

[50] Marco Liu. Optimal Number of Trials for Monte Carlo Simulation. *https://www.valuationresearch.com/wp-content/uploads/kb/SpecialReport_MonteCarloSimulationTrials.pdf*, [accessed 23.09.2021], 2017.

[51] Joel L. Horowitz. Bootstrap methods in econometrics: Theory and numerical performance. *Advances in Economics and Econometrics: Theory and Applications*, 3:188–222, 1997.

[52] Stephanie MC Fook-Chong and Robin Choo. Introduction to Bootstrap. *Proceedings of Singapore Healthcare*, 20(3):236–240, 2011.

[53] Steven J. Miller. The method of least squares. *Mathematics Department Brown University*, pages 1–7, 2006.

[54] Alistair B. Forbes. Design of experiment associated with laser tracker calibration (unpublished). 2020.

[55] Mike Meyers. Introducing basic network concepts. *https://www3.nd.edu/~cpoellab/teaching/cse40814_fall14/networks.pdf*, [accessed 08.02.2018], 2010.

[56] Nancy Katz, David Lazer, Holly Arrow, and Noshir Contractor. Network theory and small groups. *Small Group Research*, 35(3):307–332, 2004.

[57] Paul Butler. Visualizing facebook friends. *https://paulbutler.org/2010/visualizing-facebook-friends/*, [accessed 08.06.2019], 2010.

[58] Derek A. Holton and J. Sheehan. *The Petersen Graph.* Cambridge University Press, Cambridge, UK, 1993.

[59] Michele Benzi and Christine Klymko. On the limiting behavior of parameter-dependent network centrality measures. *SIAM Journal on Matrix Analysis and Application*, 32(2):686–706, 2015.

[60] Srijan Kumar, Francesca Spezzano, V.S. Subrahmanian, and Christos Faloutsos. Edge weight prediction in weighted signed networks. *In IEEE 16th International Conference on Data Mining*, pages 221–230, 2016.

[61] Ernesto Estrada and Philip A. Knight. *A First Course in Network Theory.* Oxford University Press, UK, 2015.

[62] Robert Geisberger. *Better approximation of betweenness centrality.* PhD thesis, Institute for Theoretical Computer Science, Algorithmics II Universit ät Karlsruhe (TH), 2008.

[63] Erjia Yan and Ying Ding. Applying centrality measures to impact analysis: A coauthorship network analysis. *Journal of the American Society for Information Science and Technology*, 60(10):2107–2118, 2009.

[64] Justin Zhan, Sweta Gurung, and Sai P. K. Parsa. Identification of top-k nodes in large networks using Katz centrality. *Journal of Big Data*, 4(16), 2017.

[65] Francis Bloch, Matthew O. Jackson, and Pietro Tebaldi. Centrality measures in networks. *Small Group Research*, 2016.

[66] Philip Bonacich and Paulette Lloyd. Eigenvector-like measures of centrality for asymmetric relations. *Social Networks*, 23(3):191–201, 2001.

[67] Sueyon Khim. The Frobenius-Perron Theorem. *https://www.scribd.com/document/71078313/Khim.* , [accessed 29.03.2020], 2007.

[68] Carl D. Meyer. Matrix analysis and applied linear algebra. *SIAM*, page 673, 2008.

[69] Michelle Girvan and Mark E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[70] David Gleich. Compute the Betweenness Centrality. *https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/10922/versions/2/previews/matlab_bgl/betweenness_centrality.m/index.html* , [accessed 29.02.2022], 2008.

[71] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* Johns Hopkins University, Baltimore, USA, 2007.

[72] Stephen P. Borgatti and Martin G. Everett. Models of core / periphery structures. *Social Networks*, 21(4):375–395, 1999.

[73] Wayne W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.

[74] Lester R. Ford and Delbert R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[75] Desmond J. Higham and Nataša Pržulj. Random graph models and their application to protein-protein interaction networks. *In Handbook of Statistical Systems Biology*, 2011.

[76] Luke Hakes, John W. Pinney, David L. Robertson, and Simon C. Lovell. Protein-protein interaction networks and biology – What's the connection? *Nature Biotechnology*, 26(1):69–72, 2008.

[77] Desmond J. Higham, Marija Rasajski, and Natasa Przulj. Fitting a geometric graph to a protein-protein interaction network. *Bioinformatics*, 24(8):1093–1099, 2008.

