

Identification using Convexification and Recursion

This Page will be Replaced before Printing



UPPSALA
UNIVERSITET

Abstract

System identification studies how to construct mathematical models for dynamical systems from the input and output data, which finds applications in many scenarios, such as predicting future output of the system or building model based controllers for regulating the output the system.

Among many other methods, convex optimization is becoming an increasingly useful tool for solving system identification problems. The reason is that many identification problems can be formulated as, or transformed into convex optimization problems. This transformation is commonly referred to as the convexification technique. The first theme of the thesis is to understand the efficacy of the convexification idea by examining two specific examples. We first establish that a l_1 norm based approach can indeed help in exploiting the sparsity information of the underlying parameter vector under certain persistent excitation assumptions. After that, we analyze how the nuclear norm minimization heuristic performs on a low-rank Hankel matrix completion problem. The underlying key is to construct the dual certificate based on the structure information that is available in the problem setting.

Recursive algorithms are ubiquitous in system identification. The second theme of the thesis is the study of some existing recursive algorithms, by establishing new connections, giving new insights or interpretations to them. We first establish a connection between a basic property of the convolution operator and the score function estimation. Based on this relationship, we show how certain recursive Bayesian algorithms can be exploited to estimate the score function for systems with intractable transition densities. We also provide a new derivation and interpretation of the recursive direct weight optimization method, by exploiting certain structural information that is present in the algorithm. Finally, we study how an improved randomization strategy can be found for the randomized Kaczmarz algorithm, and how the convergence rate of the classical Kaczmarz algorithm can be studied by the stability analysis of a related time varying linear dynamical system.

Abbreviations

| | |
|-------|--------------------------------------|
| KF | Kalman Filter |
| PF | Particle Filter |
| SMC | Sequential Monte Carlo |
| PMH | Particle Metropolis-Hastings |
| MCMC | Markov Chain Monte Carlo |
| KA | Kaczmarz Algorithm |
| RKA | Randomized Kaczmarz Algorithm |
| RDWO | Recursive Direct Weight Optimization |
| DWO | Direct Weight Optimization |
| PDF | Probability Density Function |
| LP | Linear Programming |
| SDP | Semi-Definite Programming |
| LTI | Linear Time Invariant |
| BLUE | Best Linear Unbiased Estimate |
| LSE | Least Squares Estimate |
| MSE | Mean Squared Error |
| ML | Maximum Likelihood |
| PE | Persistent Excitation |
| SVD | Singular Value Decomposition |
| SysID | System Identification |

Contents

| | | |
|-------|--|----|
| 1 | Introduction | 9 |
| 1.1 | Background and brief overview | 9 |
| 1.2 | Preliminaries | 11 |
| 1.2.1 | Convexification | 11 |
| 1.2.2 | Recursive Bayesian methods | 17 |
| 1.3 | Thesis contribution and outline | 23 |
| 1.3.1 | Chapter 2 | 24 |
| 1.3.2 | Chapter 3 | 24 |
| 1.3.3 | Chapter 5 | 25 |
| 1.3.4 | Chapter 4 | 25 |
| 1.3.5 | Chapter 6 | 25 |
| 1.4 | Other contributions | 26 |
| 2 | Sparse estimation from an overdetermined linear system | 28 |
| 2.1 | Problem formulation | 28 |
| 2.2 | Algorithm description | 30 |
| 2.3 | Algorithm analysis | 33 |
| 2.4 | Illustrative experiments | 38 |
| 2.4.1 | Experiment 1 | 38 |
| 2.4.2 | Experiment 2 | 40 |
| 2.5 | Conclusion | 44 |
| 2.6 | Appendix | 44 |
| 3 | Hankel matrix completion with convexification | 46 |
| 3.1 | Problem introduction | 46 |
| 3.2 | Analysis and the main result | 47 |
| 3.2.1 | Proof of Theorem 3.1 | 47 |
| 3.3 | Discussion and numerical illustration | 52 |
| 3.4 | Conclusion | 53 |
| 3.5 | Appendix | 55 |
| 3.5.1 | Proof of Fact 1 | 55 |
| 3.5.2 | Proof of Fact 2 | 57 |

| | | |
|-------|---|----|
| 3.5.3 | Proof of Fact 3 | 57 |
| 3.5.4 | Proof of Fact 4 | 58 |
| 4 | Score function estimation for systems with intractable transition kernels | 59 |
| 4.1 | Problem introduction | 59 |
| 4.2 | The convolution operator and Stein's identity | 61 |
| 4.3 | Score function estimation | 62 |
| 4.3.1 | Asymptotic analysis | 62 |
| 4.3.2 | Convergence rate analysis | 64 |
| 4.4 | Estimating higher order derivatives | 65 |
| 4.5 | Numerical illustration | 66 |
| 4.6 | Conclusion | 68 |
| 4.7 | Appendix | 68 |
| 5 | On the Recursive Direct Weight Optimization | 71 |
| 5.1 | Problem formulation | 71 |
| 5.2 | The RDWO method | 72 |
| 5.3 | Analysis and the main result | 73 |
| 5.3.1 | Derivation | 73 |
| 5.3.2 | Interpretation | 77 |
| 5.4 | Conclusion | 77 |
| 6 | On the Kaczmarz Algorithm | 79 |
| 6.1 | Problem formulation | 79 |
| 6.2 | Convergence rate of the KA | 82 |
| 6.3 | Optimized RKA | 84 |
| 6.4 | Discussion | 87 |
| 6.5 | Experiments | 89 |
| 6.6 | Conclusion | 89 |
| 7 | Summary and future work | 91 |
| 7.1 | Thesis summary | 91 |
| 7.2 | Future work | 91 |
| | References | 96 |

Chapter 1

Introduction

This chapter starts with a brief overview of system identification. After that, preliminaries on the convexification and the recursive Bayesian algorithms will be reviewed. In the end, contributions and outline of the rest of the thesis will be introduced.

1.1 Background and brief overview

System identification (SysID) is a subject about constructing dynamical models of the underlying system from the observed input and output data. The constructed mathematical models often can provide further understanding of the system, and be used for practical applications as well, such as forecasting climate change [20], building adaptive controllers [4] or for Model based Predictive Control [68].

Research on SysID can be roughly divided into two categories, namely the linear and the nonlinear SysID. Understanding for the linear SysID has been relatively matured and the achievements have been summarized from different perspectives in a number of textbooks, see e.g. [63, 91, 74, 100, 103]. The most popular approaches for linear SysID include the Maximum Likelihood (ML) method, the Prediction Error Method (PEM) and the Subspace Identification (SI) method. The ML method finds the parameters by maximizing the so-called likelihood function of the observed data. The PEM is a generalization of the ML approach, which identifies the parameters by optimizing a suitably selected cost function of the prediction error. The SI method is different from the previous two, which directly estimates a state space model for the system from the input and output data using linear algebra tools.

Unlike linear SysID, nonlinear SysID is a relatively new area and still remains active today. Early developments are surveyed in [54, 89], and the recent results are summarized in the edited book [35]. Some of the existed methods are briefly reviewed as follows. The most classical approach is based

on the Volterra series representation [85, 86] of the nonlinear system. This approach often leads to a large number of parameters to estimate, which restricts its applicability to identify highly nonlinear systems. Another popular approach is based on basis function expansion of the nonlinear part. Typical basis functions include the Fourier basis, wavelets, orthonormal polynomials [99, 17, 5]. This type of methods often leads to a linear-in-parameters problem, which can be relatively easy to solve. The kernel based method [28] is another attractive algorithm for nonlinear SysID, which does not make strong structure assumptions on the system, while instead certain smoothness properties of the nonlinear part are assumed.

Now we will give a general overview of the topics studied in this thesis. One theme of the thesis is on understanding the effectiveness of the convexification technique. For many SysID tasks, in the end, the problems can be formulated as (or transformed into) finding a low rank matrix or a sparse vector under certain conditions [41, 90, 102, 26, 46, 66, 65, 29], which are usually given as non-convex optimization problems. The non-convex optimization problems are generally hard to solve. Intuitively, in order to find the optimal low rank matrix or sparsest vector solution, one needs to enumerate all the possibilities to find the best one, which is often computationally prohibitive. Convexification [64] is one useful technique to get around the difficulties — Instead of solving the hard problem directly, it approximates the original non-convex optimization problem with a convex one, which then can be solved efficiently [7]. Specifically, in chapter 2 and chapter 3, we will study two cases where the convexification idea can be applied.

The other theme of the thesis is to study several recursive (sequential, or iterative) algorithms and their applications to solve SysID and signal processing problems. One of them is the Recursive Direct Weight Optimization algorithm advocated in [6]. This algorithm is a local modeling based approach, which updates the estimated model in an online fashion when new data is available. In chapter 5, we give some new insights on this algorithm by exploiting further structural information inherent in the algorithm formulation. The Kaczmarz algorithm [55] is an iterative method for solving a system of linear equations by implementing sequential projections. In chapter 6, we will investigate how different projection strategies will affect the performance of the algorithm. The recursive Bayesian methods, including the particle filter [36] and the particle MCMC [1] algorithm, are popular computational tools for state inference and parameter estimation for nonlinear non-Gaussian dynamical systems. In chapter 4, we will study how these methods could be adapted for parameter estimation for systems with intractable state transition kernels.

1.2 Preliminaries

1.2.1 Convexification

In this section, we will give a brief overview of convexification, and illustrate the idea with several applications arising in SysID.

As mentioned before, for many tasks, such as finding the minimum order system approximation [30], or identifying the network structure [101], the problems finally can be formulated as finding minimal rank matrices (or sparsest vectors) under certain conditions. However, finding the minimal rank matrix (or the sparsest vector) often turns out to be a non-convex optimization problem, hence difficult to solve [34]. As will be explained in more detail (with concrete examples) later on, when the nuclear norm [14] is applied to approximate the matrix rank, or the l_1 norm [18, 97, 15] is applied to approximate the vector sparsity (i.e. the number of nonzero elements in the vector), the original non-convex optimization problems become convex. This type of techniques is commonly referred to as the *convexification*¹. The resulting convex optimization problem can be solved efficiently using either gradient based methods, interior point methods, or other approaches which are more adapted for large scale problems [101]. Moreover, the solution to the convexified problem also possesses nice robustness and stability guarantees under certain conditions [13, 12, 108, 9, 10, 80].

Now we start introducing the definitions of the convex set and the convex function.

Definition 1.1. A set \mathcal{S} is convex if and only if for any two points $x_1 \in \mathcal{S}$ and $x_2 \in \mathcal{S}$, it holds true that $\alpha x_1 + (1 - \alpha)x_2 \in \mathcal{S}$ when $0 \leq \alpha \leq 1$.

Definition 1.2. A function $f : \mathcal{S} \rightarrow \mathbb{R}$ is convex if \mathcal{S} is convex, and for any two points $x_1 \in \mathcal{S}$ and $x_2 \in \mathcal{S}$, $f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2)$ holds when $0 \leq \alpha \leq 1$.

An illustration of the convex set and the convex function is given in Figure 1.1 and 1.2. An optimization problem is called convex if both its objective function and the constraint set are convex. The Linear Programming (LP) problem, the Second-Order Cone Programming (SOCP) problem and the Semi-Definite Programming (SDP) problem are three commonly used members of the convex optimization family. There exist several efficient toolboxes for solving the convex optimization problems, such as MOSEK², SeDuMi [96] or SDPT3 [98]. In depth introduction of the theories and applications of convex optimization are covered in [7].

¹Sometimes it is also called 'convex relaxation'.

²<https://www.mosek.com/>

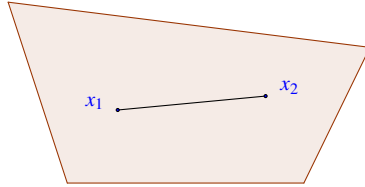


Figure 1.1: An illustration of a convex set, which is given as the gray area. When x_1 and x_2 move freely inside the set, the line segment $[x_1, x_2]$ lies inside the set as well.

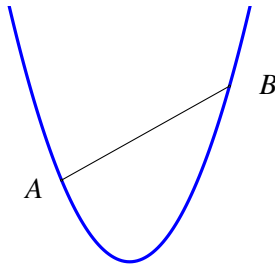


Figure 1.2: An illustration of a convex function $f(x)$. The segment AB , where $A = (x_1, f(x_1))$ and $B = (x_2, f(x_2))$, always upper bounds the function curve between x_1 and x_2 when A and B move freely on the curve.

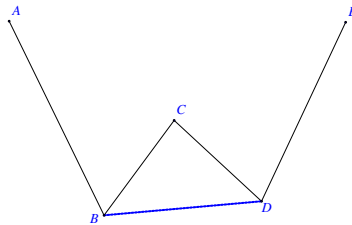


Figure 1.3: An illustration of convex envelope. The function given by the segments A, B, D, E is the convex envelope of the function given by the segments A, B, C, D, E .

The nuclear norm of matrix A is defined as $\|A\|_* \triangleq \sum_{i=1}^k \sigma_i$, where $\{\sigma_i\}_{i=1}^k$ denotes the singular values of A . The matrix nuclear norm has close relation to the matrix rank, which is built upon the idea of the *convex envelope* [47], defined as follows:

Definition 1.3. *The convex envelope $\phi_{\text{env}}(x)$ of a function $f(x)$ over the set \mathcal{S} is defined as the largest convex function $g(x)$ such that $g(x) \leq f(x)$ for all $x \in \mathcal{S}$. More precisely*

$$\phi_{\text{env}}(x) = \sup\{g(x) : g(x) \leq f(x) \text{ for } \forall x \in \mathcal{S} \text{ and } g(\cdot) \text{ is convex.}\}$$

An illustration of the intuition behind the definition of *convex envelope* is given in Figure 1.3. The key to derive the convex envelope is the conjugate function [47], which is defined as follows

Definition 1.4. *The conjugate $f^*(\cdot)$ of function $f(\cdot) : \mathcal{C} \rightarrow \mathbb{R}$ is defined as $f^*(y) = \sup_{x \in \mathcal{C}} \{y^T x - f(x)\}$.*

It can be proven that the convex envelope of a function is given by the conjugate of the conjugate (i.e. the bi-conjugate) of the function [45]. Using this result, the following theorem established in [30], gives the connection between the nuclear norm and the rank of a matrix.

Theorem 1.1. *The convex envelope of the rank of matrix A on $\mathcal{K} = \{A \in \mathbb{R}^{m \times n}, \|A\|_2 \leq 1\}$ is given by $\phi_{\text{env}}(A) = \|A\|_*$, where $\|\cdot\|_2$ indicates the operator norm of a matrix.*

From this result, we can intuitively think that the matrix nuclear norm gives the *tightest* convex approximation to the matrix rank.

The following theorem illustrates how the constraint $\|A\|_* \leq t$ can be transformed into a set of Linear Matrix Inequalities (LMI), hence representing the nuclear norm related optimization problems as Semi-Definite Programming problems.

Theorem 1.2. *For $A \in \mathbb{R}^{m \times n}$, and $t \in \mathbb{R}$, we have $\|A\|_* \leq t$ if and only if there exist matrices $B \in \mathbb{R}^{m \times m}$ and $C \in \mathbb{R}^{n \times n}$ such that*

$$\begin{bmatrix} B & A \\ A^T & C \end{bmatrix} \succeq 0 \quad \text{and} \quad \text{tr} B + \text{tr} C \leq 2t.$$

This result can be elegantly proven by making use of the fact that the nuclear norm is the dual norm of the matrix operator norm [80].

Next, we will discuss more about the convexification idea by providing three concrete examples. One noticeable common feature of these examples is that they all involve a specific low rank Hankel matrix. The example 1.1

is about designing a Linear Time Invariant (LTI) filter by making use of the nuclear norm minimization heuristic [31].

Example 1.1: Filter design using nuclear norm heuristic

The example is about the design of a low order discrete time LTI system, with linear constraints on its step response. To make use of the convexification idea, we need to identify a certain low rank structure in the problem. One key observation is that, if the transfer function of a discrete time LTI system has r stable poles, i.e. the order of the system is r , the related Hankel matrix H_n defined as $H_n(i, j) = h_{i+j-1}$, $1 \leq i, j \leq n$, will be of rank r whenever $n \geq r$, where h_i , $i = 1, \dots, 2n-1$ denotes the first $2n-1$ entries of the system impulse response. The structure of H_n is given in (1.1).

$$H_n = \begin{bmatrix} h_1 & h_2 & h_3 & \cdots & h_{n-1} & h_n \\ h_2 & h_3 & \vdots & h_{n-1} & h_n & h_{n+1} \\ h_3 & \vdots & h_{n-1} & h_n & h_{n+1} & \vdots \\ \vdots & h_{n-1} & h_n & h_{n+1} & \vdots & \vdots \\ h_{n-1} & h_n & h_{n+1} & \vdots & \vdots & h_{2n-2} \\ h_n & h_{n+1} & \cdots & \cdots & h_{2n-2} & h_{2n-1} \end{bmatrix}. \quad (1.1)$$

In the following, we will let \mathcal{H}_n denote the set of $n \times n$ Hankel matrices. Inspired by the observation that the rank of H_n reflects the order of the system, to find a low order system, the design problem can be formulated as follows [31]

$$\begin{aligned} \min_{H_n \in \mathcal{H}_n} \quad & \text{rank}(H_n) \\ \text{s.t.} \quad & l_i \leq \sum_{k=1}^i h_k \leq u_i, \forall i = 1, \dots, n, \end{aligned}$$

where l_i and u_i for $i = 1, \dots, n$ represent the lower and upper bounds for the step response of the desired system. These quantities are specified by the designer to constrain the dynamical characteristic of the system, for instance the time delay, rise time, static gain, and so on.

However, this formulation is non-convex due to the property of the matrix rank, and hence hard to solve. The idea of convexification is to convexify the objective by approximating the matrix rank with the matrix nuclear norm, which results in

$$\begin{aligned} \min_{H_n \in \mathcal{H}_n} \quad & \|H_n\|_* \\ \text{s.t.} \quad & l_i \leq \sum_{k=1}^i h_k \leq u_i, \forall i = 1, \dots, n. \end{aligned}$$

Using results in Theorem 1.2, we can reformulate the above problem as the following SDP problem

$$\begin{aligned}
& \min_{H_n \in \mathcal{H}_n, B, C} \quad \text{tr} B + \text{tr} C \\
& \text{s.t.} \quad \begin{bmatrix} B & H_n \\ H_n^T & C \end{bmatrix} \succeq 0, \\
& \quad \quad \quad l_i \leq \sum_{k=1}^i h_k \leq u_i, \quad \forall i = 1, \dots, n.
\end{aligned}$$

The example 1.2 below is on identifying an Output Error (OE) system with the nuclear norm minimization heuristic [38, 46].

Example 1.2: Output Error system identification

The output sequence $\{y_t\}_{t=1}^{\infty}$ is generated through the following Output Error system

$$y_t = \sum_{i=1}^{\infty} h_i u_{t-i} + v_t, \quad (1.2)$$

in which $\{v_t\}_{t=1}^{\infty}$ is a white noise sequence, i.e. a sequence of uncorrelated random variables with zero mean and finite variance; $\{u_t\}_{t=1}^{\infty}$ denotes the input sequence, and we assume that $u_t = 0$ when $t \leq 0$; $\{h_t\}_{t=1}^{\infty}$ denotes the impulse response of the system. Given the input and output data, the task is to estimate (identify) the impulse response sequence.

In practice, a sufficiently high order finite impulse response, say $\{h_i\}_{i=1}^{2K-1}$, is used to approximate the infinite impulse response sequence. Let H_K be defined in the same manner as in the previous example, that is, $H_K(i, j) = h_{i+j-1}$ for $i, j = 1, \dots, K$, whose rank reflects the order of the corresponding linear system.

Hence, to find a low order system which can fit the output data $\{y_t\}_{t=1}^N$ well, the following heuristic can be formulated

$$\min_{H_K \in \mathcal{H}_K} \sum_{t=1}^N \left(y_t - \sum_{i=1}^{2K-1} h_i u_{t-i} \right)^2 + \lambda \text{rank}(H_K),$$

where λ is the tuning parameter to balance the trade-off between the two terms — the data fitting term and the model complexity term. Note that \mathcal{H}_K denotes the set of $K \times K$ Hankel matrices.

Again, this formulation is a non-convex optimization problem. Using the same idea as in the previous example, the non-convex formulation can be convexified into

$$\min_{H_K \in \mathcal{H}_K} \sum_{t=1}^N \left(y_t - \sum_{i=1}^{2K-1} h_i u_{t-i} \right)^2 + \lambda \|H_K\|_*.$$

Solving this optimization problem gives us an estimation of the impulse response of the system.

In the two examples above, we illustrate how convexification can be utilized to linear SysID problems. Actually, the idea may also be applied to nonlinear SysID tasks, such as for identifying a Hammerstein system in [26], and for identifying a monotone Wiener system in [73].

Example 1.3: Monotone Wiener SysID using convexification

A Wiener system consists of a linear dynamical part followed by a nonlinear static part. In particular, in this example, the nonlinear part $f(\cdot)$ is assumed to be non-decreasing, i.e. for $x, x' \in \mathbb{R}$, we have that

$$(x - x')(f(x) - f(x')) \geq 0. \quad (1.3)$$

Analogously to the previous example, we denote the input and output of the system as $\{u_t\}_{t=1}^{\infty}$ ($u_t = 0$ for $t \leq 0$) and $\{y_t\}_{t=1}^{\infty}$ respectively, and the infinite impulse response of the linear part as $\{h_i\}_{i=1}^{\infty}$. A sufficiently high order, say $2K - 1$, finite impulse response is used to approximate the infinite impulse response sequence. Given this, the output y_t can be written as (assume noiseless case for simplicity)

$$y_t = f\left(\sum_{i=1}^{2K-1} h_i u_{t-i}\right). \quad (1.4)$$

Assume that $\{y_t\}_{t=1}^N$ are collected. Using the similar idea as in previous examples, to find a low order linear part, the following convex optimization problem can be formulated

$$\begin{aligned} \min_{H_K \in \mathcal{H}_K} \|H_K\|_* & \quad (1.5) \\ \text{s.t.} \quad (y_t - y_{t'}) \left(\sum_{i=1}^{2K-1} h_i u_{t-i} - \sum_{i=1}^{2K-1} h_i u_{t'-i} \right) & \geq 0, \forall 1 \leq t < t' \leq N, \\ \sum_{k=1}^{2K-1} h_k & = 1, \end{aligned}$$

where $H_K(i, j) = h_{i+j-1}$ for $i, j = 1, \dots, K$. In (1.5), the first constraint is given by the non-decreasing property of $f(\cdot)$. The second constraint enforces normalization to avoid the trivial solutions. Solving this optimization problem gives an estimation of the impulse response of the linear part of the system.

Note that by assuming the Lipschitz property of the nonlinear part, the first constraint can be further tightened, see e.g. the discussions in [73].

From these examples, it is evident that the convexification technique is a quite convenient tool to use when it is applicable. However, since the convex problem is an approximation to the original non-convex problem, a relevant question is to understand how well (or poor) the approximation can be. There have been many results in this direction, see e.g. [80, 9, 14], but most of them are established for general non-structured matrix cases, and usually strong stochastic assumptions are made to make certain 'restricted isometry' property to hold. When those assumptions do not hold, establishing the performance guarantees often becomes more challenging.

Finally, let us point out that the l_1 norm of a vector $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\|_1 \triangleq \sum_{i=1}^n |x_i|$, can be given as the nuclear norm of the matrix $\text{diag}(\mathbf{x})$, that is

$$\|\mathbf{x}\|_1 = \|\text{diag}(\mathbf{x})\|_*$$

It is also clear that

$$\|\mathbf{x}\|_0 = \text{rank}(\text{diag}(\mathbf{x})),$$

where $\|\mathbf{x}\|_0$ indicates the nonzero elements of vector \mathbf{x} . Using this relation, it can be established that $\|\mathbf{x}\|_1$ is a tight convex relaxation of $\|\mathbf{x}\|_0$ as well, see the derivations in [29].

1.2.2 Recursive Bayesian methods

This section introduces two methods from the family of recursive Bayesian methods. First, the particle filter, specially the bootstrap particle filter, will be introduced; afterwards, the idea of the particle Metropolis-Hastings sampler will be reviewed. These two methods will be used in chapter 4.

Particle Filter

The basic idea underlying a Particle Filter (PF) is to approximate the filtering Probability Density Function (PDF) of a dynamical system's state vector as a set of samples (the particles) with proper weights. And given by the dynamic nature of the system, the particle-based approximated PDF can then be propagated sequentially as time goes on.

In the following, we will review some basic ideas of the bootstrap particle filter [36]. See also the recent tutorial paper [87], which gives a nice review of the (general) particle filter and its application to system identification.

Assume that the underlying dynamical system is described by the following equations

$$x_{t+1}|x_t \sim f_\theta(x_{t+1}|x_t), \tag{1.6}$$

$$y_t|x_t \sim g_\theta(y_t|x_t), \tag{1.7}$$

$$x_1 \sim \mu_\theta(x_1), \tag{1.8}$$

where the states are denoted by $x_t \in \mathbb{R}^n$ and observations are denoted by $y_t \in \mathbb{R}^m$, and θ represents the system parameters. The state filtering task is to infer the PDF $p_\theta(x_t|y_{1:t})$. When the model is linear and $f_\theta(\cdot)$ and $g_\theta(\cdot)$ are Gaussian, the filtering problem is solved by the Kalman filter [56]. However, in non-linear non-Gaussian case, the filtering problem is not analytically solvable in general.

Now we start to introduce how the bootstrap particle filter [36] is derived for estimating the filtering PDF $p_\theta(x_t|y_{1:t})$. Let us begin with the *initialization* step at $t = 1$ to get the particle approximation of $p(x_1|y_1)$. To do that, a set of particles $\{x_1^i\}_{i=1}^N$ is simulated according to $x_1^i \sim \mu_\theta(x_1)$ and their corresponding unnormalized and normalized weights are given by $\bar{w}_1^i = g(y_1|x_1^i)$ and

$$w_1^i = \frac{\bar{w}_1^i}{\sum_{k=1}^N \bar{w}_1^k},$$

which gives the particle approximation of $p(x_1|y_1)$ as

$$\hat{p}_\theta(x_1|y_1) = \sum_{i=1}^N w_1^i \delta(x_1 - x_1^i).$$

Remark 1.1. Note that if there is no y_1 available in the initialization step, the unnormalized and normalized weights for particles $\{x_1^i\}_{i=1}^N$ are given by $\bar{w}_1^i = 1$ and $w_1^i = 1/N$.

For $t \geq 2$, we start by noticing the following recursions which are given by the dynamics of the system:

$$p_\theta(x_t|y_{1:t}) = \frac{g_\theta(y_t|x_t)p_\theta(x_t|y_{1:t-1})}{p_\theta(y_t|y_{1:t-1})}, \quad (1.9)$$

and

$$p_\theta(x_t|y_{1:t-1}) = \int f_\theta(x_t|x_{t-1})p_\theta(x_{t-1}|y_{1:t-1})dx_{t-1}. \quad (1.10)$$

The previous two equations describe the relation between $p_\theta(x_t|y_{1:t})$ and $p_\theta(x_{t-1}|y_{1:t-1})$, and given by this relation, the bootstrap PF runs in the following recursive way. Suppose a particle approximation of $p_\theta(x_{t-1}|y_{1:t-1})$ is given by

$$\hat{p}_\theta(x_{t-1}|y_{1:t-1}) = \sum_{i=1}^N w_{t-1}^i \delta(x_{t-1} - x_{t-1}^i),$$

in which $\{x_{t-1}^i\}_{i=1}^N$ are the particles, and $\{w_{t-1}^i\}_{i=1}^N$ are their corresponding normalized weights, representing the relative importance of the corresponding particles.

Inserting this approximation into the recursions in (1.9) and (1.10), we have that

$$p_{\theta}(x_t|y_{1:t}) \cong \frac{g_{\theta}(y_t|x_t)}{p_{\theta}(y_t|y_{1:(t-1)})} \sum_{i=1}^N w_{t-1}^i f_{\theta}(x_t|x_{t-1}^i). \quad (1.11)$$

Notice that (1.11) can be thought as a mixture of PDFs, hence sampling from this PDF can be accomplished by the following steps.

First, a set of N indexes $\{a_t^i\}_{i=1}^N$ are sampled independently according to the following distribution. For each $i \in [1, N]$

$$\mathbb{P}(a_t^i = j) = w_{t-1}^j, \text{ for } j = 1, \dots, N,$$

where a_t^i is called the ancestor index. This step is commonly referred to as the resampling step.

Remark 1.2. *Practically, this step will be able to avoid the weight collapse phenomena (which could happen when a_t^i is set to be i), saying that a small number of the weights dominate all the others. On the other hand, this step also causes the issues that: 1) Some particles will lose their descendants, leading to the so-called path degeneracy phenomena, which introduces additional difficulties for the state smoothing problems [62]; 2) Computationally, this step also brings difficulties for parallel implementation of the algorithm, see e.g. [70, 43] for more discussions.*

Next, for each $\hat{x}_{t-1}^i \triangleq x_{t-1}^{a_t^i}$, where $i \in [1, N]$, generate a new sample (particle) x_t^i according to (commonly referred to as 'propagation' step)

$$x_t^i \sim f_{\theta}(x_t|x_{t-1}^i),$$

and after that, assign the new sample x_t^i with weight (commonly referred to as 'weighting' step)

$$\bar{w}_t^i = g_{\theta}(y_t|x_t^i).$$

After all the N samples $\{x_t^i\}_{i=1}^N$ are generated, the weights $\{\bar{w}_t^i\}_{i=1}^N$ are normalized to $\{w_t^i\}_{i=1}^N$ according to

$$w_t^i = \frac{\bar{w}_t^i}{\sum_{k=1}^N \bar{w}_t^k}.$$

In conclusion, the distribution

$$\hat{p}_{\theta}(x_t|y_{1:t}) = \sum_{i=1}^N w_t^i \delta(x_t - x_t^i) \quad (1.12)$$

represents the particle approximation of $p_{\theta}(x_t|y_{1:t})$, the filtering distribution at time t . An illustration of the 'Resampling', 'Propagation', 'Weighting' steps is given in Figure 1.4.

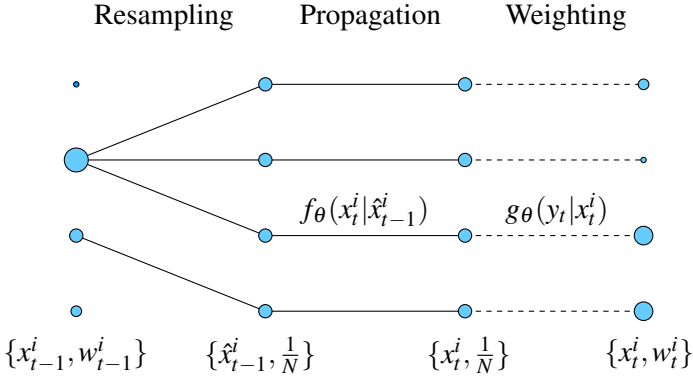


Figure 1.4: This figure illustrates the idea of the 'Resampling + Propagation + Weighting' steps for the bootstrap particle filter, where $N = 4$ and the particles are numbered from 1 to 4 from top to bottom. The size for each 'particle' represents its corresponding normalized weight. The particles x_t^1 , x_t^2 and x_t^3 share a common ancestor, which is x_{t-1}^2 , i.e. $a_t^1 = a_t^2 = a_t^3 = 2$. In addition, $a_t^4 = 3$, thus particles x_{t-1}^1 and x_{t-1}^4 are not resampled.

Remark 1.3. Whenever we have (1.12), a Monte Carlo approximation of the integral $\mathbb{E}\{g(x_t)\}$ for any test function $g(x_t)$, where the expectation is with respect to $p_\theta(x_t|y_{1:t})$, can be computed as

$$\mathbb{E}\{g(x_t)\} = \int g(x_t) p_\theta(x_t|y_{1:t}) dx_t \approx \sum_{i=1}^N w_t^i g(x_t^i). \quad (1.13)$$

Under certain conditions, the estimate in (1.13) satisfies a central limit theorem as N (t is fixed) tends to infinity, see e.g. [19]. If the system satisfies certain 'forgetting' conditions, in the sense that the dependence between x_t and x_s gets smaller as $|t - s|$ increases, the variance of the estimate in (1.13) can be uniformly bounded over time, see e.g. [104].

Note that more efficient (in the sense of having smaller variance of the normalized weights) proposal distributions (i.e. the PDFs which are used to generate samples in the initialization and propagation steps) can be designed for the particle filter. The key is to incorporate information about the measurement y_t . See for example the auxiliary particle filters introduced in [76] and the MIS particle filter in [57]. In example 1.4, we illustrate the performance of the bootstrap PF for state inference in a toy linear Gaussian model.

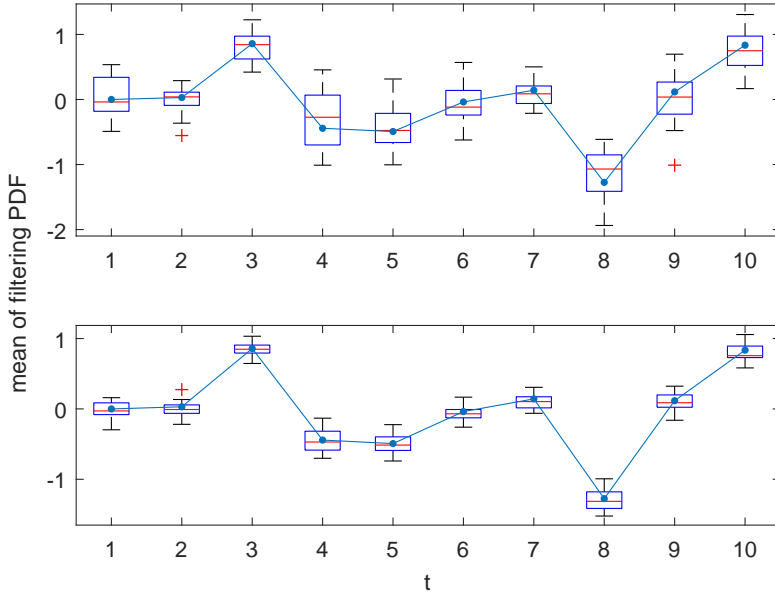


Figure 1.5: The dots in both plots represent the mean value (of the filtering PDF) given by the Kalman filter. Both plots also give the boxplot of 20 approximations to the mean value (of the filtering PDF) by running 20 independent bootstrap PFs, with 10 particles (the upper one) and 20 particles (the lower one). It can be observed that, as the number of particles increases, the approximation gets closer to the analytical result given by the Kalman filter.

— **Example 1.4: Bootstrap PF applied to a linear Gaussian model** —

In this example, we will apply the bootstrap PF for the state inference in the following first order autoregressive system

$$x_t = 0.8x_{t-1} + v_t, \tag{1.14}$$

$$y_t = x_t + e_t, \tag{1.15}$$

where $\{v_t\}_{t=2}^T$, $\{e_t\}_{t=2}^T$ and x_1 are all standard normal distributed random variables. For such a system, the exact filtering PDF $p(x_t|y_{1:t})$ can be analytically calculated by the Kalman filter. The mean value of the filtering PDF given by both the Kalman filter and the particle filter are displayed in Figure 1.5 for $T = 10$ measurements. More discussions can be found in the figure's caption.

In the following, some useful facts from the construction of the bootstrap PF are stated. For notational convenience, let

$$\mathbf{x}_t \triangleq \{x_t^1, \dots, x_t^N\} \quad \text{for } t = 1, \dots, T$$

and

$$\mathbf{a}_t \triangleq \{a_t^1, \dots, a_t^N\} \quad \text{for } t = 2, \dots, T$$

and $\mathbf{u} = \{\mathbf{x}_{1:T}, \mathbf{a}_{2:T}\}$. Given by the procedure of the bootstrap particle filter, the distribution of \mathbf{u} can be written as

$$p_{\theta}(\mathbf{u}) = \prod_{i=1}^N \mu_{\theta}(x_1^i) \prod_{t=2}^T \prod_{i=1}^N w_{t-1}^{a_t^i} f_{\theta}(x_t^i | x_{t-1}^i).$$

An interesting fact is that the particle filter can provide an unbiased estimate of the likelihood $p_{\theta}(y_{1:T})$, i.e.

$$\mathbb{E}_{p_{\theta}(\mathbf{u})}(\hat{p}(y_{1:T})) = p_{\theta}(y_{1:T}), \quad (1.16)$$

where $\hat{p}_{\theta}(y_{1:T}) = \prod_{s=1}^T \{\frac{1}{N} \sum_{i=1}^N \bar{w}_s^i\}$, see e.g. [60, 75]. This result is of fundamental importance when deriving the particle Metropolis-Hastings sampler.

Particle Metropolis-Hastings sampler

In this part, we consider the case when the parameter θ is modeled as a random variable, and the randomness is described by a prior distribution as $\pi(\theta)$. When $y_{1:T}$ are obtained, what people often interested in is to draw samples from the posterior distribution $p(\theta|y_{1:T})$, which for example can be used to estimate the posterior mean and variance.

It is not always an easy task to draw samples from this distribution, for the reason that the likelihood of the data, i.e. $p_{\theta}(y_{1:T})$, is hard to evaluate. However, as explained before, we do have an unbiased estimate of the likelihood from the PF. This observation motivates us to consider the following function $\phi(\theta, \mathbf{u})$

$$\phi(\theta, \mathbf{u}) = \frac{\pi(\theta)p_{\theta}(\mathbf{u})\hat{p}_{\theta}(y_{1:T})}{p(y_{1:T})}, \quad (1.17)$$

which can be further proven to be a PDF, and leaves $p(\theta|y_{1:T})$ as its marginal.

Given (1.16), it is easy to see that

$$\int \phi(\theta, \mathbf{u}) d\mathbf{u} = \frac{\pi(\theta)p_{\theta}(y_{1:T})}{p(y_{1:T})} = p(\theta|y_{1:T}). \quad (1.18)$$

Given these facts, the Particle Metropolis-Hastings (PMH) sampler then runs a standard Metropolis-Hastings sampler to draw samples from (1.17). Given a sample $(\theta[m], \mathbf{u}[m])$, the PMH sampler generates samples for the next step according to the following proposal

$$\theta' \sim q(\theta'|\theta[m]), \quad \mathbf{u}' \sim p_{\theta'}(\mathbf{u}'). \quad (1.19)$$

The sample (θ', \mathbf{u}') will be accepted, i.e. $(\theta[m+1], \mathbf{u}[m+1]) = (\theta', \mathbf{u}')$, with probability α given by

$$\alpha = \min\left(1, \frac{\hat{p}_{\theta'}(y_{1:T})p_{\theta'}(\mathbf{u}')\pi(\theta')/p(y_{1:T})}{\hat{p}_{\theta[m]}(y_{1:T})p_{\theta[m]}(\mathbf{u}[m])\pi(\theta[m])/p(y_{1:T})} \frac{q(\theta[m]|\theta')p_{\theta[m]}(\mathbf{u}[m])}{q(\theta'|\theta[m])p_{\theta'}(\mathbf{u}')}\right),$$

which can be simplified to

$$\alpha = \min\left(1, \frac{\hat{p}_{\theta'}(y_{1:T})\pi(\theta')}{\hat{p}_{\theta[m]}(y_{1:T})\pi(\theta[m])} \frac{q(\theta[m]|\theta')}{q(\theta'|\theta[m])}\right).$$

If (θ', \mathbf{u}') is rejected, set $(\theta[m+1], \mathbf{u}[m+1]) = (\theta[m], \mathbf{u}[m])$. This step will be run for certain number of iterations to avoid the burn-in period for the convergence to the stationary distribution (1.17). Finally, we make several remarks:

- Inspecting the derivations above, it can be observed that the only requirement to assure the exact sampling from $p(\theta|y_{1:T})$ is that a nonnegative unbiased estimate of $p_{\theta}(y_{1:T})$ is given. This observation can be used to derive new algorithms for generating samples from $p(\theta|y_{1:T})$, which is referred as the pseudo-marginal MCMC framework [2]. In the previous derivations, an unbiased estimate of $p_{\theta}(y_{1:T})$ is obtained by running a bootstrap PF.
- To generate samples from the joint distribution $p(\theta, x_{1:T}|y_{1:T})$, the Particle Marginal Metropolis-Hastings (PMMH) sampler and the Particle Gibbs (PG) sampler can be utilized. These methods are derived by considering the following PDF, which is an extended version of (1.17), given as

$$\phi(\theta, \mathbf{u}, k) = \frac{\pi(\theta)p_{\theta}(\mathbf{u})\hat{p}_{\theta}(y_{1:T})w_T^k}{p(y_{1:T})}, \quad (1.20)$$

where $k \in [1 : N]$ indexes one of the particles at time T , which also implicitly indexes one ancestral trajectory given by $\{x_{b_1}, \dots, x_{b_T}\}$, where $b_T = k$, and $b_t = a_{t+1}^{b_{t+1}}$ for $t = 1, \dots, T-1$. It can be established that (1.20) admits $p(\theta, x_{1:T}|y_{1:T})$ as its marginal. With this fact, a Metropolis-Hastings sampler or a Gibbs sampler, working together with a particle filter, can be utilized to generate samples from (1.20), hence samples from $p(\theta, x_{1:T}|y_{1:T})$. For more details, we refer to [1].

1.3 Thesis contribution and outline

Generally speaking, this thesis makes contributions in the following two directions:

- To understand how the convexification idea can be useful for some problems in system identification.
- Provide additional insights, either new derivations or new observations, to some existing recursive algorithms.

In this following, we will state our contributions for the problems studied in the thesis, chapter by chapter.

1.3.1 Chapter 2

This chapter studies an approach for efficiently estimating a sparse vector from an overdetermined linear system which are perturbed by Gaussian noise. In particular, we assume that the overdetermined observation matrix satisfies certain property which is close to the Persistent Exciting condition.

The proposed estimator consists of three steps : 1) An ordinary Least Squares Estimate (LSE); 2) The support set is estimated by solving a Linear Programming (LP) optimization problem, whose solution is given by a soft thresholding step; 3) A de-biasing step using an ordinary LSE based on the estimated support set from the second step.

The main result of this chapter establishes that when the number of observations goes to infinity, the proposed estimator is able to detect the true support set almost surely, under certain assumptions of the observation matrix. This chapter is based on the following work

- Liang Dai and Kristiaan Pelckmans, Sparse estimation from noisy observations of an overdetermined linear system, *Automatica*, vol. 50, no. 11, pp. 2845-2851, 2014.

1.3.2 Chapter 3

As illustrated in the previous sections, when convexification is applied to some SysID tasks, the matrices appearing in the final optimization problems will be of low rank, and of Hankel structure as well. However, there are limited results in understanding when and how the nuclear norm heuristic will work in such cases. This chapter focuses on the study of the completion of a low rank Hankel matrix using the nuclear norm heuristic, which is related to the recovery of the impulse response of a stable linear system.

For a stable system with a single real pole, it is proven that the nuclear norm heuristic can successfully complete the related Hankel matrix; However, for a slightly more complicated case, i.e. when the stable system has two real poles, it is found that the nuclear norm heuristic can not always complete the Hankel matrix, which illustrates some limitations of the nuclear norm heuristic when applied to such problems.

The main part of this chapter is spent on building the *certificate* to guarantee the successful completion of the related Hankel matrix, which relies heavily on the structure information in the problem setting. This chapter is based on the following work

- Liang Dai and Kristiaan Pelckmans, On the nuclear norm heuristic for a Hankel matrix completion problem, *Automatica*, vol. 51, no. 1, pp. 268-272, 2015.

1.3.3 Chapter 5

Recursive Direct Weight Optimization (RDWO) is an online local modeling algorithm to build the mathematical model for the underlying system. The idea behind this approach is to minimize the probability that an upper bound of the estimation error is larger than a given threshold. It has the nice properties that the optimization problem has a closed form solution, and the solution can be updated online as new observations are obtained.

Though the RDWO algorithm is elegant and useful, its existing derivation is relatively involved. A closer look into the algorithm reveals that there is certain structure information inherent in the algorithm formulation which is under-exploited. The contribution of this chapter lies in proposing a novel and simpler derivation of the RDWO method by making use of the under-exploited information. A by-product is that a more compact algorithm description is obtained as well. This chapter is based on the following work

- Liang Dai and Thomas B. Schön, A new structure exploiting derivation of Recursive Direct Weight Optimization, *IEEE transactions on Automatic Control*, vol. 60, no. 6, pp. 1683-1685, 2015.

1.3.4 Chapter 4

When gradient based algorithms are used for Maximum Likelihood estimation, the score function, i.e. the gradient of the log-likelihood function, is often needed. In some cases, it is possible to estimate the score function by Fisher's identity [78]. However, when the system state transition kernel is not known explicitly, this approach is not applicable anymore.

By building upon a recent idea in [23], this chapter gives some new insights for the score function estimation for systems with intractable state transition kernels. The main idea of the work is to take a probabilistic view of a basic property of the convolution operator. By this viewpoint, the score function can be estimated by sampling from the parameter posterior distribution, which can be accomplished by an application of the particle MCMC method. This chapter is based on the following work

- Liang Dai and Thomas B. Schön, Using convolution to estimate the score function for intractable state transition models, to appear in *IEEE Signal Processing Letters*, 2016.

1.3.5 Chapter 6

The Kaczmarz Algorithm (KA) [55] is an iterative algorithm for solving a system of linear equations $\{\mathbf{a}_i^T \mathbf{x} = b_i\}_{i=1}^m$, by sequentially projecting onto the hyperplanes defined by these equations. Recently, the Randomized Kaczmarz Algorithm (RKA) [95] improves the convergence rate over the classical KA by implementing a random projection at each step.

In the original work of RKA in [95], the probability to project onto the hyperplane $\{\mathbf{a}_i^T \mathbf{x} = b_i\}$ is proportional to $\|\mathbf{a}_i\|^2$. However, this probability distribution is questioned in [16] about its optimality. One contribution of this chapter establishes the result that it is possible to find better probability distributions, which can improve the convergence rate of the RKA further. The key underlying the work is to optimize a tight upper bound to the convergence rate of the RKA.

It is pointed in [72] that, 'although the Kaczmarz method is popular in practice, theoretical results on the convergence rate of the method have been difficult to obtain'. Another contribution of this chapter lies in presenting an approach to study the convergence properties of the KA, by relating the process of the algorithm to a linear time varying dynamical system. Using this relation, the convergence rate of the KA is obtained by studying the stability property for the related dynamical system. This chapter is based on the following work

- Liang Dai, Mojtaba Soltanalian and Kristiaan Pelckmans, On the randomized Kaczmarz algorithm, *IEEE Signal Processing Letters*, vol. 21, no. 3, pp. 330-333, 2013.
- Liang Dai and Thomas B Schön, On the exponential convergence of the Kaczmarz algorithm, *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1571-1574, 2015.

1.4 Other contributions

Besides the contributions listed above, the author has also contributed to the following results during his Ph.D. studies.

- Kristiaan Pelckmans and Liang Dai, A simple recursive algorithm for learning a monotone Wiener system, In Proceedings of the *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando, Florida, USA, December, 2011.
- Liang Dai and Kristiaan Pelckmans, An ellipsoid based, two-stage screening test for BPDN, In Proceedings of the *20th European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania, August, 2012.
- Liang Dai and Kristiaan Pelckmans, An online algorithm for controlling a monotone Wiener system, In Proceedings of the *24th Chinese Control and Decision Conference (CCDC)*, Taiyuan, China, May, 2012.
- Thomas B. Schön, Fredrik Lindsten, Johan Dahlin, Johan Wägberg, Christian A. Naesseth, Andreas Svensson, and Liang Dai, Sequential Monte Carlo methods for system identification, In Proceedings of the *17th IFAC Symposium on System Identification (SYSID)*, Beijing, China, October, 2015.

- Liang Dai, Kristiaan Pelckmans and Er-Wei Bai, Identifiability and convergence analysis of the MINLIP estimator, *Automatica*, vol. 51, no. 1, pp. 104-110, 2015.

Chapter 2

Sparse estimation from an overdetermined linear system

2.1 Problem formulation

This chapter considers the estimation of a sparse parameter vector from noisy observations of a linear system. The formal definition and assumptions of the problem are given as follows. Let $n > 0$ be a fixed number, denoting the dimension of the underlying parameter vector, and let $N > 0$ denote the number of equations ('observations'). The observed signal $\mathbf{y} \in \mathbb{R}^N$ obeys the following relation:

$$\mathbf{y} = \mathbf{A}\mathbf{x}^0 + \mathbf{v}, \quad (2.1)$$

where the elements of the vector $\mathbf{x}^0 \in \mathbb{R}^n$ are considered to be the fixed, but unknown parameters of the system. Moreover, it is assumed that \mathbf{x}_0 is s -sparse (i.e. there are s nonzero elements in the vector). Let $\mathcal{T} \subset \{1, \dots, n\}$ denote the support set of \mathbf{x}^0 (i.e. $\mathbf{x}_i^0 = 0 \Leftrightarrow i \notin \mathcal{T}$) and \mathcal{T}^c be the complement of \mathcal{T} , i.e. $\mathcal{T} \cup \mathcal{T}^c = \{1, 2, \dots, n\}$ and $\mathcal{T} \cap \mathcal{T}^c = \emptyset$. The elements of the vector $\mathbf{v} \in \mathbb{R}^N$ are assumed to follow the distribution

$$\mathbf{v} \sim \mathcal{N}(0, c\mathbf{I}_N), \quad (2.2)$$

where $0 < c \in \mathbb{R}$.

Applications of such a setup appear in many places, to name a few, see the applications discussed in [59] on the detection of nuclear material, and in [58] on model selection for aircraft test modeling. In the numerical illustrations provided in Section 2.4, we will demonstrate an example about the line spectral estimation [93].

The matrix $\mathbf{A} \in \mathbb{R}^{N \times n}$ with $N > n$ is called the sensing (or observation) matrix. Such a setting (\mathbf{A} is a 'tall' matrix) makes it different from the setting studied in compressive sensing, where the sensing matrix is 'fat', i.e. $N \ll n$. For an introduction to the compressive sensing theory, see e.g. [21, 15].

In this chapter, we assume that the matrix \mathbf{A} is always of full column rank. Denote the Singular Value Decomposition (SVD) of matrix $\mathbf{A} \in \mathbb{R}^{N \times n}$ as

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (2.3)$$

in which $\mathbf{U} \in \mathbb{R}^{N \times n}$ satisfies $\mathbf{U}^T \mathbf{U} = \mathbf{I}_n$, $\mathbf{V} \in \mathbb{R}^{n \times n}$ satisfies $\mathbf{V}^T \mathbf{V} = \mathbf{I}_n$, and $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ is a diagonal matrix $\mathbf{\Sigma} = \text{diag}(\sigma_1(\mathbf{A}), \sigma_2(\mathbf{A}), \dots, \sigma_n(\mathbf{A}))$, where $\sigma_i(\mathbf{A})$ denotes the i -th singular value of the matrix \mathbf{A} . We further make the following assumptions on \mathbf{A} :

Definition 2.1. *The matrix sequence $\{\mathbf{A}_N \in \mathbb{R}^{N \times n}\}_N$ is sufficiently rich if there exist a finite N_0 and $0 < c_1 \leq c_2$, such that for all $N > N_0$, the matrices $\{\mathbf{A}_N\}_{N > N_0}$ satisfy*

$$c_1 \sqrt{N} \leq \sigma_1(\mathbf{A}_N) \leq \sigma_2(\mathbf{A}_N) \leq \dots \leq \sigma_n(\mathbf{A}_N) \leq c_2 \sqrt{N}. \quad (2.4)$$

To avoid notational overload, the dependence of \mathbf{A} on N is not stated explicitly when no confusion occurs in what follows.

Remark 2.1. *In [81] and [109], the authors make the assumption on \mathbf{A} that the sample covariance matrix $\frac{1}{N} \mathbf{A}^T \mathbf{A}$ converges to a finite, positive-definite matrix:*

$$\lim_{N \rightarrow \infty} \frac{1}{N} \mathbf{A}^T \mathbf{A} = \mathbf{D} \succ 0. \quad (2.5)$$

This assumption is also known as Persistent Excitation (PE), see e.g. [91]. Note that the assumption in (2.4) covers a wider range of cases. For example, it does not require the singular values of $\frac{1}{\sqrt{N}} \mathbf{A}$ to converge, while only requiring that these singular values lie in the interval $[c_1, c_2]$ when N increases.

Classically, given by the Gauss-Markov theorem [77], the ordinary Least Square Estimate (LSE) of \mathbf{x}^0 under the model described in (2.1) and (2.2) is the Best Linear Unbiased Estimate (BLUE) of \mathbf{x}^0 . However, the ordinary LSE does not utilize the 'sparsity' information of \mathbf{x}^0 , which raises the question of whether it is possible to improve on the ordinary LSE by exploiting the 'sparse' property of \mathbf{x}^0 .

In the literature, several approaches have been suggested to improve the classical LSE by making use of the sparsity information, which can perform as if the true support of \mathbf{x}^0 were known. Such a property is coined as the ORACLE property in [27]. In what follows, we will have a brief overview of these approaches. In [27], the SCAD (Smoothly Clipped Absolute Deviation) estimator is presented, which is given by solving a non-convex optimization problem. Later in [109], the ADALASSO (Adaptive Least Absolute Shrinkage and Selection Operator) estimator is presented. The ADALASSO estimator consists of two steps, which implements an ordinary LSE in the first step,

and then a reweighted Lasso optimization problem is solved in the next step. Recently, in [81], two LASSO-based estimators, namely the 'A-SPARSEVA-AIC-RE' method and the 'A-SPARSEVA-BIC-RE' method, are suggested. Both methods need to do the ordinary LSE in the first step, after that, a Lasso optimization problem is solved, and finally re-estimate the non-zeros of \mathbf{x}^0 using the ordinary LSE.

In this chapter, another approach to estimate the sparse vector \mathbf{x}^0 will be presented, which also possesses the ORACLE [27] property. The proposed method consists of three steps, in the first step, an ordinary LSE is conducted, the second step is to solve a Linear Programming problem, whose solution is given by a soft-thresholding step, finally, redo the LSE to improve the estimate of the non-zero elements of \mathbf{x}^0 . Details will be given in Section 2.2.

In the following part of the chapter, the lower bold case will be used to denote a vector and capital bold characters are used to denote matrices. The subsequent sections are organized as follows. In Section 2.2, we will describe the algorithm and an analytical solution to the LP problem will also be given. In Section 2.3, we will analyze the algorithm in detail. In Section 2.4, we conduct several examples to illustrate the efficacy of the proposed algorithm and compare the it with some other algorithms. Finally, we conclude this chapter in Section 2.5.

2.2 Algorithm description

The algorithm consists of the following three steps:

1. *LSE*: Compute the ordinary LSE of \mathbf{x}^0 , denoted as \mathbf{x}^{ls} .
2. *LP*: Select $0 < \varepsilon < 1$, and let $\lambda = \sqrt{\frac{2n}{N^{1-\varepsilon}}}$, then solve the following Linear Programming problem:

$$\mathbf{x}^{\text{lp}} = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ s.t. } \|\mathbf{x} - \mathbf{x}^{\text{ls}}\|_{\infty} \leq \lambda, \quad (2.6)$$

and find the support set \mathcal{T}^{lp} of \mathbf{x}^{lp} as an estimate of \mathcal{T} .

3. *RE-LSE*: Compute the LSE of \mathbf{x}^0 based on \mathcal{T}^{lp} . Form the matrix $\mathbf{A}_{\mathcal{T}^{\text{lp}}}$, which consists the columns of \mathbf{A} indexed by \mathcal{T}^{lp} and let $\mathbf{A}_{\mathcal{T}^{\text{lp}}}^{\dagger}$ denote its pseudo-inverse. Then the final estimation \mathbf{x}^{rels} is given by

$$\mathbf{x}_{\mathcal{T}^{\text{lp}}}^{\text{rels}} = \mathbf{A}_{\mathcal{T}^{\text{lp}}}^{\dagger} \mathbf{y},$$

and

$$\mathbf{x}_{\mathcal{T}^{\text{lp}}^c}^{\text{rels}} = \mathbf{0},$$

in which $\mathcal{T}^{\text{lp}^c}$ denotes the complement set of \mathcal{T}^{lp} .