[78] Ernesto Estrada. Virtual identification of essential proteins within the protein interaction network of yeast. *Proteomics 6*, pages 35–40, 2006.

[79] Claudia Romano, Giovanni Corsetti, Vincenzo Flati, Evasio Pasini, Anna Picca, Riccardo Calvani, Emanuele Marzetti, and Francesco Saverio Dioguardi. Influence of Diets with Varying Essential/Nonessential Amino Acid Ratios on Mouse Lifespan. *Journal Nutrients*, 11(1367), 2019.

[80] Sergei Kucherenko and Shufang Song. Different numerical estimators for main effect global sensitivity indices. *Reliability Engineering and System Safety*, 165:222–238, 2017.

[81] Andrea Saltelli and Paola Annoni. How to avoid a perfunctory sensitivity analysis. *Environmental Modeling and Software*, 25(12):1508–1517, 2010.

[82] Andrea Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. *Global sensitivity analysis*. The Primer and John Wiley and Sons, UK, 2008.

[83] Per-Anders Ekstrom. Eikos: A Simulation Toolbox for Sensitivity Analysis in Matlab. *http://ecolego.facilia.se/ecolego/show/Sensitivity+analysis+toolbox/*, [accessed 08.11.2018], 2005.

[84] Toshimitsu Homma and Andrea Saltelli. Importance measure in global sensitivity analysis of nonlinear models. *Reliability Engineering and System Safety*, 52(1):1–7, 1996.

[85] Thomas Bassine, Bryan Cooley, Kenneth Jutz, and Lisa Mitchell. Accuracy of data-based sensitivity indices. *SIAM*, 9:348–366, 2016.

[86] Benjamin Durakovic. Design of experiments application, concepts, examples: state of the art. *Periodicals of Engineering and Natural Sciences*, 5(3):421–439, 2017.

[87] Filipa Paulo and Lucia Santos. Design of experiments for microencapsulation applications: A review. *Materials Science and Engineering: C*, 77:1327–1340, 2017.

[88] Peigen Yu, Mei Y. Low, and Weibiao Zhou. Design of experiments and regression modelling in food flavour and sensory analysis: A review. *Trends in Food Science and Technology*, 71:202–215, 2018.

[89] Arno Schlueter and Philipp Geyer. Linking BIM and design of experiments to balance architectural and technical design factors for energy performance. *Automation in Construction*, 86:33–43, 2018.

[90] Benjamin Durakovic and Muris Torlak. Simulation and experimental validation of phase change material and water used as heat storage medium in window applications. *Journal of Materials and Environmental Science*, 8(5):1837–1846, 2017.

[91] Sushant Garud, Iftekhar A. Karimi, and Markus Kraft. Design of computer experiments: A review. *Computers and Chemical Engineering*, 106:71–95, 2017.

[92] Huairui Guo and Adamantios Mettas. Design of experiments and data analysis. *Reliability and Maintainability Symposium*, 2010.

[93] William B. Penzes. Time line for the definition of the meter. *http://www.mel.nist.gov/div821/museum/timeline.html/*, [accessed 08.04.2019], 2003.

[94] G. N. Peggs, Paul G. Maropoulos, Ben Hughes, Alistair B. Forbes, Marek Ziebart, and Bala Muralikrishnan. Recent developments in large-scale dimensional metrology. *Journal of Engineering*, 223(6):571–595, 2009.

[95] R. H. Schmitt, M. Peterek, Edward P. Morse, W. Knapp, M. Galetto, F. Härtig, F. G. Goch, Ben Hughes, Alistair B. Forbes, and W. T. Estler. Advances in large-scale metrology – review and future trends. *CIRP Annals – Manufacturing Technology*, June 2016.

[96] Alistair B. Forbes and Hoang D. Minh. Design of linear calibration experiments. *Measurement*, 46(9):3730–3736, 2013.

[97] Chan Basaruddin Asep Sugih Nugraha. Analysis and comparison of QR decomposition algorithm in some types of matrix. *Proceedings of the Federated Conference on Computer Science and Information Systems*, pages 561–565, 2012.

[98] Zheng Wang, Alistair B. Forbes, and Paul G. Maropoulos. Laser tracker position optimization. *Journal of the CMSC*, 9(1):8–12, 2014.

[99] Alistair B. Forbes. Weighting observations from multi-sensor coordinate measuring systems. *Measurement Science and Technology Journal*, 23(2), 2012.

[100] Alistair B. Forbes. Private communication at NPL (August – September). 2018.