Note that the LP problem in (2.6) has an analytical solution. The reasoning is as follows. Writing the ∞ -norm constraint out explicitly, we get

$$\begin{aligned} \mathbf{x}^{\text{lp}} &= \arg \min_{x_1, \dots, x_n} \sum_{i=1}^n |x_i| \\ \text{s.t. } & |x_i - x_i^{\text{ls}}| \leq \lambda, \text{ for } i = 1 \dots n. \end{aligned} \quad (2.7)$$

From the new formulation, we can see that there are no cross terms in both the objective function and the constraint inequalities, hence each component can be optimized separately. From this observation, the solution of the LP problem is given as

$$x_i^{\text{lp}} = \begin{cases} 0, & \text{if } |x_i^{\text{ls}}| \leq \lambda \\ x_i^{\text{ls}} - \lambda, & \text{if } x_i^{\text{ls}} > \lambda \\ x_i^{\text{ls}} + \lambda, & \text{if } x_i^{\text{ls}} < -\lambda \end{cases}$$

for $i = 1, 2, \dots, n$. Such a solution \mathbf{x}^{lp} is also referred to as an application of the soft-thresholding (ST) operation to \mathbf{x}^{ls} , see e.g. [22]. Several remarks are given as follows.

Remark 2.2. *The order of λ chosen as $-\frac{1}{2} + \frac{\varepsilon}{2}$ is essential to make the asymptotic oracle property hold. Intuitively speaking, such a choice can make the following two facts hold.*

1. *Whenever $\varepsilon > 0$, \mathbf{x}^0 will lie in the feasible region of (2.6) with high probability.*
2. *The threshold decreases 'slower' (in the order of N) than the variance of the pseudo noise term $\mathbf{V}\Sigma^{-1}\mathbf{U}^T\mathbf{v}$. With such a choice, it is possible to get a good approximation of the support set of \mathbf{x}^0 in the second step.*

Remark 2.3. *Note that the formulation in (2.6) is inspired by the Dantzig selector in [8]. The similarities and differences between them are given as follows.*

1. *Both (2.6) and the Dantzig selector lie in the following class*

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \|\mathbf{W}(\mathbf{x} - \mathbf{x}^{\text{ls}})\|_\infty \leq \lambda. \quad (2.8)$$

If \mathbf{W} is chosen as the identity matrix, we obtain the proposed method; If \mathbf{W} is chosen as $\mathbf{A}^T\mathbf{A}$, then we obtain the same formulation as given by the Dantzig selector.

2. *As pointed out in [25], the solution path of the Dantzig selector behaves erratically with respect to the value of the regularization parameter. However, the solution path of (2.6) with respect to the value of λ behaves regularly. This is due to the fact that, given λ , the solution to (2.6) is given by the soft-thresholding operation (with the threshold λ) to the ordinary LSE \mathbf{x}^{ls} . When λ increases, the solution will decrease*

(or increase) linearly and when it hits zero, it will remain to be zero. An illustration of the solution path is given as follows. Assume that $n = 4$ and $\mathbf{x}^{ls} = [2, 0.5, -1, -1.5]^T$, then the solution path to (2.6) w.r.t. λ is given in Figure 2.1.

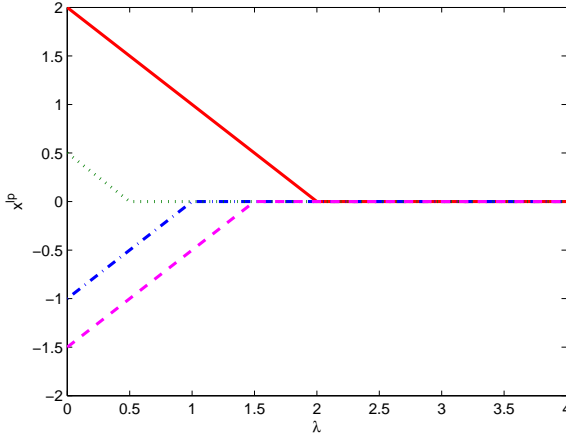


Figure 2.1: An illustration of the solution path to (2.6) w.r.t. λ . When λ equals zero, the solution to (2.6) is \mathbf{x}^{ls} ; when λ increases, the solution trajectory shrinks linearly towards zero and then remains zero.

Remark 2.4. *From a computational point of view, the SCAD needs to solve a non-convex optimization problem which will suffer from multiple local minimas [42]. Regarding this, the proposed scheme is mainly compared with the methods which are formulated as convex optimization problems. In Table 2.1, we list the computational steps needed for different methods. In the table, the term ST means the soft-thresholding operation, the term Re-LSE means 'redo the LSE after detecting the support set of the estimate obtained from the second step'. From this table, we can see that in the first step, all the methods need to do the ordinary LSE; in the second step, except the proposed method (which is denoted by LP + Re-LSE), the other methods need to solve a LASSO optimization problem, which could be more computationally involved than a simple soft-thresholding operation; except the ADALASSO method, the other methods all need to do a Re-LSE step.*

Remark 2.5. *Note that the proposed method does not need an "adaptive step" (i.e. to reweigh the cost function) in order to achieve the ORACLE property, which is different from the methods presented in [81] and [109].*

Table 2.1: Computational steps needed for different methods

| | Step 1 | Step 2 | Step 3 |
|-------------------|--------|--------|--------|
| LP + Re-LSE | LSE | ST | Re-LSE |
| ADALASSO | LSE | LASSO | |
| A-SPARSEVA-AIC-RE | LSE | LASSO | Re-LSE |
| A-SPARSEVA-BIC-RE | LSE | LASSO | Re-LSE |

2.3 Algorithm analysis

In this section, we will discuss the properties of the proposed method. For convenience, the smallest singular value of \mathbf{A} will be denoted as σ in the following, that is $\sigma \triangleq \sigma_1(\mathbf{A})$.

Remark 2.6. *In the following, we assume that the noise variance equals one, i.e. $c = 1$, for the following reasons:*

1. *When the noise variance is given in advance, one can always re-scale the problem accordingly.*
2. *Even if the noise variance is not known explicitly (but is known to be a finite value), the support of \mathbf{x}^0 will be recovered asymptotically. This is a direct consequence of the fact that finite, constant scalings do not affect the asymptotic statements, i.e. we can use the same λ for any level of variance without influencing the asymptotic behavior.*

The following facts (Lemma 2.1-2.3) will be useful for the subsequent analysis.

Lemma 2.1. *The Least Square Estimate \mathbf{x}^{ls} of \mathbf{x}^0 is given by*

$$\mathbf{x}^{\text{ls}} = \mathbf{x}^0 + \mathbf{V}\Sigma^{-1}\mathbf{U}^T\mathbf{v}.$$

Proof. This fact is due to the relation between the LSE of \mathbf{x}^0 and the property of the pseudo inverse of matrix \mathbf{A} . The calculation goes as follows.

$$\begin{aligned} \mathbf{x}^{\text{ls}} &= \mathbf{A}^\dagger \mathbf{y} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T (\mathbf{A} \mathbf{x}^0 + \mathbf{v}) \\ &= \mathbf{x}^0 + (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{v} \\ &= \mathbf{x}^0 + \mathbf{V}\Sigma^{-1}\mathbf{U}^T \mathbf{v} \end{aligned}$$

□

Lemma 2.2. *Let $\mathbf{b} = \Sigma\mathbf{V}^T\mathbf{x}^{\text{ls}} - \Sigma\mathbf{V}^T\mathbf{x}^0$, then \mathbf{b} is a Gaussian random vector with distribution $\mathcal{N}(0, \mathbf{I}_n)$.*

Proof. This fact is a direct consequence of the previous fact and the assumption that $\mathbf{v} \sim \mathcal{N}(0, \mathbf{I}_N)$. The calculation goes as follows.

$$\begin{aligned}\Sigma \mathbf{V}^T \mathbf{x}^{\text{ls}} - \Sigma \mathbf{V}^T \mathbf{x}^0 &= \Sigma \mathbf{V}^T (\mathbf{x}^{\text{ls}} - \mathbf{x}^0) \\ &= \Sigma \mathbf{V}^T \mathbf{V} \Sigma^{-1} \mathbf{U}^T \mathbf{v} \\ &= \mathbf{U}^T \mathbf{v}.\end{aligned}$$

Since $\mathbf{v} \sim \mathcal{N}(0, \mathbf{I}_N)$, it gives that $\mathbf{U}^T \mathbf{v}$ is also Gaussian distributed, with mean zero and variance $\mathbb{E}(\mathbf{U}^T \mathbf{v} \mathbf{v}^T \mathbf{U}) = \mathbf{I}_n$, which concludes the result. \square

Lemma 2.3. *Given $d > 0$ and $c > 0$, then it holds that*

$$\int_{|t|>d} \frac{1}{c\sqrt{2\pi}} e^{-\frac{t^2}{2c^2}} dt \leq e^{-\frac{d^2}{2c^2}}.$$

When $c = 1$, a proof of this result can be found in Lemma 10.1 of [79] and the general case ($c \neq 1$) can be proven by change of variables based on the case when $c = 1$.

In the following, we will first analyze the probability that \mathbf{x}^0 lies in the constraint set of the LP problem given by (2.6). Then we give an error estimate of the results given by (2.6). After that, we will discuss the capability of recovering the support set of \mathbf{x}^0 by (2.6), which will lead to the discussion about the asymptotic ORACLE property of the proposed estimator.

Lemma 2.4. *For all $\lambda > 0$, it holds that*

$$\mathbb{P}\left(\|\mathbf{V}^T \mathbf{x}^{\text{ls}} - \mathbf{V}^T \mathbf{x}^0\|_\infty > \frac{\lambda}{\sqrt{n}}\right) \leq ne^{-\frac{\lambda^2 \sigma^2}{2n}}.$$

Note that σ denotes the minimal singular value of matrix \mathbf{A} .

Proof. By Lemma 2.2, and noticing that $\mathbf{b} = \Sigma \mathbf{V}^T \mathbf{x}^{\text{ls}} - \Sigma \mathbf{V}^T \mathbf{x}^0$ is a random vector with distribution $\mathcal{N}(0, \mathbf{I})$, we have that

$$\begin{aligned}&\mathbb{P}\left(\|\mathbf{V}^T \mathbf{x}^{\text{ls}} - \mathbf{V}^T \mathbf{x}^0\|_\infty > \frac{\lambda}{\sqrt{n}}\right) \\ &\leq \mathbb{P}\left(\|\Sigma \mathbf{V}^T \mathbf{x}^{\text{ls}} - \Sigma \mathbf{V}^T \mathbf{x}^0\|_\infty > \frac{\lambda \sigma}{\sqrt{n}}\right) \\ &= \mathbb{P}\left(\|\mathbf{b}\|_\infty > \frac{\lambda \sigma}{\sqrt{n}}\right) \\ &= \mathbb{P}\left(\exists i, \text{ such that } |b_i| > \frac{\lambda \sigma}{\sqrt{n}}\right) \\ &\leq \sum_{i=1}^n \mathbb{P}\left(|b_i| > \frac{\lambda \sigma}{\sqrt{n}}\right).\end{aligned}$$

Application of Lemma 2.3 gives the desired result. \square

Lemma 2.5. For $\lambda > 0$, if $\|\mathbf{V}^T \mathbf{x}^{ls} - \mathbf{V}^T \mathbf{x}^0\|_\infty \leq \frac{\lambda}{\sqrt{n}}$, then $\|\mathbf{x}^{ls} - \mathbf{x}^0\|_\infty \leq \lambda$.

Proof. Define \mathbf{c} as $\mathbf{c} = \mathbf{V}^T \mathbf{x}^{ls} - \mathbf{V}^T \mathbf{x}^0$, resulting in $\|\mathbf{x}^{ls} - \mathbf{x}^0\|_\infty = \|\mathbf{V}\mathbf{c}\|_\infty$. Analyzing the i th element of $\mathbf{V}\mathbf{c}$ gives that

$$|\mathbf{V}_i \mathbf{c}| \leq \|\mathbf{c}\|_2 \leq \|\mathbf{c}\|_\infty \sqrt{n} \leq \lambda.$$

The first inequality is by definition, the second inequality comes from the fact that $\sum_i \mathbf{c}_i^2 \leq n \|\mathbf{c}\|_\infty^2$ and the last inequality is due to the assumption of the lemma. \square

Based on the previous results, we have that

Lemma 2.6. $\mathbb{P}(\|\mathbf{x}^{ls} - \mathbf{x}^0\|_\infty \leq \lambda) \geq 1 - ne^{-\frac{\lambda^2 \sigma^2}{2n}}$.

Proof. The proof goes as follows

$$\begin{aligned} & \mathbb{P}(\|\mathbf{x}^{ls} - \mathbf{x}^0\|_\infty \leq \lambda) \\ & \geq \mathbb{P}\left(\|\mathbf{V}^T \mathbf{x}^{ls} - \mathbf{V}^T \mathbf{x}^0\|_\infty \leq \frac{\lambda}{\sqrt{n}}\right) \\ & = 1 - \mathbb{P}\left(\|\mathbf{V}^T \mathbf{x}^{ls} - \mathbf{V}^T \mathbf{x}^0\|_\infty > \frac{\lambda}{\sqrt{n}}\right) \\ & \geq 1 - ne^{-\frac{\lambda^2 \sigma^2}{2n}} \end{aligned}$$

The first inequality comes from Lemma 2.5, and the second inequality follows from Lemma 2.4. \square

By choosing λ as the one given in the proposed algorithm, the following result is obtained.

Theorem 2.1. Given $0 < \varepsilon < 1$, and let $\lambda^2 = \frac{2n}{N^{1-\varepsilon}}$, we have that

$$\mathbb{P}\left(\|\mathbf{x}^{ls} - \mathbf{x}^0\|_\infty \leq \lambda\right) \geq 1 - ne^{-c_1^2 N^\varepsilon},$$

in which c_1 is defined in (2.4).

This theorem tells that with a proper choice of λ , \mathbf{x}^0 will lie in the feasible set of (2.6) with high probability. Next, we will derive an error bound (in the l_2 sense) of the estimate obtained by the LP formulation.

Lemma 2.7. For a given λ , if $\|\mathbf{x}^{ls} - \mathbf{x}^0\|_\infty \leq \lambda$ holds (i.e. \mathbf{x}^0 lies in the feasible set of (2.6)), then we have that $\|\mathbf{h}\|_2^2 \leq 4s\lambda^2$, in which $\mathbf{h} = \mathbf{x}^{lp} - \mathbf{x}^0$, and s denotes the number of non-zeros of \mathbf{x}^0 .

Proof. We first consider the error vector on \mathcal{T}^c which is given by $\mathbf{h}_{\mathcal{T}^c}$. Since $\|\mathbf{x}^{\text{ls}} - \mathbf{x}^0\|_\infty \leq \lambda$ and $\mathbf{x}_{\mathcal{T}^c}^0 = \mathbf{0}$, we have that $\|\mathbf{x}_{\mathcal{T}^c}^{\text{ls}}\|_\infty \leq \lambda$.

It follows from the previous discussions that \mathbf{x}^{lp} is obtained by an element-wise application of the soft-thresholding operator with the threshold λ to \mathbf{x}^{ls} , hence we obtain that $\mathbf{x}_{\mathcal{T}^c}^{\text{lp}} = \mathbf{0}$. This implies that $\mathbf{h}_{\mathcal{T}^c} = \mathbf{0}$.

Next we consider the error vector on the support \mathcal{T} , denoted as $\mathbf{h}_{\mathcal{T}}$. From the property of the soft-thresholding operation, it follows that

$$\|\mathbf{x}_{\mathcal{T}}^{\text{ls}} - \mathbf{x}_{\mathcal{T}}^{\text{lp}}\|_\infty \leq \lambda.$$

By the triangle inequality, we have that

$$\|\mathbf{h}_{\mathcal{T}}\|_\infty = \|\mathbf{x}_{\mathcal{T}}^0 - \mathbf{x}_{\mathcal{T}}^{\text{lp}}\|_\infty \leq \|\mathbf{x}_{\mathcal{T}}^{\text{ls}} - \mathbf{x}_{\mathcal{T}}^{\text{lp}}\|_\infty + \|\mathbf{x}_{\mathcal{T}}^{\text{ls}} - \mathbf{x}_{\mathcal{T}}^0\|_\infty \leq 2\lambda.$$

Finally, it follows that

$$\|\mathbf{h}\|_2^2 = \|\mathbf{h}_{\mathcal{T}}\|_2^2 + \|\mathbf{h}_{\mathcal{T}^c}\|_2^2 \leq |\mathcal{T}| \|\mathbf{h}_{\mathcal{T}}\|_\infty^2 \leq 4s\lambda^2.$$

□

Having obtained \mathbf{x}^{lp} , the support set of \mathbf{x}^0 can be estimated by the support set of \mathbf{x}^{lp} (as in the second step of the algorithm). The following results will be about the asymptotic (in the sense of when N becomes large) behavior of the support estimation. In the following, we will denote the estimated support set as $\mathcal{T}^{\text{lp}}(N)$ when we have N observations.

We will first get a weak support recovery result, and based on this, we can further prove that the support as recovered by the LP formulation will converge to the true support \mathcal{T} with probability 1 when N goes to infinity.

Lemma 2.8. *Given $0 < \varepsilon < 1$, and assume that the matrix \mathbf{A}^1 satisfies (2.4) with constants c_1, c_2 . Let $\lambda^2 = \frac{2n}{N^{1-\varepsilon}}$ as given in the algorithm, then*

$$\lim_{N \rightarrow \infty} \mathbb{P}(\mathcal{T}^{\text{lp}}(N) = \mathcal{T}) = 1.$$

Proof. Let $\bar{\mathbf{v}} = \mathbf{V}\Sigma^{-1}\mathbf{U}^T\mathbf{v}$. Since $\mathbf{x}^{\text{ls}} = \mathbf{x}^0 + \mathbf{V}\Sigma^{-1}\mathbf{U}^T\mathbf{v}$, we have that $\mathbf{x}^{\text{ls}} = \mathbf{x}^0 + \bar{\mathbf{v}}$, in which $\bar{\mathbf{v}}$ follows a normal distribution $\mathcal{N}(0, \mathbf{V}\Sigma^{-2}\mathbf{V}^T)$.

Without loss of generality, we assume that $x_1^0, x_2^0, \dots, x_s^0$ are the nonzero elements of \mathbf{x}^0 and are all positive. Let $0 < x_0 \triangleq \min\{x_i^0, i \in \mathcal{T}\}$. Since λ decreases when N increases, there exists a number $N_1 \in \mathbb{N}$, such that $\lambda < \frac{x_0}{2}$ for all $N \geq N_1$.

In the following derivations, we use $v_{i,j}$ to denote the element in the i th row, j th column of \mathbf{V} and \bar{v}_i denotes the i th element of $\bar{\mathbf{v}}$. When $N > N_1$, we have

¹More precisely, the matrix sequence $\{\mathbf{A}_N\}_N$.

the following bound of $\mathbb{P}(\mathcal{I}^{\text{lp}}(N) \neq \mathcal{I})$:

$$\begin{aligned}
& \mathbb{P}(\mathcal{I}^{\text{lp}}(N) \neq \mathcal{I}) \\
&= \mathbb{P}(|x_1^0 + \bar{v}_1| < \lambda, \text{ or } |x_2^0 + \bar{v}_2| < \lambda, \dots, \text{ or } |x_s^0 + \bar{v}_s| < \lambda; \\
&\quad \text{or } |\bar{v}_{s+1}| > \lambda, \text{ or } |\bar{v}_{s+2}| > \lambda, \dots, \text{ or } |\bar{v}_n| > \lambda) \\
&\leq \sum_{i=1}^s \mathbb{P}(-\lambda - x_i^0 < \bar{v}_i < \lambda - x_i^0) + \sum_{i=s+1}^n \mathbb{P}(|\bar{v}_i| > \lambda) \\
&\stackrel{(I)}{\leq} \sum_{i=1}^s \frac{2\lambda e^{-(2\sum_{j=1}^n \sigma_j^{-2} v_{ij}^2)^{-1}(-x_i^0 + \lambda)^2}}{\sqrt{2\pi(\sum_{j=1}^n \sigma_j^{-2} v_{ij}^2)}} + \sum_{i=s+1}^n e^{-(2\sum_{j=1}^n \sigma_j^{-2} v_{ij}^2)^{-1}\lambda^2} \\
&\leq \sum_{i=1}^s \frac{2c_2\sqrt{N}\lambda}{\sqrt{2\pi}} e^{-\frac{1}{2}c_1^2 N(-x_i^0 + \lambda)^2} + \sum_{i=s+1}^n e^{-\frac{1}{2}c_1^2 N\lambda^2} \\
&\leq 2c_2s\sqrt{n}N^{\frac{\varepsilon}{2}} e^{-\frac{1}{8}(c_1x_0)^2N} + ne^{-c_1^2 nN^\varepsilon} \\
&= CN^{\frac{\varepsilon}{2}} e^{-\frac{1}{8}(c_1x_0)^2N} + ne^{-c_1^2 nN^\varepsilon}, \tag{2.9}
\end{aligned}$$

where $C = 2c_2s\sqrt{n}$. The inequality (I) in the chain holds due to the fact that the probability distribution function of \bar{v}_i is monotonically increasing in the interval $[-\lambda - x_i^0, \lambda - x_i^0]$ (since $\lambda < \frac{x_0}{2}$ when $N > N_1$), and together with the results in Lemma 2.3.

Then we can see that both terms in (2.9) will tend to 0 as $N \rightarrow \infty$ for any fixed $\varepsilon > 0$, i.e. $\lim_{N \rightarrow \infty} \mathbb{P}(\mathcal{I}^{\text{lp}}(N) = \mathcal{I}) = 1$. \square

Based on the previous result, we further have that

Theorem 2.2. *Given $0 < \varepsilon < 1$, and assume that the matrix \mathbf{A}^2 satisfies (2.4) with constants c_1, c_2 . Let $\lambda^2 = \frac{2n}{N^{1-\varepsilon}}$, then it holds that*

$$\mathbb{P}(\exists N', \text{ such that } \{\mathcal{I}^{\text{lp}}(N) = \mathcal{I}\} \text{ for all } N \geq N') = 1.$$

Proof. Recall that $0 < x_0 \triangleq \min\{x_i^0, i \in \mathcal{I}\}$. From the proof in the previous lemma, we have that when $N > N_1$

$$\begin{aligned}
& \mathbb{P}(\mathcal{I}^{\text{lp}}(N) \neq \mathcal{I}) \\
&\leq CN^{\frac{\varepsilon}{2}} e^{-\frac{1}{8}(c_1x_0)^2N} + ne^{-c_1^2 nN^\varepsilon} \\
&= Ce^{-\frac{1}{8}(c_1x_0)^2N + \frac{\varepsilon}{2}\ln(N)} + e^{\ln(n) - c_1^2 nN^\varepsilon} \\
&= Ce^{(c_1x_0)^2N(\frac{\varepsilon\ln(N)}{2(c_1x_0)^2N} - \frac{1}{8})} + e^{c_1^2 nN^\varepsilon(\frac{\ln(n)}{c_1^2 nN^\varepsilon} - 1)}.
\end{aligned}$$

²More precisely, the matrix sequence $\{\mathbf{A}_N\}_N$.

Since $0 < \varepsilon < 1$ and $x_0 > 0$, we have that $\frac{\varepsilon \ln(N)}{2(c_1 x_0)^{2N}}$ and $\frac{\ln(n)}{c_1^2 n N^\varepsilon}$ will tend to zero if $N \rightarrow \infty$. Hence there exists a number $N_2 \in \mathbb{N}$ such that for all $N > N_3 \triangleq \max(N_1, N_2)$ one has that $\frac{\varepsilon \ln(N)}{2(c_1 x_0)^{2N}} < \frac{1}{16}$ and $\frac{\ln(n)}{c_1^2 n N^\varepsilon} < \frac{1}{2}$. Therefor we have that

$$\begin{aligned}
& \sum_{N=N_3}^{\infty} \mathbb{P}(\mathcal{I}^p(N) \neq \mathcal{I}) \\
& \leq \sum_{N=N_3}^{\infty} C e^{-\frac{1}{16}(c_1 x_0)^{2N}} + \sum_{N=N_3}^{\infty} e^{-\frac{1}{2}c_1^2 n N^\varepsilon} \\
& \leq \int_{N_3-1}^{\infty} C e^{-\frac{1}{16}(c_1 x_0)^{2t}} dt + \int_{N_3-1}^{\infty} e^{-\frac{1}{2}c_1^2 n t^\varepsilon} dt \\
& = A + B.
\end{aligned}$$

Furthermore, it can be seen that

$$A = \int_{N_3-1}^{\infty} C e^{-\frac{1}{16}(c_1 x_0)^{2t}} dt < \infty.$$

In the following, we will show that

$$B = \int_{N_3-1}^{\infty} e^{-\frac{1}{2}c_1^2 n t^\varepsilon} dt < \infty.$$

By change of variables using $x = \frac{1}{2}c_1^2 n t^\varepsilon$, we have that

$$B = \frac{1}{c_1^2 n \varepsilon} \int_{\frac{1}{2}c_1^2 n (N_3-1)^\varepsilon}^{\infty} x^{\frac{1}{\varepsilon}-1} e^{-x} dx < \frac{1}{c_1^2 n \varepsilon} \Gamma\left(\frac{1}{\varepsilon}\right) < \infty$$

where Γ is the Gamma function. Hence

$$\sum_{N=N_3}^{\infty} \mathbb{P}(\mathcal{I}^p(N) \neq \mathcal{I}) < \infty.$$

Application of the Borel-Cantelli lemma [4] implies that the events in $\{\mathcal{I}^p(N) \neq \mathcal{I}\}_{N=N_3}^{\infty}$ will not happen infinitely often, which concludes the result. \square

2.4 Illustrative experiments

This section supports the findings in the previous section with numerical examples and make the comparisons with some other algorithms which also possess the ORACLE property.

2.4.1 Experiment 1

This example is taken from [109]. The setups are repeated as follows.

- \mathbf{x}^0 is set to be $(3, 1.5, 0, 0, 2, 0, 0, 0)^T$;
- Rows of matrix A are i.i.d. normal vectors;
- The correlation between the j_1 -th and the j_2 -th elements of each row are given as $0.5^{|j_1-j_2|}$;
- The noise term $\mathbf{v} \in \mathbb{R}^N$ follows the distribution $\mathcal{N}(0, \mathbf{I}_N)$.

Based on these setups, the proposed method and also the methods presented in [81] (the A-SPARSEVA-AIC-RE method and the A-SPARSEVA-BIC-RE methods) and [109] (the ADALASSO method) are applied to recover \mathbf{x}^0 . In this experiment, ε for the proposed method is set as $\frac{1}{3}$; λ_N for 'ADALASSO' is chosen as $N^{1/2-\gamma/4}$ (this choice satisfies all the assumptions in Theorem 2 in [109]), and γ is set to be 1; the thresholding value (for detecting zero components from the solution of the Lasso problem) for the 'A-SPARSEVA-AIC-RE' and 'A-SPARSEVA-BIC-RE' are set to be 10^{-5} as suggested in [81]. For the comparison, we also include the experiment result obtained by using the LASSO method, in which we set the tuning parameter as \sqrt{N} . In Figure 2.2, for every N , the experiment is repeated 50 times (with independently generated noise terms) to get the estimated MSE. The following abbreviations are used in Figure 2.2.

1. The curve with tag 'LSE' gives the MSE of the estimates by the LSE algorithm;
2. The curve with tag 'LP + RE-LSE' gives the MSE of the estimates given by the proposed algorithm;
3. The curve with tag 'ORACLE-LSE' gives the MSE of the estimates by the ORACLE LSE;
4. The curves with tags 'A-SPARSEVA-AIC-RE' and 'A-SPARSEVA-BIC-RE' give the MSE of the estimates by the methods presented in [81];
5. The curve with tag 'ADALASSO' gives the MSE of the estimates by the ADALASSO method presented in [109];
6. The curve with tag 'LASSO' gives the MSE of the estimates of the LASSO method.

Note that, when N becomes large, the curves 'LP + RE-LSE' and 'ORACLE-LSE' exactly match each other.

Figure 2.3 demonstrates the efficacy of support recovery of the LP formulation in (2.6) for different choices of ε . In the plot, 'portion' is defined as the ratio of successful trials over the total number of trials. We conclude the empirical observations for this experiment in the caption of the figure.

In the following, we compare the performances of the ADALASSO and the proposed method when the tuning parameters are selected using the 5-fold cross-validation (see [42]). The ADALASSO algorithm and the proposed algorithm have one tuning parameter each, namely γ for the ADALASSO, and ε for the proposed method. ε and γ are selected from $\{1/8, 1/4, 1/2\}$ and $\{1/2, 1, 2\}$ separately. For each N , we run 100 independent realizations. In each realization, we record the value $\|\hat{\mathbf{x}} - \mathbf{x}^0\|_2^2$, where $\hat{\mathbf{x}}$ denotes the estimate

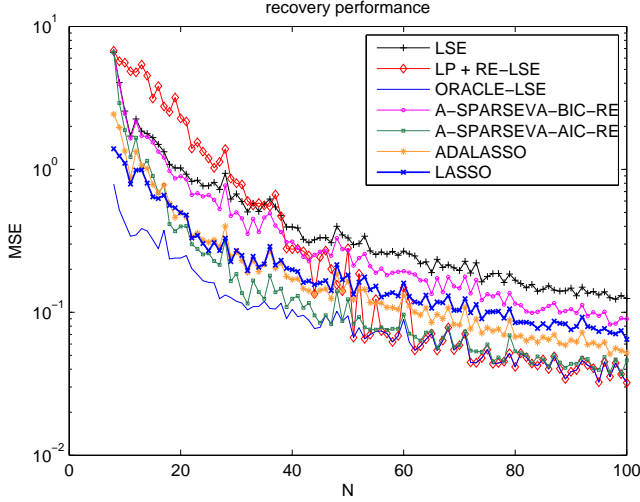


Figure 2.2: Performance of the different estimators from N observations to estimate \mathbf{x}^0 . This picture indicates that the proposed estimator will give exactly the same performance as the ORACLE estimator for a large N ($N \approx 75$).

obtained by the estimator. The results are reported in Figure 2.4 and see also the discussions in the caption of the figure.

2.4.2 Experiment 2

In this part, we perform an experiment for recovering the sinusoids from noisy measurements. The data is generated as follows:

$$y(t) = \sum_{k'=1}^{n'} c_{i_{k'}} \sin(w_{i_{k'}} t) + v(t).$$

Here both $\{w_{i_{k'}}\}_{k'}$ and $\{c_{i_{k'}}\}_{k'}$ are unknown, but we know that the frequencies do belong to a (larger, but of constant size) set $\{w_k\}_{k=1}^n$ of n elements. By sampling the system with period t_s , we obtain the system

$$\mathbf{y} = \mathbf{A}\mathbf{c}^0 + \mathbf{v}, \quad (2.10)$$

where $\mathbf{y} = [y(t_s), \dots, y(Nt_s)]^T$. The matrix $\mathbf{A} \in \mathbb{R}^{N \times n}$ is defined as follows. The i -th row of \mathbf{A} is given by

$$\mathbf{A}_i = [\sin(iw_1 t_s), \sin(iw_2 t_s), \dots, \sin(iw_n t_s)]. \quad (2.11)$$

The parameter vector and noise vector are defined as $\mathbf{c}^0 = [c_1, c_2, \dots, c_n]^T$ and $\mathbf{v} = [v(t_s), v(2t_s), \dots, v(Nt_s)]^T$.

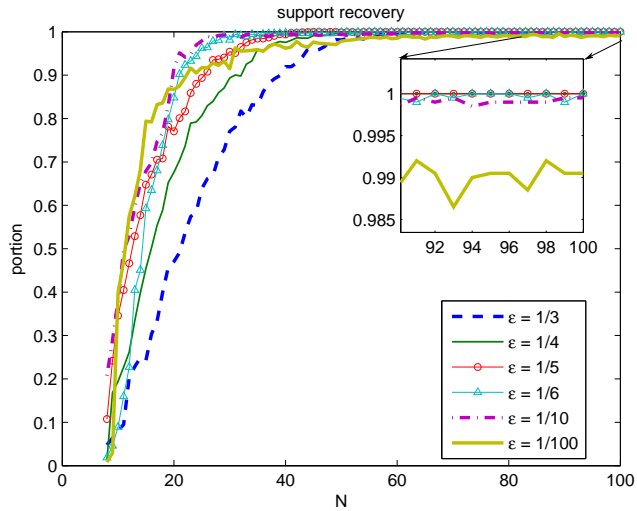
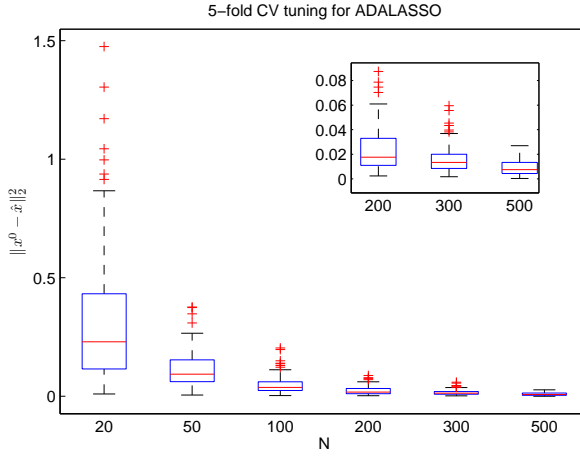
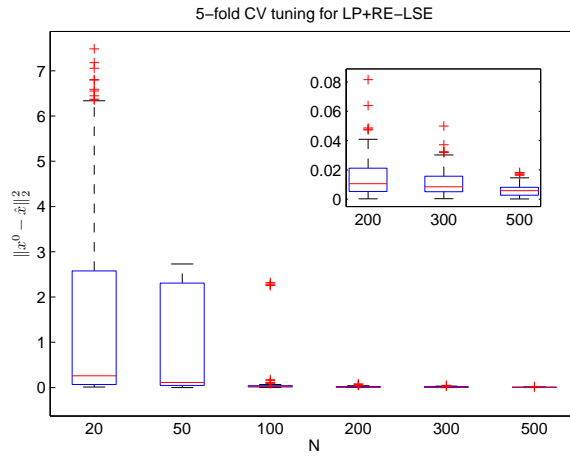


Figure 2.3: Support recovery performance of (2.6) for different choices of ϵ . Empirically, we observe that: 1) When ϵ is chosen to be small, the ratio for successful support recovery will be larger when N is small; but when N is large, the ratio for successful support recovery will converge slower to 100% and oscillation exists. This can be observed in the zoomed-in part. 2) When ϵ is chosen to be large, the ratio for successful support recovery will be smaller when N is small; but when N is large, the ratio for successful support recovery will go faster to 100% and no oscillation exists, see also the zoomed-in part in the figure.



(a)



(b)

Figure 2.4: This figure demonstrates the boxplots of the recovery error obtained by the ADALASSO estimator and the proposed estimator when the tuning parameters are chosen by the 5-fold cross validation method. From this figure, we can see that performances of both methods are similar when N is large, see the zoomed part in the figures. It can also be observed that when N is small, the ADALASSO method has smaller recovery error compared with the proposed method.

In this experiment, $n = 10$ and $\mathbf{c}^0 = (1, 1, 1, 0, \dots, 0)^T$, $w_k = k$ for $k = 1, 2, \dots, n$. We increase N up to 500 and the noise vector \mathbf{v} satisfies $\mathbf{v} \sim \mathcal{N}(0, \mathbf{I}_N)$. We also assume that only the first three entries in $\{w_k\}_{k=1}^n$ occur effectively in the system of (2.10) and the corresponding amplitudes are set to 1, i.e. $n' = 3$ and $i_1 = 1, i_2 = 2, i_3 = 3$. The sampling period t_s is set to $0.1s$.

The result using the proposed algorithm to recover \mathbf{x}^0 is displayed in Figure 2.5. It is again clear that the proposed estimator is as efficient as the ORACLE estimator if there are enough samples.

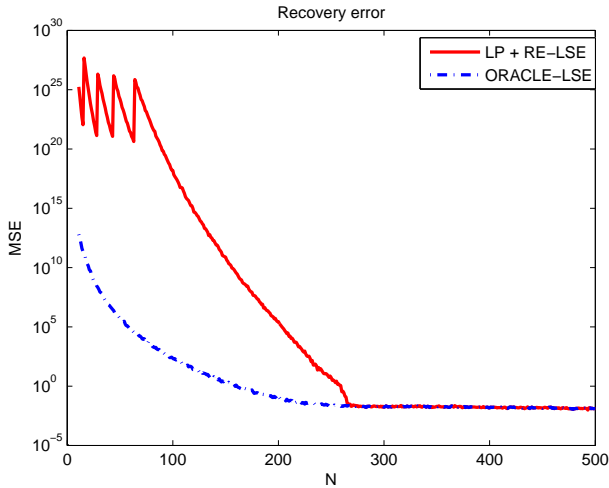


Figure 2.5: Performance of applying the proposed estimator to recover sinusoidal functions from N observations in Experiment 2. This example also indicates that after a finite number of observations, the estimate is exactly equal to the ORACLE estimator.

This is indeed predicted by the theory above since the \mathbf{A} in (2.10) obeys the assumption of (2.4). This follows from the proposition given as:

Proposition 2.1. *There exist constants $\{C_{i,j}\}_{0 \leq i,j \leq n}$ which do not depend on N , such that the following results hold. For any $1 \leq i \neq j \leq n$, it holds that:*

$$\left| (\mathbf{A}^T \mathbf{A})_{i,j} \right| = \left| \sum_{t=1}^N \sin(tw_it_s) \sin(tw_jt_s) \right| \leq C_{i,j} \quad (2.12)$$

and for any $1 \leq i \leq n$ that:

$$(\mathbf{A}^T \mathbf{A})_{i,i} = \sum_{t=1}^N (\sin(tw_it_s))^2 \geq \frac{N}{2} - C_{i,i}. \quad (2.13)$$

The proof is given in the Appendix. With this proposition, an application of the Geršgorin circle theorem implies that the eigenvalues of $\mathbf{A}^T \mathbf{A}$ will increase with the order of N , which in turn implies that the matrix \mathbf{A} in (2.10) satisfying the property in (2.4).

2.5 Conclusion

This chapter presents an algorithm for solving an over-determined linear system from noisy observations, specializing to the case where the true 'parameter' vector is sparse. The proposed method does not require an optimization problem to be solved. Furthermore, it is established that the proposed method possesses the ORACLE property. More discussions, including the limitations and future work about the method, can be found in Chapter 7.

2.6 Appendix

Proof. The proof of (2.12) goes as follows. First we note that

$$\begin{aligned} & \left| \sum_{t=1}^N \sin(tw_i t_s) \sin(tw_j t_s) \right| \\ &= \frac{1}{2} \left| \sum_{t=1}^N (\cos(t(w_i - w_j)t_s) - \cos(t(w_i + w_j)t_s)) \right| \\ &\leq \frac{1}{2} \left| \sum_{t=1}^N \cos(t(w_i - w_j)t_s) \right| + \frac{1}{2} \left| \sum_{t=1}^N \cos(t(w_i + w_j)t_s) \right|. \end{aligned}$$

We focus on bounding the term $\left| \sum_{t=1}^N \cos(t(w_i - w_j)t_s) \right|$, the bound of the other term will follow along the same lines.

$$\begin{aligned} \left| \sum_{t=1}^N \cos(t(w_i - w_j)t_s) \right| &= \left| \operatorname{Re} \left(\frac{1 - e^{j(N+1)(w_i - w_j)t_s}}{1 - e^{j(w_i - w_j)t_s}} \right) - 1 \right| \\ &\leq \left| \frac{1 - e^{j(N+1)(w_i - w_j)t_s}}{1 - e^{j(w_i - w_j)t_s}} \right| + 1 \\ &\leq \frac{2}{\left| 1 - e^{j(w_i - w_j)t_s} \right|} + 1, \end{aligned}$$

which is a constant which does not depend on N , so the inequality (2.12) is obtained.

In order to prove the inequality (2.13), we observe that

$$\sum_{t=1}^N (\sin(tw_it_s))^2 = \frac{1}{2} \sum_{t=1}^N (1 - \cos(2tw_it_s)) \geq \frac{N}{2} - \frac{1}{2} \left| \sum_{t=1}^N \cos(2tw_it_s) \right|.$$

Using the previous bounding method, $\frac{1}{2} \left| \sum_{t=1}^N \cos(2tw_it_s) \right|$ is also bounded by a constant $C_{i,i}$ which does not depend on N . This concludes the proof. \square

Chapter 3

Hankel matrix completion with convexification

3.1 Problem introduction

Convexification using the nuclear norm heuristic is becoming increasingly popular, as illustrated by the examples in chapter 1. Apart from them, more examples can be found in e.g. [101], [66]. This chapter studies the performance of the nuclear norm heuristic for a Hankel matrix completion problem [11, 37, 80], which relates to the recovery of the impulse response of a stable linear system. To be precise, we make the following assumptions throughout the chapter: (1) the first n entries of the impulse response are provided while the last $n - 1$ entries are to be completed, (2) the provided entries are exact, i.e. there is no noise present.

Analogously to what has been done in those examples in chapter 1, the related Hankel matrix can be constructed from the impulse response sequence. Then given by the previous assumptions, the entries in the upper triangle part¹ of the related Hankel matrix are known, and the entries in the lower triangle part are to be completed, using the nuclear norm minimization heuristic. Note that in this case, the revealed entries of the matrix are given deterministically and the underlying matrix is of Hankel structure. These characters make the theories in previous work [11, 37, 80] not directly applicable to analyze the problem considered here.

This chapter tries to provide some insights for understanding how the nuclear norm heuristic performs on this specific problem. Note that although the considered problem is simplified, it still captures many aspects of the more general cases, see for example the problems discussed in chapter 1.

The following notational conventions will be used for this chapter. Vectors are denoted in boldface, scalars are denoted in lowercase, matrices as capital letters, and sets are represented as calligraphic letters. \mathcal{H}_n denotes the set of $n \times n$ Hankel matrices, I_n denotes the identity matrix of size $n \times n$, \mathbf{e}_i denotes

¹In this chapter, the upper triangle part of a square matrix $A \in \mathbb{R}^{n \times n}$ means the set of entries $a_{i,j}$ such that $i + j \leq n + 1$, $a_{i,j}$ denotes the (i, j) -th element of A .

the unit vector with only the i -th element to be one and all the other elements zero, $\|\cdot\|_*$ represents the nuclear norm (sum of all the singular values) of a matrix, $\|\cdot\|_2$ represents the spectral norm of a matrix, and $\|\cdot\|_F$ represents the Frobenius norm of a matrix.

3.2 Analysis and the main result

For the problem introduced in the previous section, the following result establishes that when the underlying Hankel matrix is of rank one (the related stable linear system has one single pole), the nuclear norm heuristic can always succeed in completing the related Hankel matrix, i.e. recovering the future impulse responses.

This section focuses on the analysis for the rank one case, discussions for more general case will be given in next section (including some numerical illustrations).

Theorem 3.1. *Let $-1 < h < 1$ and define $\mathbf{h} \in \mathbb{R}^n$ as $\mathbf{h} = [1, h, h^2, \dots, h^{n-1}]^T$, and a matrix $G_0 \in \mathcal{H}_n$ as $\mathbf{h}\mathbf{h}^T$. Consider the following application of the nuclear norm heuristic:*

$$\begin{aligned} \hat{G}_0 &\triangleq \arg \min_{G \in \mathcal{H}_n} \|G\|_* \\ \text{s.t. } &G(i, j) = G_0(i, j), \forall (i + j) \leq n + 1, \end{aligned} \quad (3.1)$$

then it holds that \hat{G}_0 is unique and $\hat{G}_0 = G_0$.

3.2.1 Proof of Theorem 3.1

Based on the matrices G_0 and G in Theorem 3.1, define:

$$H = G_0 - G, \quad (3.2)$$

Notice that by construction, all the entries of H in the upper triangle part are zero, which means that H can be decomposed as

$$H = \sum_{i=1}^{n-1} v_i G_i, \quad (3.3)$$

where $\{G_i\}_{i=1}^{n-1}$ are the basis matrices with the entries in the i -th lower anti-diagonal equal to 1 and the others equal to zero and $v_i \in \mathbb{R}, \forall i = 1, \dots, n-1$. Equivalently stated, the (m, k) -th element of G_i equals one if $m + k = n + 1 + i$, zero otherwise.

Define the projection matrix

$$P = \frac{G_0}{\|\mathbf{h}\|_2^2}$$

and its complement projection matrix as $Q = I_n - P$.

The following proposition 3.1 will be used later, which characterizes the nuclear norm as the dual norm of the spectral norm [80].

Proposition 3.1. *Given matrix $A \in \mathbb{R}^{n \times n}$, we have that*

$$\|A\|_* = \sup\{\text{tr}(MA) : \|M\|_2 \leq 1, M \in \mathbb{R}^{n \times n}\}. \quad (3.4)$$

The following result will be needed in the Lemma 3.3.

Proposition 3.2. *Given H as defined in (3.2), if $H \neq 0$, then $QHQ \neq 0$.*

Proof. We prove that the only possibility for $QHQ = 0$ to hold is when $H = 0$. Notice that $H = (P + Q)H(P + Q)$, expanding this equality, we have that

$$H = PHP + PHQ + QHP + QHQ.$$

Hence, if $QHQ = 0$, we have that

$$\begin{aligned} H &= PHP + PHQ + QHP \\ &= PH + QHP. \end{aligned}$$

Since $P = \frac{\mathbf{h}\mathbf{h}^T}{\|\mathbf{h}\|_2^2}$, the previous relation implies that H can be represented as $\mathbf{h}\mathbf{a}^T + \mathbf{b}\mathbf{h}^T$ where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$. Since H is symmetric, it holds that

$$\mathbf{h}\mathbf{a}^T + \mathbf{b}\mathbf{h}^T = \mathbf{a}\mathbf{h}^T + \mathbf{h}\mathbf{b}^T,$$

or equivalently

$$\mathbf{h}(\mathbf{b} - \mathbf{a})^T = (\mathbf{b} - \mathbf{a})\mathbf{h}^T. \quad (3.5)$$

Given the fact in (3.5), the two rank-one matrices $\mathbf{h}(\mathbf{b} - \mathbf{a})^T$ and $(\mathbf{b} - \mathbf{a})\mathbf{h}^T$ will have the same row space and column space, which implies that $\mathbf{b} - \mathbf{a} = k\mathbf{h}$ for some k ².

This implies that H can be written as

$$H = \mathbf{h}\mathbf{a}^T + \mathbf{b}\mathbf{h}^T = \mathbf{h}\mathbf{a}^T + \mathbf{a}\mathbf{h}^T + k\mathbf{h}\mathbf{h}^T,$$

i.e.

$$H = (\mathbf{a} + \frac{k}{2}\mathbf{h})\mathbf{h}^T + \mathbf{h}(\mathbf{a} + \frac{k}{2}\mathbf{h})^T.$$

Let $\mathbf{c} = (c_1, c_2, \dots, c_n)^T = \mathbf{a} + \frac{k}{2}\mathbf{h}$. Notice that the i -th element of the first column of H equals $h^{i-1}c_1 + c_i$. By construction, the first column of H is a zero vector. Hence, for $i = 1$, it holds that $2c_1 = 0$, i.e. $c_1 = 0$. Thus the i -th element of the first column of H equals c_i , which implies that $c_2 = \dots = c_n = 0$, i.e. $\mathbf{c} = 0$. This concludes that $H = 0$. \square

²Multiplying both sides of the previous equation with \mathbf{h}^T , we can get $k = \frac{(\mathbf{b} - \mathbf{a})^T \mathbf{h}}{\|\mathbf{h}\|_2^2}$.

The following Lemma 3.1 provides a sufficient condition for Theorem 3.1 to hold.

Lemma 3.1. *If for any $H \neq 0$ as defined in (3.2), we have that*

$$|\text{tr}(PH)| < \|QHQ\|_*, \quad (3.6)$$

then the optimization problem (3.1) recovers G_0 exactly.

Proof. Let $V \in \mathbb{R}^{n \times (n-1)}$ be a matrix which satisfies $VV^T = Q$ and $V^TV = I_{n-1}$. For a matrix $M \in \mathbb{R}^{n \times n}$ with $\|M\|_2 \leq 1$, if the row space of M is orthogonal to the row space of P , and the column space of M is orthogonal to the column space of P , then M can be written as VBV^T , where $B \in \mathbb{R}^{(n-1) \times (n-1)}$ and $\|B\|_2 \leq 1$. Given this fact, the sub-gradient of $\|\cdot\|_*$ at G_0 is given as the set (see e.g. [80]):

$$\mathcal{S}_h = \{P + VBV^T : \|B\|_2 \leq 1\}. \quad (3.7)$$

By the property of sub-gradient, it holds true that for any H as in (3.2),

$$\|G_0 + H\|_* \geq \|G_0\|_* + \langle H, F \rangle,$$

where $F \in \mathbb{R}^{n \times n}$ is any matrix which belongs to \mathcal{S}_h .

Hence, for any H , if there exists a matrix B with $\|B\|_2 \leq 1$, such that

$$\langle H, P + VBV^T \rangle > 0$$

or equivalently

$$\text{tr}(HP) > \langle V^T HV, -B \rangle, \quad (3.8)$$

then $\|G_0 + H\|_* > \|G_0\|_*$ holds, which implies the result in Theorem 3.1.

We are left to find a matrix B with $\|B\|_2 \leq 1$ which satisfies inequality (3.8). Notice that $Q = VV^T$ is a projection matrix onto a $n - 1$ dimensional subspace and $V^TV = I_{n-1}$, then we have that

$$\|QHQ^T\|_* = \|VV^T H V V^T\|_* \stackrel{(I)}{=} \|V^T H V\|_*,$$

in which the equality (I) holds is because of the definition of nuclear norm. Together with the fact in (3.6), i.e.

$$|\text{tr}(HP)| < \|QHQ^T\|_*,$$

we have that

$$|\text{tr}(HP)| < \|V^T H V\|_*.$$

Furthermore, it follows from Proposition 3.1 that there exists a matrix B_1 with $\|B_1\|_2 \leq 1$, such that

$$\|V^T H V\|_* = \langle V^T H V, B_1 \rangle.$$

Therefore it holds that

$$|\text{tr}(HP)| < \langle V^T HV, B_1 \rangle.$$

Hence

$$\text{tr}(HP) > -\langle V^T HV, B_1 \rangle = \langle V^T HV, -B_1 \rangle$$

holds, which gives that the inequality (3.8) holds for B_1 . This concludes the proof. \square

Next, we prove that the condition (3.6) in Lemma 3.1 will always hold whenever $H \neq 0$. The following Lemma 3.2 constructs a matrix M_0 which will be used in Lemma 3.3 to construct the 'certificate' M_1 .

Lemma 3.2. *Given the matrices $G_i, P, Q \in \mathbb{R}^{n \times n}$ defined as before, then there exists a matrix $M_0 \in \mathbb{R}^{n \times n}$ with $\|M_0\|_2 < 1$, such that*

$$\text{tr}(QG_iQM_0 - G_iP) = 0, \quad \forall i = 1, 2, \dots, n-1. \quad (3.9)$$

Proof. We will give a construction of such a matrix M_0 . Let $r > 0$ denote the norm of vector \mathbf{h} , which clearly satisfies

$$r^2 = \|\mathbf{h}\|_2^2 = 1 + h^2 + \dots + h^{2n-2}.$$

We construct two matrices $Q_1 \in \mathbb{R}^{n \times n}$ and $Q_2 \in \mathbb{R}^{n \times n}$ which satisfy the following two equations:

$$r^2(Q_1 + Q_2) = r^2Q = r^2I_n - G_0, \quad (3.10)$$

and $r^2(Q_1 - Q_2)$ is given by

$$\begin{bmatrix} -h^n & -h^{n+1} & -h^{n+2} & \dots & -h^{2n-2} & s \\ -h^{n+1} & -h^{n+2} & \vdots & -h^{2n-2} & s & -1 \\ -h^{n+2} & \vdots & -h^{2n-2} & s & -1 & -h \\ \vdots & -h^{2n-2} & s & -1 & \vdots & \vdots \\ -h^{2n-2} & s & -1 & \vdots & \vdots & -h^{n-3} \\ s & -1 & -h & \dots & -h^{n-3} & -h^{n-2} \end{bmatrix}, \quad (3.11)$$

where $s = h + h^3 + \dots + h^{2n-3}$.

The matrices Q_1 and Q_2 satisfy the following properties (proofs are given in the Appendix):

- Fact 1:

$$(Q_1 + Q_2)(Q_1 + Q_2) = (Q_1 - Q_2)(Q_1 - Q_2). \quad (3.12)$$

- Fact 2:

$$(Q_1 + Q_2)(Q_1 - Q_2) = (Q_1 - Q_2)(Q_1 + Q_2). \quad (3.13)$$

• Fact 3:

$$Q_1 Q_2 = Q_2 Q_1 = 0. \quad (3.14)$$

• Fact 4:

$$Q_1^2 = Q_1, \quad Q_2^2 = Q_2. \quad (3.15)$$

These facts give that

$$(Q_1 - Q_2)^2 = Q_1 + Q_2 = Q. \quad (3.16)$$

Since Q is a projection matrix, it gives that the spectral norm of $(Q_1 - Q_2)$ is 1. These facts also lead us to consider the following choice of M_0

$$M_0 = -h^n(Q_1 - Q_2). \quad (3.17)$$

Now we can prove that matrix M_0 satisfies all the equalities given in (3.9) based on these facts. First, notice that the equalities in (3.9) are equivalent to the following equalities

$$\text{tr}(G_i(QM_0Q - P)) = 0, \quad \forall i = 1, 2, \dots, n-1. \quad (3.18)$$

The term $QM_0Q - P$ in (3.18) can be calculated as follows:

$$\begin{aligned} & QM_0Q - P \\ &= -h^n(Q_1 + Q_2)(Q_1 - Q_2)(Q_1 + Q_2) - P \\ &= -h^n(Q_1 - Q_2)(Q_1 + Q_2) - P \\ &= -h^n(Q_1 - Q_2) - P \\ &= M_0 - P. \end{aligned} \quad (3.19)$$

Hence, proving the equalities in (3.18) boils down to prove that

$$\text{tr}(G_i(M_0 - P)) = 0, \quad \forall i = 1, 2, \dots, n-1. \quad (3.20)$$

Notice that M_0 has the same elements as P in the lower triangle part, implying that (3.20) holds, which in turn implies (3.9).

Finally, notice that $\|M_0\|_2 = |h|^n$, which is less than 1. This gives that M_0 is the desired matrix and concludes the proof. \square

Based on the constructed M_0 in Lemma 3.2, we can certify that:

Lemma 3.3. *For any $H \neq 0$, we have that*

$$|\text{tr}(PH)| < \|QHQ\|_*. \quad (3.21)$$

Proof. We distinguish between two cases, namely

$$\text{tr}(PH) < \|QHQ\|_*, \quad (3.22)$$

and

$$-\text{tr}(PH) < \|QHQ\|_*. \quad (3.23)$$

We will give a derivation of (3.22), the proof of (3.23) follows along the same lines. With the application of Proposition 3.1, it follows that proving (3.22) is equivalent to proving that

$$\sup_{\|M\|_2 \leq 1} \text{tr}(QHQM - HP) > 0. \quad (3.24)$$

Notice that $H = \sum_{i=1}^{n-1} v_i G_i$, and that by construction of M_0 in Lemma 3.2, we have that

$$\text{tr}(QHQM_0 - HP) = \sum_{i=1}^{n-1} v_i \text{tr}(QG_iQM_0 - G_iP) = 0.$$

Next, observe that M_0 is strictly inside the ball

$$\mathcal{B}_M = \{M : \|M\|_2 \leq 1, M \in \mathbb{R}^{n \times n}\},$$

hence there exists a small value $\delta > 0$ such that

$$M_1 = M_0 + \delta(QHQ), \quad (3.25)$$

will also be inside \mathcal{B}_M . Since $H \neq 0$ and it follows that $QHQ \neq 0$ given by Proposition 3.2, which in turn implies that

$$\text{tr}(QHQM_1 - HP) = \delta \text{tr}(QHQQHQ) = \delta \|QHQ\|_F^2$$

is positive. This certifies (3.24) and hence (3.22), which in turn concludes the proof of Lemma 3.3. \square

In conclusion, application of Lemmas 3.1, 3.2 and 3.3 gives Theorem 3.1.

3.3 Discussion and numerical illustration

The previous section studies the rank one Hankel matrix completion problem where the revealed entries follow a deterministic pattern. It is natural to raise the question whether the nuclear norm heuristic will still work when the rank of the Hankel matrix is larger than 1. Consider a second order system with the impulse response given by $\{h_1^{i-1} + h_2^{i-1}\}_{i=1}^{\infty}$, where $h_1, h_2 \in \mathbb{R}$ and $-1 < h_1, h_2 < 1$ represent the two poles of the system. Let $n = 10$, then the matrix completion problem based on the nuclear norm heuristic is given as

$$\hat{G} = \arg \min_{G \in \mathcal{H}_{10}} \|G\|_* \quad (3.26)$$

$$\text{s.t. } G(i, j) = G_0(i, j), \forall i + j \leq 11.$$

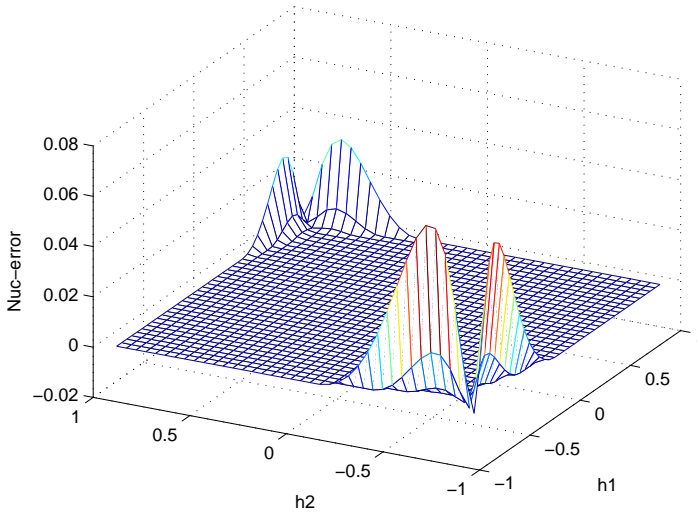


Figure 3.1: This figure displays $\|G_0\|_* - \|\hat{G}\|_*$ for a range of 2 real poles. It is seen that the nuclear norm objective value is not always minimal for the true system G_0 for many choices of h_1 and h_2 , implying that the heuristic will not always work for such systems. Note that the difference equals zero for the case where $h_1 = h_2$ as confirmed by Theorem 3.1.

Figure 3.1 displays $\|G_0\|_* - \|\hat{G}\|_*$ for different choices of h_1 and h_2 , which are chosen from $h_1 = -0.94 : 0.05 : 0.94$ and $h_2 = -0.94 : 0.05 : 0.94$. From this experiment, it becomes clear that G_0 does not always has minimal nuclear norm, and recovery by (3.26) will not necessarily succeed. It is worthwhile to mention that, in most cases (see the results reported in Figure 3.1), the nuclear norm heuristic gives the correct completion.

In Theorem 3.1, it is established that whenever $|h| < 1$, the nuclear norm heuristic will be able to complete the Hankel matrix. However, the numerically recovered matrix may not be exactly equal to G_0 , since G_0 is rank deficient which causes additional difficulties for the numerical optimization. See for example the illustrations in Figure 3.2 and 3.3.

3.4 Conclusion

In this chapter, we have examined an application of the nuclear norm heuristic for the completion of a low rank Hankel matrix. We show that, when the underlying Hankel matrix is of rank one, the nuclear norm heuristic can be successful in the matrix completion task. The proof relies on building a certificate which captures the structural properties of the problem. While in

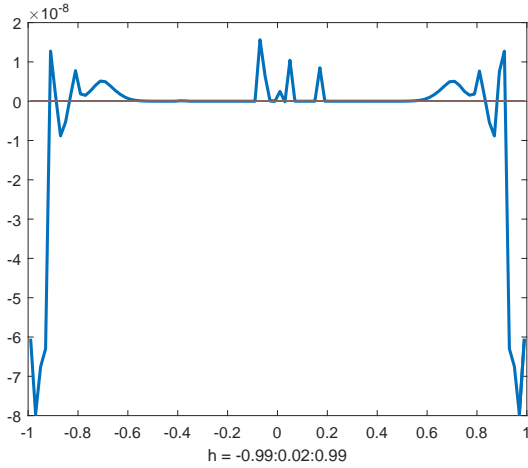


Figure 3.2: This figure displays the value of $\|G_0\|_* - \|\hat{G}\|_*$ when $h = -0.99 : 0.02 : 0.99$. When h gets closer to zero and the boundaries, the numerical error gets larger.

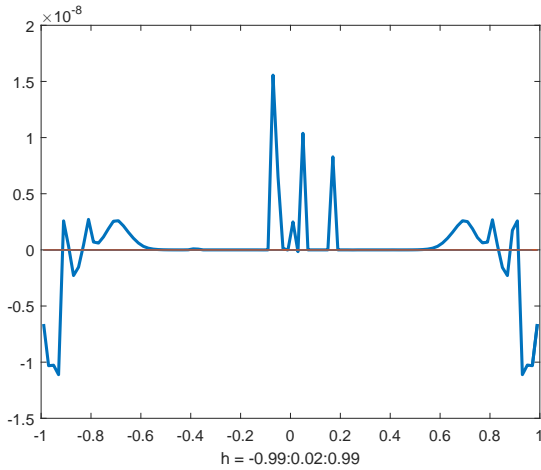


Figure 3.3: This figure displays the value of $\frac{\|G_0\|_* - \|\hat{G}\|_*}{\|G_0\|_*}$ when $h = -0.99 : 0.02 : 0.99$. When h gets closer to zero and the boundaries, the numerical error gets larger.

a slightly more complicated case, we numerically illustrate that the heuristic will not always work.

3.5 Appendix

Define $\Delta = r^2(Q_1 - Q_2)$, and its k th column as $\Delta_k = \delta_k + hr^2\mathbf{e}_{n-k+1}$, in which

$$\delta_k = \left[-h^{n+k-1}, \dots, -h^{2n-1}, -1, \dots, -h^{k-2} \right]^T.$$

3.5.1 Proof of Fact 1

Notice that proving (3.12) is equivalent to proving

$$r^2(Q_1 + Q_2)r^2(Q_1 + Q_2) = \Delta^2.$$

By definition, we have that

$$r^2(Q_1 + Q_2)r^2(Q_1 + Q_2) = (r^2I_n - G_0)^2 = r^4I_n - r^2(\mathbf{h}\mathbf{h}^T).$$

Hence, it remains to prove that

$$\Delta^2 = r^4I_n - r^2(\mathbf{h}\mathbf{h}^T).$$

We will compute the off-diagonal entries and the on-diagonal entries of Δ^2 separately.

- The off-diagonal elements.

Take for any $1 \leq k_1, k_2 \leq n$ the corresponding columns from Δ as Δ_{k_1} and Δ_{k_2} (assume that $k_1 < k_2$ without loss of generality). The (k_1, k_2) and (k_2, k_1) elements of Δ^2 are given by

$$\begin{aligned} \Delta_{k_1}^T \Delta_{k_2} &= (\delta_{k_1} + hr^2\mathbf{e}_{n-k_1+1})^T (\delta_{k_2} + hr^2\mathbf{e}_{n-k_2+1}) \\ &= \delta_{k_1}^T \delta_{k_2} + hr^2 (\delta_{k_1}^T \mathbf{e}_{n-k_2+1} + \mathbf{e}_{n-k_1+1}^T \delta_{k_2}). \end{aligned}$$

It holds that $\delta_{k_1}^T \delta_{k_2}$ is given as

$$\begin{aligned} &\left[-h^{n+k_1-1}, -h^{n+k_1}, \dots, -h^{2n+k_1-k_2-1} \right] \left[-h^{n+k_2-1}, -h^{n+k_2}, \dots, -h^{2n-1} \right]^T \\ &+ \left[-h^{2n+k_1-k_2}, -h^{2n+k_1-k_2+1}, \dots, -h^{2n-1} \right] \left[-1, -h, \dots, -h^{k_2-k_1-1} \right]^T \\ &+ \left[-1, -h, \dots, -h^{k_1-2} \right] \left[-h^{k_2-k_1}, -h^{k_2-k_1+1}, \dots, -h^{k_2-2} \right]^T, \end{aligned}$$

and also

$$\delta_{k_1}^T \mathbf{e}_{n-k_2+1} + \mathbf{e}_{n-k_1+1}^T \delta_{k_2} = -h^{2n+k_1-k_2-1} - h^{k_2-k_1-1}.$$

Hence we have that

$$\Delta_{k_1}^T \Delta_{k_2} = h^{2n+k_1+k_2-2} + h^{2n+k_1+k_2} + \dots + h^{4n+k_1-k_2-2} \quad (3.27)$$

$$+ h^{2n+k_1-k_2} + h^{2n+k_1-k_2+2} + \dots + h^{2n+k_2-k_1-2} \quad (3.28)$$

$$+ h^{k_2-k_1} + h^{k_2-k_1} + \dots + h^{k_1+k_2-4} \quad (3.29)$$

$$- h^{2n+k_1-k_2} (1 + h^2 + \dots + h^{2n-2}) \quad (3.30)$$

$$- h^{k_2-k_1} (1 + h^2 + \dots + h^{2n-2}). \quad (3.31)$$

It follows that

$$\Delta_{k_1}^T \Delta_{k_2} = -h^{2n+k_1-k_2} - h^{2n+k_1-k_2+2} + \dots - h^{2n+k_1+k_2-4} \quad (3.32)$$

$$- h^{k_1+k_2-2} - h^{k_1+k_2} \dots - h^{2n+k_2-k_1-2} \quad (3.33)$$

$$+ h^{2n+k_1-k_2} + h^{2n+k_1-k_2+2} + \dots + h^{2n+k_2-k_1-2}, \quad (3.34)$$

where (3.32) is obtained by combining (3.27) and (3.30), and (3.33) is obtained by combining (3.29) and (3.31). Furthermore, combining (3.32) and (3.34), we get that

$$\Delta_{k_1}^T \Delta_{k_2} = -h^{2n+k_2-k_1} - h^{2n+k_1-k_2+2} + \dots - h^{2n+k_1+k_2-4} \\ - h^{k_1+k_2-2} - h^{k_1+k_2} \dots - h^{2n+k_2-k_1-2},$$

which finally gives that

$$\Delta_{k_1}^T \Delta_{k_2} = -(h^{k_2+k_1-2} + h^{k_2+k_1} + \dots + h^{2n+k_2+k_1-4}) \\ = -h^{k_1+k_2-2} r^2,$$

as desired.

- The diagonal elements.

For $1 \leq k \leq n$, we need to verify that

$$\Delta_k^T \Delta_k = (1 + h^2 + \dots + h^{2n-2})^2 - (1 + h^2 + \dots + h^{2n-2}) h^{2k-2},$$

in which $\Delta_k = \delta_k + hr^2 \mathbf{e}_{n-k+1}$.

Notice that

$$\Delta_k^T \Delta_k = \|\delta_k\|_2^2 + 2hr^2 \delta_k^T \mathbf{e}_{n-k+1} + h^2 r^4 \\ = (h^{2n+2k-2} + h^{2n+2k} + \dots + h^{4n-4} + h^{4n-2}) \\ + (1 + h^2 + \dots + h^{2k-4}) - 2r^2 h^{2n} + h^2 r^4.$$

Furthermore, it holds that

$$h^{4n-2} - 2r^2 h^{2n} + h^2 r^4 = (hr^2 - h^{2n-1})^2 \\ = (h + h^3 + \dots + h^{2n-3})^2.$$

It remains to verify that

$$\begin{aligned} & (h^{2n+2k-2} + h^{2n+2k} + \dots + h^{4n-4}) \\ & \quad + (h + h^3 + \dots + h^{2n-3})^2 + (1 + h^2 + \dots + h^{2k-4}) \\ & = (1 + h^2 + \dots + h^{2n-2})^2 - (1 + h^2 + \dots + h^{2n-2})h^{2k-2}, \end{aligned}$$

which is equivalent to verifying

$$\begin{aligned} & (1 + h^2 + \dots + h^{4n-4}) + (h + h^3 + \dots + h^{2n-3})^2 \\ & = (1 + h^2 + \dots + h^{2n-2})^2. \end{aligned}$$

This follows from the following reasoning

$$\begin{aligned} & \frac{(1 - h^2)(1 - h^{4n-2}) + h^2(1 - h^{2n-2})^2}{(1 - h^2)^2} = \frac{(1 - h^{2n})^2}{(1 - h^2)^2} \\ \Leftrightarrow & 1 + h^{4n} - 2h^{2n} = (1 - h^{2n})^2. \end{aligned}$$

3.5.2 Proof of Fact 2

We firstly prove that the vector \mathbf{h} lies in the null space of Δ . Take for any $1 \leq k \leq n$ the corresponding column from the matrix Δ , we have that

$$\begin{aligned} \Delta_k^T \mathbf{h} &= \delta_k^T \mathbf{h} + hr^2 \mathbf{e}_{n-k+1}^T \mathbf{h} \\ &= -(h^{n+k-1} + h^{n+k+1} + \dots + h^{3n-k-1}) \\ & \quad - (h^{n-k+1} + h^{n-k+3} + \dots + h^{n+k-3}) \\ & \quad + (h^{n-k+1} + h^{n-k+3} + \dots + h^{3n-k-1}) \\ &= 0. \end{aligned}$$

Hence, $(Q_1 - Q_2)\mathbf{h} = 0$ and $\mathbf{h}^T(Q_1 - Q_2) = 0$ hold. Therefor it holds that $(Q_1 - Q_2)P = P(Q_1 - Q_2) = 0$. With this, the Fact 2 can be concluded by the following

$$\begin{aligned} & (Q_1 + Q_2)(Q_1 - Q_2) = (Q_1 - Q_2)(Q_1 + Q_2) \\ \Leftrightarrow & (I_n - P)(Q_1 - Q_2) = (Q_1 - Q_2)(I_n - P) \\ \Leftrightarrow & P(Q_1 - Q_2) = (Q_1 - Q_2)P. \end{aligned}$$

3.5.3 Proof of Fact 3

From (3.12), we have that

$$Q_2 Q_1 + Q_1 Q_2 = -Q_2 Q_1 - Q_1 Q_2.$$

From (3.13), we have that

$$Q_2 Q_1 - Q_1 Q_2 = -Q_2 Q_1 + Q_1 Q_2.$$

Hence we can conclude that $Q_1 Q_2 = Q_2 Q_1 = 0$.

3.5.4 Proof of Fact 4

As shown in the proof of Fact 2, we have that $(Q_1 - Q_2)P = 0$. Together with $(Q_1 + Q_2)P = 0$, we have that $Q_1P = 0$ and $Q_2P = 0$. Hence, the Fact 4 can be concluded as follows

$$\begin{aligned}Q_1 &= Q_1(Q_1 + Q_2 + P) = Q_1^2 + Q_1Q_2 + Q_1P = Q_1^2, \\Q_2 &= Q_2(Q_1 + Q_2 + P) = Q_1Q_2 + Q_2^2 + Q_2P = Q_2^2.\end{aligned}$$

Chapter 4

Score function estimation for systems with intractable transition kernels

4.1 Problem introduction

In this chapter, we consider the problem of estimating the parameters θ in the following nonlinear state space model

$$\begin{aligned}x_{t+1}|x_t &\sim f_\theta(x_{t+1}|x_t), \\y_t|x_t &\sim g_\theta(y_t|x_t),\end{aligned}$$

where x_t denotes the state and y_t denotes the measurement. Furthermore, we assume that the system transition kernel $f_\theta(\cdot)$ is intractable, which will be made clear shortly. This scenario can occur in many places, for example, when the state transition is given by nonlinear stochastic differential equations, and the observations are obtained at discrete time instances. In such scenarios, it is often impossible to obtain a closed form expression for the transition kernel between the consecutive observation times. The following example 4.1 gives an illustration of the concept of an intractable system. Note that although the expression for the transition kernel is intractable, we can still simulate the model to obtain samples of the states.

Example 4.1: The Phytoplankton-Zooplankton model

Here we will give an example, the Phytoplankton-Zooplankton (PZ) model introduced in [53, 69, 52], to illustrate the concept of an intractable model. Let t denotes time with 'day' as the unit, $p(t)$ and $z(t)$ denote the population size of phytoplankton and zooplankton, respectively.

Further, assume that the growth rate of phytoplankton $\alpha(t)$ satisfies the following stochastic process

$$\alpha(t) = \sum_{k=0}^{\infty} \alpha_k s(t-k), \tag{4.1}$$

where $s(t) = 1$ if $0 \leq t < 1$, otherwise $s(t) = 0$. In addition, $\{\alpha_k\}_{k=0}^{\infty}$ independently follow the same distribution as $\mathcal{N}(\mu_\alpha, \sigma_\alpha^2)$.

The dynamics of $p(t)$ and $z(t)$, i.e. the interactions between the Phytoplankton and Zooplankton, are characterized by the following differential equations

$$\begin{aligned}\frac{dp(t)}{dt} &= \alpha(t)p(t) - cp(t)z(t), \\ \frac{dz(t)}{dt} &= ecp(t)z(t) - m_l z(t) - m_q z^2(t),\end{aligned}$$

with the initial distributions for both species as

$$\log p_0 \sim \mathcal{N}(\mu_p, \sigma_p^2) \quad \text{and} \quad \log z_0 \sim \mathcal{N}(\mu_z, \sigma_z^2).$$

The observations are noisy measurements of the phytoplankton population given by

$$\log y(t) \sim \mathcal{N}(\log p(t), \sigma_y^2).$$

In the equations, c is the Zooplankton clearance rate, e is the Zooplankton growth efficiency, m_l and m_q are the linear and quadratic mortality rates of Zooplankton.

Assume that the observations are only collected at discrete times. Given the state at $t = k - 1$ as $x(k - 1) = (p(k - 1), z(k - 1))$, to get the state at time k , $x(k) = (p(k), z(k))$, we need to solve the previous differential equations with $x(k - 1)$ as the initial condition and $\alpha(t)$ given by the stochastic process in (4.1).

However, this differential equation can not be solved analytically, hence numerical methods has to be used to approximate the solution, which concludes that the transition kernel for this model is not analytically evaluable.

The Maximum (log-)Likelihood (ML) method is one attractive way for parameter estimation due to its large sample statistical efficiency [91]. Briefly stated, the ML approach finds the parameter estimate $\hat{\theta}_{ML}$ by maximizing the likelihood function $L(\theta) = p(y|\theta)$, or the log-likelihood function $l(\theta) = \log(p(y|\theta))$, i.e.

$$\hat{\theta}_{ML} = \arg \max_{\theta} L(\theta) = \arg \max_{\theta} l(\theta), \quad (4.2)$$

where θ represents the parameters and y denotes the measurements.

To numerically optimize (4.2), the gradient of the log-likelihood function, i.e. the score function, is useful. When the state transition kernel is tractable, Fisher's identity, together with a particle smoother can be applied [78, 62]. However, for the intractable case, the task becomes more challenging. Recently, [23] introduced an approach for score function estimation in the intractable case by introducing a pseudo prior for the parameters. The score

function can then be given as a function of the mean value of the parameter posterior distribution. Note that the method proposed in [23] is related to the work in [50, 51], which introduces the iterated filtering algorithm, a stochastic approximation based method for ML parameter estimation where the parameter is modeled as a random walk with diminishing variance.

The contribution of this chapter is that we provide an alternative derivation of the estimator in [23] by making use of a basic property of the convolution operator. Note that an idea close to the one presented here has been independently developed by the authors of [23], see their recent preprint on arXiv [24]. The main difference between this chapter and [24] lies in that the results in [24] directly utilize Stein's identity, while this chapter starts from an elementary property of the convolution operator which can offer additional insights. Note that in both the work in this chapter and the results in [24], the pseudo prior distribution for the parameter is assumed to be Gaussian.

This chapter is structured as follows. In next section, the convolution operator and Stein's identity are introduced and their connections are established. We will then illustrate how to leverage the connection to estimate the score function. After that, we briefly discuss the second order derivative estimation of the log likelihood function. A numerical example is conducted to show the efficacy of the results. Finally, we conclude the chapter with some open questions. In this chapter, we will stick to the univariate case. The time indexes will also be dropped occasionally for notational simplicity.

In the following, the double factorial $k!!$ denotes $k \times (k-2) \cdots \times 3 \times 1$ when k is odd, and $k \times (k-2) \cdots \times 4 \times 2$ when k is even. We will use $f^{(l)}(\cdot)$ to denote the l -th derivative of function $f(\cdot)$. We also use $f'(\cdot)$ and $f''(\cdot)$ to denote the first and second order derivatives of $f(\cdot)$ when they become more convenient.

4.2 The convolution operator and Stein's identity

We first describe the convolution operator and one of its relevant properties that will be used later. For two functions $f(\alpha) : \mathbb{R} \rightarrow \mathbb{R}$ and $g(\alpha) : \mathbb{R} \rightarrow \mathbb{R}$, the convolution of them is defined as

$$(f * g)(\theta_0) = \int_{-\infty}^{+\infty} f(\theta)g(\theta_0 - \theta)d\theta = \int_{-\infty}^{+\infty} f(\theta_0 - \theta)g(\theta)d\theta. \quad (4.3)$$

If both functions are differentiable and satisfying certain integrability conditions, the following basic property holds

$$(f * g')(\theta_0) = (f' * g)(\theta_0), \quad (4.4)$$

which can be written as

$$\int_{-\infty}^{+\infty} f(\theta)g'(\theta_0 - \theta)d\theta = \int_{-\infty}^{+\infty} f'(\theta)g(\theta_0 - \theta)d\theta. \quad (4.5)$$

Let $g(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} : 0, \tau^2)$, with a probabilistic view of (4.4), it gives that

$$\tau^{-2} \mathbb{E}_{\theta}[(\boldsymbol{\theta} - \boldsymbol{\theta}_0)f(\boldsymbol{\theta})] = \mathbb{E}_{\theta}[f'(\boldsymbol{\theta})], \quad (4.6)$$

where the expectation is taken with respect to $\mathcal{N}(\boldsymbol{\theta} : \boldsymbol{\theta}_0, \tau^2)$.

Note that this is commonly referred to as Stein's identity [92]. The usual *integration by parts* approach can also be used to establish Stein's identity. Benefit of the current approach is that it gives more intuition of the identity, by linking it to the well-known concept of convolution. Besides this, it can also be used to estimate the second (possibly higher) order derivative of the log-likelihood function, which will be discussed in Section IV. In next section, we will apply the property in (4.4) for estimating the score function.

4.3 Score function estimation

4.3.1 Asymptotic analysis

Notice that the following result holds

$$\lim_{\tau \rightarrow 0} \mathcal{N}(\boldsymbol{\theta} : \boldsymbol{\theta}_0, \tau^2) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_0), \quad (4.7)$$

where the $\delta(\cdot)$ indicates the Dirac delta function. This allows us to conclude that the right hand side of (4.6) can approximate $f'(\boldsymbol{\theta}_0)$ well in the limit when τ approaches zero. In turn, the left hand side is also a reasonable proxy for $f'(\boldsymbol{\theta}_0)$.

Assume that the likelihood of the data y is given by $p(y|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the system parameter. Further, we denote

$$l(\boldsymbol{\theta}) = \log(p(y|\boldsymbol{\theta}))$$

and accordingly

$$l^{(1)}(\boldsymbol{\theta}) = \frac{p'(y|\boldsymbol{\theta})}{p(y|\boldsymbol{\theta})}.$$

The task for us is to estimate $l^{(1)}(\boldsymbol{\theta})$ at $\boldsymbol{\theta}_0$.

Notice that we can link the relation given in (4.6) to the dynamical system as follows. Let $f(\boldsymbol{\theta})$ in (4.6) be given by $f(\boldsymbol{\theta}) = \frac{p(y|\boldsymbol{\theta})}{p(y)}$, we have that

$$\tau^{-2} \mathbb{E}_{\theta} \left((\boldsymbol{\theta} - \boldsymbol{\theta}_0) \frac{p(y|\boldsymbol{\theta})}{p(y)} \right) = \mathbb{E}_{\theta} \left(\frac{p'(y|\boldsymbol{\theta})}{p(y)} \right). \quad (4.8)$$

Based on (4.8), we make the following observations:

- Direct verification gives that the left hand side of (4.8) is the expectation of $(\boldsymbol{\theta} - \boldsymbol{\theta}_0)$ with respect to the posterior distribution $p(\boldsymbol{\theta}|y)$ if we regard $\mathcal{N}(\boldsymbol{\theta} : \boldsymbol{\theta}_0, \tau^2)$ as the prior distribution of $\boldsymbol{\theta}$. Later on, we will denote the left hand side of (4.8) by $\tau^{-2} \mathbb{E}_{\boldsymbol{\theta}_0, \tau}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)$, as in [23].

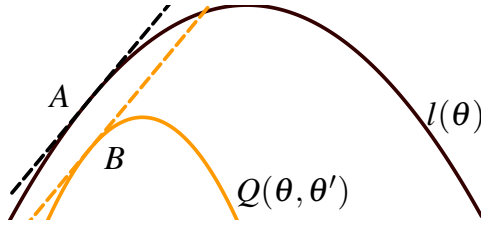


Figure 4.1: A geometrical illustration of Fisher's identity.

- Given (4.7), we have that

$$p(y) = \int p(y|\theta) \mathcal{N}(\theta : \theta_0, \tau^2) d\theta \rightarrow p(y|\theta_0),$$

when $\tau \rightarrow 0$, which means that the right hand side of (4.8) will converge to the score function at θ_0 when $\tau \rightarrow 0$. In turn, it implies that the left hand side of (4.8) will provide a reasonable estimate of the score function at θ_0 when τ is small.

To summarize, this analysis offers an intuition of the feasibility in exploiting the property in (4.4) for the score stimulation, and the estimator is given by the quantity $\tau^{-2} \mathbb{E}_{\theta_0, \tau}(\theta - \theta_0)$.

Remark 4.1. *Note that when an expression of the state transition kernel is not explicitly known, Fisher's identity [78] is not directly applicable. However, it is still possible to sample from the posterior distribution $p(\theta|y)$ by using particle MCMC methods [1], which provides a Monte Carlo approximation to the left hand side of (4.8), and in turn an approximation of the score function.*

Example 4.2: Why Fisher's identity is not applicable

In Figure 4.1, $l(\theta)$ denotes the log likelihood function, and $Q(\theta, \theta')$ is given by

$$Q(\theta, \theta') = \int \log p_{\theta}(x, y) p_{\theta'}(x|y) dx,$$

where y denotes system observations and x denotes the state variables. For simplicity, we assume θ to be univariate. The Fisher's identity gives that

$$\left. \frac{dQ(\theta, \theta')}{d\theta} \right|_{\theta=\theta'} = \left. \frac{dl(\theta)}{d\theta} \right|_{\theta=\theta'}.$$

In other words, it says that the tangent of $l(\theta)$ at the point $A = (\theta', l(\theta'))$ and the tangent of $Q(\theta, \theta')$ at the point $B = (\theta', Q(\theta', \theta'))$ are parallel to each other. See a geometrical illustration in Figure 4.1.

With this relation, score function estimation at θ' can be done by estimating the derivative of $Q(\theta, \theta')$ at θ' , given as

$$\frac{dQ(\theta, \theta')}{d\theta} \Big|_{\theta=\theta'} = \int \frac{d(\log p_{\theta}(x, y))}{d\theta} \Big|_{\theta=\theta'} p_{\theta'}(x|y) dx. \quad (4.9)$$

A Monte Carlo approximation of (4.9) can be formulated by sampling from $p_{\theta'}(x|y)$ using a particle smoother [62]. Note that to evaluate the differential inside the integral of (4.9), analytical formulations of both the state transition density and the observation density should be given, which restricts its straightforward applicability to the problem discussed in this chapter.

Remark 4.2. *Note that if we let $g(\theta)$ be a probability density function which belongs to the exponential family, then by the convolution property in (4.4), a generalized Stein's identity, similar to (4.6), can also be established, see also the results in [49].*

With this generalization, an estimator of the score function estimation can be found analogously to what was done before. Note that, in this case, the pseudo prior introduced for the parameter does not necessarily have to be Gaussian.

4.3.2 Convergence rate analysis

This section establishes the non-asymptotic analysis of the estimator. More specifically, we will establish the following result.

Proposition 4.1. *Assume that $p(y|\theta)$ is analytic and for given θ_0 and τ , there exist $C_1(\theta_0, \tau)$ and $C_2(\theta_0, \tau)$ defined as*

$$C_1(\theta_0, \tau) = \frac{1}{p(y|\theta_0)} \left| \sum_{k \text{ is even}} \frac{p^{(k+1)}(y|\theta_0) \tau^{k-2}}{k!!} \right| \quad (4.10)$$

and

$$C_2(\theta_0, \tau) = \left| \left(\sum_{k \text{ is even}} \frac{p^{(k)}(y|\theta_0) \tau^{k-2}}{k!!} \right) \frac{\int p'(y|\theta) \mathcal{N}(\theta : \theta_0, \tau^2) d\theta}{p(y|\theta_0)p(y)} \right|. \quad (4.11)$$

Further, assume that the series $\sum_{k=1}^{\infty} c_k(\theta)$ and $\sum_{k=1}^{\infty} s_k(\theta)$ defined in (4.21) and (4.25) are uniformly convergent series on \mathbb{R} , then we have that

$$|l^{(1)}(\theta_0) - \tau^{-2} \mathbb{E}_{\theta_0, \tau}(\theta - \theta_0)| \leq C(\theta_0, \tau) \tau^2, \quad (4.12)$$

where $C(\theta_0, \tau) = C_1(\theta_0, \tau) + C_2(\theta_0, \tau)$.

Proof. By making use of (4.8) and that $\mathbb{E}_\theta \left(\frac{p'(y|\theta_0)}{p(y|\theta_0)} \right) = \frac{p'(y|\theta_0)}{p(y|\theta_0)}$, the error in (4.12) can be decomposed according to

$$\begin{aligned} \left| l^{(1)}(\theta_0) - \tau^{-2} \mathbb{E}_{\theta_0, \tau}(\theta - \theta_0) \right| &= \left| \frac{p'(y|\theta_0)}{p(y|\theta_0)} - \mathbb{E}_\theta \left(\frac{p'(y|\theta)}{p(y)} \right) \right| \\ &= \left| \mathbb{E}_\theta \left(\frac{p'(y|\theta_0)}{p(y|\theta_0)} - \frac{p'(y|\theta)}{p(y)} \right) \right| \\ &\leq |\mathbb{E}_\theta T_1(\theta)| + |\mathbb{E}_\theta T_2(\theta)|, \end{aligned}$$

where

$$T_1(\theta) = \frac{p'(y|\theta_0)}{p(y|\theta_0)} - \frac{p'(y|\theta)}{p(y|\theta)}$$

and

$$T_2(\theta) = \frac{p'(y|\theta)}{p(y|\theta_0)} - \frac{p'(y|\theta)}{p(y)}.$$

It now remains to bound the terms $|\mathbb{E}_\theta T_1(\theta)|$ and $|\mathbb{E}_\theta T_2(\theta)|$ separately. These bounds are derived in the appendix. \square

Remark 4.3. *Note that although the convergence rates of the estimator in both this chapter and [24] are the same, the derivation strategies are different. In [24], the authors rely on techniques from Bayesian asymptotic theory to directly analyze the convergence rate of the estimator, i.e. the left hand side of (4.6), while the work in this chapter tries to bound the right hand side of (4.6). This allows us to decompose the error into separate terms, where each term is relatively easy to bound.*

4.4 Estimating higher order derivatives

Given (4.4), it also holds true that for $l \geq 1$, we have

$$(f^{(l)}(\theta) * g(\theta))(\theta_0) = (f(\theta) * g^{(l)}(\theta))(\theta_0), \quad (4.13)$$

where $f^{(l)}$ denotes the l -th derivative of the function f . This property makes it possible to generalize Stein's identity to higher order cases and it also opens for the possibility of estimating higher order derivatives of the log-likelihood.

Next, we discuss the case when $l = 2$ by using the property given in (4.13). Let $g(\theta) = \mathcal{N}(\theta : 0, \tau^2)$, and $f(\theta) = \frac{p(y|\theta)}{p(y)}$, then according to (4.13), we have that

$$-\tau^{-2} + \tau^{-4} \mathbb{E}_{\theta_0, \tau}(\theta - \theta_0)^2 = \mathbb{E}_\theta \left(\frac{p''(y|\theta)}{p(y)} \right), \quad (4.14)$$

where $\mathbb{E}_{\theta_0, \tau}(\cdot)$ is taken with respect to posterior distribution of θ , and $\mathbb{E}_\theta(\cdot)$ is taken with respect to the pseudo prior distribution $\mathcal{N}(\theta : \theta_0, \tau^2)$.

Furthermore, taking the square for both sides of (4.8), we have that

$$\tau^{-4} \mathbb{E}_{\theta_0, \tau}^2(\theta - \theta_0) = \mathbb{E}_\theta^2 \left(\frac{p'(y|\theta)}{p(y)} \right). \quad (4.15)$$

Subtracting (4.15) from (4.14), we have that

$$-\tau^{-2} + \tau^{-4} \mathbb{V}_{\theta_0, \tau}(\theta) = \mathbb{E}_\theta \left(\frac{p''(y|\theta)}{p(y)} \right) - \mathbb{E}_\theta^2 \left(\frac{p'(y|\theta)}{p(y)} \right), \quad (4.16)$$

where $\mathbb{V}_{\theta_0, \tau}(\theta)$ denotes the variance of θ taken with respect to the posterior distribution of θ . Notice that when $\tau \rightarrow 0$, the right hand side of (4.16) will converge to

$$\frac{p''(y|\theta_0)}{p(y|\theta_0)} - \left(\frac{p'(y|\theta_0)}{p(y|\theta_0)} \right)^2 = l^{(2)}(\theta) |_{\theta=\theta_0}.$$

Hence, the left hand side of (4.16) is a reasonable proxy for $l^{(2)}(\theta) |_{\theta=\theta_0}$, which can also be sampled using particle MCMC methods analogously to what was done previously in the first order case.

4.5 Numerical illustration

In this section, we will test the estimator by applying it to a simple toy example. We consider the first order autoregressive system as follows

$$x_{t+1} = \theta x_t + v_t, \quad (4.17)$$

$$y_t = x_t + w_t, \quad (4.18)$$

in which $\{v_t\}_{t=1}^T, \{w_t\}_{t=1}^T$ and x_1 are all standard normal distributed random variables, $T = 5$, and the true θ that was used in generating the data is given by $\theta_0 = 0.8$.

We want to estimate the score function at θ_0 given the data $\{y_t\}_{t=1}^T$. In the experiment, we make use of the Particle Metropolis-Hastings (PMH) sampler [1] to sample from the posterior distribution for the system parameter, that is $p(\theta|y_{1:T})$. Inside PMH, the bootstrap particle filter is used with $N = 50$ particles. The Markov chain is run for $M = 30000$ iterations, and the first 15000 samples are discarded to avoid the burn-in period. Note that, for such a linear Gaussian system, it is possible to evaluate the exact score function by making use of the so-called sensitivity derivatives advocated in [3]. This method is also used in this chapter to compute the exact value of the score function. The experimental results are reported in the Figure 4.2, further details are given in the figure's caption.

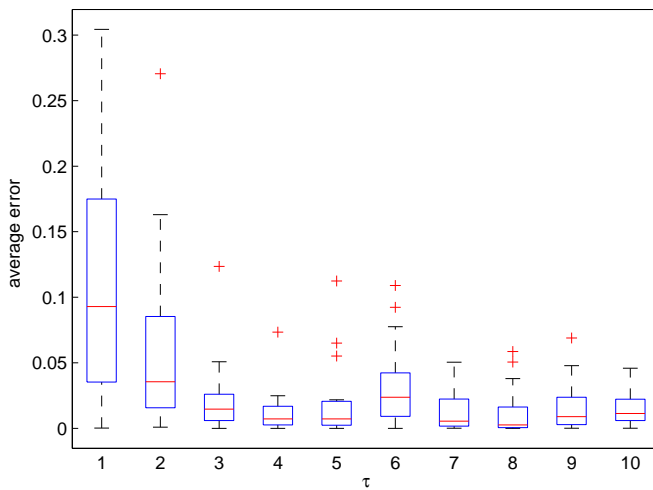


Figure 4.2: Boxplot of the average squared error for the estimates of the score function (at θ_0) when PMH is used to sample from the posterior distribution. The score value at θ_0 obtained by the sensitivity analysis method is -1.1960 . The pseudo prior introduced is $\mathcal{N}(\theta : \theta_0, \tau)$, where τ ranges from 0.005 to 0.050 with the increment 0.005. For each τ , we run 20 independent simulations. Note that, as we can see from the plot, the best result is obtained when τ is 0.02.

4.6 Conclusion

In this chapter, we revisited the problem of score function estimation by making use of a basic property of the convolution operator. We also illustrate the results with a toy example for which the true score function can be analytically computed. Empirically, we have observed that the numerical error will increase when the variance of the pseudo prior shrinks towards zero or becomes large. How to find the optimal value for the variance of the pseudo prior is currently an open problem. Note that the relation between the convolution property and the score function estimation (derivative estimation) is not specific for state space models. It would be interesting to study how the connection could be generalized to other applications.

4.7 Appendix

We first bound $|\mathbb{E}_\theta T_1(\theta)|$ as follows.

$$\begin{aligned}
 |\mathbb{E}_\theta T_1(\theta)| &= \left| \int \left(\frac{p'(y|\theta_0)}{p(y|\theta_0)} - \frac{p'(y|\theta)}{p(y|\theta)} \right) \mathcal{N}(\theta : \theta_0, \tau^2) d\theta \right| \\
 &= \frac{1}{p(y|\theta_0)} \left| \int (p'(y|\theta_0) - p'(y|\theta)) \mathcal{N}(\theta : \theta_0, \tau^2) d\theta \right| \\
 &= \frac{1}{p(y|\theta_0)} \left| \int \left(\sum_{k=1}^{\infty} \frac{p^{(k+1)}(y|\theta_0)}{k!} (\theta - \theta_0)^k \right) \mathcal{N}(\theta : \theta_0, \tau^2) d\theta \right| \tag{4.19}
 \end{aligned}$$

$$\begin{aligned}
 &= \frac{1}{p(y|\theta_0)} \left| \sum_{k=1}^{\infty} \frac{p^{(k+1)}(y|\theta_0)}{k!} \int (\theta - \theta_0)^k \mathcal{N}(\theta : \theta_0, \tau^2) d\theta \right| \\
 &= \frac{1}{p(y|\theta_0)} \left| \sum_{k \text{ is even}} \frac{p^{(k+1)}(y|\theta_0) \tau^k (k-1)!!}{k!} \right| \tag{4.20} \\
 &= C_1(\theta_0, \tau) \tau^2.
 \end{aligned}$$

In the previous derivations, in (4.19), we have made use of the Taylor expansion of the function $p(y|\theta)$ around θ_0 . Furthermore, let

$$c_k(\theta) = \frac{p^{(k+1)}(y|\theta_0)}{k!} \mathcal{N}(\theta : \theta_0, \tau^2) (\theta - \theta_0)^k, \tag{4.21}$$

then by the uniformly convergence assumption of $\sum_{k=1}^{\infty} c_k(\theta)$, the order of the integration and the infinite summation in (4.19) can be interchanged. In (4.20), we have applied the central moments of the normal distribution functions [106].

Remark 4.4. Notice that if we assume $C_1(\theta_0, \tau)$ to exist, then we have that $\lim_{\tau \rightarrow 0} C_1(\theta_0, \tau) = \frac{p^{(2)}(y|\theta_0)}{2p(y|\theta_0)}$.

Next, we proceed to establish a bound for $|\mathbb{E}_\theta T_2(\theta)|$. Denote

$$\Delta = p(y) - p(y|\theta_0),$$

then we have that

$$\begin{aligned} |\mathbb{E}_\theta T_1(\theta)| &= \left| \int \left(\frac{p'(y|\theta)}{p(y|\theta_0)} - \frac{p'(y|\theta)}{p(y)} \right) \mathcal{N}(\theta : \theta_0, \tau^2) d\theta \right| \\ &= \left| \frac{\Delta}{p(y|\theta_0)p(y)} \right| \left| \int p'(y|\theta) \mathcal{N}(\theta : \theta_0, \tau^2) d\theta \right| \\ &= |\Delta| \left| \frac{\int p'(y|\theta) \mathcal{N}(\theta : \theta_0, \tau^2) d\theta}{p(y|\theta_0)p(y)} \right| \end{aligned} \quad (4.22)$$

We will proceed by deriving a bound on $|\Delta|$. Let us start by noting that

$$\begin{aligned} |\Delta| &= |p(y) - p(y|\theta_0)| \\ &= \left| \int (p(y|\theta) - p(y|\theta_0)) \mathcal{N}(\theta : \theta_0, \tau^2) d\theta \right| \\ &= \left| \int \left(\sum_{k=1}^{\infty} \frac{p^{(k)}(y|\theta_0)}{k!} (\theta - \theta_0)^k \right) \mathcal{N}(\theta : \theta_0, \tau^2) d\theta \right| \end{aligned} \quad (4.23)$$

$$\begin{aligned} &= \left| \sum_{k=1}^{\infty} \frac{p^{(k)}(y|\theta_0)}{k!} \int (\theta - \theta_0)^k \mathcal{N}(\theta : \theta_0, \tau^2) d\theta \right| \\ &= \left| \sum_{k \text{ is even}} \frac{p^{(k)}(y|\theta_0) \tau^k}{k!} \right|. \end{aligned} \quad (4.24)$$

Similar as before, due to the uniformly convergence assumption on $\sum_{k=1}^{\infty} s_k(\theta)$, where

$$s_k(\theta) = \frac{p^{(k)}(y|\theta_0)}{k!} \mathcal{N}(\theta : \theta_0, \tau^2) (\theta - \theta_0)^k, \quad (4.25)$$

the order of integration and the infinite summation in (4.23) can be interchanged. By inserting (4.24) into (4.22), we obtain

$$|\mathbb{E}_\theta T_1(\theta)| = C_2(\theta_0, \tau) \tau^2. \quad (4.26)$$

Remark 4.5. Similar to the remark on $C_1(\theta_0, \tau)$, if we also assume that, for given θ_0 , $C_2(\theta_0, \tau)$ exists, then we have that

$$\lim_{\tau \rightarrow 0} C_2(\theta_0, \tau) = \frac{p'(y|\theta_0)p^{(2)}(y|\theta_0)}{[p(y|\theta_0)]^2}.$$

Combing all the previous results, we have that

$$\left| l^{(1)}(\boldsymbol{\theta}_0) - \tau^{-2} \mathbb{E}_{\boldsymbol{\theta}_0, \tau}(\boldsymbol{\theta} - \boldsymbol{\theta}_0) \right| \leq C(\boldsymbol{\theta}_0, \tau) \tau^2. \quad (4.27)$$

Chapter 5

On the Recursive Direct Weight Optimization

5.1 Problem formulation

Local modeling is an attractive approach for building nonlinear dynamical models. The general idea of this approach is to estimate the function value at a specific point by making use of the neighboring observed data. The kernel estimator and the local polynomial method are two classical ways for implementing the local modeling idea, for which the final estimator is given by a linear combination of the observations with suitably chosen weights. The choice of weights in these methods often relies on asymptotic arguments as the number of observations goes to infinity, which is not practical in reality. To mitigate this drawback, the Direct Weight Optimization (DWO) method [82, 83, 84] estimates the function value with a direct linear combination of the observations, and then finds the corresponding weights by optimizing a suitably chosen criteria (an upper bound of the worst case mean squared error).

Similar to the spirit of DWO, in [6], the authors proposed the so called Recursive Direct Weight Optimization (RDWO) approach. The idea of RDWO is based on minimizing the probability that an upper bound of the estimation error is larger than a predefined threshold. Notable properties of this approach include: (1) the solution to the final optimization problem can be solved analytically; (2) the optimal solution can be updated recursively when new observation is available, which only involves simple algebraic calculations. However, though the algorithm possesses nice properties and is easy to implement, its original derivation turns out to be quite involved. The contributions of this chapter lie in the following two points: (1) A simpler derivation of the RDWO method is presented; (2) A novel interpretation of the method is provided as well.

The chapter will be organized as follows. Section 5.2 outlines the main idea of the RDWO method; Section 5.3 introduces the new findings; Finally in section 5.4, we summarize the chapter.

5.2 The RDWO method

We will give a brief overview of the RDWO method, following the notations introduced in [6]. As in [6], we will illustrate the idea under the univariate case. The nonlinear system considered is given by

$$y(k) = f(\varphi(k)) + e(k), \quad k = 1, \dots, N, \quad (5.1)$$

where the nonlinear function $f(\cdot)$ is assumed to be differentiable with Lipschitz constant L_1 , i.e.

$$\left| \frac{df(\varphi)}{d\varphi} \right| \leq L_1, \quad \varphi \in \mathbb{R}.$$

For $k = 1, \dots, N$, $y(k) \in \mathbb{R}$ are observations, $\varphi(k) \in \mathbb{R}$ are inputs, and $e(k) \in \mathbb{R}$ are noise terms, which are assumed to be independent and identically distributed (i.i.d.) Gaussian noise with zero mean and variance σ_e^2 .

For point x , the estimate $\hat{f}(x)$ of $f(x)$ is given by

$$\hat{f}(x) = \sum_{k=1}^N w_x(k) y(k), \quad (5.2)$$

where $\sum_{k=1}^N w_x(k) = 1$. The RDWO method is one way to decide the weights $\{w_x(k)\}_{k=1}^N$ in certain optimal sense. Further define that

$$\tilde{\varphi}_x(k) = |x - \varphi(k)|,$$

for $k = 1, \dots, N$. The following upper bound of the squared estimation error $(\hat{f}(x) - f(x))^2$ can be obtained:

$$(\hat{f}(x) - f(x))^2 \leq z^2, \quad (5.3)$$

where

$$z = L_1 \sum_{k=1}^N |w_x(k)| \tilde{\varphi}_x(k) + \left| \sum_{k=1}^N w_x(k) e(k) \right|.$$

In [6] it is pointed out that this bound is in fact tight for some functions and noises for which the inequality becomes an equality. Notice that z is a random variable, whose density function can be computed analytically. The RDWO method then finds the weights by minimizing the probability of z being greater than a predefined threshold δ' , that is to solve

$$\min_{\{w_x(k)\}_{k=1}^N} \text{Prob}(z \geq \delta'), \quad \text{s.t.} \quad \sum_{k=1}^N w_x(k) = 1, \quad (5.4)$$

where δ' satisfies

$$\delta' > L_1 \sum_{k=1}^N |w_x(k)| \tilde{\varphi}_x(k) \geq L_1 \min \{ \tilde{\varphi}_x(k) \}_{k=1}^N.$$

After some algebraic simplifications as done in [6], it turns out that the optimized weights can be obtained by solving the following optimization problem.

For given $\delta = \frac{\delta'}{L_1} > 0$, N , $x \in \mathbb{R}$, solve:

$$\begin{aligned} \widehat{w}_x &= \arg \max_{w_x(k)} \frac{\delta - \sum_{k=1}^N |w_x(k)| \widetilde{\varphi}_x(k)}{\sqrt{\sum_{k=1}^N w_x(k)^2}} & (5.5) \\ \text{s.t. } & \sum_{k=1}^N w_x(k) = 1. \end{aligned}$$

The main result in [6], which gives the analytical solution to the problem (5.5), is phrased as follows.

Theorem 5.1. *Suppose that $\delta > \min_{1 \leq k \leq N} \widetilde{\varphi}_x(k)$. Let $M_x \triangleq \{m_1, m_2, \dots, m_l\}$ be a set such that $m \in M_x \Leftrightarrow \delta > \widetilde{\varphi}_x(m)$. Then the solution to (5.5) is unique and given by*

$$\widehat{w}_x(k) = \begin{cases} \frac{\delta - \widetilde{\varphi}_x(k)}{l\delta - \sum_{i=1}^l \widetilde{\varphi}_x(m_i)}, & k \in M_x, \\ 0, & k \notin M_x. \end{cases} \quad (5.6)$$

Based on Theorem 5.1, a recursive scheme can be designed for updating the weights when new observations are obtained [6].

5.3 Analysis and the main result

In this section, we will first present the new derivation for the RDWO, after that, a new interpretation of the RDWO based on the new derivation will be discussed.

5.3.1 Derivation

Lemma III.1 in [6] will be reused for our analysis, which is restated as Lemma 5.1 here.

Lemma 5.1. *The problem given in (5.5) is equivalent to the following optimization problem: For given $\delta = \frac{\delta'}{L_1} > 0$, N , $x \in \mathbb{R}$, solve:*

$$\begin{aligned} \widehat{w}_x &= \arg \max_{w_x} \frac{\delta - \sum_{k=1}^N w_x(k) \widetilde{\varphi}_x(k)}{\sqrt{\sum_{k=1}^N w_x(k)^2}} & (5.7) \\ \text{s.t. } & \sum_{k=1}^N w_x(k) = 1, \\ & w_x(k) \geq 0, \quad \text{for } k = 1, \dots, N. \end{aligned}$$

Let us introduce the quantities

$$\widehat{\varphi}_x(k) \triangleq \delta - \widetilde{\varphi}_x(k), \quad k = 1, \dots, N, \quad (5.8)$$

which will play important roles in the following derivations. A fact relating the different notations introduced so far is given by

$$m \in M_x \Leftrightarrow \delta > \widetilde{\varphi}_x(m) \Leftrightarrow \widehat{\varphi}_x(m) > 0. \quad (5.9)$$

Note that $\{\widehat{\varphi}_x(k)\}_{k=1}^N$ reveals a particular structure in the problem. Notice the following fact

$$\delta - |x - \varphi(k)| = \min\{\varphi(k) - (x - \delta), (x + \delta) - \varphi(k)\},$$

so when $k \in M_x$, i.e. when $\varphi(k)$ lies in the interval $(x - \delta, x + \delta)$, the values $\varphi(k) - (x - \delta)$ and $(x + \delta) - \varphi(k)$ measure the distances between $\varphi(k)$ and the points $x - \delta$ and $x + \delta$ separately. An illustration of $\widehat{\varphi}_x(k)$ is given in Figure 5.1.

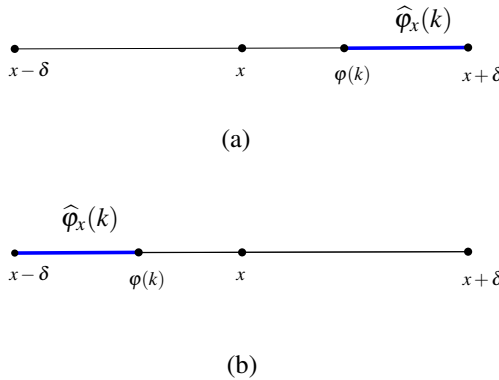


Figure 5.1: This figure illustrates the meaning of $\widehat{\varphi}_x(k)$ when $k \in M_x$ in different cases. We can see that, $\widehat{\varphi}_x(k)$ measures the distance between $\varphi(k)$ and the set of endpoints of the interval $(x - \delta, x + \delta)$ which is illustrated by the lengths of the thick segments.

Next, we will transform the problem in (5.7) into an equivalent formulation using $\widehat{\varphi}_x(k)$. Note that

$$\begin{aligned} \delta - \sum_{k=1}^N w_x(k) \widetilde{\varphi}_x(k) &= \delta \sum_{k=1}^N w_x(k) - \sum_{k=1}^N w_x(k) \widetilde{\varphi}_x(k) \\ &= \sum_{k=1}^N w_x(k) \widehat{\varphi}_x(k), \end{aligned}$$

which implies that the problem (5.7) can be rewritten as

$$\begin{aligned} \widehat{w}_x &= \arg \max_{w_x} \frac{\sum_{k=1}^N w_x(k) \widehat{\varphi}_x(k)}{\sqrt{\sum_{k=1}^N w_x(k)^2}} \\ \text{s.t. } \quad &\sum_{k=1}^N w_x(k) = 1, \\ &w_x(k) \geq 0, \quad \text{for } k = 1, \dots, N. \end{aligned} \quad (5.10)$$

The sketch of the following part is as follows. Based on the new formulation in (5.10), we will first establish a lemma, which makes it possible to determine the zero elements of \widehat{w}_x before actually solving the problem in (5.10). When the zero elements of \widehat{w}_x have been identified beforehand, the remaining task is to determine those nonzero elements, which are finally determined by an application of the Cauchy-Schwartz inequality.

Lemma 5.2. *For the problem given in (5.10), if $\widehat{\varphi}_x(k) \leq 0$, then it holds that $\widehat{w}_x(k) = 0$.*

Proof. The proof is given by a contradiction argument. Assume that $\widehat{\varphi}_x(k) \leq 0$, but $\widehat{w}_x(k) \neq 0$.

First, we show that $\widehat{w}_x(k) < 1$ holds. According to the assumptions given in Theorem 5.1, we have that

$$\delta > \min_{1 \leq j \leq N} \widehat{\varphi}_x(j),$$

which gives that the maximum value of the objective function in (5.7) and also in (5.10) will be positive. Together with the fact that $\sum_{1 \leq j \leq N} \widehat{w}_x(j) = 1$, we have that $\widehat{w}_x(k) \neq 1$, otherwise the maximum value of the objective function in (5.10) will be non-positive.

Since $\widehat{w}_x(k) \neq 1$, we can construct another point \bar{w}_x satisfying the constraints to (5.10), which is given as:

$$\bar{w}_x(i) = \begin{cases} \frac{\widehat{w}_x(i)}{1 - \widehat{w}_x(k)}, & i \neq k, \\ 0, & i = k, \end{cases}$$

for $i = 1, \dots, N$.

It will be proven shortly that the objective function will increase at $\bar{w}_x = (\bar{w}_x(1), \dots, \bar{w}_x(N))$, which contradicts the fact that \widehat{w}_x is the optimal value, thus the proof is concluded.

The reasoning is given by the following arguments.

$$\begin{aligned}
\frac{\sum_{i=1}^N \bar{w}_x(i) \widehat{\varphi}_x(i)}{\sqrt{\sum_{i=1}^N \bar{w}_x(i)^2}} &= \frac{\sum_{i=1, i \neq k}^N \frac{\widehat{w}_x(i)}{1 - \widehat{w}_x(k)} \widehat{\varphi}_x(i)}{\sqrt{\sum_{i=1, i \neq k}^N \left(\frac{\widehat{w}_x(i)}{1 - \widehat{w}_x(k)}\right)^2}} \\
&= \frac{\sum_{i=1, i \neq k}^N \widehat{w}_x(i) \widehat{\varphi}_x(i)}{\sqrt{\sum_{i=1, i \neq k}^N \widehat{w}_x(i)^2}} \\
&\stackrel{(I)}{>} \frac{\sum_{i=1}^N \widehat{w}_x(i) \widehat{\varphi}_x(i)}{\sqrt{\sum_{i=1}^N \widehat{w}_x(i)^2}}. \tag{5.11}
\end{aligned}$$

The inequality (I) follows from the following facts. By assumption, $\widehat{\varphi}_x(k) \leq 0$ and $\widehat{w}_x(k) > 0$ hold, so we have

$$\sum_{i=1, i \neq k}^N \widehat{w}_x(i) \widehat{\varphi}_x(i) \geq \sum_{i=1}^N \widehat{w}_x(i) \widehat{\varphi}_x(i).$$

Also, since $\widehat{w}_x(k) \neq 0$, it follows that

$$0 < \sqrt{\sum_{i=1, i \neq k}^N \widehat{w}_x(i)^2} < \sqrt{\sum_{i=1}^N \widehat{w}_x(i)^2}.$$

These two facts conclude the inequality (I), and finalize the proof. \square

From Lemma 5.2, we can conclude that to optimize (5.10), we only need to optimize over the weights for those $\widehat{\varphi}_x(i)$ which are positive. With the facts given in Lemma 5.2, the remaining steps for the derivation of Theorem 5.1 are as follows.

Proof. The optimization problem (5.10) can be translated into the following problem:

$$\begin{aligned}
\widehat{w}_x &= \arg \max_{w_x(k)} \frac{\sum_{k \in M_x} w_x(k) \widehat{\varphi}_x(k)}{\sqrt{\sum_{k \in M_x} w_x(k)^2}} \\
\text{s.t. } &\sum_{k \in M_x} w_x(k) = 1, \\
&w_x(k) \geq 0.
\end{aligned}$$

Applying the Cauchy-Schwartz inequality, we have that

$$\begin{aligned}
\frac{\sum_{k \in M_x} w_x(k) \widehat{\varphi}_x(k)}{\sqrt{\sum_{k \in M_x} w_x(k)^2}} &\leq \frac{\sqrt{\sum_{k \in M_x} w_x^2(k)} \sqrt{\sum_{k \in M_x} \widehat{\varphi}_x^2(k)}}{\sqrt{\sum_{k \in M_x} w_x(k)^2}} \\
&= \sqrt{\sum_{k \in M_x} \widehat{\varphi}_x(k)^2}, \tag{5.12}
\end{aligned}$$

and the equality holds when $\frac{w_x(k)}{\widehat{\varphi}_x(k)} = C, k \in M_x$, in which C is a constant which will be determined shortly. Since $\sum_{k \in M_x} w_x(k) = 1$, we have that $\sum_{k \in M_x} C \widehat{\varphi}_x(k) = 1$, which in turn gives that

$$C = \frac{1}{\sum_{k \in M_x} \widehat{\varphi}_x(k)}.$$

Finally, we conclude that

$$\widehat{w}_x(k) = \begin{cases} \frac{\widehat{\varphi}_x(k)}{\sum_{i \in M_x} \widehat{\varphi}_x(m_i)}, & k \in M_x, \\ 0, & k \notin M_x, \end{cases} \quad (5.13)$$

which is equivalent to the solution in (5.6). \square

5.3.2 Interpretation

In this section, we will give an alternative interpretation of the RDWO method based on $\widehat{\varphi}_x(k)$ introduced in (5.8), which will make the algorithm description more compact, which is given in Algorithm 1.

The following time-dependent sets are defined. For a given x , M_x^N is defined as the index set for the inputs $\{\varphi(k)\}_{k=1}^N$ which lie in the interval $(x - \delta, x + \delta)$, and $\{w_k^N(x)\}_{k=1}^N$ are the weights obtained by the RDWO method when N observations are obtained, $f_N(x)$ is the approximated function value at time N .

According to (5.13), when a new observation is obtained at time $N + 1$, the distance from $\varphi(N + 1)$ to x is calculated. If the distance is greater than δ , then the previous weights keep unchanged and no weight will be assigned for the current observation (corresponding to steps 4 to 6 in Algorithm 1). Otherwise, all the previous weights are reweighed by a factor λ_{N+1} , which is defined in (5.14), and the current observation is assigned with the weight $1 - \lambda_{N+1}$ (corresponding to steps 9 to 12 in Algorithm 1).

5.4 Conclusion

This chapter presented a novel derivation of the recursive direct weight optimization algorithm by introducing new quantities which can exploit useful structural information inherent in the problem. Based on the formulation provided by the new derivation, a new interpretation of the algorithm is also obtained. We end the discussions by remarking the following two points:

- The studies about the consistency properties and other related issues of the RDWO in [6] are also valid to the new formulation derived in this chapter, since the two formulations are mathematically equivalent.
- The difference compared to the earlier result in [6] is that by introducing the new structure exploiting quantities, the derivation and the interpretation of the RDWO are made more transparent and compact.

Algorithm 1 Recursive Direct Weight Optimization (RDWO)

- 1: Collect new data $y(N+1)$, $\varphi(N+1)$.
- 2: Calculate

$$\widehat{\varphi}_x(N+1) = \delta - |x - \varphi(N+1)|.$$

- 3: **if** $\widehat{\varphi}_x(N+1) \leq 0$ **then**
- 4: Set $M_x^{N+1} = M_x^N$.
- 5: Update w_k^{N+1} according to

$$w_k^{N+1}(x) = \begin{cases} w_k^N(x), & \text{if } k = 1, 2, \dots, N, \\ 0, & k = N+1. \end{cases}$$

- 6: Set $f_{N+1}(x) = f_N(x)$.
- 7: Set $N \leftarrow N+1$, and go back to iterate from step 1.
- 8: **else**
- 9: Set $M_x^{N+1} = M_x^N + m_{l+1}$, where $m_{l+1} = N+1$.
- 10: Calculate

$$\lambda_{N+1} = \frac{\sum_{j=1}^l \widehat{\varphi}_x(m_j)}{\sum_{j=1}^{l+1} \widehat{\varphi}_x(m_j)}, \quad (5.14)$$

- 11: Update w_k^{N+1} according to

$$w_k^{N+1}(x) = \begin{cases} \lambda_{N+1} w_k^N(x), & \text{if } k = 1, 2, \dots, N, \\ 1 - \lambda_{N+1}, & k = N+1. \end{cases}$$

- 12: Update $f_{N+1}(x)$ according to

$$f_{N+1}(x) = \lambda_{N+1} f_N(x) + (1 - \lambda_{N+1}) y(N+1).$$

- 13: Set $l \leftarrow l+1$ and $N \leftarrow N+1$, and go back to iterate from step 1.
 - 14: **end if**
-

Chapter 6

On the Kaczmarz Algorithm

6.1 Problem formulation

In this chapter, we study another type of recursive algorithm for parameter estimation — the Kaczmarz algorithm (KA) [55] and its randomized version, the Randomized Kaczmarz algorithm (RKA) advocated in [94]. The KA is an algorithm to solve a system of linear equations

$$\mathbf{Ax} = \mathbf{b}, \quad (6.1)$$

where $\mathbf{x} \in \mathbb{R}^n$ denotes the unknown parameter vector, $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m \geq n$, $\text{rank}(\mathbf{A}) = n$ and $\mathbf{b} \in \mathbb{R}^m$. Define the hyperplanes $\{H_i\}_{i=1}^m$ as

$$H_i = \{\mathbf{x} | \mathbf{a}_i^T \mathbf{x} = b_i\},$$

where the i -th row of \mathbf{A} is denoted by \mathbf{a}_i^T and the i -th element of \mathbf{b} is denoted by b_i . Geometrically, the KA finds the solution to (6.1) by projecting (or approximately projecting) onto the hyperplanes cyclically from an initial approximation \mathbf{x}_0 , which reads as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda \frac{b_{i(k)} - \mathbf{a}_{i(k)}^T \mathbf{x}_k}{\|\mathbf{a}_{i(k)}\|_2^2} \mathbf{a}_{i(k)}, \quad (6.2)$$

where $i(k) = \text{mod}(k, m) + 1$, in which $\text{mod}(\cdot, \cdot)$ denotes the *modulus after division* operation and $\|\cdot\|_2$ denotes the matrix spectral norm. In the update equation (6.2), λ is the relaxation parameter, which satisfies $0 < \lambda < 2$. Figure 6.1 illustrates the algorithm in a low dimensional case when $\lambda = 1$.

One typical application of the KA is on the computerized tomography [48], and a description of it can be found in the following example 6.1.

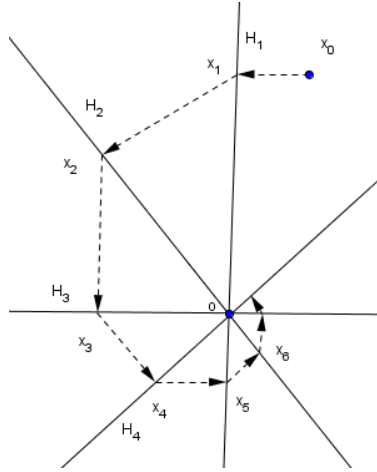


Figure 6.1: An illustration of the KA with $\lambda = 1$. Here, $m = 4$ and $n = 2$, and the solution \mathbf{x} to $A\mathbf{x} = \mathbf{b}$ is represented by the point o . We can see that by the sequence of projections, \mathbf{x}_k converges to the solution \mathbf{x} .

— **Example 6.1: Algebraic model for X-ray computerized tomography** —

The X-ray computerized tomography [48, 71] is a computer-based imaging technique which makes use of many X-rays from different directions to scan the inside of an object. Here, we will illustrate a simplified (only 2 dimensional) algebraic model of the idea underlying computerized tomography, for which the Kaczmarz algorithm can be used to estimate the parameters. In Figure 6.2, we assume that the object (to be scanned) is given by 9 squares, and x_1, \dots, x_9 represent their attenuation (damping) coefficients for the X-ray (different attenuation coefficients represent different matters). It is also assumed that the attenuation coefficient for each square is a constant. After the i -th X-ray has passed through the object, its intensity will be attenuated by the object, and this attenuation is denoted as b_i . Algebraically, b_i can also be written as a linear combination of x_1, \dots, x_9 as follows

$$a_{i,1}x_1 + a_{i,2}x_2 + \dots + a_{i,9}x_9 = b_i, \quad (6.3)$$

where $a_{i,j}$ represents the length of the i -th X-ray in block j , and $a_{i,j}x_j$ represents how much the i -th X-ray is attenuated in block j . Note that $a_{i,j}$ is determined by the geometry of the object and the vector

$$\mathbf{a}_i^T = (a_{i,1}, a_{i,2}, \dots, a_{i,9})$$

will be sparse with $a_{i,3}, a_{i,5}$ and $a_{i,6}$ nonzero in this case.

Let $\mathbf{x} = (x_1, x_2, \dots, x_9)^T$ and $\mathbf{b} = (b_1, b_2, \dots, b_9)^T$, we can then stack all the measurements from m X-rays into the equation $A\mathbf{x} = \mathbf{b}$, for which the Kaczmarz algorithm can be applied to find the vector \mathbf{x} . Note that the sparse property of matrix A makes the inner product in the Kaczmarz algorithm very efficient to compute.

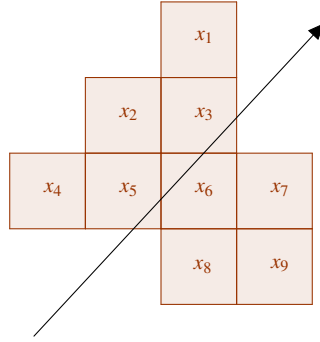


Figure 6.2: An illustration of the X-ray computerized tomography.

It is well-known that the KA is sometimes rather slow to converge. This is especially true when several consecutive row vectors of the matrix \mathbf{A} are in some sense "close" to each other. In order to mitigate this drawback, the RKA was introduced in [44, 95].

The key of the RKA is that, instead of performing the projections cyclically, the projections are performed in a random order. More specifically, at time k , the hyperplane H_p will be selected to be projected onto with probability proportional to $\|\mathbf{a}_p\|_2^2$, for $p = 1, \dots, m$. Intuitively speaking, the involved randomization is performing a kind of "preconditioning" to the original matrix equations [94], resulting the following exponential convergence rate:

$$\mathbb{E}(\|\mathbf{x}_k - \mathbf{x}\|_2^2) \leq (1 - \kappa(\mathbf{A})^{-2})^k \|\mathbf{x}_0 - \mathbf{x}\|_2^2, \quad (6.4)$$

in which $\kappa(\mathbf{A}) = \|\mathbf{A}\|_F \|\mathbf{A}^\dagger\|_2$, and with \mathbb{E} concerning all the random choices up to time k in the RKA.

As explained before, the deterministic ordering of the projections in (6.2) makes the algorithm slow to converge in certain cases. It is also challenging to study the performance bound of the method. One contribution of this chapter lies in presenting a convenient way to study the convergence property of the KA. The key underlying the approach is that we can interpret the solution path of the KA as the output of a particular dynamical system. By this connection, convergence results for the KA can be obtained by studying the stability property of the related dynamical system.

For the RKA, it is argued in [16] that 'Assigning probabilities corresponding to the row norms is in general certainly not optimal. The second contribution of this chapter attempts to find an improved probability distribution for

implementing the random projections, so that a better performance can be obtained. The distribution vector is found by minimizing a tight upper bound to the convergence rate of the RKA, which ends up with solving a convex optimization problem.

For notational convenience, we introduce the matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$, for which the i -th row \mathbf{b}_i^T is defined as

$$\mathbf{b}_i \triangleq \frac{\mathbf{a}_i}{\|\mathbf{a}_i\|_2}, i = 1, \dots, m.$$

Also let $\mathbf{P}_i \triangleq \mathbf{b}_i \mathbf{b}_i^T$ for $i = 1, 2, \dots, m$ and

$$\boldsymbol{\theta}_k \triangleq \mathbf{x}_k - \mathbf{x}$$

for $k \geq 0$.

The chapter will be organized as follows. In the subsequent section, we will establish an upper bound of the convergence rate of the KA. In particular, we study the convergence behavior of the sub-sequence $\{\boldsymbol{\theta}_{jm}\}_{j=0}^\infty$. In section 6.3, we then move on to the study of the RKA. After that, some numeric experiments are presented along with discussions. In the end, we draw the conclusion.

6.2 Convergence rate of the KA

Notice that, we can rewrite (6.2) as

$$\boldsymbol{\theta}_{k+1} = (\mathbf{I} - \lambda \mathbf{P}_{i(k)}) \boldsymbol{\theta}_k, \quad (6.5)$$

which can be interpreted as a discrete time-varying linear dynamical system. This observations inspires us to study the convergence of KA by employing the techniques for analyzing the stability properties of time varying linear systems, see e.g. [61, 39].

In what follows we will focus on analyzing the convergence rate of the sub-sequence $\{\|\boldsymbol{\theta}_{jm}\|_2^2\}_{j=0}^\infty$. Given the fact that $i(k) = \text{mod}(k, m) + 1$, we have

$$\boldsymbol{\theta}_{(j+1)m} = \left(\prod_{i=1}^m (\mathbf{I} - \lambda \mathbf{P}_i) \right) \boldsymbol{\theta}_{jm} \triangleq \mathbf{M}_m \boldsymbol{\theta}_{jm}.$$

The following theorem provides an upper bound on the spectral norm of \mathbf{M}_m .

Theorem 6.1. *Let $\rho \triangleq \|\mathbf{M}_m\|_2$ and $0 < \lambda \leq 2$, then it holds that*

$$\rho^2 \leq \rho_1 \triangleq 1 - \frac{\lambda(2-\lambda)}{(2+\lambda^2 m^2) \|\mathbf{B}^\dagger\|_2^2}, \quad (6.6)$$

where \mathbf{B}^\dagger denotes the pseudo-inverse of the matrix \mathbf{B} .

Proof. Let $\mathbf{v}_0 \in \mathbb{R}^n$ be a vector satisfying $\mathbf{M}_m \mathbf{v}_0 = \rho \mathbf{v}_0$, $\|\mathbf{v}_0\|_2 = 1$ and let $\mathbf{v}_i = (\mathbf{I} - \lambda \mathbf{P}_i) \mathbf{v}_{i-1}$ for $i = 1, \dots, m$. It follows that $\mathbf{v}_m = \mathbf{M}_m \mathbf{v}_0$ and $\|\mathbf{v}_m\|^2 = \rho^2$.

Note that since $\mathbf{P}_i^2 = \mathbf{P}_i$, we have that

$$(\mathbf{I} - \lambda \mathbf{P}_i)^2 = \mathbf{I} - (2\lambda - \lambda^2) \mathbf{P}_i,$$

for $i = 1, \dots, m$. Hence, it holds that

$$\begin{aligned} \|\mathbf{v}_i\|^2 &= \mathbf{v}_{i-1}^T (\mathbf{I} - \lambda \mathbf{P}_i)^2 \mathbf{v}_{i-1} \\ &= \mathbf{v}_{i-1}^T (\mathbf{I} - \lambda(2 - \lambda) \mathbf{P}_i) \mathbf{v}_{i-1} \\ &= \|\mathbf{v}_{i-1}\|^2 - \lambda(2 - \lambda) \|\mathbf{P}_i \mathbf{v}_{i-1}\|^2, \end{aligned}$$

which in turn implies that

$$\lambda(2 - \lambda) \sum_{i=1}^m \|\mathbf{P}_i \mathbf{v}_{i-1}\|^2 = \|\mathbf{v}_0\|^2 - \|\mathbf{v}_m\|^2 = 1 - \rho^2. \quad (6.7)$$

Also, for any $i \in \{1, \dots, m\}$, we have that

$$\begin{aligned} &\|\mathbf{v}_i - \mathbf{v}_0\| \\ &= \left\| \sum_{k=1}^i (\mathbf{v}_k - \mathbf{v}_{k-1}) \right\| = \lambda \left\| \sum_{k=1}^i \mathbf{P}_k \mathbf{v}_{k-1} \right\| \\ &\leq \lambda \sum_{k=1}^i \|\mathbf{P}_k \mathbf{v}_{k-1}\| \leq \lambda \sqrt{i} \sqrt{\sum_{k=1}^i \|\mathbf{P}_k \mathbf{v}_{k-1}\|^2} \\ &\leq \sqrt{\lambda i} \sqrt{\lambda \sum_{k=1}^m \|\mathbf{P}_k \mathbf{v}_{k-1}\|^2} \end{aligned}$$

Together with (6.7), we get

$$\|\mathbf{v}_i - \mathbf{v}_0\|^2 \leq \frac{\lambda i}{2 - \lambda} (1 - \rho^2). \quad (6.8)$$

Meanwhile, we have that

$$\begin{aligned} &\lambda \mathbf{v}_0^T \mathbf{B}^T \mathbf{B} \mathbf{v}_0 \\ &= \lambda \sum_{k=1}^m \mathbf{v}_0^T \mathbf{P}_k \mathbf{v}_0 = \lambda \sum_{k=1}^m \|\mathbf{P}_k \mathbf{v}_0\|^2 \\ &= \lambda \sum_{k=1}^m \|\mathbf{P}_k [\mathbf{v}_{k-1} + (\mathbf{v}_0 - \mathbf{v}_{k-1})]\|^2 \\ &\leq 2\lambda \sum_{k=1}^m \|\mathbf{P}_k \mathbf{v}_{k-1}\|^2 + 2\lambda \sum_{k=1}^m \|\mathbf{P}_k (\mathbf{v}_{k-1} - \mathbf{v}_0)\|^2 \\ &\leq 2\lambda \sum_{k=1}^m \|\mathbf{P}_k \mathbf{v}_{k-1}\|^2 + 2\lambda \sum_{k=1}^m \|\mathbf{v}_{k-1} - \mathbf{v}_0\|^2. \end{aligned} \quad (6.9)$$

Together with (6.7) and (6.8), we have that

$$\lambda \mathbf{v}_0^T \mathbf{B}^T \mathbf{B} \mathbf{v}_0 \leq \frac{2(1-\rho^2)}{2-\lambda} + 2\lambda \sum_{k=1}^m \frac{\lambda(k-1)}{2-\lambda} (1-\rho^2)$$

or equivalently

$$\lambda \mathbf{v}_0^T \mathbf{B}^T \mathbf{B} \mathbf{v}_0 \leq \frac{1-\rho^2}{2-\lambda} (2 + \lambda^2 m(m-1)), \quad (6.10)$$

and hence it follows that

$$\rho^2 \leq 1 - \frac{\lambda(2-\lambda) \mathbf{v}_0^T \mathbf{B}^T \mathbf{B} \mathbf{v}_0}{2 + \lambda^2 m(m-1)}.$$

Since $\mathbf{v}_0^T \mathbf{B}^T \mathbf{B} \mathbf{v}_0 \geq \frac{1}{\|\mathbf{B}^\dagger\|_2^2}$, we conclude that

$$\rho^2 \leq 1 - \frac{\lambda(2-\lambda)}{(2 + \lambda^2 m(m-1)) \|\mathbf{B}^\dagger\|_2^2}. \quad (6.11)$$

Finally, notice that $m(m-1) \leq m^2$ holds for any natural number m , which concludes the proof. \square

6.3 Optimized RKA

Now we move on to study how a probability distribution vector could be found to obtain an improved convergence performance of the RKA. Note that in this section, we always let $\lambda = 1$. Let $\mathbf{p} \in \mathbb{R}^m$ be a probability distribution vector (i.e. $\mathbf{p} \geq 0$, $\mathbf{1}^T \mathbf{p} = 1$) for selecting the rows in the RKA, and let p_i denote the i -th element of \mathbf{p} .

Suppose that we have \mathbf{x}_{k-1} , and based on \mathbf{x}_{k-1} , the next approximation \mathbf{x}_k is updated according to (6.2), in which the index $i(k-1) = j$ with probability p_j . By the property of the projection operation, we have that

$$\|\boldsymbol{\theta}_k\|_2^2 = \|\boldsymbol{\theta}_{k-1}\|_2^2 \sin^2(\alpha_i), \quad (6.12)$$

in which α_i denotes the angle between $\boldsymbol{\theta}_{k-1}$ and \mathbf{b}_i , i.e. the normal direction of the selected projecting hyperplane.

Based on the previous formula, we have that

$$\mathbb{E}_{|\mathbf{x}_{k-1}}(\|\boldsymbol{\theta}_k\|_2^2) = \|\boldsymbol{\theta}_{k-1}\|_2^2 \sum_{i=1}^m p_i \sin^2(\alpha_i), \quad (6.13)$$

in which $\mathbb{E}_{|\mathbf{x}_{k-1}}$ denotes the expectation operator conditioned on \mathbf{x}_{k-1} . It follows that:

$$\sum_{i=1}^m p_i \sin^2(\alpha_i) \leq \sup_{\mathbf{y} \in \mathbb{R}^n, \mathbf{y} \neq \mathbf{0}} \sum_{i=1}^m p_i \sin^2(\beta_i) \triangleq \Omega_1, \quad (6.14)$$

and

$$\sum_{i=1}^m p_i \sin^2(\alpha_i) \geq \inf_{\mathbf{y} \in \mathbb{R}^n, \mathbf{y} \neq \mathbf{0}} \sum_{i=1}^m p_i \sin^2(\beta_i) \triangleq \Omega_2, \quad (6.15)$$

in which β_i denotes the angle between \mathbf{y} and \mathbf{b}_i .

Given (6.13), (6.14) and (6.15), we have that

$$\mathbb{E}_{\cdot|\mathbf{x}_{k-1}}(\|\boldsymbol{\theta}_k\|_2^2) \leq \Omega_1 \|\boldsymbol{\theta}_{k-1}\|_2^2, \quad (6.16)$$

and

$$\mathbb{E}_{\cdot|\mathbf{x}_{k-1}}(\|\boldsymbol{\theta}_k\|_2^2) \geq \Omega_2 \|\boldsymbol{\theta}_{k-1}\|_2^2. \quad (6.17)$$

Iterating the relations given in (6.16) and (6.17), the following result follows

Theorem 6.2. *The norm of error vector $\boldsymbol{\theta}_k$ can be bounded in expectation by*

$$\mathbb{E}(\|\boldsymbol{\theta}_k\|_2^2) \leq \Omega_1^k \|\boldsymbol{\theta}_0\|_2^2, \quad (6.18)$$

and

$$\mathbb{E}(\|\boldsymbol{\theta}_k\|_2^2) \geq \Omega_2^k \|\boldsymbol{\theta}_0\|_2^2, \quad (6.19)$$

where the expectations are taken with respect to all the random choices up to time k .

According to Theorem 6.2, in order to get a better performance, we need to find a probability distribution vector, such that Ω_1 can be made as small as possible. In the following, we will first derive closed form expressions for Ω_1 and Ω_2 , and then introduce a convex optimization problem to calculate the probability distribution vector $\hat{\mathbf{p}}$ which minimizes Ω_1 .

Notice that

$$\sum_{i=1}^m p_i \sin^2(\beta_i) = 1 - \sum_{i=1}^m p_i \cos^2(\beta_i),$$

so in order to minimize Ω_1 , equivalently, we can maximize the following

$$\inf_{\mathbf{y} \in \mathbb{R}^n, \mathbf{y} \neq \mathbf{0}} \sum_{i=1}^m p_i \cos^2(\beta_i).$$

Restricting $\|\mathbf{y}\|_2 = 1$ will not affect $\{\beta_i\}_{i=1}^m$, and we get

$$\cos^2(\beta_i) = \mathbf{y}^T \mathbf{b}_i \mathbf{b}_i^T \mathbf{y}.$$

Therefore

$$\sum_{i=1}^m p_i \cos^2(\beta_i) = \sum_{i=1}^m p_i \mathbf{y}^T \mathbf{b}_i \mathbf{b}_i^T \mathbf{y},$$

where the right hand side can be rewritten as

$$\mathbf{y}^T \mathbf{B}^T \text{diag}(\mathbf{p}) \mathbf{B} \mathbf{y}.$$

Notice that

$$\min_{\mathbf{y} \in \mathbb{R}^n, \|\mathbf{y}\|_2=1} \mathbf{y}^T \mathbf{B}^T \text{diag}(\mathbf{p}) \mathbf{B} \mathbf{y} = \sigma_n(\mathbf{B}^T \text{diag}(\mathbf{p}) \mathbf{B}),$$

in which $\sigma_n(\cdot)$ denotes the smallest singular value of the matrix. Hence we have that

$$\Omega_1 = 1 - \sigma_n(\mathbf{B}^T \text{diag}(\mathbf{p}) \mathbf{B}), \quad (6.20)$$

and similarly, we have that:

$$\Omega_2 = 1 - \sigma_1(\mathbf{B}^T \text{diag}(\mathbf{p}) \mathbf{B}), \quad (6.21)$$

in which $\sigma_1(\cdot)$ denotes the maximal singular value of the matrix.

Notice that minimizing Ω_1 is equivalent to maximizing $\sigma_n(\mathbf{B}^T \text{diag}(\mathbf{p}) \mathbf{B})$, which means that we can solve the following problem instead:

$$\begin{aligned} \max_{\mathbf{p} \in \mathbb{R}^m} \quad & \sigma_n(\mathbf{B}^T \text{diag}(\mathbf{p}) \mathbf{B}) \\ \text{s.t.} \quad & \mathbf{1}^T \mathbf{p} = 1, \\ & p_i \geq 0, \quad i = 1, \dots, m. \end{aligned} \quad (6.22)$$

This problem can be rewritten as the following SDP problem, in which \hat{t} denotes the optimized σ_n and $\hat{\mathbf{p}}$ denotes the corresponding probability distribution vector:

$$\begin{aligned} (\hat{\mathbf{p}}, \hat{t}) = \arg \max_{\mathbf{p} \in \mathbb{R}^m, t \in \mathbf{R}} \quad & t \\ \text{s.t.} \quad & \mathbf{1}^T \mathbf{p} = 1, \\ & p_i \geq 0, \quad i = 1, \dots, m, \\ & \mathbf{B}^T \text{diag}(\mathbf{p}) \mathbf{B} - t \mathbf{I}_n \succeq 0. \end{aligned} \quad (6.23)$$

After solving the optimization problem of (6.23), $\hat{\mathbf{p}}$ is applied to the RKA to select the hyperplanes to project onto. Such a scheme will be abbreviated as Optimized RKA (ORKA) in the following part of the chapter.

Remark 6.1. *There exist cases such that $\Omega_1 = \Omega_2$, i.e. there exists a vector \mathbf{p} , such that*

$$\sigma_1(\mathbf{B}^T \text{diag}(\mathbf{p}) \mathbf{B}) = \sigma_n(\mathbf{B}^T \text{diag}(\mathbf{p}) \mathbf{B}),$$

i.e. $\mathbf{B}^T \text{diag}(\mathbf{p}) \mathbf{B} = \frac{1}{n} \mathbf{I}_n$. In such cases, we have that

$$\Omega_1 = \Omega_2 = 1 - \frac{1}{n},$$

and the optimized probability distribution obtained by solving eq. (6.23) is the same as suggested in [95]. It can be verified that when the columns of \mathbf{A} are orthogonal and of equal norms, then such property will hold.

Next, we discuss the relationship between the ORKA and the RKA. It is evident that the projection operations in (6.2) depend only on the corresponding normal directions of the hyperplanes $\{H_i\}_{i=1}^m$. Hence we can optimize

$$\kappa(\mathbf{A}) = \|\mathbf{A}\|_F \|\mathbf{A}^\dagger\|_2$$

with respect to the row norms of matrix \mathbf{A} . The optimization problem is given as

$$\min_{\{\|\mathbf{a}_i\|_2\}_{i=1}^m} \kappa(\mathbf{A})$$

Define $\mathbf{q} \in \mathbb{R}^m$, in which $q_i = \|\mathbf{a}_i\|_2^2$ for $i = 1, \dots, m$. Then the previous optimization problem can equivalently reformulated as

$$\min_{\mathbf{q}} \frac{\sqrt{\mathbf{1}^T \mathbf{q}}}{\sigma_n(\mathbf{A})}.$$

Set $\mathbf{1}^T \mathbf{q} = 1$ and notice the fact that $\mathbf{A}^T \mathbf{A} = \mathbf{B}^T \text{diag}(\mathbf{q}) \mathbf{B}$, then we can rewrite the previous problem as follows

$$\begin{aligned} (\hat{\mathbf{q}}, \hat{\sigma}_n) &= \arg \max_{\mathbf{q} \in \mathbb{R}^m, \sigma_n(\mathbf{A}) \in \mathbf{R}} \sigma_n^2(\mathbf{A}) \\ \text{s.t.} \quad &\mathbf{1}^T \mathbf{q} = 1, \\ &q_i \geq 0, \quad i = 1, \dots, m, \\ &\mathbf{B}^T \text{diag}(\mathbf{q}) \mathbf{B} - \sigma_n^2(\mathbf{A}) I_n \succeq 0. \end{aligned} \tag{6.24}$$

It can be observed that this optimization problem is equivalent to the problem in (6.23).

6.4 Discussion

Note that although the formulation in (6.23) is convex, it is still time consuming to solve this SDP optimization problem. In this section, we will discuss two possibilities to solve it approximately. One approximation of (6.23) is obtained by relaxing the constraint $\mathbf{B}^T \text{diag}(\mathbf{p}) \mathbf{B} - t I_n \succeq 0$ with the following linear constraints:

$$\mathbf{b}_i^T \text{diag}(\mathbf{p}) \mathbf{b}_i \geq t; \forall i = 1, \dots, m. \tag{6.25}$$

It is due to the fact that, for two positive semidefinite matrices $P_1, P_2 \in \mathbb{R}^{n \times n}$, if $P_1 \succeq P_2$, then $P_1(i, i) \geq P_2(i, i)$ holds for $i = 1, \dots, n$. Such relaxation reduces the SDP problem into a Linear Programming (LP) problem, which is computationally easier to solve.

In order to get a better relaxation, we introduce another approximation method which relates to the research in *Optimal Input Design* [32]. Notice that for any probability distribution vector \mathbf{p} , it holds true that

$$\text{tr}(\mathbf{B}^T \text{diag}(\mathbf{p})\mathbf{B}) = 1,$$

i.e. the summation of all the singular values of $\mathbf{B}^T \text{diag}(\mathbf{p})\mathbf{B}$ is fixed, then maximizing $\sigma_n(\mathbf{B}^T \text{diag}(\mathbf{p})\mathbf{B})$ will make all the singular values of $\mathbf{B}^T \text{diag}(\mathbf{p})\mathbf{B}$ tend to be close. This leads us to consider maximizing the product of the singular values of $\mathbf{B}^T \text{diag}(\mathbf{p})\mathbf{B}$, or maximizing the determinant of $\mathbf{B}^T \text{diag}(\mathbf{p})\mathbf{B}$. As the log function is monotonically increasing, we can optimize the following

$$\max_{\mathbf{p} \in \mathbb{R}^m} \log |\mathbf{B}^T \text{diag}(\mathbf{p})\mathbf{B}|, \quad (6.26)$$

in which $|\cdot|$ denotes the matrix determinant. Optimizing this quantity subject to the same constraints as in (6.22) boils down to solving the so-called *D-Optimal Design* problem. One simple iterative algorithm to solve such a problem has been suggested in [88], which is given as

$$\begin{aligned} p_i^0 &= \frac{\|\mathbf{a}_i\|^2}{\|\mathbf{A}\|_F^2}; & i = 1, \dots, m; \\ p_i^{t+1} &= p_i^t \frac{\mathbf{b}_i^T (\mathbf{B}^T \text{diag}(\mathbf{p}^t)\mathbf{B})^{-1} \mathbf{b}_i}{n}; & i = 1, \dots, m. \end{aligned} \quad (6.27)$$

Here, \mathbf{p}^t denotes the estimate at time t , and p_i^t denotes its i -th element. It has been proven in [107, 33] that for this algorithm, $\log |\mathbf{B}^T \text{diag}(\mathbf{p}^t)\mathbf{B}|$ increases monotonically w.r.t. t .

Remark 6.2. Here we give a brief proof of the monotonicity following the idea in [33]. Notice that the function $g(\mathbf{t}) = \log |\mathbf{B}^T \text{diag}(e^{\mathbf{t}})\mathbf{B}|$ is convex in $\mathbf{t} \in \mathbb{R}^m$ [40, 105], hence for any two vectors \mathbf{t}_1 and \mathbf{t}_2 , we have that

$$g(\mathbf{t}_2) \geq g(\mathbf{t}_1) + \nabla g(\mathbf{t}_1)^T (\mathbf{t}_2 - \mathbf{t}_1).$$

Let $\mathbf{t}_1 = \log(\mathbf{p}^t)$ and $\mathbf{t}_2 = \log(\mathbf{p}^{t+1})$, then the following fact holds true

$$\log |\mathbf{B}^T \text{diag}(\mathbf{p}^{t+1})\mathbf{B}| \geq \log |\mathbf{B}^T \text{diag}(\mathbf{p}^t)\mathbf{B}| + n \sum_{i=1}^m p_i^{t+1} \log \frac{p_i^{t+1}}{p_i^t}.$$

Notice that $\sum_{i=1}^m p_i^{t+1} \log \frac{p_i^{t+1}}{p_i^t} \geq 0$, which concludes the monotonicity of the iteration in (6.27).

In next section, we will make use of this monotonicity property to approximately solve (6.22) when the objective function is replaced by (6.26).

6.5 Experiments

In this section, we will conduct experiments to illustrate the efficacy of the presented methods for obtaining an improved randomization strategy for the RKA.

The setup of our experiment is given as follows. The matrix \mathbf{A} is first generated by $\text{randn}(m,n)$ in Matlab with $m = 200$ and $n = 20$. After that, each row is normalized, and then scaled with a random number which is uniformly distributed in $[0, 1]$. The reason for generating \mathbf{A} in this way is that in the first stage, the generated rows of \mathbf{A} will have different directions which are uniformly distributed on the sphere S^{n-1} [67], and in the second stage, different rows of \mathbf{A} will be assigned with different norms, to see how different row norms will affect the performance of the RKA. The vector \mathbf{x} is generated by $\text{randn}(n,1)$, and \mathbf{b} is generated as $\mathbf{b} = \mathbf{A}\mathbf{x}$. We will compare the Mean Squared Error (MSE) along the projection path obtained by the following methods:

1. The one suggested in [95] (abbreviated as *RKA*);
2. The one obtained by the SDP optimization given by (6.23) (abbreviated as *ORKA*);
3. The one obtained by the LP approximations in (6.25) (abbreviated as *LPORKA*);
4. The one obtained by the iterative method to solve the D-Optimal Design criteria (abbreviated as *ITEORKA*). We iterate (6.27) for 10 times in this experiment.

For each method, we run the experiment 2000 times to get the averaged performance. The CVX toolbox¹ is used to solve the SDP and LP optimization problems. From the experiment, we can observe that the time for solving the LP problem in LPORKA is close to the time needed for 10 iterations of (6.27), and the time needed for solving (6.23) in ORKA is approximately 7 times as solving the LP problem in LPORKA. The experiment results are reported in Figure 6.3, and more discussions are given in the caption therein.

6.6 Conclusion

This chapter discussed two aspects related with the Kaczmarz algorithm. We first analyzed the convergence rate for the classical KA from a dynamical system point of view. We also studied the possibility and methodology to find an improved probability distribution vector for implementing the random projections inside the Randomized Kaczmarz Algorithm. Numerical examples were also provided to show the improvements obtained by the proposed method.

¹<http://cvxr.com/>

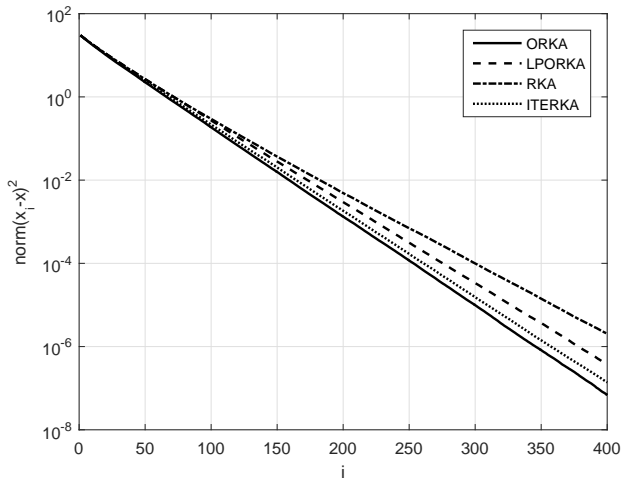


Figure 6.3: The curves demonstrate the MSE for different methods. We can see that the ORKA improves the convergence speed the most; the LPORKA method and the ITERKA method also improve the convergence speed, and the ITEORKA method improves more than the LPORKA method.

Chapter 7

Summary and future work

7.1 Thesis summary

This thesis has made contributions in the following two themes.

1. To understand the convexification idea and its application in SysID by two case studies;
2. The new look of several existed recursive identification algorithms, by providing new insights and observations.

More specifically, in Chapter 2, we showed that a l_1 norm based approach can indeed help in exploiting the sparsity information of the underlying parameter vector under certain Persistent Excitation assumptions. In Chapter 3, we studied whether and how the nuclear norm minimization heuristic can be useful in completing a low rank Hankel matrix, where one particular problem was examined. In Chapter 4, we established a relation between a basic property of convolution operator and the score function estimation, and based on this relationship, the recursive Bayesian algorithms can be exploited to estimate the score function for systems with intractable transition densities. In Chapter 5, a new derivation and a new interpretation of the Recursive Direct Weight Optimization method were proposed by exploiting certain structural information in the problem formulation. In Chapter 6, we studied how an improved randomization strategy can be found for the Randomized Kaczmarz Algorithm, and how the convergence rate of the classical Kaczmarz Algorithm can be estimated by the stability analysis of a related linear dynamical system.

7.2 Future work

Despite that some progress has been made in the thesis, there are of course still many research questions that remain to be studied based on the findings reported in the thesis, which are organized chapter by chapter as follows.

- Chapter 2

One open question is to quantify how many samples would be sufficient to guarantee exact recovery of \mathbf{x}^0 for a given sparsity level. Another open question is how to find a suitable weighting matrix \mathbf{W} (the one in Remark 2.3) which can further improve the performance of the proposed algorithm.

- Chapter 3

One open question is how to generalize the presented analysis to the stable multiple-poles system case. By inspecting the proofs for the single-pole case, we can see that Lemmas 3.1 and 3.3 are still applicable. However, the construction of M_0 becomes more complicated in this situation. Another question is that the studies in this chapter assume noiseless data, it is not clear how this assumption can be relaxed in the noisy case.

- Chapter 4

As we can see from the experiment, when the variance of the artificial prior shrinks to be very small, the estimator quality degrades. The question of how to improve the numerical accuracy when the variance of the prior distribution is small remains open. The relationship between the convolution property and derivative estimation is not specific for state space models, it would be interesting to find out other applications where this connection can be utilized.

- Chapter 6

Though an improved algorithm can be found by solving (6.22), this is still an SDP problem, which is quite computationally expensive to solve. One future research direction is to find a more computationally efficient way to solve (or approximately solve) the optimization problem in (6.22).

Summary in Swedish

Systemidentifiering (SysID) handlar om att ta fram matematiska modeller för dynamiska system från uppmätt in- och utdata. De matematiska modellerna kan ge en djupare förståelse för det underliggande systemet, och kan också användas i praktiska tillämpningar som prediktion och prediktionsreglering.

Systemidentifiering kan grovt sagt delas in i två kategorier - linjär och icke-linjär SysID. För linjär SysID finns det en välutvecklad teori som finns beskriven från olika perspektiv i flera läroböcker, te.x [63, 91]. Populära metoder för linjär SysID inkluderar maximum likelihood (ML), prediktionsfelsmetoden (engelska: prediction error method, PEM), och subspace identification (SI). ML-metoden hittar parametrarna genom att maximera den så kallade likelihood-funktionen för den observerade data. PEM är en generalisering av ML, och identifierar parametrarna genom att optimera en välvald kostnadsfunktion för prediktionsfelet. SI-metoden skiljer sig från de två andra, genom att den direkt estimerar en tillståndsmodell från in- och utdata med hjälp av verktyg från linjär algebra.

Till skillnad från linjär SysID, så är icke-linjär SysID ett relativt nytt forskningsområde som fortfarande är mycket aktivt. Genomgångar av tidiga framsteg finns i [54, 89], och nya resultat finns beskrivna i [35]. Ett sätt att representera en icke-linjär modell är Volterra-serier [85, 86]. Denna representation leder ofta till ett stort antal okända parameterer som måste uppskattas, vilket gör den svår att använda för kraftigt icke-linjära system. Ett annat angreppssätt är att använda sig av basfunktionsutveckling för den icke-linjära delen. Vanliga basfunktioner inkluderar Fourierbasen, wavelets och ortonormala polynom [99, 5]. Med basfunktioner fås ofta estimeringsproblem som är linjära i parametrarna, och därmed är de relativt enkla att lösa. Ett annat alternativ för icke-linjär SysID är att använda kernel-baserade metoder [28], då de, utöver vissa antaganden om glatthet, inte kräver några särskilt restriktiva antaganden.

Utöver metoderna som nämnts ovan, så är konvex optimering ett verktyg som används allt mer vid både linjär och icke-linjär systemidentifiering. Orsaken är att många identifieringsproblem antingen kan formuleras direkt, eller omformuleras som, ett konvext optimeringsproblem. Att transformera ett icke-

konvext problem till ett konvext problem kallas för konvexifiering. Intuitivt så innebär detta att det ursprungliga svåra problemet approximeras av ett nytt och enklare problem. Första delen av denna avhandling handlar om två problem där konvexifiering kan användas. Där visas hur denna teknik kan vara användbar för att förbättra estimering när parametervektorn är gles (dvs, endast ett fåtal element i parametervektorn är noll-skilda). Sedan analyserar vi hur minimering av nukelärnormen (engelska: nuclear norm, dvs summan av alla singularvärden i en matris) beter sig vid komplettering av Hankel-matriser med låg rang. Grundidén är att använda strukturinformation från problemställningen för att skapa ett dualt certifikat.

Rekursiva algoritmer är vanliga inom systemidentifiering. Den andra delen av denna avhandling studerar några existerande rekursiva algoritmer, och etablerar nya kopplingar, insikter och tolkningar av dem. Vi visar först på en koppling mellan grundläggande egenskaper hos faltningsoperatören och "the score function estimation". Via denna relation visar vi hur rekursiva Bayesianska algoritmer kan användas för att estimeras the score function för system med svåra överångstätheter. Vi tar också fram en ny härledning och tolkning av den rekursiva "direct optimization method", genom att utnyttja viss strukturell information i algoritmen. Till sist visar vi fram en förbättrad randomiseringsstrategi för den randomiserade Kaczmarz-algoritmen, och visar att konvergens för Kaczmarz algoritmen kan studeras via stabilitetsanalys för ett relaterat linjärt dynamiskt system.

Acknowledgments

First of all I would like to thank my supervisors Thomas Schön and Kristiaan Pelckmans for their guidance and support during the past many years. Thanks to all my past and present SysCon colleagues for creating a nice working environment. I would also like to thank Joel Kronander, Per Mattsson, Johannes Nygren and Andreas Svensson for proof reading the thesis and giving constructive suggestions. Last but not least I would like to thank all my family members for your unconditional and endless support.

Thanks to the Swedish Research Council for supporting the work under the project *Probabilistic modeling of dynamical systems* (Contract number: 621-2013-5524).

References

- [1] C. Andrieu, A. Doucet, and R. Holenstein. Particle markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.
- [2] C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient monte carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.
- [3] K. J. Åström. Maximum likelihood and prediction error methods. *Automatica*, 16(5):551–574, 1980.
- [4] K. J. Åström and B. Wittenmark. *Adaptive Control*. Courier Corporation, 2013.
- [5] E.-W. Bai. Non-parametric nonlinear system identification: a data-driven orthogonal basis function approach. *IEEE Transactions on Automatic Control*, 53(11):2615–2626, 2008.
- [6] E.-W. Bai and Y. Liu. Recursive direct weight optimization in nonlinear system identification: A minimal probability approach. *IEEE Transactions on Automatic Control*, 52(7):1218–1231, 2007.
- [7] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.
- [8] E. Candés and T. Tao. The Dantzig selector: statistical estimation when p is much larger than n . *The Annals of Statistics*, 35(6):2313–2351.
- [9] E. J. Candés and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.
- [10] E. J. Candés and Y. Plan. A probabilistic and riplless theory of compressed sensing. *IEEE Transactions on Information Theory*, 57(11):7235–7254, 2011.
- [11] E. J. Candés and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.
- [12] E. J. Candés, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.
- [13] E. J. Candés and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.
- [14] E. J. Candés and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.
- [15] E. J. Candés and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.
- [16] Y. Censor, G. T. Herman, and M. Jiang. A note on the behavior of the randomized Kaczmarz algorithm of Strohmer and Vershynin. *Journal of*

- Fourier Analysis and Applications*, 15(4):431–436, 2009.
- [17] S. Chen, S. Billings, C. Cowan, and P. Grant. Non-linear systems identification using radial basis functions. *International Journal of Systems Science*, 21(12):2513–2539, 1990.
- [18] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1):33–61, 1998.
- [19] N. Chopin. Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *Annals of statistics*, 32(6):2385–2411, 2004.
- [20] P. De Larminat. *Climate Change: Identification and Projections*. John Wiley & Sons, 2014.
- [21] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
- [22] D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995.
- [23] A. Doucet, P. E. Jacob, and S. Rubenthaler. Derivative-free estimation of the score vector and observed information matrix with application to state-space models. *arXiv preprint arXiv:1304.5768v1*, 2013.
- [24] A. Doucet, P. E. Jacob, and S. Rubenthaler. Derivative-free estimation of the score vector and observed information matrix with application to state space models. *arXiv preprint arXiv:1304.5768v3*, 2015.
- [25] B. Efron, T. Hastie, and R. Tibshirani. Discussion of the ‘Dantzig selector’. *The Annals of Statistics*, 35(6):2358–2364, 2007.
- [26] T. Falck, J. A. Suykens, J. Schoukens, and B. De Moor. Nuclear norm regularization for overparametrized hammerstein systems. In *Proceedings of the 49th IEEE Conference on Decision and Control (CDC), Atlanta, GA USA*, pages 7202–7207. IEEE, 2010.
- [27] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.
- [28] J. Fan and Q. Yao. *Nonlinear time series: nonparametric and parametric methods*. Springer Science & Business Media, 2003.
- [29] M. Fazel. *Matrix rank minimization with applications*. PhD thesis, Stanford University, 2002.
- [30] M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the American Control Conference*, volume 6, pages 4734–4739. IEEE, 2001.
- [31] M. Fazel, H. Hindi, and S. P. Boyd. Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *Proceedings of the American Control Conference (ACC), Denver, CO USA*, volume 3, pages 2156–2162. IEEE, 2003.
- [32] V. V. Fedorov. *Theory of optimal experiments*. Elsevier, 1972.
- [33] W. Gao, P. S. Chan, H. K. T. Ng, and X. Lu. Efficient computational algorithm for optimal allocation in regression models. *Journal of Computational and Applied Mathematics*, 261:118–126, 2014.
- [34] D. Ge, X. Jiang, and Y. Ye. A note on the complexity of l_p minimization. *Mathematical Programming*, 129(2):285–299, 2011.

- [35] F. Giri and E.-W. Bai. *Block-oriented nonlinear system identification*. Springer, 2010.
- [36] N. J. Gordon, D. J. Salmond, and A. F. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.
- [37] D. Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.
- [38] C. Grossmann, C. N. Jones, and M. Morari. System identification via nuclear norm regularization for simulated moving bed processes from incomplete data sets. In *Proceedings of the 48th IEEE Conference on Decision and Control (CDC), Shanghai, China*, pages 4692–4697. IEEE, 2009.
- [39] L. Guo and L. Ljung. Exponential stability of general tracking algorithms. *IEEE Transactions on Automatic Control*, 40(8):1376–1387, 1995.
- [40] L. Gurvits and A. Samorodnitsky. A deterministic algorithm for approximating the mixed discriminant and mixed volume, and a combinatorial corollary. *Discrete and Computational Geometry*, 27(4):531–550, 2002.
- [41] Y. Han and R. A. De Callafon. Hammerstein system identification using nuclear norm minimization. *Automatica*, 48(9):2189–2193, 2012.
- [42] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.
- [43] G. Hendeby, R. Karlsson, and F. Gustafsson. Particle filtering: the need for speed. *EURASIP Journal on Advances in Signal processing*, 2010:22, 2010.
- [44] G. T. Herman and L. B. Meyer. Algebraic reconstruction techniques can be made computationally efficient [positron emission tomography application]. *IEEE Transactions on Medical Imaging*, 12(3):600–609, 1993.
- [45] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms II: Advanced theory and Bundle methods*, volume 306. Springer Science & Business Media, 2013.
- [46] H. Hjalmarsson, J. S. Welsh, and C. R. Rojas. Identification of Box-Jenkins models using structured ARX models and nuclear norm relaxation. In *16th IFAC Symposium on System Identification (SYSID), Brussels, Belgium*, 2012.
- [47] R. A. Horn and C. R. Johnson. Topics in matrix analysis. *Cambridge University Press*, 1991.
- [48] G. N. Hounsfield. Computerized transverse axial scanning (tomography): Part 1. Description of system. *The British Journal of Radiology*, 46(552):1016–1022, 1973.
- [49] H. M. Hudson. A natural identity for exponential families with applications in multiparameter estimation. *The Annals of Statistics*, 6(3):473–484, 1978.
- [50] E. Ionides, C. Bretó, and A. King. Inference for nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 103(49):18438–18443, 2006.
- [51] E. L. Ionides, A. Bhadra, Y. Atchadé, and A. King. Iterated filtering. *The Annals of Statistics*, 39(3):1776–1802, 2011.
- [52] P. E. Jacob. Sequential Bayesian inference for implicit hidden Markov models and current limitations. *ESAIM: Proceedings and Surveys*, 51(10):24–48, 2015.

- [53] E. Jones, J. Parslow, and L. M. Murray. A Bayesian approach to state and parameter estimation in a Phytoplankton-Zooplankton model. *Australian Meteorological and Oceanographic Journal*, 59:7–16, 2010.
- [54] A. Juditsky, H. Hjalmarsson, A. Benveniste, B. Delyon, L. Ljung, J. Sjöberg, and Q. Zhang. Nonlinear black-box models in system identification: Mathematical foundations. *Automatica*, 31(12):1725–1750, 1995.
- [55] S. Kaczmarz. Angenäherte auflösung von systemen linearer gleichungen. *Bulletin International de l'Academie Polonaise des Sciences et des Lettres*, 35:355–357, 1937.
- [56] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.
- [57] J. Kronander and T. Schön. Robust auxiliary particle filters using multiple importance sampling. In *Statistical Signal Processing (SSP), 2014 IEEE Workshop on*, pages 268–271. IEEE, 2014.
- [58] S. L. Kukreja. Application of a least absolute shrinkage and selection operator to aeroelastic flight test data. *International Journal of Control*, 82(12):2284–2292, 2009.
- [59] P. Kump, E.-W. Bai, K.-s. Chan, B. Eichinger, and K. Li. Variable selection via RIVAL (removing irrelevant variables amidst Lasso iterations) and its application to nuclear material detection. *Automatica*, 48(9):2107–2115, 2012.
- [60] R. Laubenfels. Feynman–Kac formula: Genealogical and interacting particle systems with applications. *Journal of the American Statistical Association*, 100(472):1460–1460, 2005.
- [61] G. Lei. *Time Varying Stochastic Systems: Stability, Estimation and Control*. Jilin Science and Technology Press, 1993.
- [62] F. Lindsten and T. B. Schön. Backward simulation methods for Monte Carlo statistical inference. *Foundations and Trends in Machine Learning*, 6(1):1–143, 2013.
- [63] L. Ljung. *System Identification*. Springer, 1998.
- [64] L. Ljung. Perspectives on system identification. *Annual Reviews in Control*, 34(1):1–12, 2010.
- [65] I. Markovsky. Structured low-rank approximation and its applications. *Automatica*, 44(4):891–909, 2008.
- [66] I. Markovsky. How effective is the nuclear norm heuristic in solving data approximation problems? In *Proceedings of the 16th IFAC Symposium on System Identification (SYSID), Brussels, Belgium*, 2012.
- [67] G. Marsaglia. Choosing a point from the surface of a sphere. *The Annals of Mathematical Statistics*, 43(2):645–646, 1972.
- [68] M. Morari, C. Garcia, J. Lee, and D. Prett. *Model Predictive Control*. Prentice Hall Englewood Cliffs, NJ, 1993.
- [69] L. M. Murray, E. M. Jones, and J. Parslow. On disturbance state space models and the particle marginal Metropolis-Hastings sampler. *SIAM Journal on Uncertainty Quantification*, 1(1):494–521, 2013.
- [70] L. M. Murraya, A. Leeb, and P. E. Jacob. Parallel resampling in the particle filter. *Journal of Computational and Graphical Statistics*, 2016.
- [71] F. Natterer. *The mathematics of computerized tomography*, volume 32. Siam, 1986.

- [72] D. Needell. Randomized kaczmarz solver for noisy linear systems. *BIT Numerical Mathematics*, 50(2):395–403, 2010.
- [73] K. Pelckmans. Convex optimization for blind identification of monotone wiener systems. In *Proceedings of the 17th IFAC Symposium on System Identification (SYSID), Beijing, China*. IFAC, 2015.
- [74] R. Pintelon and J. Schoukens. *System identification: a frequency domain approach*. John Wiley & Sons, 2012.
- [75] M. K. Pitt, R. Dos Santos Silva, P. Giordani, and R. Kohn. On some properties of markov chain monte carlo simulation methods based on the particle filter. *Journal of Econometrics*, 171(2):134–151, 2012.
- [76] M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.
- [77] R. L. Plackett. Some theorems in least squares. *Biometrika*, 37(1-2):149–157, 1950.
- [78] G. Poyiadjis, A. Doucet, and S. S. Singh. Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98(1):65–80, 2011.
- [79] H. Rauhut. Compressive sensing and structured random matrices. *Theoretical foundations and numerical methods for sparse recovery*, 9:1–92, 2010.
- [80] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.
- [81] C. R. Rojas and H. Hjalmarsson. Sparse estimation based on a validation criterion. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando, Florida USA*, pages 2825–2830. IEEE, 2011.
- [82] J. Roll, A. Nazin, and L. Ljung. A non-asymptotic approach to local modelling. In *Proceedings of the 41st IEEE Conference on Decision and Control (CDC), Las Vegas, NV USA*, pages 638–643. IEEE, 2002.
- [83] J. Roll, A. Nazin, and L. Ljung. A general direct weight optimization framework for nonlinear system identification. 2005.
- [84] J. Roll, A. Nazin, and L. Ljung. Nonlinear system identification via direct weight optimization. *Automatica*, 41(3):475–490, 2005.
- [85] W. J. Rugh. *Nonlinear System Theory*. Johns Hopkins University Press, 1981.
- [86] M. Schetzen. *The Volterra and Wiener theories of nonlinear systems*. John Wiley & Sons, 1980.
- [87] T. B. Schön, F. Lindsten, J. Dahlin, J. Wågberg, C. A. Naesseth, A. Svensson, and L. Dai. Sequential monte carlo methods for system identification. In *Proceedings of the 17th IFAC Symposium on System Identification (SYSID), Beijing, China*, 2015.
- [88] S. Silvey, D. Titterton, and B. Torsney. An algorithm for optimal designs on a design space. *Communications in Statistics Theory and Methods*, 7(14):1379–1389, 1978.
- [89] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995.
- [90] R. S. Smith. Frequency domain subspace identification using nuclear norm

- minimization and hankel matrix realizations. *IEEE Transactions on Automatic Control*, 59(11):2886–2896, 2014.
- [91] T. Söderström and P. Stoica. *System Identification*. Prentice-Hall, Inc., NJ, USA, 1989.
- [92] C. M. Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, 9(6):1135–1151, 1981.
- [93] P. Stoica and R. L. Moses. *Introduction to spectral analysis*. Prentice hall Upper Saddle River, 1997.
- [94] T. Strohmer and R. Vershynin. Comments on the randomized Kaczmarz method. *Journal of Fourier Analysis and Applications*, 15(4):437–440, 2009.
- [95] T. Strohmer and R. Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15(2):262–278, 2009.
- [96] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999.
- [97] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.
- [98] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95(2):189–217, 2003.
- [99] P. Van den Hof and B. Ninness. System identification with generalized orthonormal basis functions. In *Modelling and Identification with Rational Orthogonal Basis Functions*, pages 61–102. Springer, 2005.
- [100] P. Van Overschee and B. De Moor. *Subspace identification for linear systems: Theory, Implementation, Applications*. Springer Science & Business Media, 2012.
- [101] L. Vandenberghe. Convex optimization techniques in system identification. In *Proceedings of the IFAC Symposium on System Identification (SYSID), Brussels, Belgium*, pages 71–76, 2012.
- [102] M. Verhaegen and A. Hansson. Nuclear norm subspace identification (N2SID) for short data batches. *arXiv preprint arXiv:1401.4273*, 2014.
- [103] M. Verhaegen and V. Verdult. *Filtering and system identification: a least squares approach*. Cambridge University Press, 2007.
- [104] N. Whiteley et al. Stability properties of some particle filters. *The Annals of Applied Probability*, 23(6):2500–2537, 2013.
- [105] A. Wiesel. Geodesic convexity and covariance estimation. *IEEE Transactions on Signal Processing*, 60(12):6182–6189, 2012.
- [106] A. Winkelbauer. Moments and absolute moments of the normal distribution. *arXiv preprint arXiv:1209.4340*, 2012.
- [107] Y. Yu. Monotonic convergence of a general algorithm for computing optimal designs. *The Annals of Statistics*, 38(3):1593–1606, 2010.
- [108] P. Zhao and B. Yu. On model selection consistency of Lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006.
- [109] H. Zou. The adaptive Lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

